15TH INTERNATIONAL CONFERENCE

ON

INTELLIGENT GAMES AND SIMULATION

GAME-ON® 2014

EDITED BY

Patrick Dickinson

September 9-11, 2014

University of Lincoln
Lincoln
UNITED KINGDOM

A Publication of EUROSIS-ETI

15TH International Conference

on

Intelligent Games and Simulation

LINCOLN, UNITED KINGDOM

SEPTEMBER 9 - 11, 2014

Organised by

ETI

Sponsored by

EUROSIS

Co-Sponsored by

Binary Illusions

Ghent University

Larian Studios

University of Skövde

University of Lincoln

Lincoln, United Kingdom

EXECUTIVE EDITOR

PHILIPPE GERIL
(BELGIUM)

EDITOR

General Conference Chair

Patrick Dickinson
University of Lincoln, Lincoln, United Kingdom

Local Programme Committee

Andrew Elliott, University of Lincoln, Lincoln, United Kingdom
John Murray, University of Lincoln, Lincoln, United Kingdom
Greg Cielniak, University of Lincoln, Lincoln, United Kingdom
Olivier Szymanezyk, University of Lincoln, Lincoln, United Kingdom

INTERNATIONAL PROGRAMME COMMITTEE

Game Development Methodology

Óscar Mealha, University of Aveiro, Portugal
Esteban Clua, Universidade Federal Fluminense, Brasil

Physics and Simulation

Graphics Simulation and Techniques
Ian Marshall, Coventry University, Coventry, United Kingdom
Marco Roccetti, University of Bologna, Bologna, Italy

Facial, Avatar, NPC, 3D in Game Animation
Yoshihiro Okada, Kyushu University, Kasuga, Fukuoka, Japan
Marcos Rodrigues, Sheffield Hallam University, Sheffield, United Kingdom
Joao Manuel Tavares, FEUP, Porto, Portugal

Rendering Techniques
Joern Loviscach, Hochschule Bremen, Bremen, Germany

Artificial Intelligence

Artificial Intelligence and Simulation Tools for Game Design
Stephane Assadourian, UBISOFT, Montreal, Canada
Flavio Soares Correa da Silva, USP, Sao Paulo, Brazil
Patrick Dickinson, Lincoln University, Lincoln, United Kingdom
Antonio J. Fernandez, Universidad de Malaga, Malaga, Spain
Marc-Philippe Huget, University of Savoie, Le-Bourget-du-Lac, France
Joseph Kehoe, Institute of Technology Carlow, Carlow, Ireland
David Moffat, Glasgow Caledonian University, Glasgow, United Kingdom
Sam Redfern, National University of Ireland, Galway, Ireland
Miguel Tsai, Ling Tung University, Taichung, Taiwan

Learning & Adaptation
Christian Bauckage, University of Bonn, Sankt Augustin, Germany
Christos Bouras, University of Patras, Patras, Greece
Adriano Joaquim de Oliveira Cruz, Univ. Federal de Rio de Janeiro, Rio de Janeiro, Brazil
Andrzej Dzielinski, Warsaw University of Technology, Warsaw, Poland
Maja Pivec, FH JOANNEUM, University of Applied Sciences, Graz, Austria

INTERNATIONAL PROGRAMME COMMITTEE

Intelligent/Knowledgeable Agents
Nick Hawes, University of Birmingham, United Kingdom
Wenji Mao, Chinese Academy of Sciences, Beijing, China P.R.
Marco Remondino, University of Turin, Turin, Italy

Collaboration & Multi-agent Systems
Victor Bassilious, University of Abertay, Dundee, United Kingdom
Sophie Chabridon, Groupe des Ecoles de Telecommunications, Paris, France

Opponent Modelling
Pieter Spronck, University of Maastricht, Maastricht, The Netherlands
Ingo Steinhauser, Binary Illusions, Braunschweig, Germany

Peripheral

Psychology, Affective Computing and Emotional Gaming

Myriam Abramson, US Naval Research Laboratory, USA
Eva Hudlicka, Psychometrix Associates, Blacksburg, USA

Artistic input to game and character design

Anton Eliens, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
Richard Wages, Nomads Lab, Koln, Germany

Storytelling and Natural Language Processing

Jenny Brusk, Gotland University College, Gotland, Sweden
Ruck Thawonmas, Ritsumeikan University, Kusatsu, Shiga, Japan
Clark Verbrugge, McGill University, Montreal, Canada

Online Gaming and Security Issues in Online Gaming
Marco Furini, University of Modena and Reggio Emiliano, Modena, Italy
Pal Halvorsen, University of Oslo, Oslo, Norway
Fredrick Japhet Mtenzi, School of Computing, Dublin, Ireland
Andreas Petlund, University of Oslo, Oslo, Norway
Jouni Smed, University of Turku, Turku, Finland
Knut-Helge Vik, University of Oslo, Oslo, Norway

MMOG's

Michael J. Katchabaw, The University of Western Ontario, London, Canada
Jens Mueller-Iden, University of Munster, Munster, Germany
Alice Leung, BBN Technologies, Cambridge, USA
Mike Zyda, USC Viterbi School of Engineering, Marina del Rey, USA

Serious Gaming

Wargaming Aerospace Simulations, Board Games etc....
Roberto Beauclair, Institute for Pure and Applied Maths., Rio de Janeiro, Brazil
Joseph M. Saur, Georgia Tech Research Institute, Atlanta, USA
Jaap van den Herik, Tilburg University, Tilburg, The Netherlands

Games for training
Michael J. Katchabaw, The University of Western Ontario, London, Canada
Jens Mueller-Iden, Universitaet Muenster, Muenster, Germany

INTERNATIONAL PROGRAMME COMMITTEE

Games Applications in Education, Government, Health, Corporate, First Responders and Science
Russell Shilling, Office of Naval Research, Arlington VA, USA

Games Interfaces - Playing outside the Box

Games Console Design
Chris Joslin, Carleton University, Ottawa, Canada

GAME ON®
2014

All articles published in these Proceedings have been peer reviewed

EUROSIS-ETI Publications are ISI-Thomson and IET referenced

GAMEON proceedings are indexed on SCOPUS and Elsevier Engineering Village

A CIP Catalogue record for this book is available from the Royal Library of Belgium under nr.12620

For permission to publish a complete paper write EUROSIS, c/o Philippe Geril, ETI Executive Director, Greenbridge Science Park, Ghent University, Ostend Campus, Wetenschapspark 1, Plassendale 1, B-8400 Ostend, Belgium.

EUROSIS is a Division of ETI Bvba, The European Technology Institute, Torhoutsesteenweg 162, Box 4, B-8400 Ostend, Belgium

Printed in Belgium by Reproduct NV, Ghent, Belgium
Cover Design by Grafisch Bedrijf Lammaing, Ostend, Belgium
Cover Pictures by Larian Studios, Ghent, Belgium

EUROSIS-ETI Publication

ISBN: 978-9077381-85-4
EAN: 978-9077381-85-4

Preface

Dear Participants,

It gives me the greatest pleasure to welcome you to the School of Computer Science at the University of Lincoln, and to the 15th edition of GAMEON, which we are hosting between 9th and 11th of September 2014.

Computer games were already well established as the cornerstone of a global entertainment industry when the first GAMEON conference took place; since then they have have become not only a pervasive part of our everyday lives, but also an ever more exciting area of academic research. The range of topics presented this year, including AI, education, simulation and narrative, demonstrates the rich opportunities for study, which are already being undertaken in this field.

I would like to express my gratitude to all those who have contributed to this event: firstly to those who have submitted papers, and will present them over the next couple of days and; of course, to Philippe Geril who is the driving force behind GAMEON; and to the programme committe who have reviewed papers, and contributed to organising this event.

Finally, I would like to thank and highlight our prestigious keynote and invited speakers, and also workshop organisers, who together have put together an excellent and diverse set of talks and activities, which I hope you will find both interesting and enjoyable.

Lincoln, September 2014

Patrick Dickinson
GAMEON'2014 General Conference Chair

CONTENTS

CONTENTS

SCIENTIFIC PROGRAMME

GAME HEURISTICS

VOX POPULI - A Case Study of User Comments on Contemporary Video Games in Relation to Video Game Heuristics

Björn Strååt, Harko Verhagen
Dept of Computer and Systems Sciences
Stockholm University
E-mail: [bjor-str|verhagen]@dsv.su.se

KEYWORDS

Video Games, User comments, User reviews, Game Design Evaluation, Heuristics

ABSTRACT

Video game reviews and user comments on released video games are common. Sales figures can often be related to the judgement from such comments. Before release, video game designers can refine their design by assessing it using different tools, such as heuristic evaluations or user tests. However, do heuristic lists designed for video games really measure the same issues that players react on? The answer might lie in the user comments online.

This paper presents a case study examines the relation between user comments and a set of heuristics designed to measure video gameplay. By conducting a linguistic analysis on video game user comments on three games of different user score, we matched common words and comments to a set of video game heuristics. Thus we could conclude which issues were most important to the users. The findings of this study can help designers and researchers to refine user studies.

INTRODUCTION

Sales figures are an undeniable measure of the success or failure of a video game production. The market outcome does not however tell us the underlying story. If we want to answer the "why", we will have to study the "how" – how the player community receives the product. Pre-release, one way to gauge the quality of a video game is expert evaluations, such as heuristic evaluations. Heuristic evaluations will provide game developers with information on which pre-defined issues they should address. Post release, the finished product will inevitably be subject to online discussion and assessment. Expert reviews and player comments matter. According to Livingston et al. (2011), players will down rate games that have poor reviews or user comments, and they seem to trust negative reviews more than positive.

In a study conducted parallel to this paper, we examined which issues were most frequent in modern character based video games, such as role playing games, action adventure games and first person shooters. To that end, we used a heuristic evaluation method designed to find as many issues as possible in the gameplay of the aforementioned game genres. The result showed that one of the most frequent issue was loss of control of the game. We now need to examine if this view is shared by regular users.

Metacritic.com is a website where third-party reviews are gathered and weighted into a so-called metacritic score. The metacritic score will provide insight into how a game measures up from the point of view of professional reviewers as well as regular users/players. Apart from the rating, metacritic.com also allows contributors to write personal, free text comments on the game. By analysing the contents of these comments we will have an indication of which factors the users find to be critical for a quality video game, and also whether these critical success factors are connected to the issues that the heuristic evaluation tools are designed to find and remedy.

This paper is organized as follows. First we present the heuristic evaluation tools in more detail and confine these to a relatively short list without ambiguity. Following this we present the games selected using the metacritic scores. Then we used linguistic analysis to find the most common aspects commented by the users. Finally, we connect these to the heuristic evaluation list and present our conclusions.

PREVIOUS WORK

In Strååt et al. (2013) we examined the most commonly available video game design heuristics (detailed below in section Game Design Heuristics), and more specifically which of these heuristics that would be appropriate to measure the video game world itself. The study (ibid) produced a list of heuristics with focus on gameplay evaluations. We called this list the *Net Heuristic List*. The list contains 14 heuristics, derived from Desurvire et al (2004) Desurvire and Wiberg (2009) and Pinelle et al (2008).

Game Design Heuristics

The following authors and articles were all considered when we worked with the Net Heuristic List .

Federoff (2002) was one of the first to publish a set of heuristics for games, in cooperation with a game developer team. Federoff collected heuristics from the 400 project list **?** and from literature on the subject. She compared her list with a real work place situation by following a team of game designers for a week. If the team used one of the heuristics on her list, she recorded a hit, and if they did something not on the list, she added it. This resulted in a list of 40 heuristics divided into three categories; Game Play, Game Mechanics and Game Interface.

Laitinen (2008) used Nielsen's heuristics Nielsen (2005) to evaluate a computer game under production. He found that the Nielsen heuristic method did find more problems than an evaluation performed without heuristics. However, he points out that a heuristic list tailored for games would be even more efficient to use.

Desurvire et al. (2004) created the HEP (Heuristic Evaluation for Playability), where they refined the evaluation method by categorizing 43 heuristics into Game Play, Game Story, Mechanics, and Usability. The HEP list carries a strong focus on game design issues and learnability of the game (Isbister & Shaffer, 2008, pp. 79-88), and is, according to Desurvire et al. (2004), widely used by game designers.

Pinelle et al. (2008) identified usability issues based on game reviews on the GameSpot.com website. They grouped similar problems into categories, and created heuristics describing how to avoid the problems. Their list contains ten heuristics.

Desurvire and Wiberg (2010) used a survey method on a game convention to define the PLAY heuristic list based on the HEP study. It is "... a more refined and updated list of Game Playability Principles (PLAY)..." The authors wanted a generalized foundation, modifiable for each game. The PLAY list contains 48 heuristics.

The Net Heuristic List

The Net Heuristic List was compiled from the compilation of design principles by Pinelle et al. (2008), the HEP list by Desurvire et al. (2004) and the PLAY list by Desurvire and Wiberg (2010).

- Pinelle et al
  1. Provide visual representations that are easy to interpret and that minimize the need for micromanagement. Visual representations, such as radar views, maps, icons, and avatars, are frequently used to convey information about the current status of the game. Visual representations should be designed so that they are easy to interpret, so that they minimize clutter and occlusion, and so that users can differentiate important elements from irrelevant elements. Further, representations should be designed to minimize the need for micromanagement, where users are forced to interactively search through the representation to find needed elements.
- HEP
  1. Make effects of the Artificial Intelligence (AI) clearly visible to the player by ensuring they are consistent with the player's reasonable expectations of the AI actor.
  2. The Player has a sense of control over their character and is able to use tactics and strategies.
  3. Provide consistency between the game elements and the overarching setting and story to suspend disbelief.
  4. The game transports the player into a level of personal involvement emotionally (e.g., scare, threat, thrill, reward, punishment) and viscerally (e.g., sounds of environment).
- PLAY
  1. The game is paced to apply pressure without frustrating the players. The difficulty level varies so the players experience greater challenges as they develop mastery
  2. Changes the player make in the game world are persistent and noticeable if they back-track to where they have been before
  3. There is an emotional connection between the player and the game world as well as with their "avatar."
  4. The game utilizes visceral, audio and visual content to further the players' immersion in the game
  5. Status score Indicators are seamless, obvious, available and do not interfere with game play
  6. Game provides feedback and reacts in a consistent, immediate, challenging and exciting way to the players' actions.
  7. The game gives rewards that immerse the player more deeply in the game by increasing their capabilities, capacity or, for example, expanding their ability to customize.
  8. Players should be given context sensitive help while playing so that they are not stuck and need to rely on a manual for help
  9. Game story encourages immersion (If game has story component).

About AntConc

AntConc, by Anthony (2012), is a freeware concordance- and text analysis tool written in Perl 5.8. It runs under any windows environment, Linux and Macintosh OS X (up to 10.9) and fairly easy to use. It was created by Dr Laurence Anthony at the Faculty of Science and Engineering at Waseda University, Japan. Our primary use of AntConc is word frequency listing and the concordance Key-Word-In-Context (KWIC) function. The KWIC lists any selected word in the actual sentences it was used, which in turn allows for word-by-word analysis.

SELECTED GAMES

For the purpose of this study, we chose three games with similar interaction, play-style and genre. As we wanted to see whether player comments were different based on the ranking of the game, we selected a high, a medium and a low ranked video game, according to metacritic.com. All games were released in 2013, and all user comments were gathered in may 2014.
Assassins Creed IV: Black Flag (ACIV). At the time of writing, ACIV the latest installation of the Assassins Creed franchise. The player can travel around and perform tasks (like assassinations)or follow to a pre-written storyline. The view is third - person / over the shoulder , that is, the player looks at the avatars back, and over the shoulder. The game-play encourages the player to use stealth techniques and alternative paths over running straight into the enemy. ACIV set in the world of Caribbean pirates, around the mid-1600s . ACIV got 90/100 on metacritic and thus is regarded as a highly ranked game.
DARK. DARK is a third - person / over the shoulder stealth game in a modern vampire environment. The player is encouraged to solve the task (infiltrate buildings et c.) with stealth. The game offers some aspects of character development, where the player can unlock special abilities as the game story progresses. The game was ranked 40/100 on metacritic and thus is viewed as a medium ranked game.
Ride to Hell : Retribution (RTHR) is a third person/over the shoulder games in outlaw motorcycle gang environment . The game missions vary between fighting enemies in close combat or fire fights and riding a motorcycle . There are some aspects character development, such as upgrades on both weapons and motorcycle. The game was ranked 14/100 on metacritic and thus becomes our low-ranked game

RESULT

For each game we gathered all player comments from metacritic.com. The player comments were then used to generate word lists in AntConc for each game. Word lists contain all the words of a text listed in descending order of frequency. The word lists were filtered through a common-words stop list, to remove common words such as I, is, am, if et c.
From the word lists, we picked the first five words according to the following categories:

- Words that were most frequently used to describe the players' feelings towards the game.
- Words that were most frequently used to describe the game or aspects of the game.
- Words that describe specific actions and activities within the game.

The Words we found were:

- ACIV:
  - Adjective/Adverb: *good, great, best, bad, repetitive*
  - Game descriptive: *story, graphics, storyline, control (controls), mechanics*
  - Gameplay specifics/activity: *combat, sailing, missions, battles, gameplay*
- DARK:
  - Adjective/Adverb: *good, bad, better, hard, fun*
  - Game descriptive: *story, graphics, animations (animation), difficulty, characters*
  - Gameplay specifics/activity: *stealth, powers, gameplay, kill, blood*
- RTHR:
  - Adjective/Adverb: *bad, worst, terrible, good, awful*
  - Game descriptive: *story, acting, sound, scenes, controls*
  - Gameplay specifics/activity: *gameplay, combat, gaming, enemies, effects*

We then generated(KWIC)concordance lists for each of the words. The KWIC list shows, row by row, the selected word in its context. That way it is possible to analyse how the writer intended for the word to be interpreted.
Finally, we compared the KWIC with Net Heuristic List to see if we could detect any heuristic matches in the players' comments. Each comment was classified as either positive or negative.
Table 1 shows the relative frequency of comments / heuristic, positive or negative, based on the player comments for the three games. The heuristics from the Net Heuristic List are abbreviated to category and number e.g. the second heuristic of HEP *The Player has a sense of control over their character and is able to use tactics and strategies* is called HEP 2 in Table 1.

Table 1: Results from the heuristic analysis of the KWIC list. Columns show positive and negative frequency of heuristics from the Net Heuristic list, per game.

Game:	**ACIV**		**DARK**		**RTHR**	
Heuristic	Pos %	Neg %	Pos %	Neg %	Pos %	Neg %
Pinelle et al 1	0.0	0.0	0.0	0.8	0.0	0.0
HEP 1	0.4	0.0	0.0	2.5	0.0	3.7
HEP 2	3.4	9.6	5.8	5.8	0.0	12.2
HEP 3	10.0	3.6	0.8	4.1	0.4	12.2
HEP 4	8.4	7.2	1.7	5.8	0.0	37.1
PLAY 1	1.4	9.2	2.5	8.3	0.0	3.3
PLAY 2	0.0	0.0	0.0	0.0	0.0	0.0
PLAY 3	2.6	0.8	0.8	2.5	0.0	0.4
PLAY 4	15.8	1.6	3.3	16.5	0.8	23.3
PLAY 5	0.0	0.0	0.8	0.8	0.0	0.0
PLAY 6	0.0	4.8	0.0	4.1	0.0	0.4
PLAY 7	0.4	0.4	4.1	5.0	0.4	0.4
PLAY 8	0.0	0.0	0.0	0.0	0.0	0.0
PLAY 9	13.2	6.2	5.0	12.4	0.0	9.0

CONCLUSION

As can be seen in table 1, HEP 4 - *The game transports the player into a level of personal involvement emotionally*, PLAY 4 *The game utilizes visceral, audio and visual content to further the players' immersion in the game* and PLAY 9 *Game story encourages immersion* received the highest frequency total. In the high rated game, the positive values are slightly higher than the negative, but for the medium and low rated games the negative values are very strong compared to the positives. This indicates that the game story, the visual content and the personal immersion are strongly considered by the players. Players seem to be disappointed when these parts of the video game is poorly implemented and, in some what less degree, impressed when it is implemented in a good way.

The authors were surprised to see that HEP 2 *The Player has a sense of control over their character and is able to use tactics and strategies* did not receive stronger values. Comments about controls were predominantly negative, which indicates that players mention game controls when they are poorly implemented. If the controls are good, the players don't seem to mention them much at all. It would be useful to know which heuristics share that notion; can a future study classify essential game design elements, that must not ever be poorly implemented, from other, less essential, game elements?

Some of the heuristics were not mentioned at all. This might be because those heuristics are rarely violated, or that video game designers find it easy to adhere to them. For future studies, measuring the need for each heuristic would be valuable when trying to create new and improved evaluation tools.

The findings mentioned above can easily be developed into survey or interview questions. If a video game developer conducts user studies with focus groups et c. they can benefit from our findings. For future studies, it would be interesting to base interviews on our findings when interviewing players who beta test games before release.

Players are not expert evaluators. They don't use heuristics to assess if a design is flawed or not, they just react to what they see. We believe that it is worth to take the players word into account even before creating a user study; what comments should we pay attention too, and where do players put their focus. As mentioned above, quite a few of the heuristics were not involved at all. Is this an indication that using a player focus group will give us parallel information to what we can gain from expert evaluations? Future studies should bring clarity to this.

Another aspect, which is not reflected in the heuristic analysis, is that many players commented on game and video graphic optimization. Comments on this aspect was often complaints; the users were disappointed that the game didn't run well on their computers etc. This is interesting, even if it is not a real finding of the study; none of the heuristic lists we have been working with considers technical issues, such as described by the users in this case. This might open for a new venue of studies, where game optimization/technical issue heuristics might be detailed.

It should be taken into consideration that the players who write on metacritic.com not necessarily are representative for the entire population of players. With thin in mind, the metacritic.com website is a common source of information for consumers who are interested in new games, not unlike the IMDB.com site for movies. Reviews from regular users are sometimes even more trusted than reviews from professional writers, as the regular users do not stand anything to gain from writing reviews. Whether word-of-mouth and online reviews have a decisive impact on the purchase decision or may be debatable, but we found the method of using player comments as base for analysis of player values both intriguing and valuable. It seems that if players may choose between good interaction and a good gaming experience, they seem to put more value on the game itself.

REFERENCES

Anthony L., 2012. *Laurence Anthony's Software.* web page. Retreived June 3 , 2014, from `http://www.antlab.sci.waseda.ac.jp/index.html`.

Desurvire H.; Caplan M.; and Toth J.A., 2004. *Using heuristics to evaluate the playability of games.* In *CHI'04 extended abstracts on Human factors in computing systems.* ACM, 1509–1512.

Desurvire H. and Wiberg C., 2010. *User Experience Design for Inexperienced Gamers: GAP–Game Approachability Principles.* In *Evaluating User Experience in Games*, Springer. 131–147.

Federoff M.A., 2002. *Heuristics and usability guidelines for the creation and evaluation of fun in video games.* Ph.D. thesis, Citeseer.

Laitinen S., 2008. *Usability and Playability Expert Evaluation.* In K. Isbister and N. Schaffer (Eds.), *Game usability: Advancing the player experience*, CRC Press. 91–94.

Livingston I.J.; Nacke L.E.; and Mandryk R.L., 2011. *The impact of negative game reviews and user comments on player experience.* In *ACM SIGGRAPH 2011 Game Papers.* ACM, 4.

Nielsen J., 2005. *Ten usability heuristics.*

Pinelle D.; Wong N.; and Stach T., 2008. *Heuristic evaluation for games: usability principles for video game design.* In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* ACM, 1453–1462.

Strååt B.; Johansson M.; and Warpefelt H., 2013. *Evaluating Game Heuristics for Measuring Player Experience.* In W.L. Anton Eliens (Ed.), *gameOn 2013.* Eurosis.

An Investigation into the use of a Gamified Revision Solution for Primary School Mathematics and its Experimental Comparison to Traditional Revision Methods

Tom Pendle
School of Computer Science
University of Lincoln, UK
Email: 11204467@students.lincoln.ac.uk

Keywords

Gamification, Educational Games, Applied Behavioural Analysis.

Abstract

This paper discusses the design and implementation of a gamified revision application based on a comprehensive literature review and in-depth analysis of previous case studies in the field of educational games. An experiment was devised to quantitatively compare the application with traditional paper-based revision methods, and an extensive set of results were collected and evaluated. It was concluded that not only can the use of gamified solutions provide superior motivation and engagement when used in an educational setting, but that they have the potential to provide through immediate feedback a system that encourages learning on a question-by-question basis, an aspect which cannot be rivalled by traditional techniques.

Introduction

Education is an ever-changing science, and new ways of creating motivation and improving learning are constantly being adopted in the classroom environment. At the same time, casual gaming has really picked up steam over the last few years, with more and more people of all ages playing games in their spare time, especially on mobile devices. In fact, figures show a year-on-year growth in mobile gaming revenue in the U.K of 63% last year (Casual Games Association, 2013, 4).

Taking these two main factors into account, it seems logical that if there was a way to utilise the captivating, motivating and encouraging nature of games in an educational setting, it would provide a rewarding and entertaining experience for pupils, but would it necessarily create a productive learning environment? That is the question that this paper aims to quantitatively answer, through the creation of an application which can be directly compared to traditional revision techniques.

Related Work

Borys and Laskowski (2013) used gamification in the form of a PBL system (Points, achievement Badges and Leader boards) to reward students for various tasks throughout a semester, like attending classes and completing tests, to determine whether the gamification had any impact on the student's performance or attendance levels. Interestingly, they discovered that whilst the gamified group initially performed better, their motivation was shown to decrease towards the end of the semester. They also implemented a leader board, as did Domínguez *et al.* (2013), and both case studies found that whilst the creation of competition between students did create motivation for some, it caused more to lose motivation as they began to realise they could not keep up with the leaders.

In terms of designing the revision application, the work of Linehan *et al.* (2011) was extremely valuable, as they presented a series of 'Practical, Appropriate, Empirically-Validated Guidelines for Designing Educational Games' (p1). They focused on the merits of Applied Behaviour Analysis, a psychological framework which has had multiple previous successful implementations, most notably by Saville *et al.* (2006) and Olympia *et al.* (1994). Typically used to assist the teaching of children with learning difficulties such as Autism, Linehan *et al.* (2011) highlight the features of ABA that make it a good foundation for the design of a successful educational game.

The Method

Using these case-studies and guidelines, as well as a wealth of other literature, a carefully selected set of requirements for the game design was produced (see Table 1). In addition, an initial focus group was run with the pupils that would be involved in the investigation, and data from that was also used for the requirements list.

ID	Requirement
1	The design of the application should be performed from the ground-up, i.e. not an adaptation of an existing game.
2	The application should feature a competitive aspect, or at least a means of determining score for use in a leader board, high score competition, etc.
3	The learning elements of the application must be intrinsic to gameplay; answering questions correctly must allow the player to progress.
4	The answering of questions should be met with immediate feedback, either positive reinforcement via a gain in score/ progression in the game or via positive/negative punishment via being unable to continue in the game or losing score previously gained respectively.
5	Questions presented to the player must be of a difficulty that adapts to the player's performance in answering previous questions.
6	The design of the game must follow the four stages of the ABA framework as closely as possible.
7	Graphics should be bright and colourful to enhance the visual enjoyment of the application.

Table 1: Proposed Requirements List for the gamified application.

ID	Requirement
8	The question generation system must be able to swiftly evaluate a player's ability and use it to encourage them to progress from that level upwards.
9	All four operations must be included, and questions must adhere to the national curriculum guidelines.
10	Multiple themed levels should be created, each with a different character for the player to control. The player should be able to choose from different themes at the start of the application.

Table 2: Further Requirements for the gamified application.

Using these requirements, two prototypes were developed, and the children were given the opportunity to play them and answer a questionnaire about them, which asked them to rate each prototype in terms of visual appeal, the amount of entertainment provided, and the game idea itself. They were also given space to provide any future improvements/changes that they would like to see.

Following the prototype feedback session, the top rated game overall was chosen to be carried forward, and three more requirements were proposed (See Table 2).

Using the prototype as a foundation, the final game was designed and developed and met all ten requirements. Three separate levels were created to give the children more variety and choice, each with a bright, vibrant and engaging colour scheme (See Figure 1).

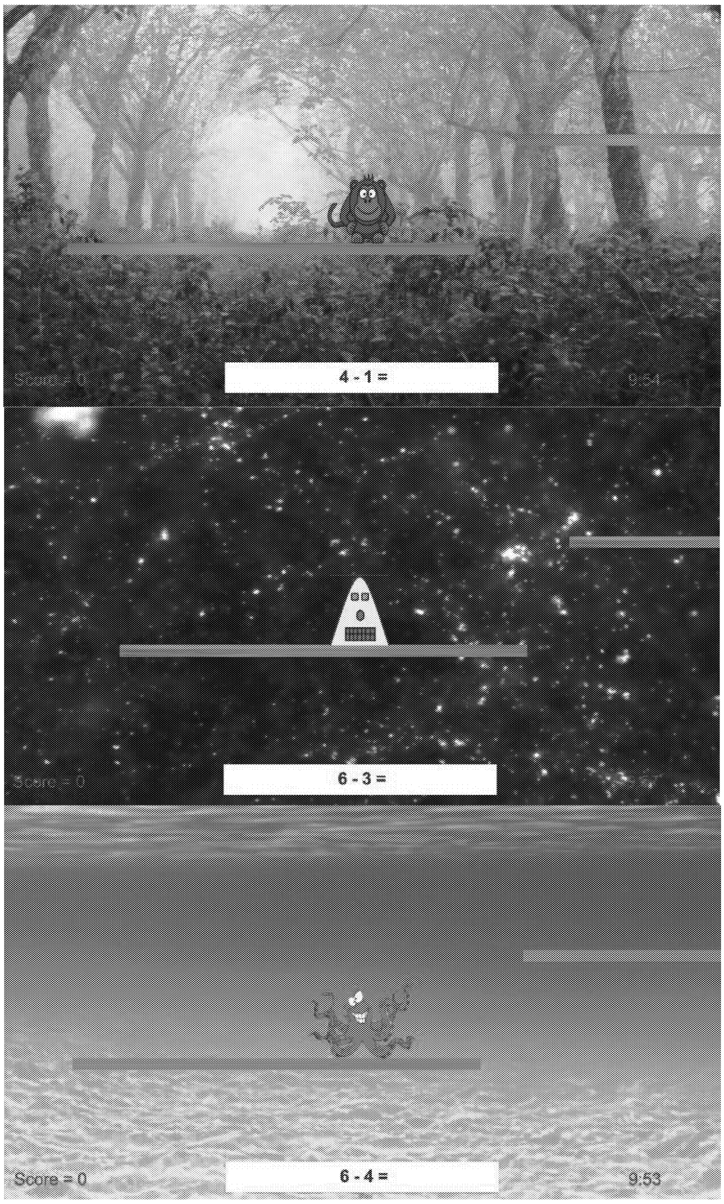


Figure 1: The three themed levels in the final application.

The main feature of the game in terms of creating a successful learning environment was the adaptive question generation system. The player would be assigned a difficulty level for addition, subtraction, multiplication and division independently, based on how many questions they got correct before getting one wrong (questions got more difficult after each successful answer and more simplistic after an incorrect answer). The system therefore was able to calculate the level of an individual player and encourage them to progress their learning at the appropriate difficulty. In addition to this, the player can only progress through the level if they answer a question correctly, if they answer incorrectly the game lets them know and they have time to answer more than once, therefore encouraging them to immediately realise that they have made a mistake and learn from it.

The final experiment was designed as follows:

- Test all of the students on the chosen topic to begin with. Use their scores to set them into 2 even groups in terms of mathematical ability.

- The 2 groups will perform a different activity for the duration of the testing period.

    - Group 1 – Gamified application (10 minutes per day) and reading
    - (10 minutes per day).

    - Group 2 – Traditional revision (10 minutes per day) and a non-educational game (10 minutes per day).

- Students will then be tested again at the end of the testing period, with the same test.

Results

Whilst a wealth of results were collected including scores for each person per day on both games and on the traditional paper revision, this paper is primarily concerned with the direct comparison between the improvement in performance on the initial and final tests for both groups. As such, the most important results are the test results, specifically in terms of total questions attempted by each pupil, total questions answered correctly and accuracy of answers provided.

Firstly, when comparing the total number of questions answered (see Table 3), the results show that members of the control group improved the number of questions they answered between the tests by 29%, whereas the group using the gamified app improved by only 14%.

Group:	A (Gamified)	B (Control)
Initial Questions Answered:	35.0	40.9
Final Questions Answered:	39.4	49.3
Difference:	4.4	8.4
As Percentage:	14%	29%

Table 3: Test results in terms of total questions attempted.

We can suggest that this is due to the nature of the traditional revision, a sheet full of practice questions, which could be daunting for pupils as they attempt to answer as many of the questions as possible; they feel more urgency to answer questions than the pupils using the application who only see one question at a time.

As a way of confirming this theory, we can look at the accuracy results for the two groups (see Table 4). That is, the percentage of questions answered that were answered correctly.

Group:	A (Gamified)	B (Control)
Initial Accuracy:	85%	85%
Final Accuracy:	87%	80%
Difference:	2%	-5%

Table 4: Test results in terms of accuracy of questions answered.

Looking at these results, we can see that whilst the control group answered more questions, they answered them far less accurately. This shows that the traditional revision promotes speed rather than accuracy. In this respect, we can suggest that because the application provides immediate feedback to the player as to whether they have answered a question correctly or incorrectly, it encourages them to immediately learn from their mistakes and therefore maintain a higher level of accuracy.

Finally, we can compare the groups purely in terms of their scores on the tests (see Table 5). These results show great promise for the application, as in our experiment the application actually improved pupil's performance by more than the traditional revision.

Group:	A (Gamified)	B (Control)
Initial Score:	29.8	34.4
Final Score:	34.5	39.6
Difference:	4.8	5.1
As Percentage:	19%	17%

Table 5: Test results in terms of scores.

This shows that although the pupils in the control group answered almost 20% more questions in the final test, they still performed worse because they had a lesser degree of accuracy.

Conclusion

Due to the ability of a well-designed revision application (such as the one created for this project) to provide immediate feedback to its user upon the answering of a question, an effect of 'run-time learning' can be engineered. Via a simple process of reinforcement, the user can be challenged at the exact moment when they make a mistake. This makes them stop, assess what they did wrong, and try again, all in the space of a couple of seconds after they provide their answer. This process cannot be replicated by traditional revision, and it suffers for it. With traditional revision methods, for example practise questions, feedback is only usually provided after all questions have been answered, which creates a long, drawn-out feedback cycle which is not efficient for learning as part of revision. Thus, this project can infer the conclusion that the gamification of revision, when approached systematically and scientifically, has the potential to encourage a higher degree of accuracy and fluency in its users than traditional revision methods.

Finally, with respect to how the findings of this research can be viewed in the wider context of using games and game design theory as an educational device, we can be confident that the results of this study demonstrate the real potential of educational games and gamified educational tools, especially in terms of revision. This study has found that at the very least the gamification of traditional methods can lead to solutions that are equally effective, but more interesting and motivational for the users. This study can make one final conclusion then, that if the educational world is ready to allow gamification in, gamification is ready to show its true prowess as a valid tool for providing motivating, encouraging, and innovative ways to enhance the education of children. It will take more than one study in one classroom to convince sceptics, but for now, this study has already convinced one classroom and its pupils that this new technology may one day be present in classrooms everywhere. The class involved intend to continue using the application as part of their mathematics revision, and given the hard time games have notoriously faced in breaking into the classroom environment, this has to be taken as one of the greatest successes of this project.

References

Borys, M., and Laskowski, M. (2013). Implementing Game Elements into Didactic Process: A Case Study. *Make Learn International Conference 2013*, Zadar, Croatia, June 19-21. [unpublished]. 822-823. Available from http://www.toknowpress.net/ISBN/978-961-6914-02-4/papers/ML13-326.pdf [Accessed 15/04/14].

Casual Games Association (2013) *Smartphone and Tablet Gaming 2013: Games Market Sector Report.* [online] Available from http://www.proelios.com/wp-content/uploads/2013/11/CGA-Smartphones-and-Tablets-2013-Games-Market-Sector-Report_V1.pdf [Accessed 07/04/14].

Domínguez, A., Saenz-de-Navarrete, J., de-Marcos, L., Fernández-Sanz, L., Pagés, C., and Martínez-Herráiz, J. (2013) Gamifying learning experiences: Practical implications and outcomes. *Computers and Education* (63) 380-392. Available from http://ac.els-cdn.com/S0360131513000031/1-s2.0-S0360131513000031-main.pdf?_tid=8626fdd0-c62e-11e3-abdb-00000aab0f26&acdnat=1397739067_cacaa8add031fc3e4c74e26276c019b4 [Accessed 17/04/14].

Linehan, C., Kirman, B., Lawson, S., and Chan, G. (2011). Practical, appropriate, empirically-validated guidelines for designing educational games. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* Vancouver, BC, May 7 – 12, 1979-1988. Available from http://eprints.lincoln.ac.uk/4475/1/linehan_et_al_ABA_games_authors_version.pdf [Accessed 14/04/14].

Saville, B. K., Zinn, T. E., Neef, N. A., Van Norman, N., and Ferreri, S. J. (2006) A comparison of interteaching and lecture in the college classroom. *Journal of applied behavior analysis*, 39 (1), 49–61. Available from http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1389610/pdf/jaba-39-01-49.pdf [Accessed 23/04/14].

Olympia, D. E., Sheridan, S. M., Jenson, W. R., and Andrews, D. (1994) Using student-managed interventions to increase homework completion and accuracy. *Journal of applied behavior analysis,* 27 (1), 85 – 99. Available from http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1297779/pdf/jaba00007-0087.pdf [Accessed 23/04/14].

Author Biography

TOM PENDLE was born in York, England and is currently a 4th Year Games Computing MComp student studying at the University of Lincoln. His academic interests lie in the areas of serious games and gamification, specifically the use of games and game design theory as an educational device.

Email Address: 11204467@students.lincoln.ac.uk

Youtube: https://www.youtube.com/user/TomPendleGameDesign

PLAYING WITH SCIENCE: GAMISED ASPECTS OF GAMIFICATION FOUND ON THE ONLINE CITIZEN SCIENCE PROJECT - ZOONIVERSE

Anita Greenhill, Kate Holmes, Manchester Business School, Booth Street West Manchester, M15 6PB.	Chris Lintott, Brooke Simmons, Department of Physics, Denys Wilkinson Building, Keble Road, Oxford, OX1 3RH	Karen Masters, Joe Cox, University of Portsmouth, Portland Street, Portsmouth, P01 3DE	Gary Graham, University of Leeds, Moorland Road, Leeds, West Yorkshire, LS6 1AN

KEYWORDS
Gamise, gamification, play, citizen science, Zooniverse

ABSTRACT
This paper examines incidents of play, socialisation, fun and amusement to consider how these forms of social interaction relate to the serious gaming elements of the citizen science platform. Through an ethnographic study we reveal how participants of citizen science projects demonstrate aspects of 'Gamised' behaviour. 'Gamised' behaviour is defined as user generated play in a digital platform and contrasts to incidents of 'gamification' where a platform designer purposely embeds games into a computer platform. The paper therefore examines incidents of play, socialisation, fun and amusement and compares them with the serious gaming elements of the citizen science platform.

INTRODUCTION

This study explores aspects of gamification when interacting with fellow citizens in online citizen science projects. As a means of situating the work we define user generated play in a digital platform as a 'Gamised' activity. In particular we examine incidents of play, socialisation, fun and amusement and consider these forms of social interaction in relation to the serious gaming elements of the citizen science platform. Through an ethnographic study we reveal how participants of citizen science projects demonstrate aspects of gamification when interacting amongst online platforms and forums. The key question asked is what is the relationship between knowledge building - the overall objective of the site design and 'play' as a means for building interest and on-going commitment from the users. The specific platform under exploration is the crowdsourcing citizen science platform, Zooniverse.org; the specific projects discussed including Galaxy Zoo and Snapshot Serengeti. Other networked citizen science projects are drawn upon to broaden applied examples of the play and gamification.

Citizen science is the name given to scientific investigations conducted by amateur or non-professional scientists. Usually conducted by volunteers, citizen science has been implemented across a variety of scientific disciplines in order to provide a solution towards the demands of conducting data rich scientific research amongst professional scientists; some of these demands include time, material costs and labour incurred, particularly for tasks which are not suitable for analysis using computer algorithms (Silvertown, 2009). Citizen science provides opportunities for people to collectively contribute to investigating large data sets therefore easing the demands that would otherwise slow the research process (Raddick et al. 2009). Many examples of citizen science projects in action can be found on the Zooniverse.org platform, which utilises crowdsourcing to engage and bring together citizen scientists to actively contribute towards investigations and research.

Crowdsourcing is a term originally coined by Howe (2006) to describe the activity of a large group of people, usually an online community, to collectively contribute towards content, funding, a project or service. Examples of some of the more than 20 projects found at Zooniverse.org are Galaxy Zoo (www.galaxyzoo.org), Old Weather (www.oldweather.org) and Snapshot Serengeti (www.snapshotserengeti.org). Galaxy Zoo asks participants to classify galaxies appearing in images taken by professional astronomical facilities, such as the Hubble Space Telescope and the Sloan Digital Sky Survey. The interface of the website can be considered to be fairly self-explanatory, with an image of the galaxy to be classified on the right and multiple choice questions about the features and characteristics of the galaxy on the left. The questions are purposefully kept simple and do not require specialised scientific knowledge in order for the participant to engage with the project. The Galaxy Zoo science team uses the information to search for rare types of galaxies or analyse the galaxy population statistically. Snapshot Serengeti displays images of animals gathered from camera traps at the Serengeti National Park in Tanzania. The purpose is to study how a variety of species interact with each other and how they are distributed across the landscape. This also relies on an interface similar to that of Galaxy Zoo, asking the participant a series of questions on the animals they can see on the photo. Snapshot Serengeti also has a support mechanism in place to assist participants if they are having trouble identifying the animals.

Contemporary interest in crowdsourcing, citizen science and online gaming all have one thing in common, they are enabled through the networked capacity of digitized human interaction. All three also tread a fine balance when it comes to keeping their specific community of users coming back and continuing to contribute to a final objective that is predefine by computer and platform developers. This paper considers this balancing and therefore the significance of the use of gamification to build and maintain motivation in terms of an online citizen science platform. Furthermore we examine 'Gamised' activity as a means to explore user generated play and gaming as opposed to design embedded games and playing on a web platform. The site we examine is the Zooniverse.org with its stated objective "to produce projects that use the efforts and ability of volunteers to help scientists and researchers deal with the flood of data that confronts them" (Zooniverse, 2014). The Zooniverse currently has a registered over one million "Zooites", who contribute to the serious objective of categorizing and processing data so it can then be applied to wider research projects and build towards scientific knowledge. The Zooniverse in its entirety not only contains a suite of citizen science projects, but also "Talk" platforms from which further interaction can talk place, a schools project element for within the educations system (www.zooteach.org), and links to publications detailing some of the discoveries and subsequent discussion that emerge from the processed data. The question of serious or playful gaming swims just under the surface of the contribution pool that the citizen scientists provide. Firstly there is a balance between 'real' and 'citizen' science; and secondly there is a balance between 'work' and 'play'.

Play and gamification

Abt (1987) argues that for something to be considered a game it must possess certain qualities, these are outlined as an activity, an aspect of decision-making, an objective, and rules to limit the structure and activity of the game. Although this definition may be limited, games as a form of entertainment evolve, progress and differ in meaning depending on the context. For example, serious gaming adapts the role of play by changing the role of the game. Serious gaming describes games for uses other than solely entertainment purposes (Michael and Chen, 2006); the use of serious gaming could be applied to education, planning, health and even scientific exploration. Serious games are platforms that have been specifically designed to be a game in order to achieve a serious output (Connolly et al, 2012). For example, scientists at the University of Exeter have developed a game for citizen scientists to play to collect data on camouflage called 'Where is that Nightjar?' (Project Nightjar, 2014), which times participants as they attempt to find camouflaged birds. Another example is 'Foldit', which is an online puzzle game developed by The University of Washington, where players 'fold' the structure of virtual protein in order to discover new solutions that can be applied to scientific investigations (Khatib et al. 2009). Serious gaming differs from Gamification, which can be defined as "applying game-related ideas to non-game processes, issues and situations" (Shea, 2014 p. 4). This is supported by Deterding et al. (2011) who claims gamification is adding a sense of play and game design to something that is not a game. Therefore in regards to this study, it is important to make clear that the original intention of the Zooniverse interface was not designed as a serious game but as a tool to perform citizen science. This paper focuses on how users apply game like aspects to projects and tools within the Zooniverse as an online citizen science platform (examples of this can be found under section 'Citizen science, play and gamification' to illustrate this point further).

Burke (1971, p. 33) reminds us that the 'Random House Dictionary' lists fifty-three different definitions for "play' and thirty-nine for "work". Furthermore that if both are defined in economic terms an unsatisfactory dichotomy emerges, where work is whatever you get paid for and play is everything else. Burke concludes that the only way to define either 'play' or 'work' is to find formulations which include as many of their usual uses as possible, especially the most common ones, under as few as possible clear, consistent concepts. To define either is a lengthy process. Gray (2008) on the other hand emphasises the importance of play for societies and in particular for children's development, physically, intellectually, emotionally, socially, and morally. Furthermore, he stresses that it is important to have the freedom to quit in relation to play. In contrast to Burke (1971), Gray (2008) states there are five key characteristics to play: firstly play is self chosen and self directed; secondly play is activity in which means are more valued than ends; thirdly play has structure or rules, which are not dedicated to physical necessity but emanate from the minds of the player; fourthly play is imaginative, non-literal, mentally removed in some what from “real” or “seriousness” of life; and lastly play involves an active, alert, but non-stressed frame of mind. For Gray (2008, p. 2) “play is actively conducted primarily for its own sake”, believing that all characteristics of play have to do with motivation and attitude.

There is a growing body of research examining the blurring of work and play. Yee (2005) for example discusses the blurring between videogames and work play boundaries, expressing that the user now can average 20 hours per week playing video games to a point where they describe their participation in terms of obligation and tedium, which are the antithesis of fun and enjoyment. Extending these blurring concepts are the considerations presented by Zhang & Fung (2014) who question the State’s role (in their case China) as an influencing force in consumer behaviour of both online game player in terms of labour being exploited and entrepreneurial invention as a secondary industry. A well considered and very reflexive paper that considers the economic significance of mass participation in games by Bundy (1992) also explores the notion of play as a leisure activity describing it as “a transaction or activity in which we engage only because we want to, not because we feel we must” (p.217). However it is determined by the player whether something can be seen as a game and the circumstances in which the activity is carried out because “without playfulness, all activities become work” (Bundy, 1992, p.217). The relationship between work and play has been explored by Greenhill and Fletcher (2013), who argue that as the difference between real and digital environments are becoming less apparent so are the differences between work and play. This argument is supported by Anderson et al. (2013) who explore how some online gaming platforms may be seen to subtly influence the player into enjoying the work undertaken. This is achieved by the careful use of in-game mechanics such as the timing and layering of reward systems. Reward systems can include badges to encourage the player to continue playing amongst simulated ecosystems. The result is that “work and play are starting to become indistinguishable from each other” (Yee, 2005, p.70). According to Danbridge (1986) the value of organisations is to blur the boundaries between work and play to enable workers to experience the benefits of ‘flow’ associated with play activities. Danbridge argues that by de-emphasising the dichotomy between work and play within the workplace, workers are then able to draw upon ‘fun’ and ‘enjoyment’ into the ceremonies of work. This de-emphasizing therefore enables the incorporation of elements of playfulness into their daily working lives and improves job satisfaction. Danbridge (1986) goes on to explain that the work/ play dichotomy is enforced when we describe play as a ‘process’ and work as ‘the end product’ (p. 159).

With these thoughts in mind it might be argued that the seemingly blurred relationship between work and play may be applied in a similar manner to the participation and contributions towards citizen science. When a dichotomy is established between ‘the process’ of data categorization and science as ‘end product’, could an understanding of play as categorization and work as science emerge? If the definition is carried through in terms of understanding citizen science participation in an online crowdsourcing platform, a lowering of enjoyment must ensue and the sense of fun and enjoyment diminish. The blurring of the relationship between work and play is therefore a key consideration to this paper. The relationship and understanding of gamification as play should therefore equally not be considered as a dichotomy to that of the work towards the science outcome. We will return to these arguments later in the paper when we explore the complex relationship of seriousness and play elements to achieve as an end product. It is also important for our case to ask whether seriousness and play can coexist; i.e a serious outcome with play as a form of motivation? Or does play in the scientific ‘end product’ somehow reduce or lessen the validity of the outcome if play stimulates the activity? For the Zooniverse we ask ‘Are people playing when they are categorising on the Zooniverse? Could the Zooniverse legitimately use online gaming to build ‘real’ science? And can the online game be disentangled from the serious science outcomes?’

Serious ‘Real’ Science

Chalmers (1976) states;

> “Science is highly esteemed. Apparently it is a widely held belief that there is something special about science and its methods. The naming of some claim or line of reasoning or piece of research "scientific" is done in a way that is intended to imply some kind of merit or special kind of reliability. But what, if anything, is so special about science?” (1976, p. i)

Ziman (2004) tells us science amongst other things is a social institution and that it produces quantities knowledge. He also states, “The peculiarity of science is that knowledge as such is deemed to be its principal product and service. This not only shapes its internal structure and its place in society. It also strongly colours the type of knowledge that it actually produces" (p. 5). Ziman (2004) questions whether the confidence and respect once held for genuine scientific enquiry may be becoming less apparent and acknowledges that not all scientific practice may be revered as faultless. However, Ziman also argues that there are defining aspects to real science in that it must be rigorous in intent and command respect in regards to the methods it employs by legitimately making a contribution towards a body of knowledge. Jackson (2013) considers the practice of citizen science to lack in this approach, that it may be considered just a novelty or trend with too much emphasis on the gaming aspects making science appear trivial. Concluding, it may be precarious to rely on its findings without fully knowing the potential impact it may have.

Others highlight the issues of accessibility in regards to citizen science in that it is only available to people with the knowledge and access to the technological platforms such as computers and access to the Internet (Mathieson, 2013). According to Bonney et al. (2009) lack of specialist knowledge or misclassification may result in errors within the data produced, although within Galaxy Zoo great lengths are taken to ensure the quality of the galaxy classifications, which are released for use in the wider scientific community (Lintott et al. 2011, Willett et al. 2013)

Although the practice of citizen science may not be entirely without issue, there is an element of control in regards to the data produced by the participants as they work within a system created with them in mind. For example, the interface on Galaxy Zoo provides limited options as to how galaxies can be classified, and guides the participants therefore controlling the margin of error. Any unusual activity is then filtered and flagged for further review. For Cohn (2008), the issue of whether citizen scientists can conduct real research is questioned through the reliability of the data recorded/produced, concluding that work achieved through these methods can contribute towards scientific studies overall by helping to gather data, develop guidelines and save resources. Citizen scientists have also been known to serendipitously make discoveries as people involved in a classification task continue to look for anything unusual and it also provides a multifaceted approach to exploring the data (Christian et al. 2012). This indicates that the practise of citizen science can legitimately make a contribution towards real science.

Citizen Science, play and gamification

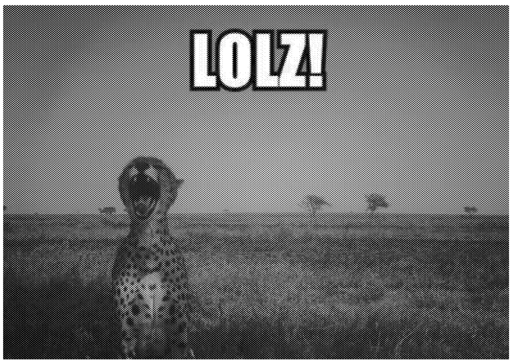
Eveleigh et al. (2011) explore the concept of gamification in regards to Zooniverse project Old Weather, a project that asks users to assist in the transcription of handwritten ship logs from the 19th century. The team behind the project realised that many users found the task repetitive and the handwriting difficult to read, so they developed a ranking system to encourage participation and sustain volunteer's engagement. This system ranked volunteers by contribution from Cadet to Captain allowing for the "top transcribers in each ship to compete for captain" (p.1). The study discovered that some of volunteers enjoyed an aspect of gamification within citizen science for reasons such as to validate their work, motivate contributions and track personal progress. But they also found this particular system may have been too competitive and perhaps went against the ethos of collective achievement that encourages many to be part of a crowdsourced project in the first place.

Across a variety of online citizen science platforms multiple examples of play, socialisation, fun and amusement can be found which may be surprising considering the practice of science is usually regarded to be a serious discipline (van DijK, 2011). As previously stated, this study focuses on Galaxy Zoo and Snapshot Serengeti. These particular projects from the Zooniverse have been chosen to be studied for a variety of reasons. Galaxy Zoo was the first project of the Zooniverse and has since evolved over the years to meet the needs of the contributors. Galaxy Zoo has a strong and loyal following that consistently dedicate their time and labour to the project and as a result it has developed a rich history. Another reason why we decided to study the Galaxy Zoo project is because it focuses on a grand subject matter and by exploring the workings of this unique project it provides us with an exciting opportunity to be part of that. Much of the same reasoning can be applied as to why we decided to explore Snapshot Serengeti, as it is also allows us to be part of another unique project and allows us to view extraordinary photography of rare animals. It is something we consider to be fun and entertaining. We also feel that Snapshot Serengeti supports a noble cause by collecting information about wild animals in order to help protect them.
When exploring the surface of the Zooniverse it may at first appear to be a straightforward platform to conduct citizen science. However, due to the legions of committed contributors residing within an active and developed community, as well as the opportunities provided for rich social interactions throughout it's forums, blogs and other examples of social media, it appears that there is far more hidden within the Zooniverse than an initial view may imply. It has been highlighted that some members have found the classification systems

within online citizen science projects to be dull and repetitive (Prestopnik and Crowston, 2011), so some members of the Zooniverse have been reported to invent their own games within the classification process to help motivate themselves and other users making it more interesting. For example, some members attempt to find and collect photos of all forty-eight animals listed within Snapshot Serengeti, while other participants have even been reported to attempt to find the rare Zorilla in order to complete their collection (Daily.zooniverse.org, 2014).

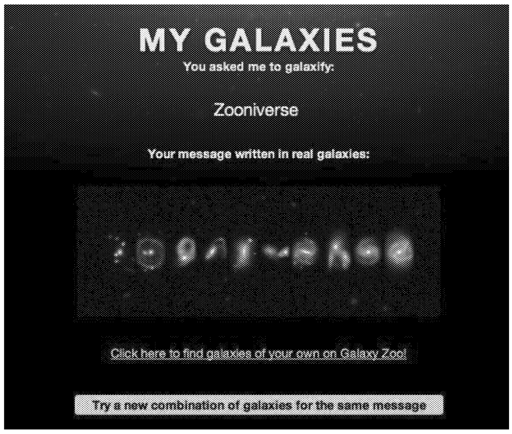
Photo of Zorilla - Snapshot Sunday (2014)

The team behind Snapshot Serengeti realised the popularity surrounding the element of play for their community. To encourage play, they drew on photographs that existed on the site and built a meme generator to allow contributors to create their own memes. The meme generator was designed to attract interest and encourage new users towards contributing to the project. The image below is meme which has been created by a participant portraying a photo of a leopard looking like it is laughing with the caption "LOLZ".



Snapshot Serengeti blog (2013)

Another example of play which has been adapted from the photos from The Zooniverse is a website called MyGalaxies (www.mygalaxies.co.uk), created by a Galaxy Zoo team scientist, and which allows participants to create messages from photos of galaxies that resemble letters. Below is an example created through the website spelling the word "Zooniverse" through images of galaxies:



www.mygalaxies.co.uk (2014)

A further example of play using this format can be found in Pedbost et al. (2009) who for an April fool prank claimed that a new galaxy cluster had been discovered which spells "So long and thanks for all the fish". These examples provide citizen scientists with opportunities to have fun and be creative with the images and data collected through the platform. They demonstrate how citizen scientists actively engage in play and gamification when participating within the Zooniverse. This study further explores aspects of gamification throughout online citizen science projects and discusses the relationship and impact this may have on real science.

RESEARCH DESIGN AND METHODOLOGY

To gain an in depth understanding of aspects of gamification within online citizen science projects we will be using multiple qualitative methods in order to collect data with an interpretivist approach. An ethnographic study is currently being carried out where the researchers are actively participating within the citizen science platform Zooniverse.org and keeping a daily diary of findings. This will also include content analysis of examples found across citizen science platforms such as games created specifically for the citizen science projects, blog posts, discussions on forums and other examples of play found within this domain. These examples will be used to further illustrate points and support arguments throughout the study.

RESEARCH AGENDA

We suggest such a research agenda shaped by the points raised above would have the following objectives:

1. To provide systematic empirical evidence concerning serious gaming and the relation play has

when motivating to a serious networked outcome and a critical examination of extant diverse secondary data

2. To develop an understanding of the processes of social interaction in the gaming context via:

i. In-depth interviews regarding the kinds of play citizen scientists and online contributors undertake in relation to other serious' forms of scientific categorizations in similar online platforms.
ii. Narratives of (play) self motivation in relation to online gaming communities, science communities and other organizational communities via a series of visual ethnographies.
iii. Analysis of the resulting data sets in order to assess the importance of issues of fun, entertainment, satisfaction, motivation, volunteering continuity, pride in contribution, a sense of connection with other citizen scientists.
iv. Exploration of the resulting data set as to how these issues intersect with other demands on citizen scientists time and long term commitment and motivations.

3. To provide insights of practical and policy relevance to core social and scientific issues by communicating our assessment of the significance of citizen science and gaming in online environments as those agencies and business draw on the online platform to process manageable form of scientific data; and by providing evidence based recommendations which can be used to inform the development of strategies for organizing disparate citizen scientists.

CONCLUSION

Within the paper we have presented a number of examples of play, socialisation, fun and amusement that is happening in the Zooniverse citizen science platform. These examples are predominantly examples of 'Gamised' activity. It is also clear from the examples presented that play, amusement and entertainment, as a form of social interaction, is important for some of the participants of the citizen science platform. It is also clear that the question of serious or playful gaming swims just under the surface of the contribution pool that the citizen scientists provide within this serious platform. There is a need to consider the balance between 'real' and 'citizen' science; and secondly there is need to further consider the balance between 'work' and 'play' when attempting to design for a serious objective within an online platform. The research proposed will therefore help to provide clearer answers to what is the best way to achieve a balance between 'play' and 'work' both from the users and developers perspectives. It will also build a robust body of work to further understand the importance of 'Gamised' and or 'Gamification' for citizen science and others considering using crowdsourcing a platform for online engagement.

ACKNOWLEDGEMENTS
We would like to acknowledge the significant contribution of EPSRC and NEMODE who are currently funding the Volcrowe research and Zooniverse.org their co-operation.

REFERENCES

Abt, C. (1987) *Serious Games*. Boston, USA: University Press of America.

Anderson, A. Huttenlocher, D. Kleinberg, J. and Leskovec, J. (2013) 'Steering User Behaviour with Badges' *Proceedings of 22nd International Conference on World Wide Web*, pp. 95 -106.

Bonney, R. Cooper, C. Dickinson, J. Kelley, S. Phillips, T. Rosenberg, K. and Shirk, J. (2009) 'Citizen Science: A Developing Tool for Expanding Science Knowledge and Scientific Literacy' *Bioscience*, Vol. 59, No. 11, pp. 977 - 984.

Bundy, A. (1992) 'Assessment of Play and Leisure: Delineation of the Problem'. *The American Journal of Occupational Therapy*, Vol. 47, pp. 217 – 222.

Burke, R. (1971) '"Work" and "Play"', *Ethics*, Vol. 82. No. 1 (Oct 1971) pp. 33 - 47.

Chalmers, A.F. (1999) *What is this Thing called Science?* University of Queensland Press: Australia.

Cohn, J. (2008) 'Citizen Science: Can Volunteers Do Real Research?' *Bioscience*, Vol. 58, No. 3.

Christian, C. Lintott, C. Smith, A. Fortson, C. and Bamford, S. (2012) 'Citizen Science: Contributions to Astronomy Research' *Organisations, People and Strategies in Astronomy*, Vol. 1, pp. 183 -197.

Connolly, T. Boyle, E. MacArthur, E. Hainey, T. and Boyle, J. (2012) 'A Systematic Literature Review of Empirical Evidence in Computer Games and Serious Games' *Computers and Education*, Vol. 59, pp. 661 - 686.

Danbridge, T.C. (1986) 'Ceremony as an Integration of Work and Play', *Organisation Studies*, Vol. 7, No. 2, pp. 159 - 170.

Deterding, S. Dixon, D. Khaled, R. Lennart, N. (2011) 'From Game Design Elements to Gamefulness: Defining "Gamification"' *Proceedings of Mindtrek '11*. September 28 - 30, 2011. Tampere, Finland.

Eveleigh, A. Jennet, C. Lynn, S. Cox, A. (2013) '"I want to be a Captain! I want to be a Captain!": Gamification in the Old Weather Citizen Science Project' *Proceedings of Gamification '13*. October 2-4, 2013. Stratford, Canada.

Greenbaum, J. (1996) 'Back to Labour: Returning to Labour Process in the Study of Work. *Proceedings of the Computer Supported Work Conference 96,* pp. 229 - 237.

Greenhill, A. and Fletcher, G. (2013) 'Labouring Online: Are There "New" Labor Processes in Virtual Game Worlds?' *Journal of Association for Information Systems*, Vol. 14, No. 11, pp. 672 - 693.

Khatib, F. Cooper, S. Tyka, M. Xu, K. Makedon, I. Popovic, Z. Baker, D. and Foldit Players (2011) ' Algorythm Discovery by Protein Folding Game Players' *PNAS*, Vol. 100, No. 47, pp. 18949 - 18953.

Lintott, C. Schawinski, K. Slosar, A. et al. (2008) 'Galaxy Zoo: Morphologies Derived from Visual Inspection of Galaxies from the Sloan Digital Sky Survey', *MNRAS*, Vol. 389, Issue. 3, pp. 1179 – 1189.

McGonigal, J. (2011) *Reality is Broken.* London, Great Britain: Jonathan Cape.
Michael, D. and Chen, S. (2006) *Serious Games: Games that educate, Train and Inform*. Ohio, USA: Cengage Learning.

Pedbost, M. Pomalgu, T. and The Galaxy Zoo Team (2009) 'Galaxy Zoo: An Unusual New Class of Galaxy Cluster', *MNRAS.*

Prestopnik, N. and Crowston, K. (2011) 'Gaming for (Citizen Science) Science: Exploring Motivations and Data Quality in the Context of Crowdsourced Science through the Design and Evaluation of a Social-Computational System' Proceedings of 2011 IEEE Conference, pp. 28 - 33.

Raddick, M. Bracey, G. Carney, K. Gyuk, G. Borne, K. Wallin, J. Jacoby, S. (2009) 'Citizen Science: Status and Research Directions for the Coming Decade' *Astro 2010 Decadal Survey Position Paper.*

Shea, T. (2014) *Gamification: Using Gaming Technology for Achieving Goals*. New York: The Publishing Group.

Silvertown, J. (2009) 'A New Dawn for Citizen Science' *Trends in Ecology and Evolution*, Vol. 29, Issue 9, pp. 467- 471 (last accessed 4 February 2014)

Van Dijk, E. (2011) 'Portraying Real Science in Science Communication' *Science Education*, Vol. 95, Issue. 6, pp. 1086 - 1100.

Willett, K. Lintott, C. Bamford, S. et al. (2013) 'Galaxy Zoo 2: Detailed Morphological Classifications for 304 122 Galaxies from the Sloan Digital Sky Survey', *MNRAS*, Vol. 435, pp. 2835 – 2860.

Yee, N. (2005) 'The Labor of Fun: How Video Games Blur the Boundaries of Work and Play' *Games and Culture*. Vol. 1, No. 1, pp. 68 – 71.

Zhang, L. and Fung, A. (2014) 'Working as Playing? Consumer Labor, Guild and the Secondary Industry of Online Gaming in China', *New Media and Society*, Vol. 16, No. 1, pp. 38 - 54

Ziman, J. (2004) *Real Science: What it is, and what it means*. Cambridge: Cambridge University Press.

WEB REFERENCES

Daily.zooniverse.org (2014) http://daily.zooniverse.org/2014/02/09/snapshot-sunday-what-do-you-get-if-you-cross-a-zebra-a-gorilla/ (last accessed 26 March 2014)

GalaxyZoo (2014) www.galaxyzoo.org (accessed 3 April 2014)
Gray, P. (2008) The Value of Play 1: the definition of play gives insights, Psychology Today, http://psychologytoday.com/blog/freedom-learn/200811/the-value-play-i-the-definition-play-gives-insights (last accessed 10th March 2014)

Foldit (2014) https://fold.it/portal/ (last accessed 1 April 2014)

Jackson, N. (2013) Citizen Science: Proper Science? http://www.britishscienceassociation.org/people-science-magazine/september-2013/regular-items-september/citizen-science-proper-science (last accessed 28 March 2014)

Mathieson, K. (2013) 'Citizen Science for Schools: What Teachers Need' http://www.britishscienceassociation.org/blog/citizen-science-schools-what-teachers-need (last accessed 29 March 2014)

MyGalaxies (2014) http://www.mygalaxies.co.uk/ (last accessed 26 March 2014)

Old Weather (2014) http://www.oldweather.org/ (last accessed 3 April 2014)

Project Nightjar (2014) http://nightjar.exeter.ac.uk/story/nightjar_game (last accessed 29 March 2014)

Save the Memes (2014) http://www.savethememes.org/ (last accessed 3 April 2014)

Snapshot Serengeti (2014) www.snapshotserengeti.org/ (last accessed 3 April 2014)

Snapshot Serengeti blog (2013) http://blog.snapshotserengeti.org/2013/07/31/save-the-memes/ (last accessed 28 March 2014)

Zooniverse (2014) www.zooniverse.org (last accessed 3 April 2014)

IN-GAME SIMULATION

UTILISATION OF VIDEO GAME PHYSICS TECHNIQUES IN REAL TIME SIMULATION OF THE WHEEL RAIL INTERFACE FOR PREDICTED DERAILMENT OF RAIL VEHICLES

Michael Simpson, William Blewitt, Gary Ushaw, Graham Morgan
School of Computing Science, Newcastle University,
Newcastle-upon-Tyne, United Kingdom

KEYWORDS

Real time simulation, Wheel rail interface

ABSTRACT

We apply real time physics techniques from video game development to the simulation of rail vehicles for rapid analysis of the stability of the wheel rail interface. We introduce a fast simulation algorithm for the wheel rail interface, using a spline-based approach to approximate the gravitational stiffness force. We compare our simulation to results achieved from solving the Nadal equation for a range of rail vehicle speeds. We extend the technique to multiple railway bogies, and use the simulation to gather data on flange collisions. As we use a physics library and techniques designed for real time gaming, our results are achieved considerably more quickly than rail industry standard simulations, allowing for multiple scenarios to be modelled and analysed.

INTRODUCTION

The simulation of rolling stock on railway tracks is vitally important to the rail industry in terms of both safety and cost. New rail vehicle designs, and new track sections, undergo extensive simulation before being green-lit for construction. Traditional simulation techniques are slow and expensive (although still considerably cheaper than building rail vehicle prototypes). The time consuming nature of these simulations prohibits the number of different scenarios which can be considered, leading to conservative options being modelled from the outset.

The video game industry employs physics simulation techniques which operate in real time for multiple agents on relatively lightweight computational platforms. Consequently simulation is both cheaper and faster than more traditional techniques. However this speed comes at the cost of the accuracy of the simulation results. It is our contention that real time video game simulation techniques can be used to model multiple scenarios quickly and cheaply in order to identify the parameters with the most promising results. These chosen scenarios can then be targeted for higher fidelity simulation. In this manner a much wider range of parameters can be included in the simulations at an early stage, with the high cost time-consuming simulations only targeting the most promising combinations.

A key aspect of rail vehicle simulation is the wheel rail interface. Analysing the contact point between each wheel and the rail leads to prediction techniques for when a piece of rolling stock will derail. Wheel climb derailment occurs when the gravitational stiffness force is unable to keep the rails on the track. This is usually caused by a rail vehicle travelling at too high a velocity around a curve in the track. Rail climb derailment is believed to be the cause of sixteen derailments in the USA between 1998 and 2000 [Iwnicki, 2003]. The Nadal formula [Nadal, 1896] describes wheel-climb behaviour of a flanged pair of wheels on a set of rails, calculating the minimum conditions at which derailment will occur. In this paper we utilise NVidia's PhysX libraries to simulate railway bogies on tracks of different curvature, and employ a spline-based algorithm to simulate the gravitational stiffness force. We measure the speed at which the simulated wheel rail interface fails due to rail climb derailment, and compare the results to those calculated from the Nadal formula.

As our simulation runs in real time, it can be used to analyse many different scenarios in order to identify the parameters most worthy of high-fidelity simulation. Further to this, we show that the work can be extended to more complex scenarios involving multiple bogies and carriages. We also demonstrate that the simulation can be used to gather data on crucial indicators of rail degradation such as flange collisions. It is anticipated that this work will lead to a much wider range of simulation scenarios being considered when designing rolling stock and rail sections.

BACKGROUND

The wheel rail interface is described, and the Nadal formula for calculating wheel-climb behaviour is presented. Current techniques for simulating rail vehicle dynamics are then reviewed, which are not real time solutions. This leads to the contribution of the paper: utilising a commercial physics package from game development for real time simulation of the wheel rail interface.

Wheel Rail Interface

A wheel-set comprises wheels, flanges and axle, as shown in Figure 1. Flanges only prevent the derailment during normal operation and do not guide the wheels around bends. Instead, the conical shape of the wheels produces a self-centring effect. As the wheel drifts away from the centre of the track, the rolling radius becomes larger on the outer wheel than on the inner wheel, changing the effective size of the wheels and causing the wheel-set to yaw about the vertical axis and centre itself between the rails. This phenomenon is sometimes known as the Gravitational Stiffness Force [Iwnicki, 2003] and allows the wheel-set to corner without wear and tear on the flanges or rails, which would be potentially dangerous and expensive.

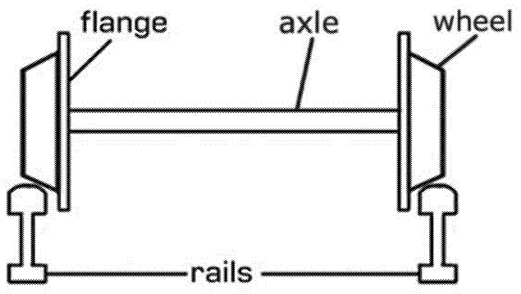


Figure 1: The Wheel Rail Interface.

When the gravitational stiffness force is not sufficient to keep the wheels on the rail, wheel climb will occur, leading to derailment if the wheel rises above the extent of the rail. The most common causes of wheel climb derailment are when the rail vehicle is travelling at too high a speed around a curve in the track, or as a result of hunting oscillation (which is a side effect of the gravitational stiffness force itself) [Marquis and Greif, 2011].
The Nadal formula describes wheel climb behaviour and represents the minimum conditions at which wheel climb derailment will occur [Nadal, 1896].

$$\frac{L}{V} = \frac{\tan\delta - \mu}{1 + \mu \tan\delta}$$

L and V represent the lateral and vertical forces acting on the wheelset, μ represents the coefficient of friction and δ represents the wheel's maximum angle of attack. The angle of attack is the angle (from horizontal) between the wheel and the rail at the contact point. Figure 2 shows the angle of attack δ between the wheel and rail during normal operation.

Rail Vehicle Simulation Techniques

Little work has been carried out to date on developing real time solutions for rail simulation. The challenges involved in simulating rail vehicle dynamics are described in [Evans and Berg, 2009]. The work does not address real-time simulation directly, but rail vehicle dynamic simulation in general is discussed. The key challenges are identified as modelling the wheel rail contact surfaces, suspension components, car body flexibility, inter-vehicle connections and track models. It is also pointed out that validation of a model is important, but problematic due to the difficulties measuring real world force data at the required fidelity.

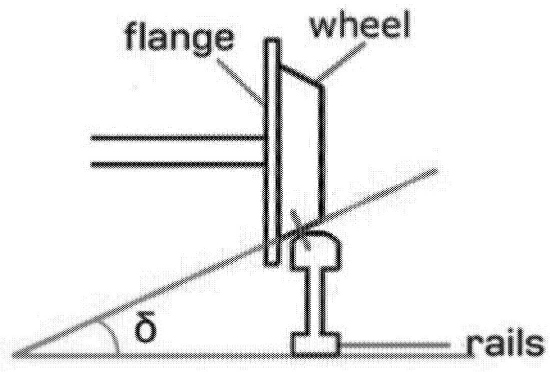


Figure 2: The Angle of Attack.

Existing work has focused on the contact surface between the rail and wheel, for application within a wider simulation for high fidelity results. [Polach, 2000] models the contact surface as an ellipsoid with normal stress distribution. This gives shorter computing time than previous approaches, and has been applied to multiple simulations successfully, but is not a real-time solution. A multi-layer spline function algorithm is described in [Shabana et al., 2001], for calculation of the third derivatives with respect to the contact surface parameters. A numerical integration method is employed in [Anyakwo et al., 2012] to solve first order differential equations representing a two degree of freedom model, taking into account the lateral displacement of the wheel-set and the yaw angle. The contact point between the rail and wheel is also the focus of [Wang and Li, 2012] which uses a technique to transform the three dimensional space of the wheel and rail surfaces into a curve of possible contact points for less complex computation. These are all computationally intense algorithms which are unlikely to be suited to real time simulation.
A method for designing and implementing real time simulation algorithms, based on the use of field-programmable gate array technology is presented in [Monga et al., 2012]. A high-performance reconfigurable platform was developed to run various simulations. Results from the simulation environment are successfully compared to a mathematical approach and the system meets the real-time constraints. This, however,

involved the use of specialised and expensive hardware; we hope that the use of PhysX will allow our simulations to run on a standard desktop PC.
Some studies have considered other applications of physics libraries for video games. In [Luo et al., 2009] a commercial physics middleware solution is employed to build a simulation and debugging environment for small, robotic cars. Two vehicles were successfully built with software that was developed in the environment, showing that the physics simulation worked well on this scale. Rail vehicles are much larger and heavier, but this work implies that video games physics software can produce meaningful results.

IMPLEMENTATION

We have produced a bespoke simulation tool called "Locomotion" for this work. The tool has been developed in C++, using the OpenGL and PhysX libraries. It simulates the progress of one or more train carriages along tracks of varying curvature at gradually increasing speed. The tool logs the number of flange collisions for each wheel, and the speed at which wheel climb derailment occurs. Figure 3 shows a screen-shot from Locomotion. The train model is based on the parameters of the Metro de Madrid 5000 Series, supplied to us by NewRail.

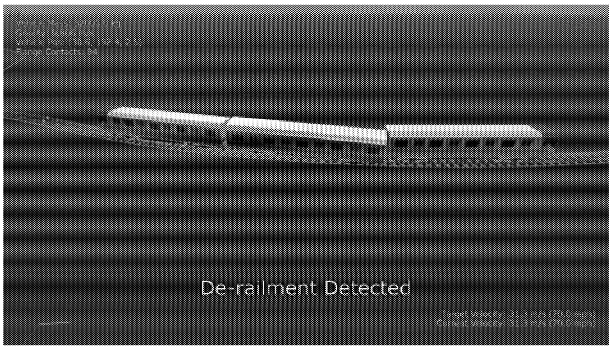


Figure 3: Screen Shot of Locomotion Simulation Tool.

To replicate the gravitational stiffness effect, we use a spline-based approach. Each section of track is represented by a spline which runs between the two rails. Each spline is defined by a Bezier curve. Every wheel/rail collision is registered, and then the nearest point on the spline to each wheel-set is calculated using an iterative search of the Bezier curve. A force is applied to the wheel-set, replicating the effect of the gravitational stiffness force. This is shown in Figure 4. The central line is the spline and the dot on the axle is the wheels global position. As the centrifugal force x causes the wheel to drift, a force y is applied to push it back towards the spline.
This force is a function of the distance to the central point (a vector), multiplied by the mass of the vehicle (divided by the number of wheel-sets) and the simulation time-step. This calculation is carried out for each wheel-set. The force is only applied while the wheels are in contact with the rails. The train will derail when the applied force is unable to counteract vehicle instability/centrifugal forces.

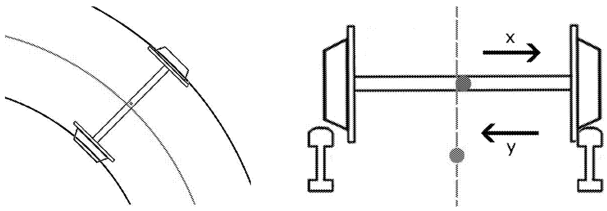


Figure 4: The Spline Based Approach.

The main body of the vehicle is constructed from a single rigid body. The mass and centre of gravity can be altered to simulate different load distributions. Each wheel is a rigid body to which the flanges are attached. The flanges are modelled as distinct rigid bodies to allow detection of flange collisions with the rails (a key signifier in rolling stock derailment and track degradation). The wheels are attached to the bogies (and the bogies to the rest of the vehicle) via revolute joints. Suspension is not currently modelled, but is approximated by an inherent, configurable flexibility in the joints. Derailment is detected by checking whether any wheel has left contact with the rail for more than a definable amount of time (in this case 0.5s), or if any component of the vehicle has come into contact with the ground. If derailment has occurred then the speed at which the vehicle was moving when last in contact with the rails is recorded as the derailment velocity.

SIMULATION RESULTS AND EVALUATION

The Nadal formula was used to predict the speed at which a single bogie is derailed, as a benchmark for comparison of our simulation results. Both the simulation and formula use the following values. Wheels have an angle of attack of 69.5°. We assume the wheel and rails are made of steel, with a coefficient of friction of 0.8. This results in a Nadal value of 0.597.
The vertical force (V) is calculated using acceleration due to gravity ($9.806 m/s^2$) and the mass of the bogie, which is 7,000kg (including wheel-sets). Our calculations also assume even distribution of forces between the bogie's two wheel-sets. We approximate the lateral force (L) using centrifugal force and calculate this for a range of speeds and curve radii.
The graph in Figure 5 shows how speed affects the L/V value. The Nadal value predicts the minimum derailment conditions (i.e. the points where the horizontal Nadal line crosses the other graphs). Solving these equations gives us the conditions for derailment shown in the table below.

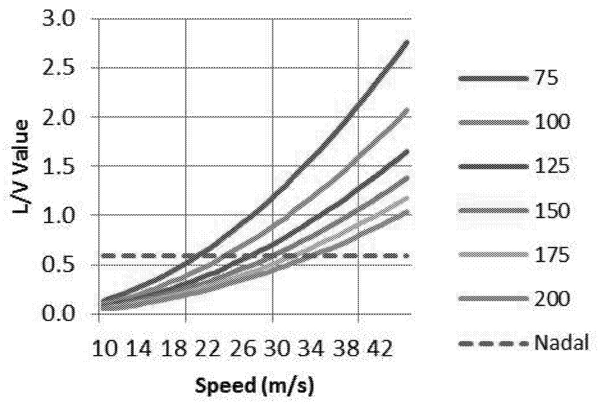


Figure 5: L/V Ratios for Varying Speed and Curve Radii.

Curve radius (m)	Predicted min derailment speed
75	20.95 m/s (46.87 mph)
100	24.19 m/s (54.12 mph)
125	27.05 m/s (60.52 mph)
150	29.63 m/s (66.29 mph)
175	32.01 m/s (71.6 mph)
200	34.22 m/s (76.55 mph)

The Locomotion software was used to construct a looped track consisting of two horizontal straight sections, and two uniformly curved sections joining them at each end. A bogie was simulated traversing this looped track, with the speed of the bogie increased by 0.5mph on each successful loop of the track until derailment occurred. The experiment was carried out with the curvature of the curved sections ranging from 75 metres to 200 metres. Each simulation was repeated 100 times, resulting in the average speed at which derailment was detected for each curvature of track. Figure 6 shows the results of these simulations, and the corresponding predicted result from solving the Nadal equation.
It can be seen from the graph that the simulated vehicle is capable of achieving higher speeds on wider curves, as expected. All simulated results are within 6mph of the predicted value from the Nadal equation. However at higher speeds, the simulation becomes less capable of successfully traversing curved tracks than the formula predicts. The range of results and standard deviations from running 100 tests of each simulation are presented in the table below.

	75m	100m	125m	150m	175m	200m
Mean	46.31	58.88	64.27	66.72	69.12	70.78
Nadal	46.87	54.12	60.52	66.29	71.6	76.55
Diff	-0.57	4.56	3.75	0.43	-2.48	-5.77
Range	5.0	16.0	11.0	11.0	8.0	4.0
Std Dv	1.09	4.44	2.11	2.27	1.76	0.83

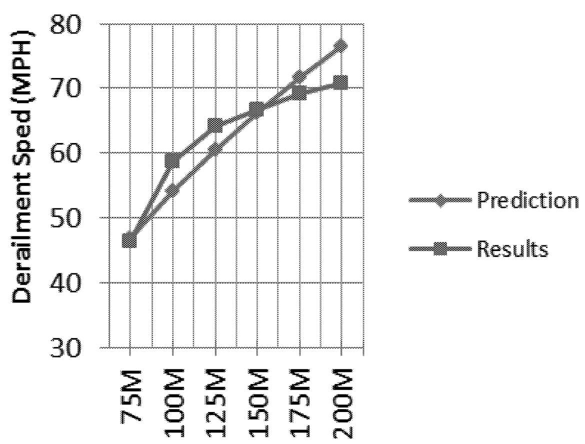


Figure 6: Predicted Speed Versus Simulated Results.

Simulating Multiple Carriages

The Locomotion tool was further used to simulate scenarios with multiple train carriages. The results in Figure 7 combine the results already considered for a single bogie with those achieved from simulating a carriage with two bogies, two carriages and three carriages (with two bogies each). In the multiple carriage scenarios, the lead carriage was pulling the rest of the train (this parameter is configurable in our software). The prediction line is from the Nadal equation for a single bogie, and is included for completeness.

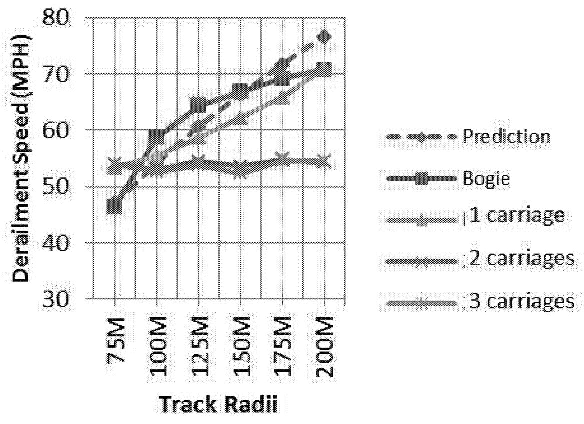


Figure 7: Derailment Speeds Compared to Bogie Predictions.

A single vehicle shows some correlation to the simulated results for a single bogie, but is less stable (and the range of derailment speeds was generally higher). Multiple vehicles are less stable due to the connection between the vehicles affecting each other's stability.
The simulation of a single bogie ran at 60 frames per second (FPS) and was therefore running in real-time. A

single carriage averaged 50 FPS, two carriages 40 FPS and three carriages 30 FPS. As we maintain a fixed time step for stability of the physics simulation, this means that the three carriage simulation is running at half speed. However, this is still significantly faster than the high fidelity simulation tools used conventionally.

Flange Collisions

As the Locomotion tool is a physical simulation of the bogie, wheels and rails, it can be utlised to collect additional data on the system. Our results for speed of derailment are sufficiently consistent with the predictions from the Nadal equation to consider the simulation as meaningful. Of particular interest to the rail industry is the number and frequency of flange collisions which occur with the track. When the flange hits the track, there is a risk of degradation to the track.

From our simulation we found that when the bogie is following a curved track to the right, 92.3% of flange collisions occur on the left side track. This is to be expected as the bogie is resting more heavily on the outside (left) wheel, creating greater likelihood of flange collisions on that side. In the case of the simulation with three carriages, it was found that the leading carriage accounted for 26.32% of flange collisions, the middle carriage 44.73%, and the trailing carriage 28.95%. This is consistent with real world expectations, as the central carriage has forces acted upon it from the two surrounding carriages.

CONCLUSIONS AND FURTHER WORK

We have successfully developed a rail dynamics simulation tool using NVidia's PhysX library and techniques from the video games industry. Our tests have shown that our spline-based method of approximating the wheel-rail interface produces results that are close to those predicted using the Nadal formula; the averages over 100 tests were within ±6mph of the predictions. While we carried out 100 tests on each curvature of track to achieve confidence in the average values, this batch testing for the bogie simulations took less than 6 hours to run, which is a considerably faster turnaround than an individual test in the more sophisticated simulation tools used in the rail industry.

Further validation is needed from real train data to determine whether our bogie simulation is sufficiently accurate. However, we believe that our results show that physics libraries for video game simulation have the potential to produce real-time engineering simulations.

Of particular interest is the "fast prototyping" aspect of the work. Utilising video game physics libraries and techniques allows us to simulate multiple scenarios in real time. This allows the rail industry to run simulations based on changing many different parameters in the design of the rolling stock and the track layout. The most successful simulations can then be earmarked for high fidelity simulation. We do not envision our approach as a replacement for existing time-consuming techniques but as an initial step in targeting those techniques. Our partners at NewRail have identified a number of fields where these techniques could be applied, including accident investigation, noise abatement, effects of load distribution and explosion modelling.

REFERENCES

[Anyakwo et al., 2012] Anyakwo, A., Pislaru, C., and Ball, A. (2012). A new method for modelling and simulation of the dynamic behaviour of the wheel-rail contact. *International Journal of Automation and Computing*, 9(3):237–247.

[Evans and Berg, 2009] Evans, J. and Berg, M. (2009). Challenges in simulation of rail vehicle dynamics. *Vehicle System Dynamics*, 47(8):1023–1048.

[Iwnicki, 2003] Iwnicki, S. (2003). Simulation of wheel–rail contact forces. *Fatigue & Fracture of Engineering Materials & Structures*, 26(10):887–900.

[Luo et al., 2009] Luo, F., Liu, C., and Sun, Z. (2009). Intelligent vehicle simulation and debugging environment based on physics engine. In *Informatics in Control, Automation and Robotics, 2009. CAR'09. International Asia Conference on*, pages 329–333. IEEE.

[Marquis and Greif, 2011] Marquis, B. and Greif, R. (2011). Application of nadal limit for the prediction of wheel climb derailment. In *2011 Joint Rail Conference*, pages 273–280. American Society of Mechanical Engineers.

[Monga et al., 2012] Monga, M., Karkee, M., Sun, S., KiranTondehal, L., Steward, B., Zambreno, J., et al. (2012). Real-time simulation of dynamic vehicle models using a high-performance reconfigurable platform. *Procedia Computer Science*, 9:338–347.

[Nadal, 1896] Nadal, J. (1896). *Theorie de la Stabilite des Locomotives*. Dunod et Vicq.

[Polach, 2000] Polach, O. (2000). A fast wheel-rail forces calculation computer code. *Vehicle System Dynamics*, 33:728–739.

[Shabana et al., 2001] Shabana, A. A., Sany, J. R., and Berzeri, M. (2001). Numerical procedure for the simulation of wheel/rail contact dynamics. *Journal of Dynamic Systems, Measurement, and Control*, 123(2):168–178.

[Wang and Li, 2012] Wang, W. and Li, G.-x. (2012). Development of high-speed railway vehicle derailment simulation–part i: A new wheel/rail contact method using the vehicle/rail coupled model. *Engineering Failure Analysis*, 24:77–92.

GAME MECHANICS OF 3D SIMULATION FOR SURGERY TRAINING SUITE

Anton Ivaschenko
Samara State Aerospace University
34 Moskovskoye shosse, 443086,
Samara Russia
E-mail: anton.ivashenko@gmail.com

Anton Skolznev
Magenta Technology
68 Lombard Street London EC3V
9LJ United Kingdom
E-mail:
skolznev@magenta-technology.ru

Andrey Kuzmin
Penza State University
40 Krasnaya, 440026,
Penza Russia
E-mail: flickerlight@inbox.ru

KEYWORDS

Surgery training, simulation, 3D modeling, game mechanics, interactive systems.

ABSTRACT

The paper describes an idea and technical approach of gaming technologies introduction in surgery training based on 3D simulation. Research is based on the authors experience in development of a specialized simulational training suite for laparotomy, laparoscopy and endovascular surgery, which was created in cooperation with Samara State Medical University. The proposed solution is based on implementation of adaptive changes in logic and surgery scenes as a reaction to the student's acting in real time. This allows improving the usability and efficiency of training suite utilization in practice.

INTRODUCTION

Intensive application of simulation tools for surgery training at clinics and medical universities requires implementation of the most up-to-date technologies in robotics, 3D modeling, electronics and software engineering. Still the combination of the most realistic visualization of surgery field and adequate haptical feedback of manipulators that simulate surgery instruments can appear to be not enough for an effective educational process. At the same time some simple versions of training suites that are comparatively cheap and easy in use can give a significant improvement to a surgery training technique.

In this paper we propose a solution vision for a computer appliance based on virtual reality implementation and used for educational purpose. The proposed approach allows the developers of new training suites to better specify the requirements, concretize the scope, prepare effective tests and improve quality assurance. The basic challenge of this research is to identify a trade off between highly realistic but laborious 3D modeling and light solutions for computer-aided education easy to operate and enter the market.

STATE OF THE ART

Simulation technologies are intensively used in medical higher education nowadays. Students employ virtual simulators to improve practical skills and enrich theoretical knowledge. Implementation of new education technologies combined with clinical work can significantly speed up the process of a surgeon's professional development.

Among the most widespread areas of simulation for surgery training there are laparoscopy, laparotomy and endovascular diagnostics and surgery (Bielser 2003, Lim 2007, Yiasemidou 2011, Zhou 2012). All the solutions in this area are based on visualization of 3D scenes that represent surgery fields for different cases and simulation of surgery intervention by means of specifically designed manipulators. The concept is pretty close to gaming simulation: the student has a certain situation described by visual model with predefined features and can perform a number of actions getting the response that simulates the real human body behavior.

Due to high complexity and uncertainty specific for real surgery intervention and uniqueness of every individual surgery scene the simulation is often simplified: the number of cases is limited to typical ones (normal and pathologic), and the possible behavior of the model is captured by a certain scenario. This helps implementing the simulation of not all the organs of a human body with realistic physiology and feedback, but only a fragment relevant to the current surgery case. Therefore the student can perform a limited number of actions at a certain moment of time; otherwise the system will terminate the game with an exception.

Besides there can be extracted two types of simulators both popular in the market:

- without force feedback: SIMENDO laparoscopy from SIMENDO and LAP-X from Epona Medical;
- providing force feedback: CAE LaparoscopyVR Simulator from CAE Healthcare, and LapSim Haptic System from Surgical Science Sweden AB.

Simbionix provides a LAP Mentor training suite with both options, either including or excluding force feedback.

There should be mentioned that each type of simulators has its own niche. Training suites without force feedback are constructively simple and relatively cheap. They allow surgeons to gain basic surgery skills and can help to evaluate a theoretical qualification and career potential of a medical student who is going to perform certain surgery interventions. Those that provide force feedback are able to offer a new range of educational techniques and bring the student closer to real life practice. One of the main problems here is to train the students to operate surgical instruments, which requires studying uncommon hand motions.

The problem of simulation-based surgery education efficiency is actively discussed in scientific community. The papers (Rodgers 2009, Munz 2004) demonstrate that in terms of educational outcome complex virtual simulators are more effective then simple models. An overview of 109 papers on simulation-based education is presented in (Issenberg 2005) that formulates key requirements for effective education. Top five requirements include realistic feedback, training repetitiveness, integration of a task to study, range of complexity levels and difference in learning strategies. Despite the fact that with no doubt the higher quality of simulation is better for educational efficiency, a number of factors is pointed out that have different impact on skills and knowledge obtained by the students. The challenges of their best combination in the process of educational process organization for various surgery specialities are studied in (Jordan 2000, Madan 2007, Hassan 2005).

MOTIVATION

However the trend in this area is to provide as much reality as possible. Modern technologies allow simulating the operating theatre with maximum possible resemblance and thereby to train novice surgeons in conditions that are as close to real life as possible. Our experience summarizes three years of development of hardware and software complex under the "Virtual surgeon" project (Ivaschenko 2013, Kolsanov 2014), and have resulted in highly realistic 3D models of the human body and adequate force feedback.

At the final stage of this project devoted to acceptance tests, delivery and deployment there was brought up an issue concerned with training suite utilization in practice. In case of high flexibility of students' actions and expansivity of scenarios how one could measure the success of surgery intervention? The training suite should provide automatic evaluation; otherwise it will require constant attendance of the trainer that reduces most of its benefits.

In real life the surgeon has to make the right decisions in real time, but in case of mistakes he can make the corresponding corrections. So the simulating software should support such a feature and introduce a sort of a point ranking system.

At the same time the simulating suite is expected to be used both for study and examination. The student who acts in training mode should have preparatory theoretical knowledge and practical skills, but a certain number of skills should be gained in the process of training. To provide this the simulated surgery intervention needs to be fascinating, i.e. attract the student's interest in the process of acting. This makes it necessary to introduce gaming mechanics to the solution and provide original strategies that will help implementing it in practice.

SOLUTION VISION

Let us consider a surgery case that can be described by a certain clinical picture and a disease pattern and has a corresponding 3D physical model simulating a human body for this case. Initially the student knows nothing about the current state except the clinical picture description and needs to perform a surgery intervention. Any mistake will cause damage and be visualized, so in case it can be mitigated the additional actions should be carried out.

For the considered case c_j, where $j = 1..N_c$ is the case number there can be defined the consecutive order of necessary and obligatory actions, represented by Boolean variables $e_{j,k} = e_{j,k}(c_j, v_{j,k}, t_{j,k}) \in \{0,1\}$, where the action $v_{j,k}$ should be ideally performed with average time interval $\Delta t_{j,k}$ after the previous action (or intervention start for the first one). Please note that the consequence $e_{j,k}$ does not specify the strict order of actions, but gives a certain number of milestones that are specific for the concerned case.

The factual surgery intervention training process can be formalized by the consecutive order of actions, represented by Boolean variables $e'_{i,j,k} = e'_{i,j,k}(u_i, c_j, v'_{i,j,k}, t'_{i,j,k}) \in \{0,1\}$, where c_j is the predefined case and u_i is the user that performs actions $v'_{i,j,k}$ at the corresponding moments of time $t_{i,j,k}$ from the intervention start. $k = 1..N'_v$ and N'_v is the number of actions.

The student's mark given by a surgery simulator is usually a grade calculated on the basis of a number of key performance indicators (KPI) characterizing intrinsic factors like human body damage and blood loss and efficiency metric, which is evaluated on the basis of agility, length of the instruments' movement track and intervention duration. In order to introduce the features of game mechanics the overall grade can be calculated in the form of a score summarizing the evaluation of student's efforts and giving him an opportunity to improve his skills.

In addition to this we propose to specify one more grade that characterizes the variation of student's actions from the original intervention scenario:

$$\rho_{i,j} = \sum_{k=1}^{N_v} \Big(e_{j,k}(c_j, v_{j,k}, t_{j,k}) \cdot t_{j,k} - \\ - \sum_{l=1}^{N'_v} e'(u_i, c_j, v'_{i,j,l}, t'_{i,j,l}) \cdot \left[v_{j,k} = v'_{i,j,l} \right] \cdot t'_{i,j,l} \Big), \quad (1)$$

where $[\ ]$ defines the Iverson bracket, a notation that denotes a number that is 1 if the condition in square brackets is satisfied, and 0 otherwise; and the other designations are introduced above.

The statement (1) allows calculating the total deviation in time between the moments of expected actions and the moments of factual actions executed by the student. This KPI can be utilized either as a component of the summarizing grade, or as a separate indicator used by the training system to adapt the complexity of the scenario to the current student. It generalizes the idea of game mechanics being applied to the process of training based on simulation. It is proposed to adapt the complexity of simulation to the student's intermediate success.

For example, in case the student is novice to work with manipulators and surgery scene he will spend much time on each action, even in case he is good in theory and knows the correct procedure. Using the statement (1) the system will identify it and simplify the case (by e.g. extending the accuracy of manipulators). In reverse for a student with enough experience of operating with manipulators the statement (1) will present negative values, and the system will also recognize the situation. In this case there can be introduced additional random events (like e.g. unexpected effusion of blood). The ways of simplification and types of unexpected events can be easily specified for certain surgical cases.

Such an approach allows obtaining the following benefits:
- the training suite becomes user friendly and adapts to different students with various experience and entry level;
- the simulated cases attract attention and interest of students in real time;
- the complexity of cases increases accordingly the student's progress in training;
- the monotonicity of training decreases by means of incorporation of random component to the case script;
- during the process of simulation the fragments of scenarios that are familiar and therefore boring are reduced;
- the system combines the periods of getting pleasure from study and satisfaction from the achievements, which is critical for a good training suite.

IMPLEMENTATION

The described solution contains hardware, including specially designed manipulators to simulate the movements of real surgical instruments with force feedback, and software that provides a realistic simulation of surgical intervention.

Laparoscopic manipulators were built using 4 Dynamixel actuators (Robotis Dynamixel site) that provide four degree-of-freedom feedback and movement. The main feature of the construction is that the entire outfit is integrated into the manipulator. This helps to reduce any transmission and as a result increases robustness and feedback sensitivity as compared with widely spread analogs. Such a solution entailed a necessity to develop specific algorithm of feedback force calculation based on transition of the physical model to forces and positions simulated by manipulators.

The software was developed in C++ using .NET C# as a platform for user interface and infrastructure support. To provide highly realistic visual and physical models there were built the scenes using PhysX (Nvidia PhysX developer site) and Bullet (Bullet physics library site) middleware. The inner parts of a human body are simulated in the scene by soft body models and surgery instruments are simulated by rigid bodies. The value of the feedback force is calculated on the basis of current geometrical position of soft and rigid bodies in the scene considering their deformation and/or topology distortion.

Both PhysX and Bullet are popular amongst 3D game developers and allow the development of highly realistic simulation scenes. These engines feature 3D collision detection, soft body dynamics, and rigid body dynamics. On the basis of this functionality there were developed special algorithms to simulate the cutting of pipes, conjoint layers and soft bodies with topology deformation, provide realistic flexibility resembling the inner parts of a human body, which is perceived in visual representation and manual feedback, and simulate liquids like blood, bile and water and their interaction with soft and rigid bodies.

The surgical cases are described by means of Lua script language. This description was separated from the simulation code. An ontology knowledge base was designed to capture basic medical knowledge about the surgical intervention stages and possible actions. In case a new event occurs that corresponds to the action in the operating scene the system checks possible actions and proceeds to the next stage. In order to provide the required performance level, the sensitivity of the operating scene is limited and differs at different stages depending on the specific skills being trained at that moment.

The developed training suite is presented in Fig. 1 - 3.

Figure 1: Laparoscopy Surgery Training Suite

The idea of game mechanics incorporation is based on the following principle. The complexity of performing of a certain training case is implemented as a dynamic value. Game logic module analyzes a consequence of student's performed actions, considering the history student's previous actions on the case and real time events. In terms of this data, Game dynamics module increases the complexity of certain training case for this student. For instance, if the student surpasses certain time threshold and his mistakes count on the case passing goes to zero at the same time, the complexity is increased by changing the calibration coefficients.

Furthermore, the level of feasibility is increased by means of incorporation of random component to the training case logic, thus making student prepared to unexpected complications of surgery intervention. According to this approach the consequence of student's actions in two different episodes of a certain training case can be significantly different. As a result the studying cases become not predetermined.

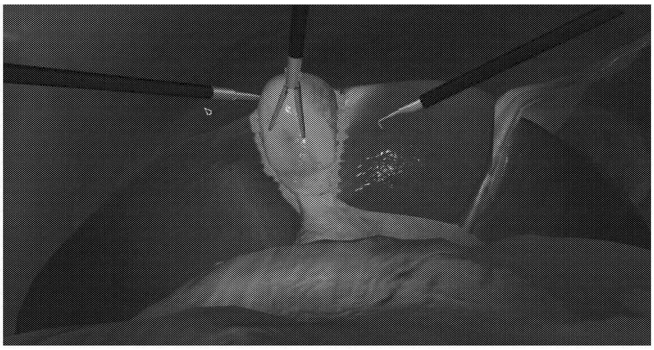


Figure 2: Simulation of Laparoscopic Cholecystectomy

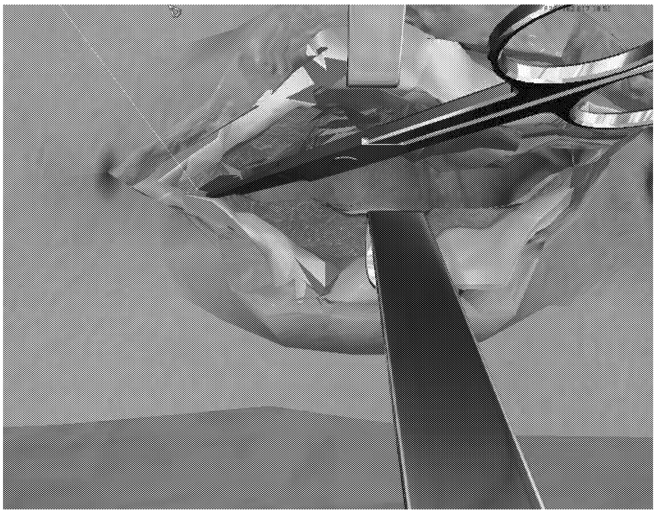


Figure 3: Simulation of Surgical Cutting

QUALITY ASSURANCE

A special hardware kit was designed and developed to provide quality assurance of the simulating suite according to the described above requirements. Firstly we need to provide a reliable behavior that depends on hardware quality and sensors adjustment. Secondly we have to test the approach based on game mechanics implementation, which is described above. Thirdly we need to test the integration of hardware and software components and configure all the elements of surgery training suite.

To solve these tasks there were developed several specialized inspection stands (see Fig. 4 – 5) that allow fulfilling test plans in automatic mode. The hardware is based on Dynamixel actuators and allows setting up the program according to which the system will perform a certain number of actions.

In order to test gaming capabilities there was carried out a number of tests with predefined student's qualification – these tests were human-aided. There was detected the system capability for adaptation to the student's behavior, which proves the technical feasibility of the provided approach. First results of deployment also illustrate an excited interest among the students' environment and validate the benefits of implementing of game mechanics in simulation education. The described suite was featured at several world-class exhibitions, such as MEDICA 2012 (Dusseldorf, Germany) and CeBIT 2013 (Hannover, Germany) and attracted the high interest of health professionals and engineers.

Figure 4: Inspection Stand and Framework for Laparoscopy Surgery Training Suite

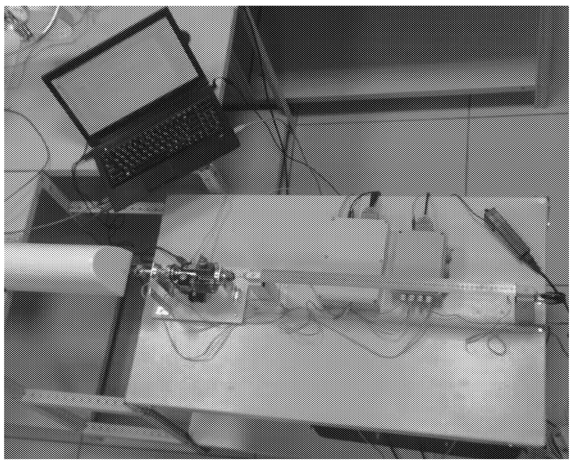


Figure 5: Inspection Stand and Framework for Endovascular Surgery Training Suite

CONCLUSION

Despite the fact that most of the surgery training suites are implemented using the same technologies that are popular in game industry, gaming strategies are yet not so popular in simulation-based education. In this paper we have proposed an idea of how simple game mechanics can be used to improve the attractibility of simulational education for students and their masters.

The introduced KPI and logic of its analysis allows adapting the complexity of surgical scenes according to individual student's experience and qualification that results in higher usability, improved educational effect and allows to use simulational training suites for self study.

The developed training suite allows its users to obtain the necessary practical surgical skills at no risk for real patients in conditions bearing considerable resemblance to the real life operating room, significantly reduce the number of surgical errors by better training; carry out trainings of various types of operative interventions; and reconstruct the real life surgery process.

REFERENCES

Bielser, D. 2003. “A framework for open surgery simulation”. A dissertation for the degree of Doctor of Technical Sciences, Diss. ETH No. 14900. Swiss Federal Institute of Technology, ETH, Zurich, 152.

Hassan, I., M. Alkhawaga, B. Gerdes, P. Langer, R. Kress, M. Rothmund et al. 2005. “Training of laparoscopic skills with virtual reality simulator: a critical reappraisal of the learning curve”. Cars 2005: International Congress Series. pp. 1281 – 1370

Issenberg S.B. et al. 2005. Med Teach London. Vol. 27. Issue 1. p. 10

Ivaschenko, A.; A. Dmitriev, A. Cherepanov, A. Vaisblat, A. Kolsanov. 2013. “Virtual Surgeon” training suite for laparoscopy, endovascular and open surgery simulation”. *Proceedings of the European Simulation and Modeling Conference 2013 (ESM 2013)*, Lancaster, UK, EUROSIS-ETI. – pp. 114 – 118

Jordan J.A., A.G. Gallagher, J. McGuigan, K. McGlade, N. McClure. 2000. “A comparison between randomly alternating imaging, normal laparoscopic imaging, and virtual reality training in laparoscopic psychomotor skill acquisition”. *Am J Surg*, Vol. 180. pp. 208 – 211

Kolsanov, A.V.; A.V. Ivaschenko, A.V. Kuzmin, A.S. Cherepanov. 2014. “Virtual Surgeon system for simulation in surgical training”. *Biomedical Engineering*, Vol. 47, No. 6, pp. 285 – 287

Lim, Y.-J.; W. Jin S. De. 2007. “On some recent advances in multimodal surgery simulation: A hybrid approach to surgical cutting and the use of video images for enhanced realism”. *Presence: Teleoperators and Virtual Environments*. Vol. 16. No. 6. – pp. 563 – 583

Madan A.K., C.T. Frantzides. 2007. “Prospective randomized controlled trial of laparoscopic trainers for basic laparoscopic skills acquisition”. *Surg Endosc*, Vol. 21. pp. 209 – 213

Munz Y.; B.D. Kumar, K. Moorthy, S. Bann, A. Darzi. 2004. “Laparoscopic virtual reality and box trainers: is one superior to the other?”. *Surg Endosc*, Vol. 18. pp. 485 – 494

Rodgers, DL et al. 2009. “The effect of hi-fi simulation on educational outcomes”. *Simulation in Healthcare*, Vol. 4, pp. 200 – 206

Yiasemidou, M.; D. Glassman, P. Vasas, S. Badiani, B. Patel. 2011. “Faster simulated laparoscopic cholecystectomy with haptic feedback technology”. *Open Access Surgery*, Vol. 4. pp. 39 – 44.

Zhou, M.; S. Tse; A. Derevianko; D.B. Jones; S.D. Schwaitzberg; C.G.L. Cao. 2012. “Effect of haptic feedback in laparoscopic surgery skill acquisition”. *Surgical Endoscopy*, Vol. 26, Issue 4. pp. 1128 – 1134.

WEB REFERENCES

Robotis Dynamixel site for professional research and development, http://www.robotis.com/xe/dynamixel_en

Nvidia PhysX developer site, https://developer.nvidia.com/physx

Bullet physics library site, http://bulletphysics.org

AUTHOR BIOGRAPHIES

Dr. Anton Ivaschenko, Doct. of Tech. Sciences, Professor at Samara State Aerospace University;

Anton Skolznev, PhD student, lead software developer at Magenta Technology (www.magenta-technology.com), a UK-based provider of real-time dynamic multi-agent solutions for intelligent decision making support.

Dr. Andrey Kuzmin, PhD, Associate professor at Penza State University.

GAME PROGRAMMING

A PROPOSAL TO PROGRAM CONTENTS FOR TEACHING CHILDREN FROM 0 TO 12 YEARS THE BASICS OF PROGRAMMING

Laura L. Dias,
Alex F. V. Machado,
Lindomar M. de Paulo,
Euler R. Q. de Alvarenga
Departamento de Computação
Instituto Federal de Educação, Ciência e Tecnologia do Sudeste de Minas Gerais - Rio Pomba, MG, Brasil
E-mail: {laura_limadias| alexcataguases} @hotmail.com and {lindomar.lmp| euler.sistemas } @gmail.com

KEYWORDS
Teaching, programming, children, syllabus.

ABSTRACT

On this paper, it is proposed a "syllabus" of simple and conventional activities that can aid children to become competent computer programmers with ease. Activities such that can be developed with the children to train the logical-mathematical way of thinking, preparing them not only to choose a major on the field of Information Technology, but, through the organization of thought process, to develop capabilities such as deduction, induction and dialectics of the professionals of the future. The goal of this paper is to gradually develop common activities focusing on both short and long-term progress, such as swimming classes, music, painting and even through toys and games to develop the potential abilities on children.

INTRODUCTION

Nowadays, most of child's education is focused on the learning of the basic skills, instead of emphasizing a potential to be achieved. Through the idea that the way of thinking can be both learned and taught, it is believed that there is the possibility to raise the general level of intelligence of any child through mediation of a teacher or parent.

This can happen by teaching children not only the basics of a course, but giving them also an extra knowledge of the world to help them in their own knowledge and abilities to efficient problem solving matters, reasoning and thinking. Thus searching for specific ways that raise the individual level of intelligence, to quicken the cognitive development and to enhance a children's capability to think and learn (Fisher 2005).

Several institutions are trying to become more focused on the students than on those who teach, to connect a school with real life situations, and encourage the students to focus on understanding and think other than just memorizing what is presented to them (Vosoiadou 2003).

Schools nowadays are concerned about children, that are the first generations that were grown on this new era of technology. They have spent their entire lives surrounded by computers, videogames, digital music players, digital cameras, mobile cellphones, and all the other toys of the digital era (Prensky 2001).

Barack Obama, president of the United States of America has recently motivated the Americans of all origins to learn how to master the technology of coding and also the computer science, which, accordingly to him, has been having huge influence on their lives.

"To learn to use these skills is not only important for your future, it is important for our country's future. If we want for the Americans to be on top, we need young North-Americas that enjoy mastering the tools and technologies that would change the way of almost everything", said President Obama on a video message to the young Americans (*Free press journal* 2014).

On another way, it is possible to notice young programmers among the mobile applications development. According to (*Tecmundo* 2014), intuitive tools can make even an 8 year child to code their own applications to any kind of platform known nowadays. Showing that there is not only motive coming from the government, but also the children and their families play a big part on it.

There are also initiatives such as "Lenoarmi", which is an integral physical education center, where children are able, on their first years to have physical activities classes, and others such as the Super Geeks, a school that teaches robotics and computer programming to children and teenagers.

Beyond these activities proposed by the governments and private initiatives, there are simple methodologies, as the simple act of playing with some games that stimulate the thought process to develop itself, (Piaget, 1973) says that children develop their logical-mathematical reasoning supported by the sensorial-motor actions over material objects and through spontaneous repetition exercises.

RELATED WORKS

Some works discuss about the importance of children to learn code language the sooner the possible. The main idea, on the other hand, is to educate our children early on how to think in a more logical way, understanding how computer programs really work, not only learn logics. The greatness of this new approach is not measurable, for these children may influence new areas on several fields, not only into the

Information Technology world, but the power of abstraction goes way beyond the computers.
Thus, which would be the best way to educate them? By analyzing previous works of great names of psychology, according to (Piaget 1989), childhood is divided into four steps, different ways of learning and notions of everything. The cognitive processes are named as motion sensor, pre-operational, formal operator and full operational.
Following this line of thought, it is possible to develop the process of teaching computational logics, as of algorithms, data structure amongst others, by shaping the learner to the cognitive process level which each children should be. Children on the motion sensor step can play with pieces shaped in order to fit in a given order, simulating sorting algorithms. Children that are able to read or understand stories will be thrilled when introduced to the world of Role-Playing Games (RPG), proposed by Facebook's engineer Carlos Bueno, on his book named Lauren Ipsum, which teaches computer science to children.
Some countries, well developed have already motivated children on this field, big companies and universities fund the idea. It is of main importance that educational institutions and developers believe on this idea to the less developed countries to achieve this learning revolution (Alvarenga 2013).

COMPUTER-FREE ACTIVITIES DIRECTED TO THE LEARNING OF COMPUTER SCIENCE

Spreadsheet skills are important, but the biggest impact over economy will be of innovative services to be built over complex and safe computational services. Systems such as YouTube, Google, Facebook, and other big names and successful stories around technology, they rely on inventive developers who could use a series of skills, including coding, security, cloud computing, and other fields of domain of Computer Science and Engineering. Although the idea is not to teach too much of specifics (complexity of algorithms, graphs theory etc), some knowledge of computer science inside schools could be valuable, even to help the students to choose a career. Instead of trying to avoid these themes, it should be transparent to students that computers are not only about slides presentations. To introduce Computer Science to children in school is difficult for many reasons: most of the teaching professionals lack of a well formed base to teach children, the administrative part of a teaching institute is divided among other courses, thus not having much information about Computer Science itself. Not only students are leaving, the situation can be far worse when it comes to the female gender compared to the male population in this course, according to North-American statistics. And these tendencies repeat themselves in another countries such as New Zealand (Bell et al. 2009). Many programs were created in order to respond to such questions, including videos showing what a career in computer science and development is about. One way to approach this problem, that has found its place of importance internationally is, the "Computer Science, Unplugged" program, with its base at Canterbury University. The project provides free resources for Computer Science dissemination and education on its website, csunplugged.org. It takes the common approach of exposing children to great ideas about Computer Science, without the use of computers. Usually, unplugged activities are about problem solving to reach a goal, and on the process, to deal with fundamental concepts of Computer Science (Bell et al. 2009).
To learn and to achieve high cognitive development, to develop physically, emotionally and socially, children must have constant interaction with other children and adults, they must explore, manipulate objects, build objects, read, listen, play, talk and build relationships. This information about children's needs is a fundamental reason why first grade teachers, many times believe that computers and "screen time" have no place in the first part of childhood, and that technology should never substitute these vital experiences. Instead of it, technology is far more productive in the lives of children when they get more enrolled with these activities, as their reflections about actions and experiences (Barron et al. 2011).

HOW DO CHILDREN LEARN?

Numerous schools are communities where children from various cultures learn together. There are cultural systematic differences about habits, doings, roles on society, etc, that usually influence the learning process. Sometimes, significant activities to the students from a social group are not that significant to students from other groups.
The research shows that students, even unconsciously can relate what is new to what they already understand, and it also points out that they learn much faster when their attention is focused on the previous knowledge as a starting point to the new one.
Children develop strategies to aid themselves into solving problems since their first years of life. These children have found that rehearsing is an efficient way to enhance their memories, without having been told to do so. Strategies are of great importance because they help students to solve problems with the most appropriate way to do so. Such strategies can enhance learning and make the process even faster. The wider the range of strategies that children may use, the most successful they will be at problem solving matters, reading, comprehension and memorizing (Vosoiadou 2003).
When dealing with small children, some factors should be taken into consideration, as an example of their physical capabilities, the development of their motor skills. Also the gender should be a matter to consider, boys are usually inclined to be stronger, while girls have more control over their motor skills. (Smith 1982).

A SYLLABUS PROPOSAL

The syllabus used for this work is presented throughout this chapter with the activities that can be easily developed.

ACTIVITY	SIMILAR ACTIVITIES	ROLE IN THE DEVELOPMENT	DISCIPLINARY COMPETENCE IN COMPUTER SCIENCE LONG OR SHORT TERM ESTIMULATED
SWIMING	Waterpolo, synchronized swimming, lifeguarding, jumping, Educational water games (Marín 2004).	motor skills; following orders; ability to control breathing; rightful movements of the limbs (Gladish 2002).	Algorithms; Compilators; Computer theory; Computer architecture.
MUSIC	reading, chess class, theater (Schellenberg 2005).	practice language patterns; interpreting books; develop comprehension through music and dramatization; learn to decode written language (Standley and Hughes 1997).	Algorithms; Programming languages; Artificial Intelligence; Computer graphics; Database.
ARTS	Dance, music and poetry (Samuelsson et al. 2009).	be more critic; development of understanding; make children more receptive to the instruncions given to them (Samuelsson et al. 2009).	All of the disciplinary fields in Computer Science.
MARTIAL ARTS	boxing, wrestling, Full Contact and Kick boxing (de Oliveira 2009).	social psychological contribution; enhance on self-confidence; self-control on social skills; higher concentration rate while performing tasks (Theeboom et al. 2009).	All of the disciplinary fields in Computer Science.
TOYS	Puzzles,LEGO, mosaic, Rubik's cube, puppets (Kishimoto 2001).	Cognitive skills; Develops verbal skills; Builds up social skills (Bomtempo 1999).	Digital circuits; Scientific methodology; Ethics and social responsibility.
ROLE-PLAYING GAMES (RPG)	Chess and card games (Passerino 1998).	Develops critical sense; Reasoning abilities; Abstraction (Simkins and Steinkuehler 2008).	All of the disciplinary fields in Computer Science.
LEGO MINDSTORMS	Robotics (Klassner 2002).	Develops hypothesis ; elaboration skills; Develops the skills to search for solutions; Develops the capability to establish relationships; Develops the skills of conclusion-taking (Benitti et al. 2009).	Algorithms; Programming languages; Artificial Intelligence.
SCRATCH		Aids on mathematical and computational learning factor; Teaches concepts related to projects in development; It is widely used in universities as the first step to programming (Resnick et al. 2009).	Algorithms; Programming languages.
ROBOCODE		Stimulates fun and creative ideas; Refines ideas into viable solutions; Transforms fragile knowledge into a concrete ability to be transferred to another scenario; Students develop competencies to each software developing step; Stimulates critical thought and Exchange of ideas (O'Kelly and Gibson 2006).	Algorithms; Programming languages.

Table 1: Syllabus

Swimming is an activity for life, but it will be taken into consideration only the first 12 years of one's life, where children do not only develop motor skills but also the psychological part. LEGO Mindstorms is indicated to children of age above 10, but if they are following a syllabus that develops motor and cognitive capabilities, it is believed that they could work with the proposed activities of LEGO Mindstorms before that age. Robocode and Scratch are digital environments that integrate with the bases of programming, together with interactivity and fun, building so, a ludic way of learning, making the whole process bearable for children, because they will think of it as fun.

The graph that follows is related to the age that each activity should be practiced, and the amount of time to be spent on a week basis. It should though be considered that such activities will only take up to four hours per day, children should have their normal lives for the syllabus to be just a way to develop their intellect without spoiling their childhood.

The electronic environments introduced in this graph follow the research of (Pires 2009), and states that on a regular basis, children stay from 1 to 5 hours per week using a computer, based on this information, it is suggested an average of 2 hours of LEGO Mindstorms, 3 hours to the first level, that represents the foundation of learning of programming-based environments and 2 hours for the second and third levels, that are for children who already have some knowledge on programming. Further on this paper each environment composed by the three levels will be explained.

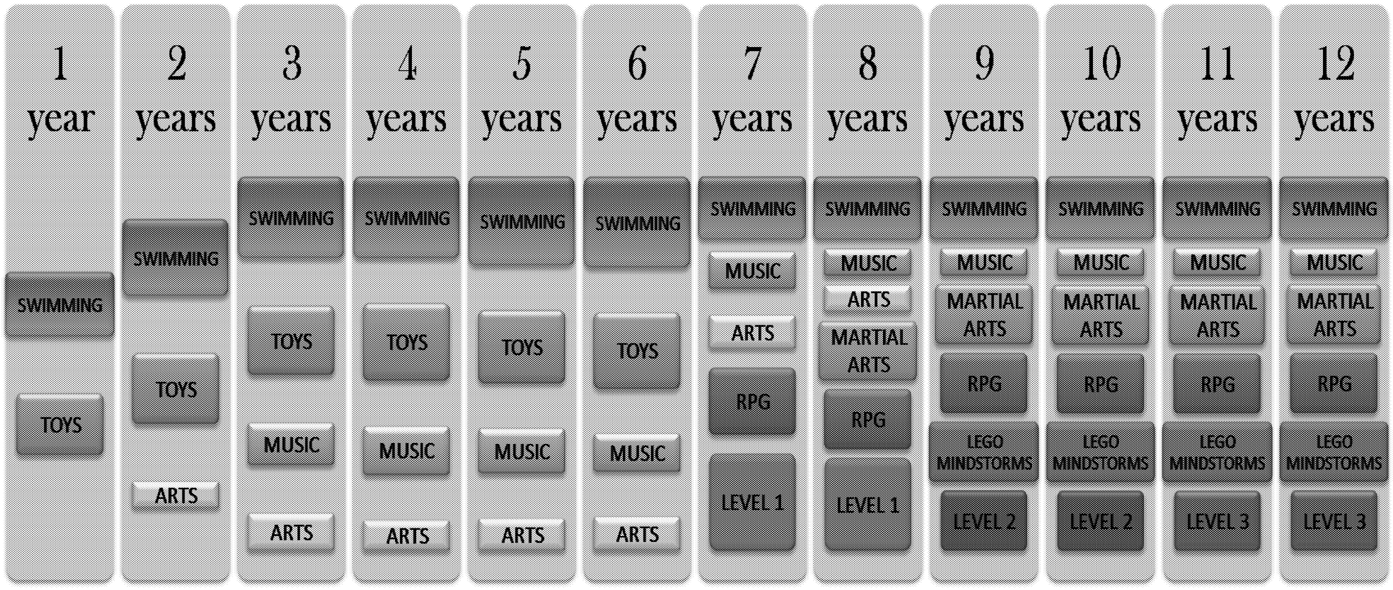


Figure 1: Syllabus by age

Considering everything proposed so far, and the enthusiasm to learn in a digital era, the following table will show games and other tools that could be useful on the learning process:

ENVIRONMENT	DESCRIPTION	LEVEL
APPINVENTOR	App Inventor: in a manner of weeks, beginners can create applications that are not only fun, but have utility in the real world. It allows to transform creative ideas into working, interactive applications that can be taken over by large companies, used by nonprofits, startups and turned into people. It has been successful, not only with children but as an initial module in a university course (Wolber 2011).	3
ARDUINO	Arduino is a widely accepted platform for physical computing. Therefore, people who want to program Arduino include designers, artists and amateur programmers. Moreover, as physical computing is an attractive tool for education, Arduino has been used in schools and universities (Kato 2010).	3
UNITY 3D	The Unity3D Game Engine has a very simple and friendly interface that facilitates the development of various genres of games and other image display systems (Passos et al. 2009).	3
ALICE 3D	The Alice Project aims to create new tools that would make creation accessible to a beginner on 3D graphics, something that current 3D tools are not able to do, since they require a prior knowledge of programming or even mathematical logic (Conway et al. 2000).	3
KODU	Kodu Game Lab, a constructionist educational software, where students can learn the concepts of logical reasoning, and playing with many objects in a three dimensional environment and multimedia elements, as an alternative to traditional teaching method (Souza and Dias 2012).	1
RPG MAKER	RPG Maker VX is a fairly inexpensive commercial application for creating video games. Intended mainly for amateurs and enthusiasts, is not a particularly robust platform for building game. Instead, it focuses on making it relatively easy to build a game based on the general conventions of the genre (Owens 2011).	2
PYGAME	Pygame is a graphically simpler program that require students to devote a large percentage of their time to the game logic (Ceder and Yergler 2003).	2
SCRATCH	The blocks are understandable and the environment allows you to explore its meaning interactively. The result is a proven success tool engaging with children and as an initial component of a university course (Wolber 2011).	1
BLOCKLY	Blockly is to encourage people who will want to become software engineers, and has a talent for it. It is a visual editor, not a language with generated code in JavaScript, Python and XML (McGettrick 2014).	2

MINIBLOQ	Minibloq is a graphical programming environment that facilitates the introducing of students to the world of programming. Students use colored blocks for programming physical computing devices very easily (Junior et al. 2013).	1
ROBOCODE	Robocode stimulates the fun element of creative ideas within the constraints of the Robocode environment with challenges to refine these ideas into a workable solution. This problem becomes fragile knowledge in a particular transferable skills that can be applied in new situations (O'Kelly and Gibson 2006).	1

Table 2: Stimulating environments

CONCLUSION

All factors above being considered, it is obvious to say that children nowadays have a completely different life from those who lived on the last century. They have much more contact with the technological universe that their grandparents had, for example. And for this reason they should be educated on a different way, but it is not healthy for them to spend much time encapsulated in their houses surrounded by electronic devices. For that reason, this work proposes a new approach about the education since the first year of one's life. Intending to prepare them for the job market that they will be part of in the future, playing a big role on it, and being intensely involved with the Computer Science universe, having had all the motor and cognitive experiences needed to achieve such baseline.

FUTURE WORK

This paper shows a new way to teach children, by inserting them into the bases of Information Technology. This is still a new technique but it gets attention because of its wide approach. The usage of this method is shown through demonstrations and theoretical analysis.
As of the results of this research, classes will be given, applying the method and practical tests on children of the correct age, that now allows the automation of various analysis presented until now. In future works, it is of interest to develop these classes and tests by using the methodology here presented on a way to prove the efficiency of the method.

ACKNOWLEDGEMENTS

The authors acknowledge the Programa de Educação Tutorial of the Ministério da Educação (MEC/SESu/Brazil) of the Instituto Federal de Educação, Ciência e Tecnologia do Sudeste de Minas Gerais, the Prefeitura de São João Nepomuceno and vereador Sebastião Barbosa, who secured the financial support this work.

REFERENCES

ALVARENGA, E. R. Q. ; MACHADO, A. F. V. . (2013) Why Children Should Program Computers without Computers. In: European GAME-ON Conference, 2013, Bruxelas. Game-on.

Barron, Brigid, et al. (2011)."Take a giant step: A blueprint for teaching young children in a digital age." *New York: The Joan Ganz Cooney Center at Sesame Workshop and Stanford University. Available at: www. joanganzcooneycenter. org/Reports-31. html.*

Bell, Tim, et al. (2009)."Computer Science Unplugged: School Students Doing Real Computing Without Computers." *New Zealand Journal of Applied Computing & Information Technology* 13.1.

Benitti, F. B. V., Vahldick, A., Urban, D. L., Krueger, M. L., & Halma, A. (2009, July). Experimentação com Robótica Educativa no Ensino Médio: ambiente, atividades e resultados. In *Anais do Workshop de Informática na Escola* (Vol. 1, No. 1, pp. 1811-1820).

Bomtempo, E. (1999). Brinquedo e educação: na escola e no lar. *Psicologia Escola e Educação*, *3*(1), 61-69.

Ceder, V., & Yergler, N. (2003). Teaching Programming with Python and PyGame. *Apresentado na PyCon*.

Conway, M., Audia, S., Burnette, T., Cosgrove, D., & Christiansen, K. (2000, April). Alice: lessons learned from building a 3D system for novices. In*Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 486-493). ACM.

de Oliveira, André Luis, Fabio Rodrigo Ferreira Gomes, and Frank Shiguemitsu Suzuki. (2009). "Conceito dos tipos de lutas a partir de uma visão de cultura corporal.".

Fisher, Robert. (2005).Teaching children to think. Nelson Thornes.

Gladish, K. (2002). Swimming programs for infants and toddlers. *Pediatrics*,*109*(1), 168-169.

Harman, M. (2010, September). Why source code analysis and manipulation will always be important. In *Source Code Analysis and Manipulation (SCAM), 2010 10th IEEE Working Conference on* (pp. 7-19). IEEE.

Junior, L. A., Neto, O. T., Hernandez, M. F., Martins, P. S., Roger, L. L., & Guerra, F. A. (2013). A Low-Cost and Simple Arduino-Based Educational Robotics Kit. Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Robotics and Control (JSRC).

Kato, Y. (2010, January). Splish: a visual programming environment for Arduino to accelerate physical computing experiences. In *Creating Connecting and Collaborating through Computing (C5), 2010 Eighth International Conference on* (pp. 3-10). IEEE.

Kishimoto, T. M. (2001). Brinquedos e materiais pedagógicos nas escolas infantis. *Educação e Pesquisa*, *27*(2), 229-245.

Klassner, F. (2002, February). A case study of LEGO Mindstorms'™ suitability for artificial intelligence and robotics courses at the college level. In*ACM SIGCSE Bulletin* (Vol. 34, No. 1, pp. 8-12). ACM.

Marín, A. M. (2004). Atividades Aquáticas como conteúdo da área de educação física. *Lecturas: Educacion Fisica y Deportes, Buenos Aires, ano,10.*

McGettrick, A., & Timanovsky, Y. (2014). Digest of ACM educational activities. *ACM Inroads*, *5*(1), 6-10.

O'Kelly, J., & Gibson, J. P. (2006). RoboCode & problem-based learning: a non-prescriptive approach to teaching programming. *ACM SIGCSE Bulletin*,*38*(3), 217-221.

Owens, T. (2011). Social videogame creation: lessons from RPG Maker. *On the Horizon*, *19*(1), 52-61.

Passerino, L. M. (1998). Avaliação de jogos educativos computadorizados.*Taller Internacional de Software Educativo*, *98*.

Passos, E. B., da Silva Jr, J. R., Ribeiro, F. E. C., & Mourão, P. T. (2009). Tutorial: Desenvolvimento de jogos com unity 3d. In *VIII Brazilian Symposium on Games and Digital Entertainment. Rio de Janeiro*.

PIAGET, J. Proceedings of the Second, (1973)b. Comments in Mathematical Education, em A. G. Howson (ed)

PIAGET, Jean. (1966-1989) - *Comentários sobre as observações críticas de Vygotsky*. Apêndice da edição italiana de: VYGOTSKY, L.S. Pensiero e Linguaggio. Firenze, Giunti, (Tradução de Agnela da Silva Ggiusta – Universidade Federal de Minas Gerais)

Pires, S. M. B. (2009). Interesses das crianças na utilização autónoma do computador, em contexto escolar e familiar.

Prensky, Marc. (2001). "Digital natives, digital immigrants part 1." On the horizon 9.5, 1-6.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. (2009). Scratch: programming for all.*Communications of the ACM*, *52*(11), 60-67.

Samuelsson, I. P., Carlsson, M. A., Olsson, B., Pramling, N., & Wallerstedt, C. (2009). The art of teaching children the arts: Music, dance and poetry with children aged 2–8 years old. *International Journal of early Years education*,*17*(2), 119-135.

Schellenberg, E. G. (2005). Music and cognitive abilities. *Current Directions in Psychological Science*, *14*(6), 317-320.

Simkins, D. W., & Steinkuehler, C. (2008). Critical ethical reasoning and role-play. *Games and Culture*, *3*(3-4), 333-355.

Smith, Frank. (1982).How children learn. Society for Learning Disabilities and Remedial Education.

Souza, P. R. D. A., & Dias, L. R. (2012). Kodu Game Labs: Estimulando o Raciocínio Lógico através de Jogos. In *Anais do Simpósio Brasileiro de Informática na Educação* (Vol. 23, No. 1).

Standley, J. M., & Hughes, J. E. (1997). Evaluation of an early intervention music curriculum for enhancing prereading/writing skills. *Music Therapy Perspectives*, *15*(2), 79-86.

Theeboom, M., De Knop, P., & Vertonghen, J. (2009). Experiences of children in martial arts. *European Journal for Sport and Society*, *6*(1), 19-34.

Vosoiadou, Stella. (2003). "How children learn?." Successful Schooling, 16.

Wolber, D. (2011, March). App inventor and real-world motivation. In*Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 601-606). ACM.

WEB REFERENCES

http://freepressjournal.in/obama-emphasises-on-computer-education-for-americans (Accessed in 30/03/2014)

http://www.tecmundo.com.br/m/52759.htm (Accessed in 30/03/2014)

BIOGRAPHIES

LAURA LIMA DIAS studied computer science at the IFSUDESTE-MG. Rio Pomba. She is a research at the group of Prof. Alex Fernandes da Veiga Machado, His main research interests are games and artificial intelligence.

ALEX FERNANDES DA VEIGA MACHADO has a PhD in Computer Science at the Universidade Federal Fluminense. He is professor of Bachelor of Computer Science at IFSUDESTE-MG, Rio Pomba Campus. Has experience and serves mainly on the following topics: game development, artificial intelligence. He is currently Coordinator of the Laboratory for Interactive Multimedia IFSUDESTE-MG.

LINDOMAR MATOS DE PAULO Computer Science student at IF Sudeste MG, former president of the Academic Directory of the Computer Science course, first student of the campus to study abroad with the Science Without Borders program, technology enthusiastic and amateur writer.

EULER RITHIELE QUEIROZ DE ALVARENGA I am a Computer Science student at the campus in Rio Pomba, codes enthusiastic and willing to create the unthinkable, currently I attend the Science Without Borders program at St Clair College, Windsor, Canada.

TEACHING INTELLIGENT VIRTUAL AGENTS PROGRAMMING THROUGH SIMULATED CHILDREN'S GAMES

Jakub Gemrot, Martin Černý, Cyril Brom
Faculty of Mathematics and Physics,
Charles University in Prague,
Malostranské náměstí 25, 118 00, Prague 1,
Czech Republic
E-mail: gemrot@ksvi.mff.cuni.cz, cernym@gmail.com, brom@ksvi.mff.cuni.cz

KEYWORDS
Educational Games, AI Education, Game AI.

ABSTRACT

Developing intelligent virtual agents (IVAs) is a great challenge for computer programmers. Lifelike virtual environments present various obstacles, especially on the lower AI level. Navigating through 3D worlds is notoriously difficult to handle properly as well as quick and appropriate reactions to rapid changes of the environment. When teaching basics of IVA development at our university, we noticed that students find the complexity of virtual environments intimidating. Although lectures on AI theory help, a substantial amount of hands-on experience is indispensable to gain proficiency. We thus searched for ways to start with very simple tasks and at the same time keep the students engaged and motivate them to experiment with AI development at home. In this paper we report on two 3D virtual environments we developed on top of Unreal Tournament 2004 for the introductory classes of our course. The environments are inspired by children's games and are focused primarily on combining simple high-level decisions with navigation. Tournaments of bots were held for both environments to conclude parts of the course. Evaluation over two years of the course shows that the environments helped students to focus on subparts of the IVA development and that the tournaments motivated the students to experiment with IVA behaviors outside the borders of the course.

INTRODUCTION

As the virtual entertainment industry grows, there is an increasing demand for education in game development and related fields. One of the very interesting areas in game development is AI and development of intelligent virtual agents (IVAs) in particular. In this context we run an IVA development course at our university since 2005 for a mix of undergraduate and postgraduate students. The course focuses on the modeling of IVA behaviors in complex 3D environments (partially observable, dynamic, continuous, non-deterministic, multi-agent) from both theoretical and practical perspective. The focus of the course is not on classical symbolic AI (A*, planning, etc.) as this is covered by prerequisite AI courses, but rather on reactive reasoning and "intelligence without representation". The theoretical part covers a broad range of topics from neuroscience and psychological background to reactive planning methodologies. In the practical part of the course, students develop bots for deathmatch mode of Unreal Tournament 2004 (Epic Games Inc. 2004).

While the course was relatively successful, we noticed that the practical part of the course was overly challenging for the students who often felt lost and frustrated. One of the most problematic areas was navigation through the 3D environment and other low-level tasks related to geometry of the environment. This was intentional in a way, because our experience indicates that it is those difficulties involved in the 3D worlds that make game AI both hard and unsuitable for classical AI approaches. But we recognized the need for environments that would retain the delicate intricacies of 3D but would accept simple scenarios and require less sophisticated higher level reasoning. Such environments will let the students progress with smaller but quicker steps and experience a sense of success and reward more often.

Furthermore, we knew that as in other areas of software development, practical experience — a lot of practical experience — is indispensable. Even more so for virtual environments, where even things that seem very easy at first sight (e.g. navigating correctly in the environment) often introduce unexpected difficulties and the actual implementation is at least as important as the general idea for the overall success of the behavior. We thus wanted to motivate students to experiment with the AI at home, as the time allocated for classes was far from sufficient to gain proficiency.

To summarize, we wanted to follow the long-known pedagogical principles: start with the simple; learning should be fun; practical experience promotes learning (Comenius 1648).

We sought inspiration in the way humans learn to navigate seamlessly in the real world. We realized that children master movement in the real world by playing simple games where fast movement in the environment is vital for victory. We found those games to be a great source of inspiration. They have simple rules and most of the students already know the rules, simple behavior is sufficient to achieve reasonable results and still there are plenty of possibilities to improve over the simple approach.

In this paper we report on our use of virtual counterparts of children games to teach navigation and basics of reactive decision making. We chose two children games: Tag! and Hide & Seek. We implemented the games in UT 2004 and used them as test environments during the introductory part of our course. To further motivate students to play with the AI at home, we introduced non-obligatory tournaments of bots in both games. We report on feedback from the students and the implications of using similar games in course curriculum.

The rest of the paper is organized as follows: First, we discuss related work on teaching AI and programming in general, than we briefly detail the curriculum of our course and motivate our use of UT2004 in the classroom followed by the discussion of requirements we imposed on the environments and details of the individual environments. Finally, we report on the evaluation of the games during two years of the course.

RELATED WORK

Multiple approaches were tried to increase engagement in general computer science/programming courses. Insights from general pedagogic research are transferred to computer science education e.g., (Lockwood and Esselstein 2013, Porter and Simon 2013, Kafai et al. 2013). These efforts are orthogonal to our research as our educational games can be incorporated in virtually all teaching methodologies. Gamification of the course was proposed (Decker and Lawley 2013) — similarly to this approach, we have implemented a flexible grading system that allows students to score points for various activities. Developing games as part of a class has also been suggested (Bayzick et al. 2013).

Bayliss (2009) shows that games have been successful in both attracting, motivating and retaining students of computer science. It is noted that correctly chosen open-ended assignments stimulate creativity and let students "play" with the task. She also reports on caveats of the approach, including the high requirements on the teacher side and possible technical problems with game technology.

In the context of AI, educational scenarios based on Pac-Man

and other simple game environments (DeNero and Klein 2010, McGovern et al. 2011, Bezakova et al. 2013) have been proposed. Those are, however not applicable to our case as they focus on classical AI techniques and do not involve an environment comparable in complexity to 3D computer games.

COURSE CLASS DETAILS

The course lectures cover various topics related to IVA development: reactive planning, subsumption architecture (Brooks 1991) , behavior oriented design (Bryson 2001), steering, evolutionary algorithms, neural networks, background in ethology, neuroscience, psychology and psychophysics; belief-desire-intention architecture (Georgeff et al. 1999) and multiagent systems. We have reported on the curriculum of the course lectures in more detail in (Brom 2009). In the following text, we focus on the practical classes we have developed and evaluated since.

To provide students with hands on experience in IVA development we have created Pogamut (Gemrot et al. 2009) — a platform for prototyping of bots' behaviors for Unreal Tournament 2004 (UT2004) (Epic Games Inc. 2004) in the Java programming language. UT2004 is first-person shooter (FPS) that was very popular in the 2000s. Even though an older game today, the graphics of UT2004 still appeal to students and the game complexity does not differ from its sequel Unreal Tournament 3 (Epic Games Inc. 2007) or other recent FPS games.

During the practical classes, students are taught how to hierarchically decompose behaviors using behavior oriented design (BOD) in a top-down manner and then implement the behavior on top of Pogamut platform using a bottom-up approach.

Respective practical classes are focusing on different technical aspects of the Pogamut platform, teaching students only a limited set of behavior primitives (sensors and effectors) available to bots every class, allowing students to gradually explore different aspects of UT2004 bot behaviors. The task of the students is to implement simple behaviors using the newly learnt primitives and to incorporate them in a bot they incrementally create. As the set of behavior primitives grows, students are able to construct more complex behaviors, starting with a simple follow-me-bot to a bot covering all the aspects of a deathmatch game.

Practical classes cover the following behavior aspects: low-level movement, environmental reasoning, navigation, item collection, combat and team work.

The ultimate objective of the practical classes is to teach students how to structure IVA behaviors for game-like tasks within UT2004 environment. The final proof of their ability to do so is a successful implementation of a death-match (DM) bot that is able to beat less-skilled human players.

Students are graded based on points they can get for multiple types of activities. Those include: attendance, homeworks, quick tests in class and several optional tournaments of bots throughout the semester.

REQUIREMENTS AND ANALYSIS

The initial course runs comprised of classes that focused on the DM mode too much. We were teaching students how to incrementally build their bots by introducing new behavior aspects that students were adding into an existing one. Even though we tried to mask as much complexity as we could in the Pogamut platform, AI for UT2004 simply needs to handle too many issues. There are numerous relevant sensory data messages UT2004 exports (21 messages, 189 attributes) that the bot has to handle as well as command messages the bot needs to use correctly (13 commands, 34 attributes). The bot has to reason about 10 weapon types and 17 item types that are available within the game.

Due to the high complexity, students could not create bots that would cope with at least the most important game aspects until late in the semester which was not very rewarding and it did not motivate students well for two reasons. Firstly, knowing that creation of DM behavior does not fit into single class and single homework, students were not experimenting with the behavior at home; they rather

waited before we explained them all behavior aspects required to create DM bot and then experimented with the DM bot only once. Secondly, students had become easily bored as the ultimate task was the same for every class even though details differed. Therefore, we decided to restructure the classes and devise new sets of tasks that are assigned to students.
From the teaching point of view, a task is characterized with knowledge (K) that is needed to solve the task and needs to be taught beforehand and with the logic (L) that the student should discover himself while solving the task. The knowledge can be further divided into three parts:

K1) the environmental mechanics involved in the task,
K2) set of behavior primitives (e.g. move, shoot, see-player) and/or higher-level actions provided by the platform (e.g. navigation) required by the solution and how to use them,
K3) reasoning techniques (e.g., non-trivial use of A*) required to solve the task.

To create the behavioral logic, a student should undergo following three steps:

L1) analyze the task,
L2) design the behavior structure,
L3) implement the behavior using the platform.

The (complicated) structure of knowledge and logic required for a deathmatch bot is shown in Fig. 1.

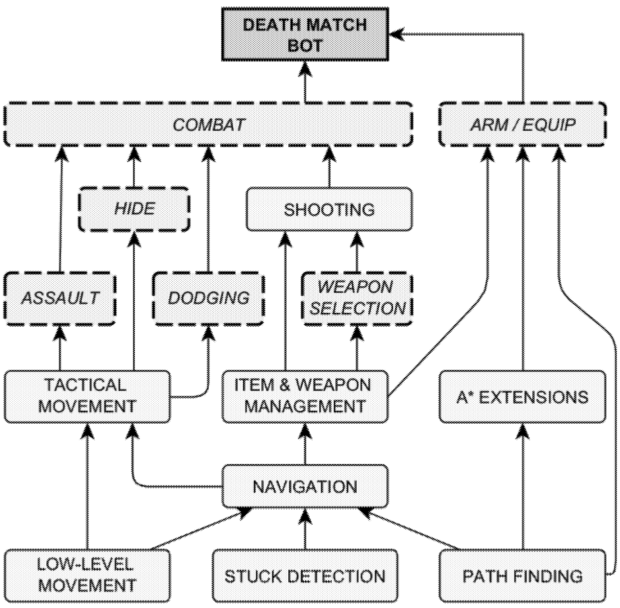


Figure 1: Structure of knowledge (full lines) and behavioral logic (dashed lines, italics) required for DM bot and their classroom dependencies.

The challenge is to choose the proper set of tasks. The tasks should be of increasing complexity and connected with each other so that students can consolidate their knowledge and skills by reusing them in portions of the more complex tasks.
The task also needs to allow for incremental buildup of the necessary knowledge and that lets the students to complete all logic development steps in reasonable time so that they stay focused and motivated. The knowledge and logic design for a task should be dealt with in the same class or in close succession: Separating the knowledge from its use in agent logic leads to poor learning performance as the theory is no longer supported by practice and separating the individual logic development steps prevents the student from getting immediate feedback on the quality of his design (e.g. wrong analysis may not be spotted until the student fails at implementing it).
We noted that the students struggled the most with the very introduction to the platform and with navigation and movement. Moreover, once classes on those topics were over, only few steps remained to their first attempts at DM bot — although there was still a lot of knowledge to master, students were already familiar with the overall design and philosophy of the Pogamut platform and thus progressed faster.
To conclude, we needed to devise tasks that would cover the basics of Pogamut and navigation and would need no other knowledge.

THE ENVIRONMENTS

Based on the requirements identified in the previous section, we have designed two environments inspired by children's games. Both environments were run in UT 2004 and the game logic was implemented as an extension to the Pogamut platform.

Tag! Game

Tag! is inspired by classical children game, where one player is the "chaser" and tries to catch other players (labeled here as "evaders") by touching them. Once a player is caught, the chaser role is passed to the caught player. The former chaser becomes an evader, but a "no-tags-back" rule is enforced: the former chaser is immune (cannot become chaser again) until the role is passed to yet another player.

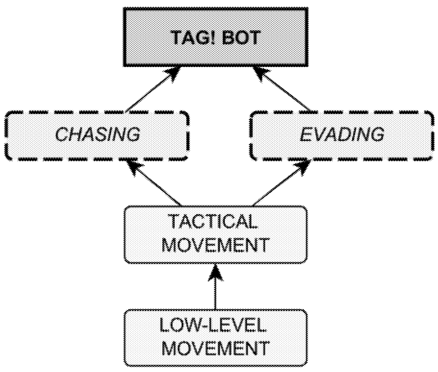


Figure 2: Structure of knowledge (full lines) and behavioral logic (dashed lines, italics) required for Tag! bot and their classroom dependencies.

The game has many interesting properties. A) The simplest strategy for chaser resp. evader is to run directly to chosen evader resp. run directly away from the chaser, therefore students can create simple Tag! bots very quickly. B) The game works well with a very simple environment (e.g. flat square or rectangle), therefore students do not need to be taught about environment representation and navigation. C) Having move, dodge and jump commands is enough to create different chasing and evading strategies (even in the simple environments), thus it provides room for students' creativity. D) The Tag! Game can be scored (how many times a bot has become the chaser, how fast a bot can pass the chaser role on) and therefore it is possible to conduct tournaments between student bots.

For reasons above, the Tag! Game makes a good candidate for the first real task for students. Technicalities required to solve the task (K1-3) can be explained quickly (45 minutes) along with coding the simplest Tag! bot implementation (L1-L3) together with students (45 minutes).

We played Tag! with four bots --- this is the minimum to allow for complex strategies to be taken: the chaser always has two possible targets (the last evader is immune to him). Since all bots move at the same speed, "tactical movement" (computing move vectors based on positions of other bots and other context) is vital factor of success. The classroom dependencies of Tag! bot are shown in Fig. 2.

The students are shown basic vector math hints and asked to create "tactical movement" separately for the chaser and evader roles. Even though a simple game, students are very creative at this part. Interestingly, the move, dodge and jump commands create a very large space of strategies and counter-strategies. One of the keys to success is the ability to predict future positions of the other bots, which is especially beneficial to the chaser who may "cut corners" to catch the evader more quickly. On the other hand a bot may decide to exploit the opponent's prediction mechanism and gain advantage by behaving unpredictably.

For instance, the bot can speed its running using dodges and jumps. However, it cannot jump too frequently as the bot cannot control its movement while in the air, therefore the opponent can reliably predict the bots position as soon as it notices that the bot is in the air. While evading, the evader can try to run smoothly in circles, which creates an endless evading behavior a simple "direct running chaser" cannot beat. It is also beneficial to implement timeouts to chasing behavior or chase only those bots one has successfully tagged before.

The bots are not aware of other bots that they do not see directly — they only know the location where they were seen for the last time — so improvements can be made by controlling the direction of bot's gaze to maximize the amount of information available for decision making. Tag! also serves as a good exercise of basic vector math, which is necessary for many more complex decisions in 3D environments.

Hide & Seek Game

Another classical children game is Hide & Seek, where one player is the "seeker" and tries to seek out others that are hiding within the environment (labeled here as "hiders"). We have implemented the variant played most commonly in Czech Republic: All players start at a designated base. The hiders are given a short time to hide, while the seeker is blindfolded (does not receive vision data and cannot issue commands). To score a point, seeker needs to find another player (see him) and then return to the base to "ground" him. However, if a hider manages to reach the base before he is grounded, he scores a point and cannot be grounded anymore. The game ends once all hiders are grounded or have reached the base or due to timeout.

Whereas Tag! focuses on low-level movement, Hide & Seek focuses on the environment representation and reasoning (path-finding and line-of-sight) and navigation (path-following). Even though the game is more complex, it still retains interesting properties. A) The simplest strategy for a seeker is to run randomly around the environment until it spots a hider, then return to the base. Analogically, for a hider, the simplest strategy is to hide at a random place and try to reach the base as soon as hider finish counting. B) The environment reasoning complexity can be lowered by, again, designing a simple environment (e.g. grid-based 2D maze without rooms). C) The Hide & Seek Game can be scored (according to the number of found hiders or the number of escapes).

One of the maps we used is shown in Fig. 3.

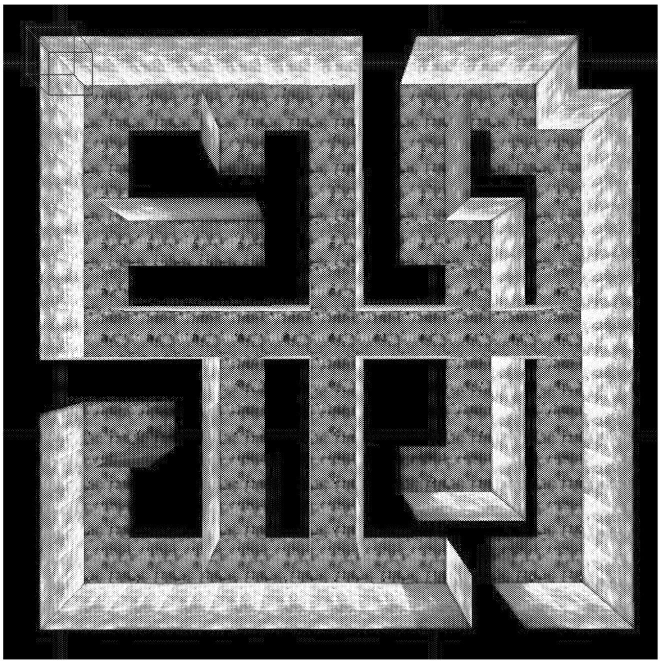


Figure 3: A sample map for Hide & Seek. The base is at the crossroads in the center of the map.

Table 1: Categories of qualitative feedback on tournaments held during the course and count of answers that belong to the category.

Category & Sample Answers	Count
Good motivation for the homework *The tournaments motivated me to try inventing unusual solutions to the problems.*	6
Creates competitive environment (in a good sense of the word) *It was nice to see some results of my work on bots and have some comparison with other students.* *It was an excellent idea to compare the bots. Everyone could see, what all can be achieved.* *Some of the resulting videos were funny.*	6
Spice the course up *Spiced the course up, nice to do a homework that is somewhat more practical.*	5
Interesting, but no time to compete *Tournaments were interesting, but due to my other activities, I did not have a time to compete truly.*	4
Interesting, but not motivating *Tournaments were interesting, but they did not motivate me really.*	1
Other *Comments about possible improvements of tournaments.*	5

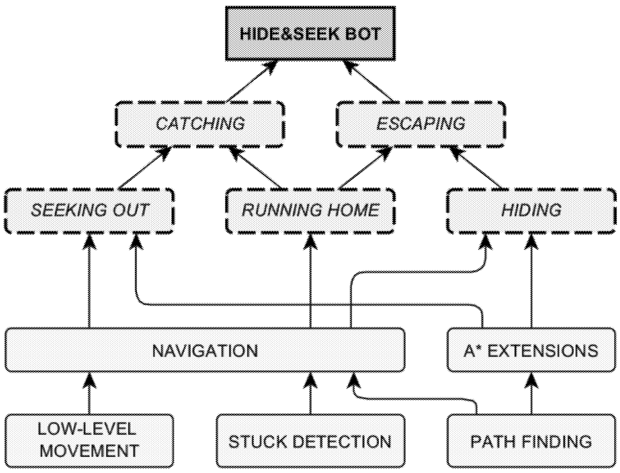


Figure 4: Structure of knowledge (full lines) and behavioral logic (dashed lines, italics) required for Hide & Seek bot and their classroom dependencies.

It is more time demanding to teach technicalities that are required to solve the task. Firstly, the navigation along with environment representation and simple path-finding needs to be taught and practiced (90 minutes). Then, the class detailing A* implementation in Pogamut and options for its adaption is required in order to provide students with tools for obtaining multiple paths towards target or searching for paths that lead through places not visible to the seeker (90 minutes).

The classroom dependencies of Hide & Seek are shown in Fig. 4. Note that low-level movement was already taught in the Tag! scenario and after that students have understood the necessary prerequisites of Tag! and Hide & Seek bots, only item and weapon management and shooting need to be explained before they can start working on a death match bot (see Fig. 1).

Hide & Seek also provides a rich strategy space. In order to gain information about positions of other players, the seeker needs to roam away from the base, making it possible that a hider reaches the base safely. Hiders on the other hand may try to spot the seeker without being spotted and have to decide when there is a reasonable chance they will make it to the base. Thus in both roles, the bot needs to balance the risks and gains of its behavior.

EVALUATION

The Tag! and Hide & Seek Games were introduced to the course curriculum in 2013 and received positive feedback. Therefore, the course ran during 2014 without changes. Here we present data from these two years from the total of 27 students (26 males, 1 female, Czechs) out of which 7 were undergraduate and 20 were postgraduate students. Importantly, the overall performance of students during the classes and the final exam improved significantly. The final exam was almost the same for the last three runs of the course and involved coding of two complex behaviors in a lab. In 2012, the average time to complete the first behavior was 2 hours, 50 minutes (sd: 30 minutes) and only two students completed the second behavior. In 2013 and 2014, the average for first task dropped to 1 hour, 29 minutes (sd: 31 minutes). All students also finished the second behavior, on average in 3 hours, 15 minutes (sd: 47 minutes). Although multiple factors may be involved (most notably innovations to Pogamut platform and prior knowledge of the tasks gained from students from previous year), the results are encouraging.

Data about tournaments was gathered through questionnaires that were part of the final exam of the course. Students were asked three quantitative questions related to the tournaments held throughout the course[1]:

1. Did you find tournaments (organized during the practice lessons) interesting? (11-Likert like scale, 0 - not at all, 5 - somewhat interesting, 10 - very interesting). The average score was 8.5 (sd: 1.69).

[1] Tournament results including all compiled bots, UT2004 replays as well as some videos can be downloaded from: pogamut.cuni.cz/pogamut-devel/doku.php?id=human-like_artifical_agents_2013-14_summer_semester

2. Did you put extra effort into homeworks that were used for tournaments? (11-Likert like scale, 0 - not at all, 5 - some effort, 10 - I have tried my best) The average score was 6.5 (sd: 1.6).
3. How many extra hours have you invested into doing your homework for a single tournament (at average)? The average was 4.25 hours (sd: 1.51).

Qualitative feedback ("Give any comments on the tournaments") answers could have been clustered to categories displayed in Table 1. The qualitative answers were very positive, with only one student that explicitly stated that he was not motivated by tournaments and over 60% of students explicitly expressing positive impacts on motivation.

CONCLUSIONS

We have presented two educational games suitable for teaching basics of IVA development. We have shown that tournaments in both games helped to motivate students to spend extra hours (over 4 on average) working on the assignment. The games themselves were instrumental in keeping students' attention and helped them learn how to navigate agents in virtual environments, reason about the environment and use vector math to calculate trajectories.
As a future work we intend to further ease development of bots for both games and start tournaments for high school student interested in game AI.

ACKNOWLEDGEMENTS

Human data were collected with APA principles in mind. This research is supported by the Czech Science Foundation under the contract P103/10/1287 (GACR), by student grants GA UK No. 655012/2012/A-INF/MFF and 559813/2013/A-INF/MFF. This research is partially supported by SVV project number 267 314.

REFERENCES

Bayliss J.D., 2009. "Using Games in Introductory Courses: Tips from the Trenches". In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*. ACM, New York, NY, USA, SIGCSE '09, 337-341.

Bayzick J.; Askins B.; Kalafut S.; and Spear M., 2013. "Reading Mobile Games Throughout the Curriculum". In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*. ACM, New York, NY, USA, SIGCSE '13, 209-214.

Bezakova I.; Heliotis J.E.; and Strout S.P., 2013. "Board Game Strategies in Introductory Computer Science". In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*. ACM, New York, NY, USA, SIGCSE '13, 17-22.

Brom C., 2009. "Curricula of the course on modelling behaviour of human and animal-like agents". In *Proceedings of the Frontiers in Science Education Research Conference*. 71-79.

Brooks R.A., 1991. "Intelligence Without Representation". *Artificial Intelligence*, 47, 139-159.

Bryson J.J., 2001. *Intelligence by design: Principles of Modularity and Coordination for Engineering Complex Adaptive Agent*. Ph.D. thesis, MIT, Department of EECS, Cambridge, MA.

Comenius J.A., 1648. Novissima Linguarum Methodus. Publisher unknown, Leszno, Poland, chap. Methodi linguarum novissimae fundamentum, Ars Didactica.

Decker A. and Lawley E.L., 2013. "Life's a Game and the Game of Life: How Making a Game out of It Can Change Student Behavior". In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*. ACM, New York, NY, USA, SIGCSE '13, 233-238.

DeNero J. and Klein D., 2010. "Teaching introductory artificial intelligence with pac-man". In *Proceedings of the Symposium on Educational Advances in Artificial Intelligence*.

Gemrot J.; Kadlec R.; Bída M.; Burkert O.; Píbil R.; Havlíček J.; Zemčák L.; Šimlovič J.; Vansa R.; Štolba M.; Plch T.; and Brom C., 2009. "Pogamut 3 Can Assist Developers in Building AI (Not Only) for Their Videogame Agents". In F. Dignum; J. Bradshaw; B. Silverman; and W. Doesburg (Eds.), *Agents for Games and Simulations*, Springer-Verlag, LNCS 5920. 1-15

Georgeff M.; Pell B.; Pollack M.; Tambe M.; and Wooldridge M., 1999. "The belief-desire-intention model of agency". In *Intelligent Agents V: Agents Theories, Architectures, and Languages*, Springer. 1-10.

Kafai Y.; Griffin J.; Burke Q.; Slattery M.; Fields D.; Powell R.; Grab M.; Davidson S.; and Sun J., 2013. "A Cascading Mentoring Pedagogy in a CS Service Learning Course to Broaden Participation and Perceptions". In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*. ACM, New York, NY, USA, SIGCSE '13, 101-106.

Lockwood K. and Esselstein R., 2013. "The Inverted Classroom and the CS Curriculum". In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*. ACM, New York, NY, USA, SIGCSE '13, 113-118.

McGovern A.; Tidwell Z.; and Rushing D., 2011. "Teaching introductory artificial intelligence through java-based games". In *AAAI Symposium on Educational Advances in Artificial Intelligence*, North America.

Porter L. and Simon B., 2013. "Retaining Nearly One-third More Majors with a Trio of Instructional Best Practices in CS1". In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*. ACM, New York, NY, USA, SIGCSE '13, 165-170.

WEB REFERENCES

Epic Games Inc., 2004. "Unreal Tournament 2004". www.unrealtournament.com. Last checked: 2014-05-25.

Epic Games Inc., 2007. "Unreal Tournament 3". www.unrealtournament.com. Last checked: 2014-05-25.

BIOGRAPHY

JAKUB GEMROT was born in Havířov, Czech Republic and went to the Charles University in Prague where we studied artificial intelligence and graduated in 2009. After the study he participated in several game development projects as an AI consultant, the most notably in the upcoming RPG Kingdom Come: Deliverence. He is one of the main authors of the Pogamut framework, which he actively maintains, and is currently finishing his doctoral thesis on controlling of intelligent virtual agents.

A Communication Tool to Support Caretaking of Senior Citizens

Rajeeka Ponrasa, Teemu Jääskeläinen, Hannu Raappana, Wang Jizhe, Eeva Leinonen, Petri Pulli
University of Oulu
Pentti Kaiteran katu 1
90570, Oulu, Finland

KEYWORDS
Serious gaming, tools, voice interaction, waypoint.

ABSTRACT

New technologies help to think in a new way. How new technologies can help senior citizens suffering from memory problems? This paper introduces a web application tool to fill the gap of health care and technology. The prototype has isometric view as an environment and is designed like a computer game. It helps caregivers to monitor and give instructions to senior citizens. Caregiver can also guide the senior citizens by calling or sending a message. The communication is much easier with the communication tool and it can also help senior citizens with their daily tasks. Caregivers can use the application to access patients' health profile. We also have developed support for a new navigation concept, smart glasses, to guide senior through the specified route.

INTRODUCTION

According to official statistics of Finland (2014) the amount of elderly people is growing quickly and the number of aging people is 83 percent of the total population. The percentage of elderly people will be double or more than double by year 2050 (United nations 2002). There are many reasons which causes the memory disorders and the aging is just a one of the reason. Many factors for example genetics and neural degeneration have a part in causing memory disorders (Rodrigue, Kennedy & Park, 2009). Memory loss can disrupts daily life for example some forgets important dates or events, some may have trouble to follow familiar recipe or driving to a familiar location and other may lose track of dates (Kopelman, 2002).

In this paper we discuss about smart technologies and services, which supports senior citizens in their daily tasks. When the society and technology is progressing at fast speed, how could we provide this new technology to aid senior citizens suffering from memory disease. One of the answers is to provide and produce a new kind of system, which can help the senior citizens with their memory loss. Furthermore these new systems could be used to help the caregivers in their job. The Old Birds is a project concerning elderly people with a condition, which affects in their performing in their daily lives. It is a computer made simulation, which can be used to gain information about senior citizens. Gathered information can then be used to construct different kind of aiding systems to help them getting along with their everyday routines. The Old Birds interface also gives a possibility for the caregivers to take care of these older citizens. The prototype, which is presented in this paper provides caregivers a smart tool for tracing and taking care of senior citizens remotely. It gives caregivers to get information about senior citizens current location and also the detail of health condition. Also relatives of senior citizens can also participate in caretaking. The tool that is presented in this paper can be a solution to this issue.

As the environments differ, a simulation is good way to illustrate the senior citizen walking into a grocery shop, or making a coffee in the kitchen for example. The system like the Old Birds, can be also used to keep track of a senior if she/he gets lost in his/her way to the grocery shop. Furthermore, with the modern day technology it's even possible to keep track of the elderly people's health condition, wirelessly. More sophisticated systems can be produced, like smart glasses, which for example can be used to guide the elderly citizen to his/her destination. The system can also be used to enrich the elderly people's social life, which is very important aspect. The system offers a solution to see if the senior falls down or gets hurt. This way the help is always near when the senior citizen needs it the most.

Yamamoto et al. (2010) describes in their studies about smart services among the elderly people. The amounts of working aged population are growing fast in Finland. It doesn't seem so realistic that there would be enough caretakers to fulfill the elderly needs or problems in the future. Also the amount of the illnesses caused by ageing for example dementia is also increasing. The social health care sector needs more new technologies, which can support the elderly people to survive with their daily tasks. The authors of the paper have been developing a ubiquitous technology for the elderly people who are suffering from various degrees of memory problems. They also describes that the purpose of the system is to motivate seniors and help them to maintain social communications. The system can be controlled by tapping the surface and this allows the elderly people to use the system without learning new control methods.

Pulli et al. (2012) have studied the safety navigation of senior citizens in their studies. The concept of smartphone-based navigation with easy-to-use functionality is dedicated to senior citizens and for their relatives. The system should be designed to use outside of the home environment to ease and to support the elderly people's life. The outdoor activities of senior citizens can be presented with the help of virtual safety borders. These borders can be determined in advance by people in charge, for example caregivers and relatives. Relatives and caregivers can observe and follow up movements of senior citizens via interfaces. These kind of proper interfaces can provide users browsing opportunities, abilities to see, review pictures and videos taken by senior citizens mobile cameras and abilities to communicate with the senior citizens. This will give a feeling of safety.

Ikeda et al. (2011) have studied about a smart kitchen system which supports the memory lost elderly people with cooking by remote supporter. The remote supporter could get a visual and verbal feature of the system to help the demented elderly people. There can be many places in the kitchen where even the best camera cannot cover it. The kitchen is also a complex environment and that is why it is hard to be monitored or controlled by a single method. Also remote supporters cannot know the kitchen environment well. Two elderly people participated in a test for making coffee in an unfamiliar place; the system gave them some instructions for making coffee. The instructions were given step by step and they were allowed to ask help from remote assistance, both participants succeeded to do their task by asking some questions from the remote assistance.

The tool of this paper is based on the work of Yamamoto et al. (2010), Pulli et al. (2012) and Ikeda et al. (2011). The game engine runs the 2.5D environment and the primary tool is GPS device, which records the data of senior citizen’s location. We added the communication tool as a new feature to the project. The communication tool allows the caregiver to make a single or multiple call to contact senior persons (Figure 2). Also to ease the communication in situations for example to remind the senior person what to buy from the shop is created with texting interface (Figure 4). Caregiver can call senior citizens to remind about some forgotten tasks and also the caregiver can send a message to senior citizens (Figure 2 and Figure 4). Caregiver can also get necessary information about the senior citizens by moving the mouse on to the avatars (Figure 1). The main idea of Old Birds avatar is to avoid privacy issues and to make the application more interesting for the caregivers. The avatars are designed cartoonish style and they demonstrate inside and outside views of senior citizens (Figure 1). The models of avatars are for both genders and different color options so that caregiver can recognize different avatars easily. We have also made the smart glasses simulation where is the scene of the Oulu city. The smart glasses simulation allows the user to see the map of Oulu as well as the bird’s moving (Figure 5). We felt also necessary to improve the movement system of Old Birds 1.0. The old movement was slow and not so intuitive. For the result to improve movements we created a waypoint system to make the movement much easier and the senior character to move in the Unity 3D environment.

IMPLEMENTATION

In this section, we describe the design and implementation process of the application. The section explains how we have used the resources from the older version of the project, and what we have changed or added in it.

The older version of the project was a Unity 3D project, which included readymade features. These features included 3D models of the birds, buildings, furniture and the indoors of Stockmann. There were some scripts for a basic movement of senior citizen “Marjatta”, represented by the Red Bird avatar, but it was decided that these scripts were insufficient and had to be replaced with a better system to save some time in the future implementation. Old Birds 1.0 also introduced a senior citizen information box that appeared on top of the birds when the user mouse hovers overs them. The feature contains information of the senior such as age, medical information, living status, memory status and technology skills. This feature is still used in Old Birds 2.0. The application environment is carried out with a 2.5D isometric environment simulation, in both Old Birds 1.0 and 2.0. (A monitoring tool to support remote caretaking of senior citizens, 2013)

The basic structure of the Old Birds 1.0 Unity project did not meet the qualifications of the Old Birds 2.0, so we did a major reconstruction of the project. All the JavaScript scripts were translated and rewritten into C#, and the simulations were transferred to their own individual scenes. The scenes were Marjatta's house, Oulu city center and Stockmann department store. The Old Birds 2.0 also lacked in documentation, as there were lacking comments on the code and some of the function names were in foreign language. Because of this reason, it took some time to re-engineer the code. The new project includes extensive documentation of the project code, and the code is rigorously commented.

Old Birds 2.0 environment is done with 2.5D isometric simulation because using isometric view instead of 2D is considered as a part of future project. The environment is divided to home, map and city environment. In the home environment is designed to look like a real home decoration. The walls of the home environment are transparent walls because the user of the web application can monitor the senior citizen through the walls. The design of the environment is important to the caregivers because nothing should block the visibility when the caregivers are monitoring the elderly persons (Figure 7). The map view shows the caregiver where the avatars are located and the map of the city is designed with 2.5D view (Figure 1). The city environment shows the most important buildings in the city (Figure 4).

Information box

The information box will appear every time when moving the mouse over each avatar. It contains health and personal information about the senior citizens and in the current version of the application the following information is included: age, medical information, living status, and status of memory and also senior citizens technology skills (Figure 1).

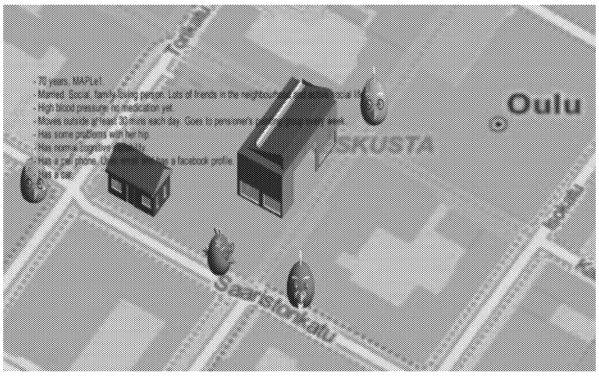


Figure 1: Information box appearing when the mouse is hovered over the chosen avatar

Caregivers' communication tool

A new feature was added to the project, which allows the caregiver to contact the seniors through a caregiver communication tool. The tool allows the caregiver to call or text a senior person through a simulated situation (Figure 2). This tool is useful in situations where a senior is in need of assistance, for example if she/he is lost, hasn't moved in a while or their relatives are worried about him/her.

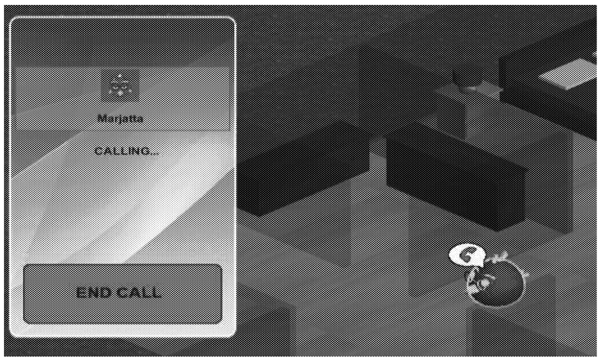


Figure 2: Making a call to a senior through the communication tool

The caregiver communication tool also allows the caregiver to make a call to many seniors at once (Figure 3). This allows the caregiver to contact groups of seniors for example in a situation where an announcement has to be made. In a situation where a storm is approaching, the caregiver can contact the seniors outdoors and tell them to go home or the nearest shelter. The green outline around the birds in figure 1, 2 and 3 indicates that the bird is selected for a call. The birds are also shown as a list in the caregiver communication tool. The right hand side GUI is a simulation of the seniors phone (Figure 3). In a situation where a senior is demented and might not recall that he/she has a caregiver, we've implemented a simulated picture of the caregiver in the senior's phone, when a call is answered.

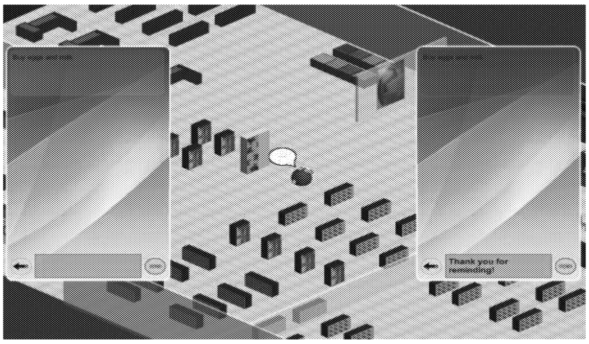


Figure 3: Making a call to multiple seniors

The texting interface (Figure 4) allows easy communication in situations where the senior might not be in immediate need of assistance, but could for example need a reminder of the groceries he/she needs to buy. The senior's phone is simulated in the answering of calls as well as answering a text message (Figure 4). Caregiver can make a call and text message to multiple seniors. This is a useful tool for the caregivers because they can give instructions anytime and anywhere to the senior citizens.

Figure 4: Text messaging to a senior

Smart glasses for senior navigation

To more enhance the simulation of technology in development in the department of information processing in the University of Oulu, we developed a scene with where a senior uses smart glasses. The idea of the smart glasses is to guide the senior through a specified route to a place where he/she needs to be. There are four different LED lights in each lens of the glasses that indicate the direction where the senior needs to go (Figure 5).

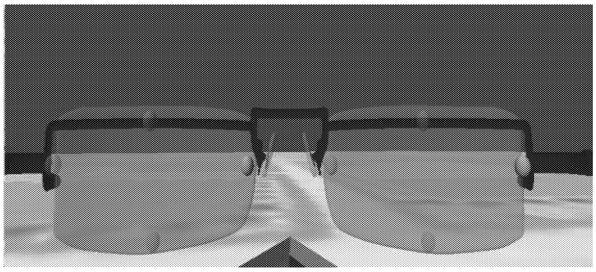


Figure 5: The smart glasses simulation

The smart glasses simulation is done in the Oulu city scene, where the user can see the map of Oulu as well as the birds' movement. There is a first person view with the glasses in the upper right corner of the picture to simulate the flashing LED lights.

Waypoint system for character movement

We felt it necessary to improve the movement system of Old Birds 1.0 dramatically. The old movement system was hardcoded in the scripts, which was not intuitive and was slow to implement. A waypoint system was created to make it easier to create new simulation where the senior character needs to move in Unity 3D environment. As shown in figure 6, it is easy to place the waypoints for the character to follow. The waypoints are the black dots, which are only shown as gizmos in the scene editor, and not the actual simulation.

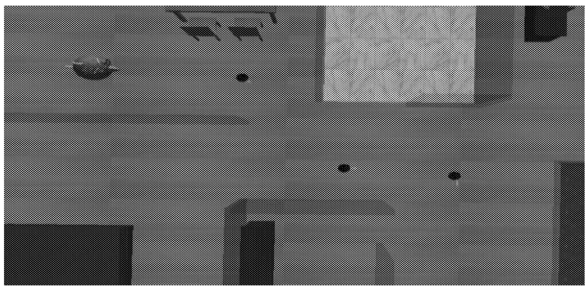


Figure 6: Moving a character using the waypoints

The waypoint system is used in both, the smart glasses and Marjatta's Stockmann visit simulations. The scene changing, for example from Marjatta's house to Oulu city scene is done through a trigger. When Marjatta touches the trigger (Figure 7), the scene is automatically changed to a scene defined in the trigger object.

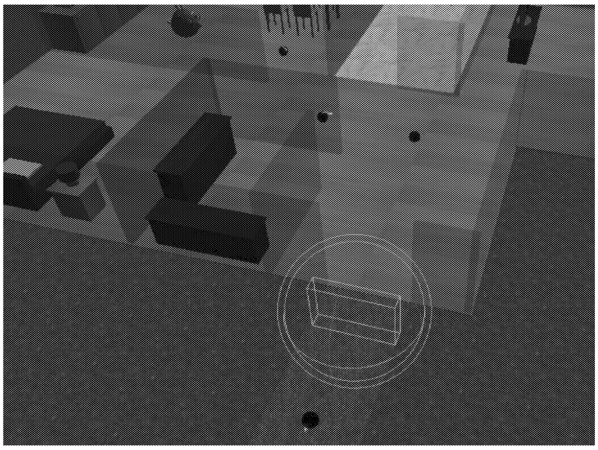


Figure 7: Changing the scene through a trigger

CONCLUSIONS AND FUTURE WORK

New technologies offer hope for many issues and this paper's web application tool fills the gap between health care and technology. In the future the waypoint system should be enhanced with new features. These features could include an editor tool where the developer could click on the ground to add a new waypoint which would automatically be assigned to the wanted character, rather than copying a waypoint object and assign it manually. The waypoints could also make the character go faster, slower or even stop for a while. In the future it is also important that the avatars are modified with a new look. The smart glass interface should be designed further and also the smartphone application for the communication tool should be also provided with a new design.

ACKNOWLEDGEMENTS

This contribution is based on results from Spring 2014 semester student project at University of Oulu, Department of Information Processing Science. We thank our supervisors Petri Pulli, Eeva Leinonen, Zeeshan Asghar, Aryan Firouzian and Tonja Molin-Juustila for ideas and guidance. Support from Academy of Finland and JST, Japan jointly funded ASTS project if acknowledged.

REFERENCES

Aryan Firouzian and Henna Nissinen, 2013, A monitoring tool to support remote caretaking of senior citizens. (GAMEON'13 paper), pp. 1- 4.

Ikeda, S., Asghar, Z., Hyry, J., Pulli, P., Pitkanen, A., Kato, H., Remote Assistance Using Visual Prompts for Demented Elderly in Cooking, (2011), 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies.

Jacelon, C. S., & Hanson, A. (2013). Older adults' participation in the development of smart environments: An integrated review of the literature. Geriatric nursing, 34, 116-121. Retrieved from: http://dx.doi.org/10.1016/j.gerinurse.2012.11.001

Kopelman, M. D., (2002). Disorders of memory. *Brain: A Journal of Neurology.* 125*(10),* 2152-2190.

Pulli, P., Asghar, Z., Siitonen, M., Niskala, R., Leinonen, E., Pitkänen, A., Lehtonen, J., Hyry, J., Kramar, V. & Korhonen, M. (2012) Mobile Augmented Teleguidance-based Safety Navigation Concept for Senior Citizens. *UAS Journal* (2), 1-9.

Rodrigue, K.M., Kennedy, K.M., & Park D.C. (2009). Beta-Amyloid Deposition and the Aging Brain. *Neuropsychological Review.*

United Nations. 2002. *World population ageing, 1950-2050*. New York: United Nations. pp. Retrieved from: http://www.un.org/esa/population/publications/worldageing19502050/

Yamamoto, G., Hyry, J., Pouke, M., Metso, A., Hickey, S. & Pulli, P. (2010). Senior Citizens' Interaction with Smart Ambient Environment.

Visual Prompts for Demented Elderly in Cooking, (2011), 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies.

Official Statistics of Finland. 2012. *Projection for the number of persons aged 65 or over slightly lower.* Retrieved September 15, 2013, from:http://tilastokeskus.fi/til/vaenn/2012/vaenn_2012_2012-09-28_tie_001_en.html.

DEVELOPMENT OF A GAME WITH KINECT FOR THE INCLUSION OF VISUALLY IMPAIRED

Paulo Roberto C. Faustino
Matheus M. Ramalho
Gabriel B. S. M. Moreira
Alex F. V. Machado
Lucas D. Silva
Instituto Federal do Sudeste de Minas Gerais, Departmento de Computação, Brasil.

KEYWORDS

Accessible Games, Games for Kinect, Audio Game

ABSTRACT

The most current and profitable electronic games have great graphics, well prepared stories and strategies with varying levels of difficulty that instigate players from all ages. However, all of these features are irrelevant to the visually impaired, limiting the public regarding electronic games. Based on this fact, audio games started being developed aiming at enlarging the access of these people to the electronic games market and at the same time promoting inclusion. In this article we present, step by step, the development of the audio game "Fuga", implemented in Unity3D with 3D sound techniques using Kinect, exploring a new perspective of assistive technology.

INTRODUCTION

The electronic games gradually have grown consequently with technology's evolution in recent years, and with that, the existence of audio games started to become a reality, although not widespread.

According to (Cordeiro, 2011) "Different brands compete between themselves by presenting solutions to the level of physical interfaces, processing power, storage and resolution (sound and image), which impacts in the own games conception, allowing them to explore new forms of interaction and increase the levels of realism and immersion achieved so far." This highlights the fact that the game industry invest in visually expensive and innovative games, but they're not accessible.

"In 10 years, over 400 audiogames have been developed, which is very small as compared to video games." affirm (Archambault et. al. 2007). With this statement, we notice that the collection of games geared for the blind are relatively small compared to other games with exclusively graphical interfaces.

Considering the small number of games already developed and that not all of the visually impaired have access to affordable technology, we notice that it would be a great challenge create a game for their entertainment. Therefore, based on research in the technological involvement of the visually impaired with accessible tools, we explored the basic of the sounds 3D concept for the development of the "Fuga" application.

Concepts about ergonomics for the correct use of Kinect were also studied, as well as the need of using basic graphical interface, quoted on the next subsection.

The existence of accessible electronic games is very important regarding technological inclusion, expanding the access of these resources visually impaired people, total impaired or partial. Such resources are included in the term of assistive technology, according to (Bersch and Tonolli 2006). Assistive Technology is still a new term, used to identify all of the resources and services that contribute to provide or enlarge functional abilities of the impaired and consequently promote independent living and inclusion.

According to WHO (World Health Organization), there are 39 million visually impaired people in the world and other 246 million suffer from moderate or severe loss of the sight. 90% of these people live in developing countries, thus highlighting that there's a market that should be explored, even though not all of the visually impaired having access to technology.

This article proposes the development of the audio game controlled by Kinect called "Fuga" to fill a part of this underexplored industry.

Here we described the engineering system and main challenges and difficulties for its implementation. Lastly, we evaluated and tested the developed application. With this assignment we tried to promote the inclusion of users who, just like everyone, have the right of all kinds of digital fun.

RELATED WORK

In the field of audio games, even having a relatively small number compared to visual games, it's possible to find a lot of categories of accessible games, educational games, simulators, musicals, reasoning, first-person, among others, using concepts and varied tools to build a certain kind of game. Given that the use of auditory interface is essential, others tools become additional to the making of an audio game.

Works developed whose focus is to use the kinect, are important be studied to observe pros and cons when you have two or more ways to play, despite the use of kinect not be the main focus of this work, has much importance in order to innovation, and will have a topic explaining the choice of the tool and selected movements.

In (Soares et al. 2013), a game (HIT CO-OP) collaborative, that is, a game in which two or more players play together benefiting one each other, was developed for use the Kinect. The goal of the game is getting the maximum points obtained collaboratively between the players. The authors state at the completion of the work that "The ability to keep players playing the longest owes much to the ability to interact with games. The Kinect has emerged as a new way to dominate them. "Just as mentioned by the authors of HIT CO-OP (Soares at. Alma 2013), in Fuga the use of kinect also was thought this way because despite using only sound to guide, the player interacts directly with the imaginary scenario using the body as an action factor for the movement of the character.

Some existing accessible games like Finger Dance (Miller at al. 2007), Guitar Hero Enabling for Blind (Yuan and Folmer 2008), Dinobas (Alves at al. 2014) and Blind Counter-Strike (Costa 2013), share the same requirements for its constructions, which is the use of the audio. However, there are other technologies that can be integrated to extras tools for the development of the game.

For example, (Costa 2013) presents a FPS (First Person Shooter) audio game, whose goal is consisted in completing five levels and each level has a certain amount of enemies to be defeated. The player detects the enemies with the assistance of the 3D sound. For its development, the perspective of the character's movement is related to the sound source of the enemy. According to the given steps, it's possible to perceive the opponent's position. In this same idea, the game "Fuga" uses the concept of 3D sound and dynamic movements for detecting the best path and identifying sound objects throughout the map that character must travel, including objects that identify obstacles.

Besides the games created from initial planning, there are adapted games generated from visual games that already exist, where there's some increase to make them accessible. An example of adaptive game is the platform game quoted in (Oren 2007) which contains the maximum of elements from the existing platform classics, having features such as jumps through void places and obstacles. The work with the audio in (Oren 2007) is notorious throughout the whole course of its execution. When the user gets closer from the edge of a platform, a sound is generated indicating the movement that should be done next: jumping to the following edge or going down another platform. The creation of enemies in (Oren 2007) also uses audio tracks that are similar to the audio signals from the classic platform games. When an enemy approaches the character, horizontally or vertically, it emits a sound that increases the volume and the execution speed until a specific sound is activated, forcing the player to make the necessary decision.

In "Fuga", the barriers also include inserted audio tracks, but it wasn't necessary to use volume functions according to the distance between the character and the obstacles, because when a character encounter some, the timing is set for them to make a decision that'll keep them in the game.

The obstacles in "Fuga" are the walls that form some kind of maze, and the audio tracks are inserted just in the paths that the character should travel. When the character gets to the end of a path, it's possible that they may set which way to turn but there's still the option to move the character forward, causing a collision with the object in front of them (which is the wall) and emitting a collision sound, showing them that they should choose another way.

DEVELOPMENT OF THE "FUGA" GAME

In this section, we'll give the sequences of the game's development, encompassing discussed methods, used software, auxiliary websites, definitions of the levels of each level and the choice of the game's mechanics.

The Game's Story

The purpose of the game is to leave a spot and get to another one while being guided by a sound, going through some obstacles and being aware of the time, simultaneously. With this guideline, we create a story that contextualizes with the original goals.

The character is a scientist who works in an unknown research for many years and that's why he has access to all of the labs in the building. In the moment that he's in the last lab, far from the exit, an accident happens in one of the sectors which causes a poisonous gas to be spread through the entire building. The lab the character is at is the only one that provides momentary protection until he can get a gas mask. However, when his eyes get exposed to the gas, he completely loses his sight. The decontamination system from the building is activated and sets the timing up for the decontamination of each lab. The character must find the exit of each lab, guided by the sound made by the sirens on the way to the exit.

This market of accessible games has as a positive point the ease of generating innovations that don't demand major research efforts, given that it was largely unexplored around the world and mainly in Brazil. (de Oliveira 2008).

There are similar stories in the market, but we're in the context of accessible games, where a lot of things that exist in the conventional games become original in the field of the 3D sound usage.

Tutorial Level

The primary process to start a game is to teach the user to execute the basic commands and present the interface to be used. This way, it is provided to those who are playing a greater control over the challenges they're going to face throughout the game.

Given that, the introduction level becomes essential on adapting the visually impaired to this new world of accessible games through the interface that explores their hearing.

As shown on figure 1, the tutorial level doesn't have a great difficulty because its main challenge is that the player control the movement's mechanic and understands the interface.

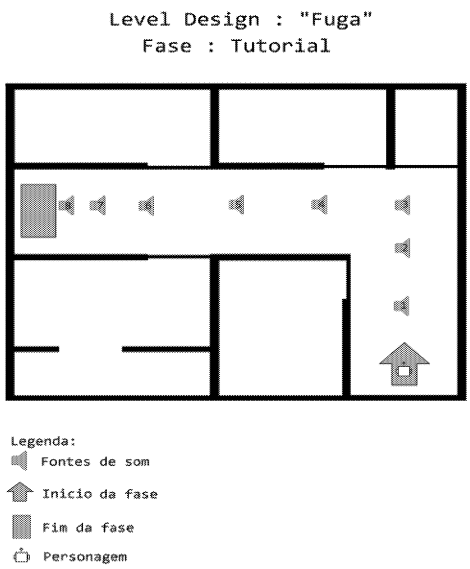


Figure 1: Tutorial Level of the Fuga Game.

3D Sound Usage

In a conventional electronic game, the 2D audio treatment helps us through some parts throughout the game, acting on the perspective of giving emotion to the levels, on alerting us when we get some special item and when we are hit by an enemy. It is not possible for all of these functions that use the 2D audio to find objects or even the character in the game, thus making a real perception of location of the character in the virtual context impossible.

Based on this idea, some developers adopted the use of the 3D sound concept to give a sense of location in their games, as assistance or as a crucial factor to the conclusion of their respective levels.

In the "Fuga" game, the use of 3D sound was vital for the making of the levels. The player's goal is simple, he's in a maze and he must find the exit, being guided by the audio emitted while he goes through the halls according to his route, as exemplified in Figure 2.

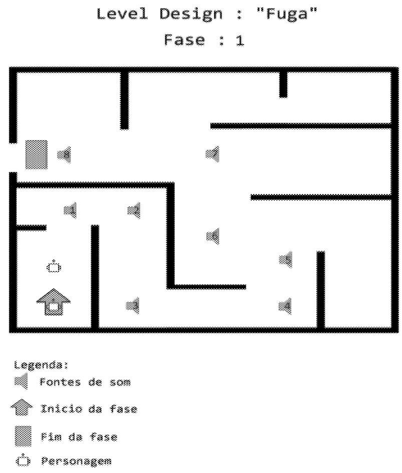


Figure 2: Level one of "Fuga"

On the beginning of each level, the player evenly listens to the instructions of what he's supposed to do or what he'll find in the course of movements. The 3D sounds is fundamental because from the moment that the character moves, the uniform sound starts to vary and he must initiate his attempt of perception to guide himself and move the character from his spot till the exit.

To work with the idea of the 3D sound interacting according to the present graphic interface, we used the Unity 3D tool for its easy handling and for being a very intuitive tool.

Creation and Editing of the Game's Audio

For the audio treatment, we adhere to the "MixCraft 6 ©" tool to edit, cut, add effects and leave the audio ready to be inserted in the making of the game on Unity.

There are two types of audio in the "Fuga" game: the auxiliary audio and the alert audio.

Auxiliary Audio

A voice that helps the player with all the necessary instructions, including information about the mechanics and menu options alternations that interact with the player and directly with the character during the levels.

For the making of this kind of audio, we created the lines giving total attention to the diction because the sentences need to be in perfect understanding, with reasonable pauses between the words. We used the "Soar Mp3" website, a free tool that converts texts into audios in Portuguese. All of the audio created in this website was loaded with normal quality.

After the conversion and loading of the audios, we used the MixCraft 6 © tool for the editing stage. The effects Classic Flanger, EZQ Equalizer and VocalZap e Acoustica – Reverb were applied in each auxiliary audio.

Alert Audio

This kind of audio includes uniform audios and 3D audios. From it, the player is forced to make his own decisions to win the game, after listening to the instructions in the beginning of each level.

The siren sound is the audio that works the ability of the player's sound location. In the halls where the character has to pass till he finds the exit, the audios are inserted only in the path that leads him to victory, therefore, he'll always know which way to guide the character just by paying attention and being agile enough to fulfil the stipulated time and divert the obstacles.

In the game's development, the alert siren audio was inserted in barriers created to occupy almost the whole width of the halls and when the object that represents the character passes these barriers, they disappear along with the audio. That makes possible that the following sound is perceived by the user.

There are also the uniform fire and electricity sounds, which are two obstacles on levels 3 and 4, respectively. These audios were extracted from YouTube videos that are free from copyrights, thus making the use of them in our application possible. After the extraction, we cut and edited them using the MixCraft 6 © tool.

Use of Kinect

"Pushing buttons on a joystick is replaced by the movement of a simplified control associated to the body movements. A natural evolution of this trend was the creation of the peripheral Kinect (Figure 3), constructed by Microsoft to be used with the XBOX 360 console", says (Paula 2008). The use of Kinect makes it possible that we eliminate the need of using a physical control, because it returns, through a set of cameras, an information quite accurate of colour and depth associated to each point. "The apparatus also stands out for developing the position (x, y and z) of a set of points associated to the main joints of the human body (head, hands, elbows, legs, etc)", complements (Paula 2008).

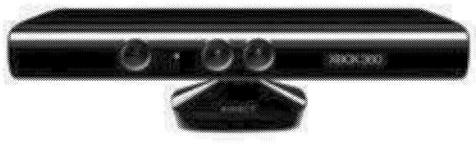


Figure 3: Microsoft Kinect Sensor

The choice of using Kinect for the "Fuga" game was an alternative that we picked to help expanding the access of this tool to the visually impaired because the existence of accessible games is already something not very available, and games that use this technology are even less disposable.

The use of Kinect is not viable in some games, according to the game's context and genre, or even for the own gameplay. In "Fuga", the theme and story of the game allow us the flexibility of developing both a game for joystick and Kinect.

Movement's Mechanic

Kinect provides to the user a better interaction with the developed applications related to it, if compared to the applications that need a physical control. However, for the development of some program or game that uses this technology, it's necessary to be alert to the final usability, like the repetition of gestures or gestures that demand too much from the user or player's body.

The final version of an electronic game developed for Kinect must have practical and comfortable movements. The developer must conciliate practicality, fun and comfort for any person that might use their product.

In "Fuga", the player must only move from a point x to a point y, going through some obstacles and having to divert them. Therefore, for the making of its movement's mechanic, we used a simple gesture to the character's movement.

To move the character from a certain position to other, the player must slightly lift one of their arms forward, not requiring a very big angle regarding their body nor staying with their arms lifted for too long. While the player has their arms lifted forward and there's no obstacle stopping them from keep going, the character will move (Figure 4).

During the character's locomotion, the alert audio of steps is activated, informing the player that the character is moving.

Based on these modelling movements, the gestures to turn the character to the right or to the left were built. During the whole game, the player has the possibility of rotating the character to follow other paths or to make him turn around. Each time that one of the arms is lifted to its respective sides, the character turns in a ninety degrees angle just once, i.e., to turn the character around the player must lift their arms to the side twice. For each turn that the character takes, an alert audio of steps is activated, informing the player that the movement was performed. These are basic movements that can be done in an overly short amount of time of execution.

In the making of the options menu, the player can choose between "Quit Game", "Start Game" and go to "Credits", in which activates an audio where the names of the developers and a brief story of the game are announced. The changing of an option to another is activated using the same mechanics as rotating the character.

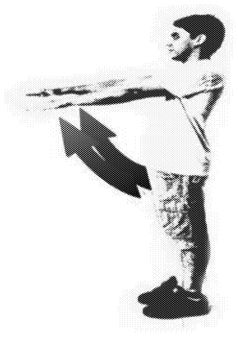


Figure 4: User lifting his arms forward.

By lifting the arm to the left, the player will be moving the box of selection to the left (Figure 5), and that also works for the right side when the right arm is lifted (Figure 6). After listen to the option they want to pick, they simply lift one of the arms forward, using the same mechanics of moving the character forward.

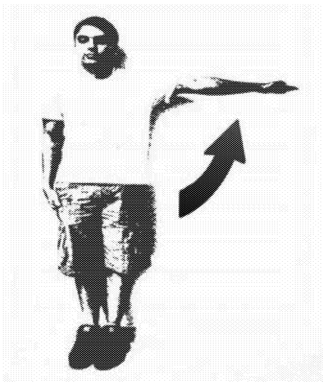


Figure 5: Player lifting his left arm to the left.

Figure 6: Player lifting his right arm to the right.

On the levels two and three, the player faces two types of obstacles that lock the locomotion mechanics of the character. In the exact moment when the mechanic is locked, the audio related to each obstacle is activated and so, the character is forced to use another kind of mechanic: crouch or jump. We inserted, contextualizing with the story of the game, two types of obstacles to raise the difficulty level of the game. The first obstacle refers to a loose energy cable hanging on the hall where the character must pass to get to the exit. In this case, the mechanic is locked, the game establishes some certain amount of time for the player to pass the situation, and otherwise he'll lose the game and restart the level. To unlock this movement, the player must quickly crouch in a small angle that doesn't cause discomfort (Figure 7).

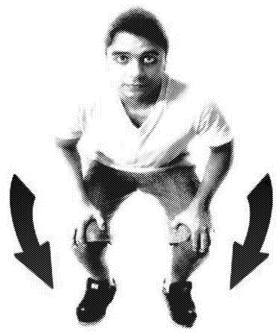


Figure 7: Player crouching.

In the second obstacle, the auxiliary audio indicates the presence of fire in front of the character, also locking the player's mechanics, just like the previous obstacle. However, the movement that unlocks this condition is a simple jump, that doesn't require a very high height to be captured by Kinect (Figure 8).

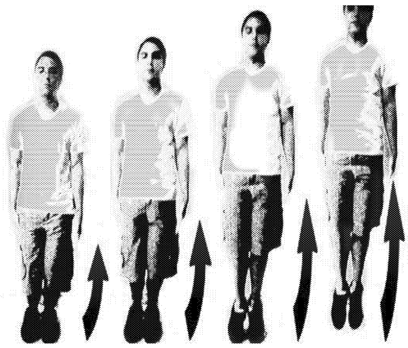


Figure 8: Player jumping.

The mechanics related to the obstacles were chosen aiming at changing the player's movements, breaking out the routine of just lifting the arm forward or to the sides during all levels. A repetitive gesture, besides causing discomfort, can lead to the player's disinterest on playing the game until the end. With these two mechanics, the player not only faces a change of movements, but also a higher level of difficulty regarding the agility to pass through the dangerous situations within a certain time in the game.

Lateralization

"Lateralization is the term used for the conduct of finding sounds to the left and to the right, having the head of the individual as reference. The intracranial localization (feeling of localization of sound inside the head) is called lateralization, opposed to the localization, which is perceived outside the head, according to the direction and distance of the sound source". (Pereira and Schochat 1997).

The individual, having total or partial loss of the sight or not, has the ability to laterize the sound, subconsciously distinguishing what the sound represents, through the distance, frequency or time that the sound gets to each ear. In the game "Fuga", this concept is extremely important to conclude the proposed challenges and complete its levels.

CHALLENGES

In this section, we'll present the immersion concept, interface, requirements and conditions to create an audio game and the challenges faced in the creation of "Fuga".

Immersion concept

Oftentimes, people search for electronic games as a form of escapism from the real world. They can allow people to take on situations and challenges that can't be faced in real life. Therefore, it's important that the developers create a high level of realism in their games, in order to create this illusion for the player. With the help of photography, realistic graphics, 3D expansive environments and spatial audio, this illusion would be possible.

This state of fake reality and feelings is known as "immersion" (Wolf and Perron 2013), that is present in our audio game, proposing to the final user a game that is closer to reality.

Interface

As stated by (Santella 2003), when two or more sources of information meet face-to-face, even if it's the encounter of a person's face with a screen's face, a human user connects with the system and the computer becomes interactive. That is the big difference that separates the tools of a program (software). Tools are meant to be used. They won't just adjust to our purposes, unless it's in a primitive physical sense. A program, unlikely, is a point of contact in which programs connect the human player to the computer processors and these intensify and modify our power of thinking. For that, the human needs to be plugged. For its side, the technology incorporates us (Santella 2003).

The interface in a game is a bridge that connects the user to it, each command that the player activates through the present interface will modify something in the game's progress. In "Fuga", the main interface is the audio, the commands are spoken by the assistant and the user chooses the next step to be taken in the game just by moving according to the given instruction. However, we also use the visual graphic interface, which is useful just for the viewer devoid of visual impairment to track the progress of the game, not influencing in any way the gameplay of the user. Although this is an application for the blind, it can be played by anyone with a normal sight without any privileges to anyone.

Among the requirements to create a game for the visually impaired, the obligation of having a graphical interface is not included, but in certain audio games the use of the interface works as an interactive bridge between the visually impaired and the not visually impaired. This interaction is important because its impact results in the popularization of these kinds of games.

"Fuga" was developed aiming at not only the visually impaired as a public target, but also all of the people capable of sound judgment, even if they have a normal sight, as long as they play with a blindfold.

Tests and Observations

Throughout the development of the audio game, a load of tests were performed for the choice of the best interaction between the user and the interface. One of the difficulties faced was the selection and editing of the audios. This is a determinant factor for the quality of the game, since it's an audio game.

In these tests, the choice of Kinect was right because it was possible to create a simple, effective and realistic mechanic in the moment of the gameplay, that just by using the movements of the body all the commands are accessed inside the game.

Figure 8: Experiments with game Fuga.

As well as the focus of this work is the inclusion of a tool in the technological universe that is an alternative for the visually impaired, tests were conducted with ten visually impaired in "Regional Association of people with disabilities of Barbacena" (Figure 9), where were observed all the important features cited above and properly evaluated. The visually impaired demonstrated ease in gameplay after the first phase followed by the tutorial stage and there was a good acceptance of ergonomics, and the use of 3D sound was characterized as an training of auditory perceptions for them.

Figure 9: Visually impaired the "Associação Regional de Pessoas Portadoras de Deficiência de Barbacena" testing the Audio Game Fuga

Talking with the blind that tested the game, we reinforce our understanding that there really is a huge shortage of virtual applications that go beyond electronic games for them.

The "Audio Game Trail" was a novelty very well received by the blind and the directors of the Association that interested in the idea of having new games and applications developed by us for your benefit.

CONCLUSION AND FUTURE WORK

The idea of creating an audio game is a challenge for the developers, but it brings a lot of benefits for the visually impaired. Between the benefits, there's the familiarization with virtual environments, causing an improved interaction with the technology, the integration of impaired people to the technological means and the diffusion of audio games.

In this application of the game "Fuga", we obtained the previous view of the requirements necessary for its creation, considering the goals that result on the benefits of the creation of an audio game. With technologies such as Kinect, there was a simplification on the gameplay because with a basic body movement, the player doesn't have the need of adaptation to any kind of joystick.

When "Fuga" was finished, we judged viable the creation of complementary levels with difficulty standards compatible with the game's level.

In these new levels, it will be necessary the creation of new auxiliary and alert audios. New challenges will be implanted, still contextualizing with the story of the game and working together with the mechanics, in a way that the usability of this current version keep functioning just as fine.

ACKNOWLEDGEMENTS

The authors acknowledge the Programa de Educação Tutorial of the Ministério da Educação (MEC/SESu/Brazil) financial support guaranteed for the accomplishment of this work.

REFERENCES

De Oliveira, G. A. G. (2012). Desenvolvimento de Jogos Utilizando Sons 3D para deficientes visuais. *Reverte-Revista de Estudos e Reflexões Tecnológicas da Faculdade de Indaiatuba*, (10).

Costa, D. (2013). *Blind counter-strike: um jogo de FPS para deficientes visuais.*

Cordeiro, J. (2011). *Murky Shooting: em busca da primazia sonora nos videojogos.*

De Paula, B. C. (2011). *Adaptando e desenvolvendo jogos para uso com o Microsoft Kinect.* Proceedings of SBGames.

Archambault, D., Ossmann, R., Gaudy, T., & Miesenberger, K. (2007). *Computer games and visually impaired people. Upgrade*, *8*(2), 43-53.

Santela, L. "*Culturas e artes do pós-humano - Da cultura das mídias à cibercultura*". São Paulo – SP, Paulus, 2003.

Pereira, L. D., & Schochat, E. (1997). Manual de avaliação do processamento auditivo central. *São Paulo: Lovise.*

Wolf, M. J., & Perron, B. (Eds.). (2013). *The video game theory reader*. Routledge.

Yuan, B., & Folmer, E. (2008, October). *Blind hero: enabling guitar hero for the visually impaired.* In Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility (pp. 169-176). ACM.

Miller, D., Parecki, A., & Douglas, S. A. (2007, October). *Finger dance: a sound game for blind people.* In *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility* (pp. 253-254). ACM.

Alves, A. G., Hostins, R. C. L., Santos, M. A., Frisoni, B. C., Cipriani, M., Bianchini, P., & Moreira, G. F. (2014). *Jogos digitais inclusivos:" Com o Dino todos podem jogar"*. Anais do Computer on the Beach, p-204.

Soares, G. C., Mossmann, J. B., & Bez, M. R., *HIT CO-OP– Um Jogo Colaborativo Utilizando Kinect*. 2013

Oren, M. A. (2007, April). *Speed sonic across the span: building a platform audio game*. In CHI'07 Extended Abstracts on Human Factors in Computing Systems (pp. 2231-2236). ACM.

Bersch, R. (2008). *Introdução à tecnologia assistiva. Porto Alegre: Centro Especializado em Desenvolvimento Infantil [CEDI]*.

Copyright © 2014 Acoustica, Inc. All rights reserved. Music Recording Software, *MixCraft 6*, Yosemite Valley.

WEB REFERENCES

Soar MP3 – Conversor online de texto e em áudio. 12 de julho de 2013, http://www.soarmp3.com.

ONUbr - Organização das Nações Unidas Brasil, 10 de outubro de 2013, http://www.onu.org.br.

BIOGRAPHIES

PAULO ROBERTO C. FAUSTINO studied computer science at the IFSUDESTE-MG. He is a research assistant at the group of the Programa de Educação Tutorial of Ministério da Educação (PET-MEC).

MATHEUS M. RAMALHO studied computer science at the IFSUDESTE-MG. He is a research assistant at the group of the Programa de Educação Tutorial of Ministério da Educação (PET-MEC).

GABRIEL B. S. M. MOREIRA studied computer science at the IFSUDESTE-MG. He is a research assistant at the group of the Programa de Educação Tutorial of Ministério da Educação (PET-MEC).

ALEX F. V. MACHADO has PhD in Computer Science at the Universidade Federal Fluminense. He is professor of Bachelor of Computer Science at IFSUDESTE-MG. and Coordinator of the Laboratory for Interactive Multimedia IFSUDESTE-MG, of the Programa de Educação Tutorial of Ministério da Educação (PET-MEC).

LUCAS D. SILVA studied computer science at the IFSUDESTE-MG. He is a research assistant at the group of the Programa de Educação Tutorial of Ministério da Educação (PET-MEC).

GAME
AI

SEMANTIC STRUCTURES FOR RTS ARMY PREDICTION

Andreas Stiegler
Stuttgart Media University
Nobelstraße 10
70569 Stuttgart, Germany
E-mail: stiegler@hdm-stuttgart.de

Daniel Livingstone
Digital Design Studio
Glasgow School of Art
G3 6RQ Glasgow, Scotland
E-mail: d.livingstone@gsa.ac.uk

KEYWORDS
Semantic Structures, StarCraft, Opponent Modelling, Cooperation, Colaboration, RTS, Reasoning

ABSTRACT

In this paper we describe a semantic structure based approach for AI for Real Time Strategy games. The semantic structures provide a shared knowledge base across the different components of an RTS AI. We outline how the a semantic structure works in conjunction with decision making algorithms such as utility systems. We also introduce a prototype implemented in StarCraft II to illustrate how the semantic structures may be used in practice, and provide some detail on how this is implemented in the chosen game platform. Tests demonstrate that the AI is able to choose actions according to high level strategies adapting according to the current game state. Additionally the test conditions used additionally involve human-AI cooperation. Finally, we present an evaluation of the AI, making a detailed study of a typical instance of game play, exploring the AI actions and decision making process.

HIGH-LEVEL PLANNING IN RTS GAMES

Real Time Strategy games offer many complex reasoning challenges for AI players. They are often split into two categories: The control of individual units or armies on the field, known as micro-management, and the large-scale strategy deployed, known as macro-management.
For many games, micro-management can be approached using situational and reactive agents that don't try to predict future game states and don't require significant memory of past actions. For example, recent research applies potential-fields, such as the approach for unit navigation shwon in (Hagelback 2012), and Bayesian models (Synnaeve 2011) to achieve effective unit control and motion. A similar potential-field pathing and formation approach, called flow-field (Treuille 2006), is for example used in the game Supreme Commander 2 to deal with large unit formations (Rabin 2013). Many commercial titles, such as StarCraft II, cover micro-management with situational approaches, such as state machines or hardcoded scripting (Robertson 2014).
Macro-management is usually a higher-level planning problem, choosing the right strategy at the right time. For many games, an important element of a strategy is the build-order, deciding on which unit, building or research item to produce at a point of time (Churchill 2012). Predicting an enemy build-order should also be an important part of RTS AI (Park 2012). Algorithms used in RTS strategy planning include Monte Carlo methods (Ching 2005) or Goal-Oriented Planning (Orkin 2003).

SEMANTIC UNDERSTANDING OF GAME MECHANICS

A predetermined set of strategies and builds can cover many situations faced in an RTS game well. In competitive situations, where it can be expected that all players are exploiting game mechanics to achieve victory, an offline learning approach of macro-management builds may work well, as many assumptions about a game can be made. Yet, even in such competitive situations, there are strategies that are hard to predict at development time. Particularly strategies that aren't very effective due to their game mechanics, but are very convenient for human players to execute due to usability, for example the way the user interface is set up.
Orkin described an interesting approach towards AI development in games especially regarding cooperative scenarios: Going away from behavior trees and state machines, but rather focusing on tagged, semantic descriptions of sample data. The job of an AI would be closer to a search engine than to a construct of algorithms (Orkin 2012).
Our approach goes into a similar direction. In many games, the semantic knowledge about game mechanics is baked into the AI algorithms and therefore static during runtime. Consider an AI in a generic Real-Time Strategy game. There might be a top-level agent, perceiving the game world as a very simplified threat-grid-map on which it runs an algorithm on to find nice target grid cells for different goals. Doing so, it assigns a certain attack priority to the enemy base and a certain defense priority to the current combat perimeter. Another algorithm might now assign workers to the defense cells, and analyze them in greater detail. The algorithm awards higher defense priorities to narrow passages and that's where the actual defense turrets are being built by the final layer of AI, which controls the individual units.

ARMY PREDICTOR PROTOTYPE

We implemented the proposed approach as a prototype in StarCraft II using a simplified version of the game, where only strategic decisions are made by players and the control over units in battle (micro management) is automated. Players are grouped in two teams and each human player is further teamed up with an AI ally with which they share technology progress and may agree on different strategies and roles. We chose to include cooperation into the prototype as we suppose that semantic structures used as

described above help solving this challenge by offering a centralized point to react on communication with a human player. Further detail on the challenges and opportunities of cooperation were illustrated in (Stiegler 2013).
The semantic structure was implemented using the StarCraft II scripting system. It contains 65 nodes, over 85 attributes and over 400 relations. The code sample below shows an example setting up the Reaper Node.

```
1     AddNode("Reaper")
2     AddRelation("Requires", "Reaper",
      "Barracks", c_requiresOwn)
3     AddRelation("Counters", "Reaper",
      "Marine", 18.513)
4     AddAttribute("Costs", "Reaper",
      (CatalogFieldValueGetAsInt(c_gameC
      atalogUnit, Reaper,
      CostResource[0], c_playerAny)))
5     LinkConstructorToNode("Reaper",
      "BarracksTrain")
```

Line 1 creates the Reaper node and registers it in the semantic structure.
Line 2 represents a requires relation, pointing towards the construction building of the Reaper infantry. The final parameter of this relation represents that the Barracks have to be controlled by the current player to allow building reapers. As human and AI players share technology progress, it is possible to fulfill parts of the requirements with buildings built by the ally. There is also a constructs relation that points from the Barracks Node towards the Reaper Node, which could be translated automatically into Requires relations at compile time.
Line 3 shows a Counters relation, a very important relation for game mechanics, indicating that the Reaper is a good choice against a Marine. The strength of the Counters relation was determined by a series of tests and is simplified at the moment. The tests include several typical game scenarios measuring a units effectiveness against alone, in small and in large groups as well as different movement formations. Especially when micromanagement is added, the Counters Relation will have to be extended or split into several new relation types.
Line 4 adds the cost attribute to the Reaper node, with the costs taken directly from the game data.
Line 5, finally, is a special command registering the button-handler of the Reaper's construction button with the Reaper node. We tried to tie the semantic structure as loosely to the actual StarCraft II internals as possible. This link allows the AI to execute the exactly same button handler that would be executed if a human pressed the "Build Reaper" button in the GUI.
This semantic structure serves as the input to all other components of the AI. The most important component is a utility system deciding which actions to pursue next. There are 11 different actions with various sub-actions, such as the "Construct Unit" action with a sub-action "Construct Reaper". The utilities for each sub-action are calculated by functions operating on the semantic structure. As illustrated above, the utility of a "Construct Reaper" sub action is greatly influenced by counters relations and what is known of the enemy army composition.
Within these utility functions, further AI components are called, like hostile army predictors. These serve the purpose of predicting hostile unit counts based on units observed in the past by creating virtual unit counts added to the observed counts for utility prediction. To do so, the predictors infer all units that are definitely present from the set of observed units. An example is shown in Figure 1, where the AI observed a hostile Space Marine (shown in black) and directly infers that at least one Barracks must be present, as there is a "Constructs" relation between Barracks and Space Marines. Based on this knowledge, the Predictor can infer predicted units (shown striped with their predicted unit counts) by walking down construction relations. Each predictor is therefore defined by the speeds at which it predicts units along a certain relation.

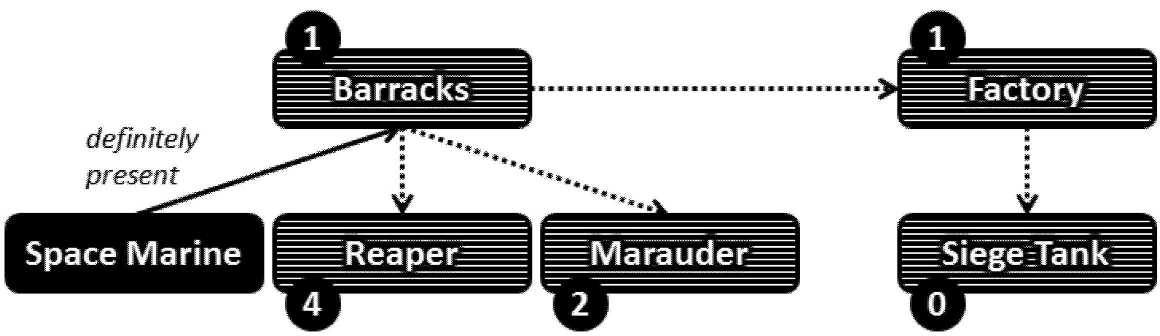


Figure 1: Simple Predictor example

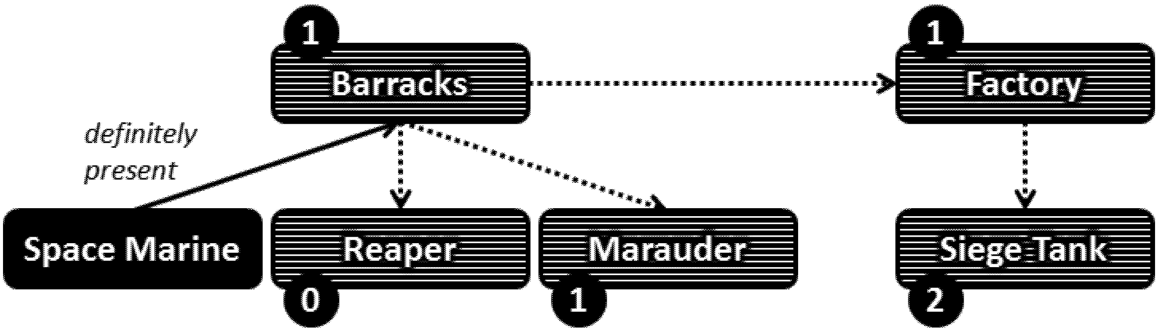


Figure 2: Siege Tank Rush Predictor example

Other predictors, like the one pictured in Figure 2, have different speeds for propagating predicted vessels along relations, for example a Siege Tank rush, leading to very different predicted units. The current implementation uses 9 handcrafted predictors. As the efficiency of a predictor does highly depend on choosing the right one for an opponents strategy, they offer a good place to utilize machine learning to infer propagation speeds from recorded games. Another option would be to represent the propagation speeds in the semantic structure themselves and use a more abstract approach similar to the predictors, to predict their actual prediction speeds.

PRELIMINARY EVALUATION

A cooperative game AI can be evaluated according to a range of criteria including believability, user-interface and interaction, performance as well as the effectiveness of the AI strategies and behaviors. In due course we intend to evaluate our work according to these different criteria, but to date only a more limited evaluation has been completed. A first evaluation of the cooperative behavior of the AI, with particular attention on the interplay between human player actions and the AI player actions is presented in (Stiegler 2013), but here we consider the results more generally.

A semantic network AI player was implemented for StarCraft II using the included scripting environment and tools. This does not make use of the inbuilt AI components, and as it runs entirely in the interpreted scripting environment can not be considered to be highly optimised. Despite this, acceptable performance was achieved across a number of games with moderate complexity and with all of the strategic algorithms using a semantic structure as their input as illustrated above. The AI focusses on the top levels of planning in StarCraft II, and does not attempt to micromanage individual units - a commonplace division between strategic (player) and tactical (unit) AI.

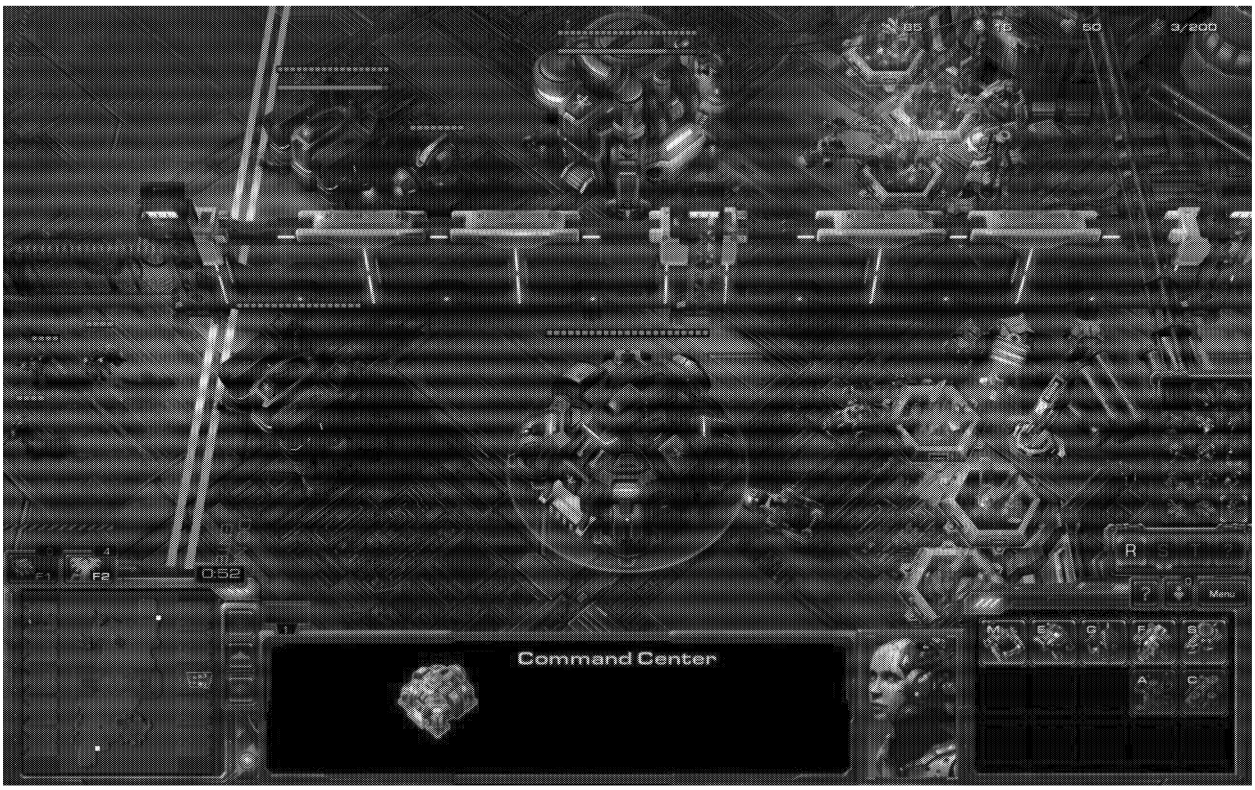


Figure 3: Prototype Screenshot

Figure 3 shows a screenshot of the StarCraft II prototype with a human base (bottom) and an AI base (top) executing a cooperative strategy. The two panels on the right are used to agree on long-term strategic and short-term tactical decisions.
The test series with the SC2 Prototype consisted of 17 games of variable complexity, depending on the strategies deployed by the two opponent teams. There were typically between 20 and 30 units active per team, but some games led to battles with only few high-tech units. In one game, the winning team built only 7 units. The number of units under AI control within a team greatly varied depending on the role the AI took. If the AI fulfilled a completely supportive role, it might not built a single unit leaving the entire army production to its human ally while focusing on unlocking research and performing upgrades to strengthen the forces. The following log shows an example of the actions performed by an AI during one of the test games. It shows the strategic decisions the AI is evaluating in 5 second intervals. The human player chose to focus on the production of larger late-game units and the AI unit opted for supporting its human ally. As the human is focusing on fulfilling technology requirements in the early game stages, the AI also had to produce military forces to compensate.

1s - 30s	Producing Building: Barracks Technology Lab
35s	Producing Worker
40s	Producing Unit: Marauder
45s - 75s	Saving for Research & Researching: Concussive Shells
80s - 100s	Saving for Unit & Producing: Marauder
105s - 120s	Saving for Unit & Producing: Reaper
125s - 135s	Saving for Building & Producing: Ghost Academy
140s - 150s	Saving for Unit: Ghost
155s	Producing Unit: Ghost
160s - 195s	Saving for Unit: Ghost
200s - 210s	Saving for Building: Engineering Bay
215s - 220s	Saving for Unit & Producing: Ghost
230s - 280s	Saving for Research & Researching: Personal Cloaking Device
290s - 325s	Saving for Unit: Ghost
330s - 335s	Saving for Building: Armory
340s - 350s	Saving for Unit: Marauder
355s	Producing Unit: Marauder
360s - 385s	Saving for Unit: Marauder
390s - 405s	Saving for Building: Armory
410s	Producing Unit: Marauder
…	…

Figure 4: Game Log Excerpt

Figure 4 shows some interesting reactions of the AI towards communication and actions performed by its ally and hostiles. At 105s, for example, the AI switched from Marauder production to the cheaper Reapers, as a large opponent force was observed. Yet, the human ally reacts by producing a few additional units, allowing the AI to switch back to its primary role at 125s: producing a research building, the Ghost Academy. Shortly after, it produces a Ghost, although it was intended to be a supportive role. This could happen because the communication output is no hard switch within the AI scripts, but just an alteration of the semantic structure. The AI is still capable of doing actions outside its designated strategy part. Observing a growing enemy force, the AI decided to save for an additional Ghost, shortly interrupted when switching to save for a research building at 200s. Growing enemy forces, however, make the AI return to its original path and produce an additional Ghost at 225s. Now the first late-game unit its human ally is ready occupying the enemy forces and giving the AI time to research the Personal Cloaking Device for its Ghosts, greatly increasing their effectiveness. Its human ally is now also producing some Ghosts. The enemy, reacting on this new threat switches over to heavily armored units making Marauders a good choice.
The battle goes on for additional 6 minutes eventually leading to victory of the AI and its ally.
The semantic structures in these tests were handcrafted with most parameters, like counter strength, drawn from actual game mechanics tests. Within this range of strategies, from a personal evaluation, the AI performed reasonable actions as expected of its chosen strategy.

CONCLUSIONS

In this paper we have outlined an approach for developing AI for RTS games based on semantic networks, and discussed how this can build on existing approaches such as utility systems. A preliminary evaluation of the performance (in terms of speed and behavior) was presented, to be extended considerably in future work. However, our preliminary evaluation has demonstrated it is possible to implement high-level RTS AI using a semantic network, and that this is able to achieve acceptable performance across both criteria. In the near future we will be releasing the prototype to the public on Blizzard Arcade to allow others to test and evaluate, or simply play with, the AI presented here.
Considering the use of predictors, and the propagation of knowledge through the semantic network, we have also thought how the current work could be extended in future from the current strategy level focus to also encapsulate more aspects of the meta-game. The function detailing how such knowledge is propagated throughout the semantic structure could itself be contained in the semantic structure, with an algorithm like a utility system or a state machine choosing between different functions. We would like to experiment with utilizing this approach in greater detail than the army estimators and using it for all propagation functions. This allows the AI to cover part of the meta-game of StarCraft II, where certain established strategies are vary in popularity over time in the manner of trends, and appear more or less often in tournaments. An observation, like the early acquisition of a Vespine resources, could lead the AI to believe that the enemy is pursuing one of a particular, popular group of strategies. This would cause the AI to select a corresponding propagation function, that creates virtual units faster that correspond to this strategy. Ultimately, this strategy could be learned, probably in an offline way analyzing recorded games from tournaments. There has already research been done regarding finding strategies in

recorded games using machine learning or other statistical approaches (Weber 2009; Hsieh 2008).
While work on such extensions has yet to begin in earnest, from our initial work we believe that the semantic network approach is an effective one, and will be continuing to explore its use, particularly in the context of cooperative AI in strategy games.

REFERENCES

Ching, M.; Buro M. and Schaeffer J. 2005. "Monte Carlo Planning in RTS Games" in *IEEE Conference on Computational Intelligence and Games (CIG)*.

Churchill, D. and Buro, M. 2011. "Build order optimization in starcraft," in *Proceedings of AIIDE*, 14-19.

Hagelback, J. 2012. "Potential-field based navigation in StarCraft," in *IEEE Conference on Computational Intelligence and Games (CIG)*, 388-393.

Hsieh, J.-L. 2008 "Building a player strategy model by analyzing replays of real-time strategy games," *in IEEE International Joint Conference on Neural Networks (IJCNN)*, 3106-3111.

Orkin, J. 2003. "Applying Goal-Oriented Action Planning to Games". In *AI Game Programming Wisdom 2*. Charles River Media, 217–228.

Orkin, J. 2012. "Data-Driven Digital Actors," in *IEEE Conference on Computational Intelligence and Games (CIG)*, Granada.

Park, H.; Lee, K.; Cho, H.-C. and Kim, K.-J. 2012. "Prediction of early stage opponents strategy for StarCraft AI using scouting and machine learning," in *WASA Proceedings of the Workshop* at SIGGRAPH Asia 7-12.

Rabin, S. (ed.) 2013: "Game AI Pro: Collected Wisdom of Game AI Professionals"

Robertson, G. and Watson, I. 2014. "A Review of Real-Time Strategy Game AI".

Stiegler, A. and Livingstone, D., "Cooperative AI in Real-Time Strategy Games", in Proceedings of the GAMEON'2013.

Synnaeve, G. and Bessiere, P. "A Bayesian model for RTS units control applied to StarCraft," in IEEE Conference on Computational Intelligence and Games (CIG) 190-196, 2011.

Treuille, A.; Cooper, S. and Popović, Z. 2006. "Continuum crowds." In *ACM Transactions on Graphics (TOG)*. Vol. 25. No. 3. ACM.

Weber, B. G. 2009. "A data mining approach to strategy prediction," in *IEEE Symposium on Computational Intelligence and Games (CIG)*, 140-147.

REGION LOAD MANAGEMENT AND ARCHITECTURAL TESTING FOR THE ALFIL CROWD SIMULATION VIRTUAL ENVIRONMENT

César García-García, Victor Larios-Rosillo and Hervé Luga
Department of Applied Information Sciences - CUTonalá
Universidad de Guadalajara
Mexico
cesar.garcia@cutonala.udg.mx

KEYWORDS

Multi-agent systems, Performance, Virtual Environments, Serious Games

ABSTRACT

This paper presents the performance evaluation of a crowd-populated platform to be used in a serious game to simulate emergency evacuations. A multi-agent system is used to control autonomous members of the crowd, while interacting with user-controlled avatars. Several experiments are performed to test resource use as well as overall performance. Finally a population cap for each game region will be decided from the evidence collected during the experiments.

INTRODUCTION

Much has been achieved in crowd simulation since the seminal works of Reynolds (1), where artificial birds – or boids– demonstrated the concept of complex flocking behaviours by aggregation of individual perceptions and actions.

To effectively model crowd behaviour, it is necessary to first model individual behavior that is coherent with the simulated virtual environment (2). Works by Perlin (3) and Ulicny (4) mention that in order to behave in a believable way these artificial actors must act according to their surrounding environment, being able to react to environmental changes and also to the actions of human-operated avatars in the virtual world. Virtual environments enhance the user experience by making it more immersive and compelling (5), and this perceived realism allows for learners to overlook the fact that they are working with artificial constructs and concentrate in the phenomena being simulated (6).

Previous work on benchmarking Multi-Agent Systems includes message-passing efficiency, responsiveness and access to databases (7). Of particular interest to us is the ability to concentrate database queries in a single agent which, in turn, provides this service to other agents.

Section describes the problem of creating realistic scenarios where agent crowds can interact with human-controlled avatars. Section presents the proposed approach for creating such scenarios. Section includes the experiment results as well as a general benchmarking of the architecture as a whole. Finally, section includes our concluding remarks and future work.

PROBLEM DESCRIPTION

We choose to create a 1-to-1 relationship between artificial humanoids in the serious game and JADE (8) agents. This choice was mostly conceptual at first, as thinking of an artificial humanoid in the game as an "agent" was very practical. When the need arose to implement several copies of the model running in parallel with the same virtual environment data, this decision proved to be correct, as migrating the code from standalone to agents was a very straightforward process given that it was initially conceived under the agent concept from the beginning.

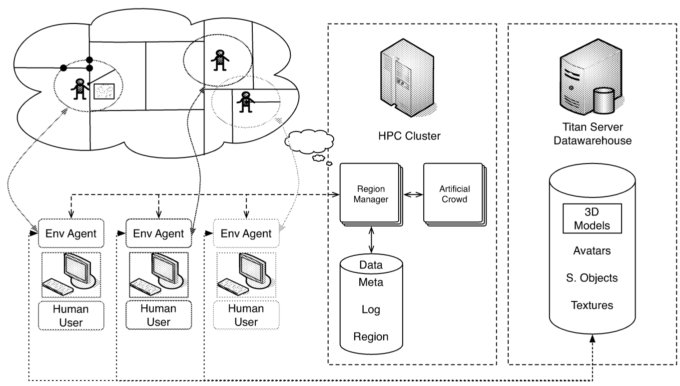


Figure 1: Multi-Agent System Architecture.

As shown in Figure 1 only three types of agents are implemented: Region Manager agents handle specific scenarios or *regions* containing sections of the virtual environment, they provide an interface between users, artificial crowds populating the regions and the system database; Environment agents reside in each client's agent container and handle the JME virtual environment; finally, Person agents represent individual artificial humanoids present in the region, usually as part of a crowd.

Virtual environment

The jMonkeyEngine platform (9) provides a very comprehensive virtual environment that is fully compatible with Java, which is very convenient given our use of JADE for the multi-agent system.
It includes an OpenGL-based graphics engine, a Bullet Physics Library engine and it is very friendly with different 3D modelling software, supporting several formats including OBJ, Collada, 3DS, OgreXML, etc. It also includes the NiftyGUI tool for creating graphical user interfaces on top of the 3D environment.
If we imagine the virtual environment as a gaming board with players, autonomous characters, smart objects and threats as game pieces; then the *environment agent* is responsible for setting up the board, assigning turns and moving the pieces around. It is also responsible for user input. It effectively translates events in the region database into meaningful events in the JME virtual environment.
The region manager is responsible of several important tasks, such as: acting as a gate-keeper for the region, only accepting new agents when the particular node has enough resources to handle it; guarantee that no agent has perfect knowledge of the environment; separate agent communication from the game environment; guarantee that only one agent per region is querying the database; and guarantee data consistency for all players across the region.
Perhaps the biggest disadvantage it creates is that the region manager agent can effectively become the system bottleneck. For this reason it is very important to assert that a particular region has enough resources to handle a new agent *before* the agent is admitted into the region.

Data as the backbone of the system

The system database acts as the backbone for the entire system preserving consistency across regions. Figure 2 shows the database schema for the system.

Regions
The region table contains information regarding the base scenario such as its size, bounding coordinates, etc. It also contains information about the region manager, its IP address, port, etc.

Assets
Persistent objects do not change during gameplay, this is used to our advantage to reduce model and texture storage by making objects represented in the scenario merely instances of assets, static objects associated to 3D models, textures, etc. In case an asset references a non-existing model or texture, the system is prepared to

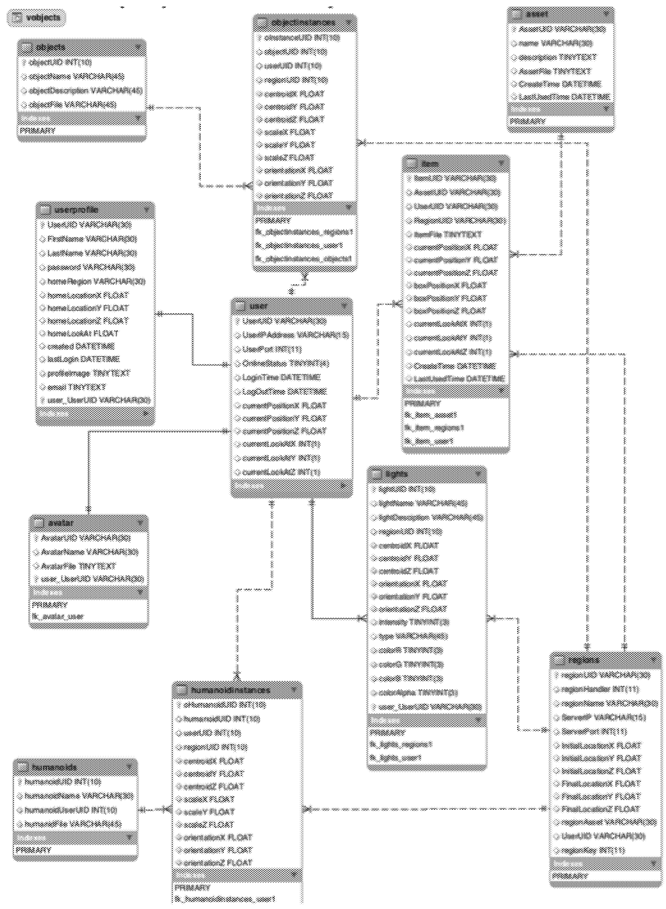


Figure 2: Database architecture.

load a placeholder box with a pre-loaded JME texture, so that the impact of the missing file is reduced.

Lights and other dynamic props
Lights and other dynamic props, such as fire, explosions and other particle-based effects have to be dynamically created in JME and thus can't be instantiated from assets. For this purpose, particle-system are stored in a different table, with relevant information such as location, orientation, intensity, type, color and transparency.

Items
Smart objects are implemented as bounding boxes around props: virtual sensors that detect proximity and collisions with other objects. The objects then execute their behaviour as appropriate.

METHOD

The first step to building the scenario is to create an empty space according to the region size, this can be achieved by running the following SQL query:

```
SELECT * FROM regions WHERE regionUID = 1
```

The resulting fields provide region-specific information such as region name, agent container address, agent container port and three pairs of X, Y, Z coordinates, indicating the size of the region. With this information, we can create a terrain to hold objects in the region. Populating the region is achieved in three steps: persistent objects, lights and smart objects.
Pesistent objects such as buildings, walls, trees and other props are stored in an object instances table and are retrieved as follows:

```
SELECT *
FROM (
objectinstances JOIN asset
ON asset.AssetUID=objectinstances.objectUID
)
WHERE objectinstances.regionUID = 1
GROUP BY objectinstances.oInstanceUID
```

Persistent objects are represented in the scenario by instantiating assets, each asset is independent of region, location, scale, rotation, and other environment concerns, such as lights; while instances are required to have such restrictions.
Lights and other particle-systems are retrieved using the following query:

```
SELECT * FROM lights WHERE regionUID = 1
```

The resulting fields contain information about the particle-system's location, rotation, intensity, and type. The type is then used to build the appropriate JME representation: a PointLight from a lamp, a DirectionalLight from the Sun, a fire, explosion or water fountain particle-system.
Fire, explosions, rain, falling dust, and similar effects are all implemented in the same way in JME: particle systems. Particle systems are collections of small objects moving together and following simple rules such as gravity, time-related position, color and size. By using this JME feature, representing a burning fire, falling water, and falling debris is essentially the same code with different parameter for particle size, gravity, and orientation. This feature is available as part of the Bullet Physics Library.
Finally, smart objects are implemented as bounding boxes around props: virtual sensors detect proximity and collisions with other objects, including artificial humanoid and user-controlled avatars. The following query is used to retrieve the bounding boxes.

```
SELECT * FROM item WHERE regionUID = 1
```

The resulting fields also contain information used to link another prop with the bounding box, effectively transforming any prop into a smart object.

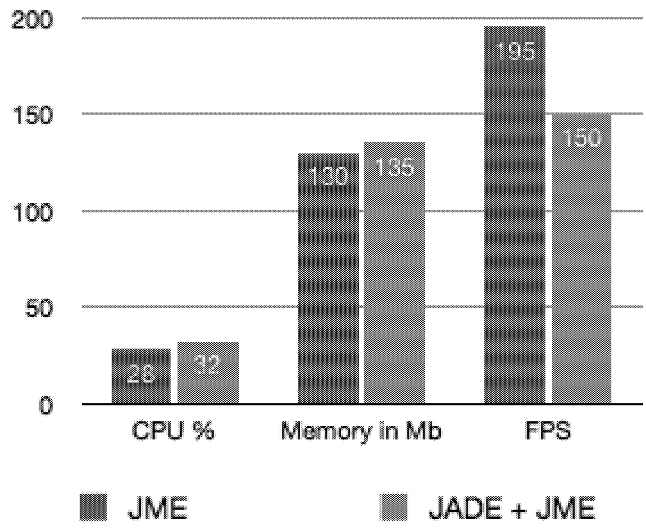


Figure 3: Comparative performance when running JME only vs. JADE-managed JME.

Equipment

All tests were performed on a MacBook Pro 2.26 GHz Intel Core 2 Duo Processor with 4 Gb of RAM, a NVIDIA GeForce 9400M graphical card with 256 Mb in VRAM and running Mac OS X 10.6.4. Using Java 1.6.0 with maximum heap size set to 1024 Mb.

RESULTS

First we want to make sure that combining the JME and JADE technologies will not cause by itself a bottleneck in performance. While JME is a good choice for our graphical needs and that JADE itself does not create a significant overhead (7), we need to know that adding an environment agent managing the virtual environment will does not severely impact graphical performance. For this benchmark we loaded a predefined scenario in JME and measured CPU usage, memory utilization and frames per second and loaded the same scenario from an environment agent, which manages the JME environment. Figure 3 illustrates the results of this initial benchmark.
Second, as a control point , we took a previously created region that is devoid of agents and added "headless" models until the environment collapsed and registered CPU usage, memory utilization and animation frames per second, all of them standard measures of game performance. A headless model is an animated character under direct control of the environment agent which only plays a walking animation, but has no stored data and is not perceiving the environment or running any kind of behavior. Figure 4 shows the results of this control point.

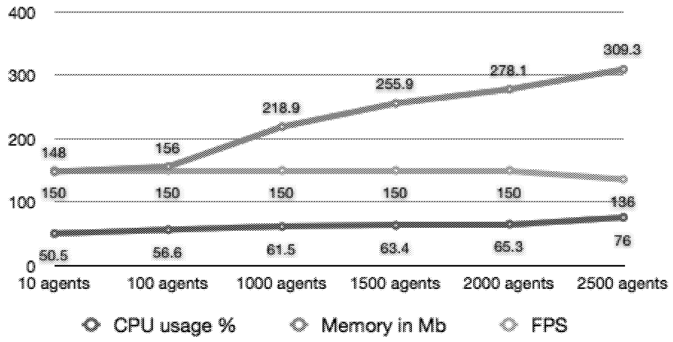


Figure 4: Resource utilization with headless agents.

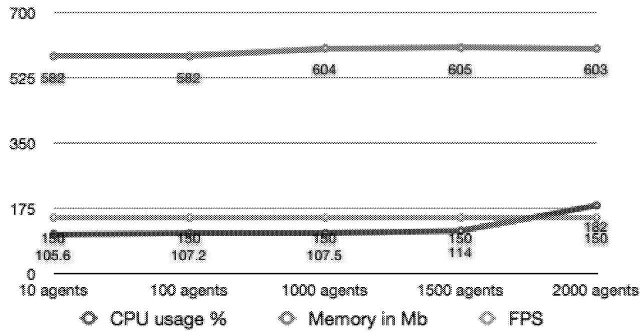


Figure 5: Resource utilization with complete agents.

Then we proceed to repeat the above benchmark but with "complete" agents which in addition to the animations are querying the environment agent for perceived data, updating their internal states, generating goals and running a single idle behavior. Figure 5 shows the result of this test.
Also, of special interest to the experiment is the time it takes for the region manager agent to process the environment-update queries from the artificial agents when under a stressful load. For this purpose, complete agents were used to constantly query the region manager for environment updates. Figure 6 shows the results of this test.

CONCLUSIONS

While our results clearly surpass those of Chmiel et. al. (7), this is easily explained by the difference in hardware resources available.
Our tests suggest that adding an environment agent does not severely impact graphical performance. Also, adding a region manager agent to concentrate database access does not inherently worsen performance. The complex behavior associated with the *person agents* clearly hinders the overall performance of the system,

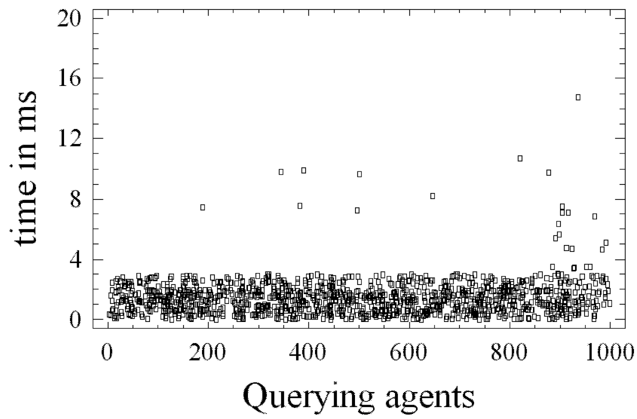


Figure 6: Time in milliseconds to perform an environment query versus number of agents in the region.

as shown in Figures 4 and 5.
From Figure 6 we can observe that the time to solve a specific query and send it's result does not seem to be affected by the number of querying agents, or rather, the inflection point is still very far away from our current ability to create agents in the same region.
With the previous observations we conclude that the maximum number of agents that a region can contain and still be operative is 2000, but this number was not constant during our tests, sometimes running into java.lang.OutOfMemoryErrors even when manipulating the heap space, and even calling the Java Virtual Machine Garbage Collector before any additional agents are created did nothing to improve the performance, even hindering it in the database queries experiment.
Finally, and taking into account the known issue with the DirectoryFacilitator in JADE becoming a bottleneck (10), we have decided to cap the population of a region to 1000 agents, including both autonomous and human-controlled agents.

ACKNOWLEDGEMENTS

This project is funded by CONACYT and COECYT-JAL grants. The DVRMedia2 framework is a co-operative effort between Universidad de Guadalajara and Institut de Recherche en Informatique de Toulouse (IRIT) trough Le Programme de Coopération Post-Gradués (PCP) Franco-Mexicain. Special thanks to Intel Guadalajara for providing the required high-performance computing support and to Unidad Estatal de Protección Civil y Bomberos Jalisco for providing required training and access to facilities.

REFERENCES

[1] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *SIGGRAPH '87:*

Proceedings of the 14th annual conference on Computer graphics and interactive techniques, pages 25–34, New York, NY, USA, 1987. ACM.

[2] Daniel Thalmann, Soraia Raupp Musse, and Marcelo Kallmann. From individual human agents to crowds. *Informatique*, 1:6–11, 2000.

[3] Ken Perlin and Athomas Goldberg. Improv: a system for scripting interactive actors in virtual worlds. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 205–216, New York, NY, USA, 1996. ACM.

[4] B. Ulicny and D. Thalmann. Crowd simulation for interactive virtual environments and VR training systems. *Europgraphics Workshop*, 2001.

[5] Gun A. Lee, Gerard Jounghyun Kim, and Chan-Mo Park. Modeling virtual object behavior within virtual environment. In *VRST '02: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 41–48, New York, NY, USA, 2002. ACM.

[6] Adam Cheng, Jonathan Duff, Estee Grant, Niranjan Kissoon, and Vincent J. Grant. Simulation in paediatrics: An educational revolution. *Paediatric Child Health*, 12(6):465–468, August 2007.

[7] Krzysztof Chmiel, Dominik Tomiak, Maciej Gawinecki, Paweł Kaczmarek, Michał Szymczak, and Marcin Paprzycki. Testing the Efficiency of JADE Agent Platform. In *ISPDC '04 Proceedings of the Third International Symposium on Parallel and Distributed Computing/Third International Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Networks*, pages 49–56, Washington, DC, US, 2004. IEEE.

[8] F. Bellifemine, G. Caire, A. Poggi, and G. Rimassa. Java Agent DEvelopment Framework: A White Paper, 2003.

[9] Mark Powell. JMonkeyEngine, 2003.

[10] Fabio Bellifemine, Giovanni Caire, and Dominic Greenwood. *Developing Multi-Agent Systems with JADE*. John Wiley Sons, Inc., West Sussex, UK, 2007.

Implementing Racing AI using Q-Learning and Steering Behaviours

Blair Peter Trusler and Christopher Child
School of Informatics
City University London
Northampton Square, London, UK
Email: btrusler@gmail.com / C.Child@city.ac.uk

KEYWORDS

Q-Learning, Reinforcement Learning, Steering Behaviours, Artificial Intelligence, Computer Games, Racing Game, Unity.

ABSTRACT

Artificial intelligence has become a fundamental component of modern computer games as developers are producing ever more realistic experiences. This is particularly true of the racing game genre in which AI plays a fundamental role. Reinforcement learning (RL) techniques, notably Q-Learning (QL), have been growing as feasible methods for implementing AI in racing games in recent years. The focus of this research is on implementing QL to create a policy which the AI agents to utilise in a racing game using the Unity 3D game engine. QL is used (offline) to teach the agent appropriate throttle values around each part of the circuit whilst the steering is handled using a predefined racing line. Two variations of the QL algorithm were implemented to examine their effectiveness. The agents also make use of Steering Behaviours (including obstacle avoidance) to ensure that they can adapt their movements in real-time against other agents and players. Initial experiments showed that both types performed well and produced competitive lap times when compared to a player.

INTRODUCTION

Reinforcement learning (RL) techniques such as Q-Learning (QL, Watkins 1989) have grown in popularity in games in recent years. The drive for more realistic artificial intelligence (AI) has increased commensurably alongside the high fidelity of experience which is now possible with modern hardware. RL can produce an effective AI controller whilst removing the need for a programmer to hard-code the behaviour of the agent.

The racing game used for performing the QL experiments was built using the Unity game engine. The game was built as a side-project in conjunction with this research. The cars in the game were created so that the throttle and steering values could be easily manipulated to control the car.

The biggest challenge when considering implementing RL is to determine how to represent and simplify the agent's state representation of the game world in an effective way to use as input for the algorithm. The information needs to be abstracted to a high level in order to ensure that only necessary details are provided. Two versions of the QL algorithm were implemented; an iterative approach and a traditional RL approach.

The results from the experiments demonstrate that when combined with steering behaviours both QL implementations produced an effective AI controller that could complete competitive lap times.

BACKGROUND

Reinforcement Learning and Steering Behaviours

RL is the method for teaching an AI agent to take actions in a given scenario. The goal is to maximise the cumulative reward, known as the *utility* (Sutton and Barto, 1988). The result of the RL process is a policy which provides the agent a roadmap of how to perform optimally. The RL process can be performed *online* or *offline*.

Online learning is the process of teaching the AI agent in real-time. Offline learning involves teaching the agent before releasing the game. Both methods have their merits and issues. For several reasons the offline version is most commonly used when RL is applied to games (and is used in this research). Primarily, it ensures that the agent will behave as expected when the game is finished. It also means there is less computational expense in real-time as the AI is behaving based on a saved policy and does not need to perform as many calculations in real-time. The offline RL process works by performing a large number of iterations (*episodes*) of a simulation in order to build up a data store of learned Q values relative to their state-action combination.

The concept of steering behaviours (SBs) was first introduced by Craig Reynolds (1999). SBs provide a mechanism of control for autonomous game agents. Reynolds proposed myriad behaviours which could be used independently of one another or holistically to achieve different behaviours.

There were three relevant SBs for this project; seek, obstacle avoidance and wall avoidance. Whilst SBs are not the focus of this paper, they were used to perform real-time avoidance techniques during the game when multiple agents were in the scene.

Q-Learning

Q-Learning is one of the most commonly used forms of RL and is a type of temporal difference learning (Sutton and Barto, 1988). QL is used to find the best action-selection policy for a finite number of states. It assigns utility values to state-action pairs based on previous actions which have led to a goal state. As the number of episodes increases, the utility estimates and predictions improve and become more reliable.

A *state* can comprise of any piece of information from the agent's environment. An *action* is the operation that the agent can perform at each state. The action selection policy is a key component to the learning process. The two common types of action selection are *greedy* and *ε-greedy* (Sutton and Barto, 1988). Greedy always chooses the optimal available action according to the current utility estimates. In contrast, ε-greedy has a small probability of selecting a random action to explore instead of choosing the greedy option.

The QL formula (1) is performed upon reaching a state. The QL formula is defined as follows:

$$Q(s, a) = (1 - \alpha)Q(s, a) + \alpha(r + \gamma \max_{a}'(Q(s', a'))) \quad (1)$$

Where:

- $Q(s, a)$ – Q value of the current state-action pair
- $Q(s', a')$ – Q value of the next state-action pair
- r – reward value associated with next state
- α – learning rate parameter
- γ – discount value parameter

The learning rate and discount value parameters are crucial in defining the learning process. The learning rate determines to what extent newly acquired information will override the previously stored information. A learning rate value of 0 will mean that the agent will not learn anything whilst a rate of 1 means that the agent will only consider the most recently acquired data. The discount parameter defines the importance of future rewards to the agent. A factor of 0 creates a short-sighted agent which only considers current rewards, whilst a factor of 1 ensures the agent will aim for the highest possible long-term reward.

Q-Learning in Games

Patel et al (2011) used QL to create an AI agent for the popular first-person shooter game *Counter-Strike*. They used QL to train a simple AI agent in order to teach it how to fight and plant a bomb. A higher reward value was assigned to the AI if it accomplished the goal of the game. For example planting the bomb produced a higher reward than killing an enemy. Their results showed that the QL bots performed competitively against the traditionally programmed bots. However, they did note that this was not tested against players. This could identify further issues that would need to be resolved in the learning process

A popular commercial racing game that makes heavy use of RL is the Forza series (*Drivatars*). The development team created a database of pre-generated racing lines for every corner on a race track (several slightly different lines per corner). For example, some racing lines will be optimal whilst others may go wide and miss the apex of the corner. The agent uses QL (offline) to learn the appropriate throttle values to follow each racing line as fast as possible. The cars also learn various overtaking manoeuvres at each part of the track. During a race, the racing lines at each corner are switched to vary the behaviour. This approach meant that the programmers were not required to hard-code the values for each track and corner and produced a reusable and effective tool for creating AI agents for each type of vehicle. This technique has resulted in the Forza series having one of the most realistic AI systems in the racing game market today.

IMPLEMENTING Q-LEARNING

Game World Representation

The first challenge was converting the three dimensional game world into a series of states for the algorithm to interpret. Firstly, a racing line was generated by positioning waypoints along the race track and creating a Catmull-Rom spline by interpolating between these points.

The states were then defined as track segments (points along the racing line). The region was implemented by placing a box collider at each of these points. The collider width was equal to that of the race track width and rotated based on the direction of the spline. The quality of the state is evaluated based on the agent's proximity to the centre of the racing line and time taken to reach the state.

Discrete Action Space

It was decided to focus the QL on learning the cars throttle values whilst using the racing line to generate the appropriate steering values. This helped to reduce the action space to an appropriate size in order to minimise the number of iterations required to perform the learning process. The action space was set to nine evenly spaced throttle values ranging from +1.0 to -1.0 (where +1.0 represents full throttle and -1.0 represents full braking or reversing).

Q-Store Data Structure

A data structure (the *Q-Store*) was implemented to store all of the data required by the learning algorithm. The Q-Store maintained a two-dimensional array of doubles. The first dimension in the array represented the state values whilst the second dimension represented the action values. This allowed for the Q value for each state-action pair to be easily stored and accessed.

Q-Learning Algorithm

As previously mentioned two versions of the QL algorithm were implemented. Both versions are very similar in nature but with some key differences as highlighted in the following sections. The algorithm works by applying each action (throttle values) at each state on the track. A reward was calculated if the car reached or did not reach the next state and the QL formula was calculated and stored. Both versions used the greedy action selection policy.

The action policy generated from each version of the algorithm was stored in a text file. This allowed the policy to be retrieved and utilised without having to re-perform the learning process each time.

First (Iterative) Version

The first version of the algorithm was based on an iterative approach. The learning agent was designed to evaluate each possible action for a state before moving on to the next state. The agent would continually reset to the starting state after each evaluation. This meant that the agent would gradually make its way along the racing line and during the process the agent would ultimately evaluate the actions between the penultimate state and the goal state. This iterative approach meant that the number of episodes could be predetermined (number of states * number of actions).

Second (Traditional) Version

The second version was based on a more traditional RL approach. Unlike the first version the learning process did not continually reset in an iterative manner. It gradually developed a policy over a number of episodes (ranging from 10 to 5000 in testing). Theoretically, an increased number of episodes will make the policy more likely to allow the agent to reach the goal in an effective way.

Reward Function

The reward function used for the agents produced a reward value based on the quality of the action performed at the current state. The value returned by the function was based on whether the action performed was good or bad. A good move would return a positive scaling reward value based on two key factors (proximity to the racing line and time taken between the two states). A final large multiplier would be added to the reward value if the car reached the goal state (the final point on the racing line). A bad move (eg crashing) would result in the function returning a negative reward value.

Execute Policy

The policy was stored in a text file that consisted of a single value (representing the action number) per line (the state). The agent would identify its current state and apply the corresponding action as specified in the file until reaching the next state.

TESTING AND RESULTS

This initial aim of this research was to investigate whether QL could be used to create a high quality controller for a racing game. Subsequent to this goal, the two versions of the QL algorithm suggested a further area of research in order to determine how they differed and which performed to a higher level. Each version of the agent was taught using the same racing line, race track and car properties. The two agents were taught using the same number of episodes (1,000) for the first two experiments. The third experiment involved varying the number of episodes for the second version of the algorithm.

State-Action Tables (Q Tables)

The first area of comparison was between the Q Tables produced by each version of the algorithm. These tables were produced after the learning process was completed by retrieving the data from the QStore. Tables 1 and 2 show that there was a difference in action selection at state 93 whilst the same action was picked at state 94.

Table 1: State-Action Table (Version 1)

State	Action	Q Value
93	6	2805597255.12183
94	0	2920734984.09786

Table 2: State-Action Table (Version 2)

State	Action	Q Value
93	0	730021813
94	0	531860033

Lap Times

The overall goal of this research was to produce a high quality AI controller for a racing game using the two variations of the QL algorithm. As a result the most tangible measurement of performance provided by the project was in terms of lap-times.

The same race track and racing line was used for each version and they both started from the same position at the beginning of each lap. Ten laps times were recorded for each version The average lap times are shown in Table 3. The lap times were performed with the obstacle avoidance and wall avoidance behaviours disabled as there were no obstacles present in the scene to check for in real-time.

Table 3: Average Lap Time Comparison

Lap Number	Version 1	Version 2
Average	42.73594	42.65832
Standard Deviation	0.52378007	1.597068

The lap times produced are very similar to the ones in the first version which, however, appeared to produce more consistent times.

Episode Variation

Unlike the first version of the implementation, the second version could be taught using an indefinite number of episodes. This raised the question of what effect would varying numbers of episodes have on the lap-time produced by the agent. Up to this point, the results produced for the second version was taught using the same number of episodes as the first version of the algorithm (approximately 1,000).

Table 4: Episode Variation Table

Episodes	Lap Time / Result
10	44.33456 (crashed into wall)
100	44.96534 (crashed into wall)
1000	42.65832
1500	41.74825
2500	***40.95938***
5000	41.46755

The policies which caused the car to crash still managed to complete their laps as the car was built with a reset function to reset the car after 2.5 seconds to a point slightly further long the racing line. Table 4 shows that the fastest lap time was produced by the 2500 iteration version whilst similar lap times were produced by the 1000, 1500 and 5000 versions.

EVALUATION

State-Action Tables (Q Tables)

The state-action tables showed that the learning agents took a different approach entering the corner. The states chosen (93 and 94) were located before the tightest corner on the track. It is interesting to note the different actions selected for state 93. The first version selected a braking action whilst the second version selected the full throttle action. This was because the first version was focused on one individual state at a time. This meant it often braked at the latest possible state as it didn't keep track of the reward based on the final end goal state. The second version had a more long-term view and as a result performed the braking action earlier (during states 89, 90 and 92) in order to achieve a better speed through the corner. This is because the QL function is aimed at achieving the highest possible long-term reward which is provided upon reaching the goal state. It would have been interesting to see the effect of different action-selection policies on the Q values produced.

Lap Times

The lap time comparison produced an interesting set of results. Table 3 shows the average and standard deviation between lap times for each version. The average lap time between the two algorithms was extremely close. The standard deviation, however, was very different. The first version appeared to produce very consistent lap times and results, whilst the second produced a wider range of very fast and relatively slow lap times. The slow lap times were often a result of going off track or hitting a wall. This would indicate that the number of episodes used to teach the second version was too low.

Episode Variation

This experiment was inspired by the standard deviation result in the lap-time test. The question raised was at what point was it that the number of episodes used cease to have an effect. Lap-times produced by the car were recorded for 10 laps. Table 5 highlights the average lap times produced and the standard deviation between them.

Table 5: Average and Standard Deviation for Episode Variation of Lap Times (Version 2 only)

Episodes	Lap Time / Result
Average	42.6889
Standard Deviation	1.62844

The results show that for 100 episodes or less, the car crashed or had an incident causing the lap-time to be increased. This was to be expected given the number of possible actions for the number of states in the game world. Interestingly, it also shows that the fastest lap time was produced from a policy created by 2500 episodes. In contrast the policies produced by 1500 and 5000 episodes produced relatively similar lap times.

One would have imagined that the lap time for 5000 episodes would have been at least as quick if not faster than the controller produced from 2500 episodes. This result is possibly due to the algorithm performing further learning and discovering that a policy for this type of lap-time would result in a crash in the tighter parts of the racetrack. Therefore it made safer choices whilst still maintaining a good overall speed.

Results Discussion

The lap-times produced by both versions are relatively competitive compared to player lap-times (with times ranging between 39 and 42 seconds on average depending on the type of player). The overall performance of the algorithm in terms of lap-time is restricted by the optimality of the racing line. The line was generated from waypoints that were implemented by hand and based on what appeared to be the best line around each corner. Better lap times would possibly have been achieved if this line was produced algorithmically to create a minimum-curvature line around the race track. It was also surprising to note that both versions produced relatively similar lap times despite the differing approach to the QL process.

CONCLUSIONS AND FUTURE WORK

This paper has presented the use of QL to produce an AI controller in a racing game. The results have shown that the controller produces reasonable lap-times and performance compared to a player. The QL formula used in this project was the standard QL approach. Other versions could have been used (eg SARSA) which may have produced differing or even improved policies for the AI controller.

There are several other areas that are open to investigation in the future. The most pertinent of these would be to utilise alternative reward functions. This could be used to create different types of AI controllers (ie varying difficulties or driving styles). A further development could have been to use multiple racing lines with differing lines into and out of corners. These lines could have been learnt and switched in real-time to produce more realistic and seemingly human behaviour. Another modification would be to increase the state-space of the game world. This would increase the size of the QStore but in turn increase the number of possible actions that can be taken around the race track. This could result in enhanced behaviour, in particular through tight or twisting corners. The state space could be expanded further by taking other factors into account such as the car velocity.

This project has shown that QL produces a reasonable controller without hard-coding a complex AI system. The racing line is the principle requirement to be implemented into the game world. In the future QL could be used to teach the agent how to steer based on its current position on the track and what lies ahead. This would then allow AI developers to focus their efforts on improving the agent's steering behaviours to create more realistic real-time interactions.

REFERENCES

Lucas, S, Togelius, J. 2007. Point-to-Point Car Racing: an Initial Study of Evolution Versus Temporal Difference Learning. *Symposium on Computational Intelligence and Games*. 1 (1), p260-267.

Moreton, H. 1983. Minimum Curvature Variation Curves, *Networks, and Surfaces for Fair Free-Form Shape Design.* United States: Berkeley. p1-213.

Patel, P, Carver, N, Rahimi, S. 2011. Tuning Computer Gaming Agents using Q-Learning. *Proceedings of the Federated Conference on Computer Science and Information Systems*. 1 (1), p581-588.

Reynolds, C. 1999. Steering Behaviors For Autonomous Characters. *Game Developers Conference*. 1 (1), p763-782.

Sutton, R and Barto, A. 1998. *Reinforcement Learning:An Introduction*. United States: MIT Press. p324-332.

Watkins, C. 1989. Learning from Delayed Rewards. London: King's College.

WEB REFERENCES

FIAS. 2010. Reinforcement Learning. Available: http://www.cs.utexas.edu/~dana/RL08.pdf. Last accessed 20th September 2013.

Microsoft. 2004. Drivatar. Available: http://research.microsoft.com/en-us/projects/drivatar/. Last accessed 16th September 2013.

Candela, J, Herbrich, R, Graepel, T. 2011. Machine Learning in Games. Available: http://research.microsoft.com/en-us/events/2011summerschool/jqcandela2011.pdf. Last accessed 16th September 2013.

Thirwell, E. 2013. Forza 5's AI is "much more engaging than anything you'll see in another racing game". Available: http://www.oxm.co.uk/62293/forza-5s-ai-is-much-more-engaging-than-anything-youll-see-in-another-racing-game/. Last accessed 20th September 2013

GAME LAYOUT DESIGN

DEVELOPING PLAYER MOVEMENT DESIGN PATTERNS IN MULTIPLAYER VIDEO GAMES

Richard Lannigan
School of Computer Science
University of Lincoln
email: r.w.lannigan@hotmail.co.uk

KEYWORDS

Game Design Patterns, Level Development, Player Movement Behaviour, Visual Identification

ABSTRACT

Game design patterns are valuable tools in understanding the effect of design decisions on player behaviour. This paper describes the identification and validation of game design patterns based around player movement in first/third person shooter multiplayer games, using a visual identification methodology. These patterns can aid level designers in creating multiplayer levels by eliciting desired actions/movement of the players. The pattern set provides a guide to key behaviours of gamers in this context, how to implement a design feature to evoke a specified behaviour, along with its predicted result. The validity of the approach is demonstrated through a level design case study. The proposed framework can be used alongside other design processes to allow for a more mature and structured approach to level design.

INTRODUCTION

An important focus of level design for multiplayer first/third person shooters is ensuring that the movement of the players around the map is fluid and enjoyable, and is one of the first considerations that is taken into account when designing a new level. Feil and Scattergood (2005) say that "As a level designer, it's imperative that you consider player movement in all your designs." (p.64).

Since a multiplayer shooter level is simply a 2D/3D environment, it holds to reason that the best way to experience it is to explore it. So designing with movement in mind will reduce the chance that it will have to be re-designed later because of major "flow" problems. In Jim Brown's presentation entitled "Legacy of Fail" at the Game Developers Conference in 2012, he stated "You never really know for sure how something will turn out until you actually play it. It's that in-game testing that allows for true iteration".

Design patterns in general are very useful in aiding the development of a product. In most cases it doesn't matter what that product is, because it will be one of three types: physical, electronic or a service, it will be directed at a user base and will be required to yield certain results. Björk and Holopainen (2005) say "The focus of architects is on the intended use of the place and the experiences people should have when crossing a bridge or being in a skyscraper" (p.33). This means that designers are always thinking about the end results of a product and the experiences that people will have with it, aiming for it to be as pleasant and memorable as possible. This will also hold true for designers of video games. Design patterns can help to achieve this, by informing an architect or designer as to the predicted results of a design implementation.

RELATED WORK

The concept of design patterns was introduced by Alexander et al, in the book "A Pattern Language: Towns, Buildings, Construction" (Alexander et al, 1977); and described patterns as:

"Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice." (ibid) (p.x).

A key piece related to this work is the paper by Milam and Seif El Nasr (2010a) entitled "Design Patterns to Guide Player Movement in 3D Games". Their paper proposes five patterns that "…serve as a contribution to understanding and evaluating level design" (*ibid*). They are intended to aid designers to 'push' and 'pull' players through a single player level. The proposed patterns are listed as: Collection, Path Target (PT), Pursue AI (PAI), Path Movement and

Resistance (PM-R) and player is Vulnerable (PV). The process taken to create and verify these patterns, took the form of visual identification, where the researchers would watch the first 10 minutes of gameplay and note when each pattern occurred. For example in one of the games they tested, Bioshock (2K Games, 2007), they noted when the player collected the wrench, radio, genetic tonic and pistol; which refers to the "Collection" pattern proposed. Having the ability to breakdown a sequence of events in this manner allows designers to better understand them. In terms of the collection pattern example, the designers may decide that they are giving the player too many items on a frequent basis; this would be seen through the visual identification process. The designers could then choose to either reduce the number of collectable items or increase the play time between them.

However, Milam and El Nasr's work only extends to single player shooter games, but can benefit multiplayer games also. In the case of many multiplayer levels, they are inspired by or even taken directly from the single player campaign; and re-purposed for competitive multiplayer. Some examples of this are "Strike" and "Bog" from Call of Duty 4: Modern Warfare (Infinity Ward, 2007), and "Sword Base" and "Spire" from Halo: Reach (Bungie, 2010). Both singleplayer and multiplayer levels are designed to encourage specific movement from the player, the aims may vary between linear and sandbox formats, but the overall desire is the same. This link suggests that Milam and El Nasr's work on single player games can be modified and added upon, to apply to multiplayer games.

METHOD

The first step in developing a set of design patterns for multiplayer shooter games, was to begin by identifying the extent at which the patterns suggested in single player games were seen in multiplayer games.

Visual Identification

Visual Identification (or Visual Analysis) in this context is where the behaviour of players is recorded. The procedure comprises of an expert researcher watching a gameplay video, and making notes of emergent behaviour patterns. The note would include the name (or abbreviated name) of the action, along with the time in which it occurred. An example of this is shown in table 1.

Table 1: Examples of the first five notes in two gameplay videos.

Halo 4	Team Fortress 2
0:21: PM-R	00:12: PPv
0:27: PTm, PM-R	00:15: PPv, PV
0:32: Co, PV	00:22: O, PTm
0:35: PT, PM-R	00:26: PV
0:44: PPv, PV	00:30: PPv, PV

This technique is based on those proposed by Milam and El Nasr (2010a and 2010b). In these papers the researchers imported gameplay video into the software 'Nvivo8'. It was then used to annotate all instances of their proposed patterns throughout chosen videos. Where the techniques differ is in the representation of the data. As shown above, the time index is displayed alongside the pattern(s). This allows a researcher to go to the exact point where the player was performing the action, and see what the player was doing and where in the level the player was doing it. For example at 0:27 in the Halo 4 gameplay video, the player is heading towards the hill in the middle of the map (PTm). He jumps over a gap and runs up the hill (PM-R). Once there at 0:32, he collects the 'Spartan Laser' weapon (Co) and almost immediately starts getting shot (PV), in which he retreats to safety.

Initial Pattern Set

In order to comprise an initial set of patterns; preliminary research was conducted. This involved searching for patterns that have a distinct effect on a player's movement. The patterns from the Milam and El Nasr (2010a) paper were the first. At this point the only changes were switching Pursue AI to Pursue Player (due to this work revolving around multiplayer and not singleplayer), and adding the 'Camping' pattern.

Visual identification of six gameplay videos was then conducted. The videos were selected from the most recent games in popular multiplayer series; each covering a specific play style. For example arena style gameplay (fast movement and jumping) and tactical style gameplay (slow movement and low health). During the analysis of these gameplay videos, gaps in the pattern set began occurring; where some of the actions performed by the players couldn't be accurately categorised by any of the six current patterns. This admitted additional patterns, along with modifications of existing ones. The derived patterns included: Reloading, Objective Action, Killstreak and Controlling Air Support. These patterns were added because they have distinct effects on the movement of players. For example if a player engages an enemy and wins the fight, they will very likely reload their weapon. Since the player will be vulnerable whilst reloading, they may slow down or completely stop and take cover.

Modified patterns included Pursue Player – visual and team (PPv and PPt), Path Target – movement and visual (PTm and PTv). These patterns were modified because they can cover two distinct behaviours each, and required extensions to illustrate which the player is performing. For example Path Target covers a player either looking at or moving towards architecture in the level. The player could be moving towards a section of the level or could be standing still and aiming at it, if they believe it is where enemy players are. Since movement is completely different in these two circumstances, the patterns should reflect this.

Results of Analysis

The finalised set of game design patterns for multiplayer shooter games was presented in a 'score card' style based on the style used by Björk and Holopainen (2005). Each score card listed the pattern's name, description, usage, consequences of use and its relations. They were written in a manner that separates each one and doesn't require the reader to understand one pattern to understand another.

The following is the full list of the finalised patterns, however to remain concise only the description is presented.

Path Movement and Resistance (PM-R)
A path movement goal that encourages players to move in a certain way, any obstruction to this movement goal is resistance that the player must overcome.

Path Target – visual (PTv)
Areas or objects in the map that stand out for players and that can be used to draw their attention through the use of a view finder e.g. a gun sight.

Path Target – movement (PTm)
Areas or objects in the map that stand out for players and that can be used to guide their movement.

Collection (Co)
Refers to the act of players exploring the map in search of collectables, such as health, armour, weapons, ammunition and objective items.

Reloading (R)
Refers to players altering their current movement path to allow them to safely reload their weapon.

Objective Action (O)
Requirements of a player to perform certain actions or be in a certain location in order to achieve an objective.

Player is Vulnerable (PV)
Anytime that the player is susceptible to injury, caused by enemy players or from environmental hazards.

Camping (Ca)
Strategically advantageous areas of the level that a player chooses to remain for a prolonged period of time.

Pursue Player (PP)
Player engagement through either direct combat or through chasing behaviour.

Pursue Player – visual (PPv)
Player engagement through visual acquisition of enemy players.

Pursue Player – team (PPt)
Friendly player engagement through flocking behaviour.

Killstreak (K)
Where a player reaches a high amount of kills or points, and then begins moving and behaving more cautiously.

Controlling Air Support (CAS)
Instances where the player stops moving and takes control of something other than their character, such as a drone, a chopper or an AC-130.

Emergence of Patterns in Different Titles

This section presents data recorded from gameplay videos of the following commercial games: Halo 4, Call of Duty: Ghosts, Battlefield 4, Team Fortress 2, Gears of War: Judgment and Unreal Tournament 3. As shown in figure 1, the proposed patterns can be used to determine what kind of architecture each level in these videos has, by simply looking at the chart. For example PM-R stands for Path Movement and Resistance, which is where the player is moving towards a goal but has to avoid physical obstacles such as walls, rocks, buildings, trees etc. The paths that the players took in the Call of Duty: Ghosts and Gears of War: Judgment videos; had more obstacles than that of Unreal Tournament 3. This shows that the level in UT3 was more open than the others.

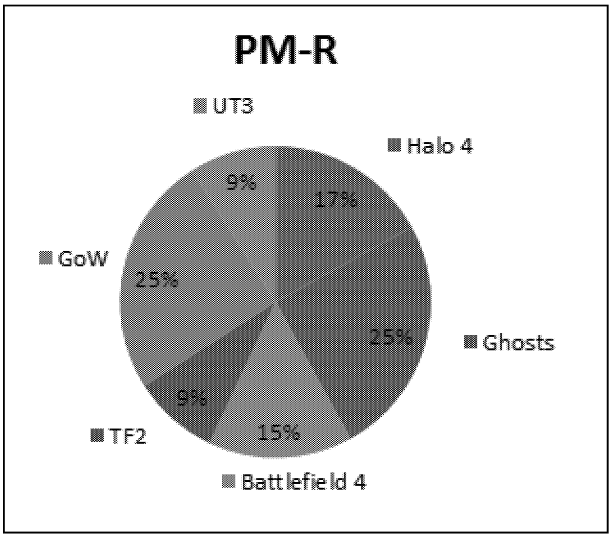


Figure 1: Data collected for the Path Movement and Resistance pattern in round one of Visual Identification.

Similarly for the Collection pattern (which involves collecting any kind of pickup in a level), some levels/games have more frequently collected pickups. As shown in figure 2, the player in Gears of War: Judgment collected the most pickups, where the player in Battlefield 4 collected none.

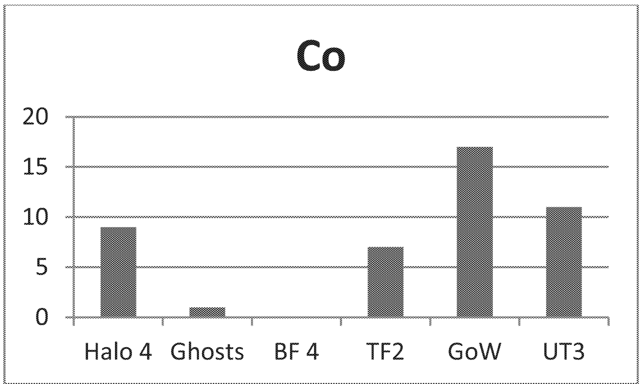


Figure 2: Data collected for the Collection pattern in round one of Visual Identification.

As mentioned previously the games selected can be categorised as either arena or tactical style shooters. The data presented in figure 3 supports this. Three of the six games have a higher frequency of the Pursue Player pattern, which means that the players were more likely to be moving whilst engaging the enemy. These three games are the 'arena' style games mentioned. UT3 is slightly lower than Halo 4 and GoW due to its larger map. The other three are the tactical style games due to the more static player engagement and reduced chasing behaviour. This is also illustrated in figure 3 through a lower PP frequency for these games.

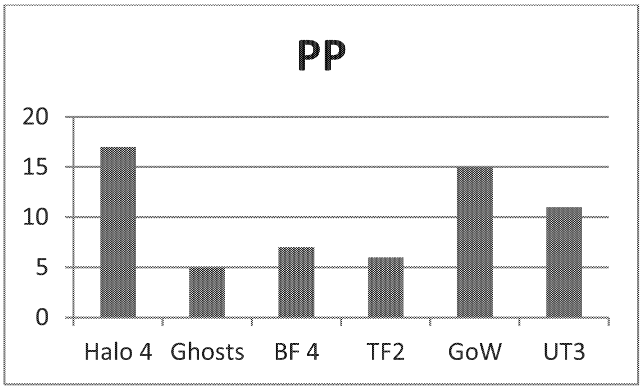


Figure 3: Data collected for the Pursue Player pattern in round one of Visual Identification.

VALIDATION

Although game design patterns are valuable in understanding player behaviour after the fact, they are also potentially useful tools for use in the design process itself. In order to effectively validate the pattern set, it was used in the iterative development of a multiplayer level. This "Testing Level" was then analysed to see if patterns led to the player behaviours predicted. The patterns were used as reference when looking at various levels in published games. For example on the level "Valhalla" featured in Halo 3 (Bungie, 2007), the focal point of the map is a hill centred mid map. This is an excellent application of the Path Target and Pursue Player patterns, as it can be used to guide movement (PTm) as well as used as a vantage point for acquiring targets (PPv). Figure 4 below illustrates the hill in Valhalla, along with the inspired creation in the Testing Level.

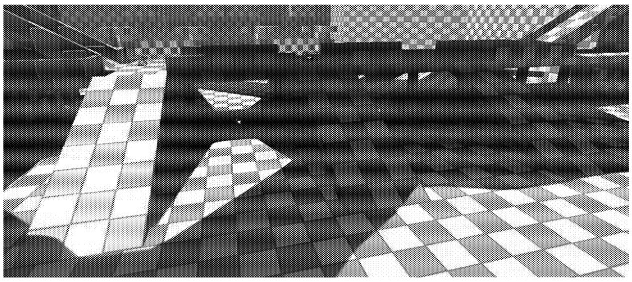


Figure 4: Images of the hill in Valhalla (top) and the bridge in the Testing Level (bottom).

The level was created using the Unreal Development Kit (Epic Games, 2009) due to its ease of use and LAN features. However the application of this framework is not dependant on the game engine nor a specific platform. The design adopted a "bare-bones" style, where the only aesthetic additions are those that influence gameplay; such as crates and team colours. Beyond that the only texturing was done using UDK's default texture. This also provided a virtual environment feel to the design.

User Questioning

To determine whether the patterns are accurately representing player behaviour and intentions, a questionnaire was created. It comprised of 20 questions, each based around an action related to a specific pattern(s). The scoring took the form of a 10 point Likert scale, allowing the users to answer how likely they are to perform the action presented in the question. For example one question comprised of: "*In capture the flag style games do you mainly defend your team's flag?*". This question related to the 'Objective Action' and 'Camping' patterns.

Play Testing (LANs)

To validate the pattern set, a multiplayer level was designed and created using the Unreal Development Kit. In this level several play sessions were held. The play testers consisted of volunteers from the University of Lincoln. They played multiple games of the Capture the Flag game type each session and their gameplay was recorded. At the end of each session they were asked to express their thoughts on the good/bad points of the level along with any suggested improvements. These suggestions were used with reference to the pattern set to make the relevant changes for the following session.

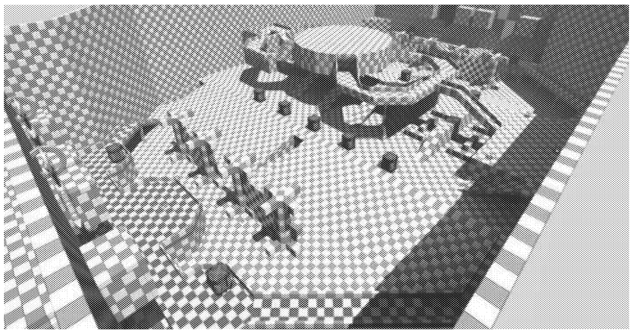


Figure 5: Image of an overhead view of the Testing Level.

This play testing was used to apply the proposed pattern set to the level design/creation scenario that it could be used for. The gameplay recorded was used in another round of Visual Identification and was used to compare the latest iteration of the level to an earlier one. The following are some examples of the patterns observed in player behaviour, and how level changes can influence how frequently these patterns occur.

With regards to the Testing Level, the frequency of the PM-R pattern increases between iteration 2 and iteration 4. This is due to the addition of an underground section, providing a more closed-in environment and more obstacles i.e. resistance. Figure 6 below displays a summation of the data collected from six different user's gameplay (three from each iteration). The PM-R pattern occurred a total of 54 times in iteration 2, and increased to 67 in iteration 4.

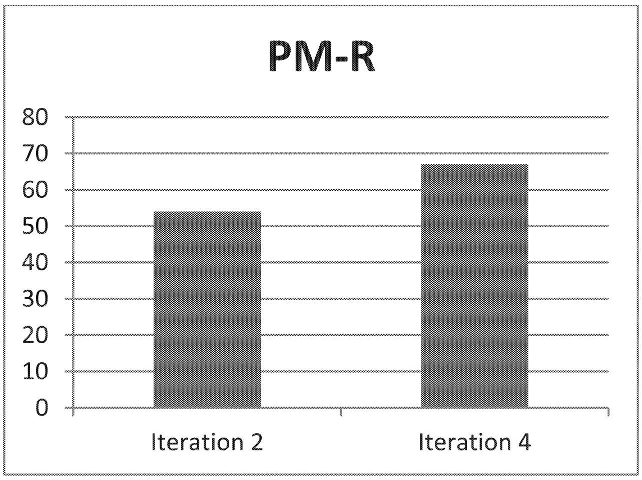


Figure 6: Data collected for the Path Movement and Resistance pattern in round two of Visual Identification.

The frequency of the Collection pattern also increased, rising from 39 to 71 as shown in figure 7 below. This was due to the increase in ammo, health and weapon pickups throughout the level.

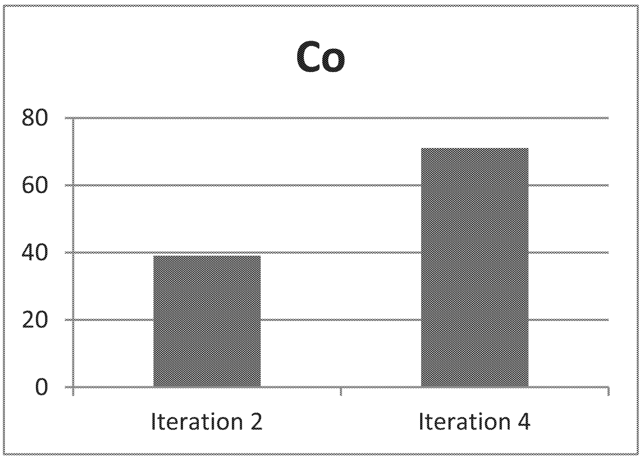


Figure 7: Data collected for the Collection pattern in round two of Visual Identification.

CONCLUSIONS

This paper presented a framework for designing for and analysing player movement and behaviour. It proposed a pattern set specifically directed at multiplayer shooter games. The processes presented aided in validating the framework and in applying it to a real world scenario; where the pattern set was used in the design, creation and testing of a multiplayer level.

It also provides examples of how a level's architecture can encourage certain behaviour from a player, and how it can influence a specific play style. It suggests that the majority of player actions have an effect on a player's movement, be it increased or reduced. Overall it proposes that understanding patterns in player behaviour can allow designers to predict how players will interact with a level; or allow designers to analyse the interactions after the fact.

Future Work

Possible future work could be to use the proposed pattern set to influence the design of several multiplayer levels; each with a distinct play style in mind. For example a small enclosed level to promote PM-R, PP and PV, a large open level for PPv, PTv, PTm and Ca, along with a mid-design such as two enclosed bases and an open area between, to promote all patterns. Recorded gameplay of these levels can then be analysed using Visual Identification to determine whether the expected design patterns are present.

Another possibility is to use the pattern set to analyse a multitude of games, other than the six already reviewed. This would further aid in validating the set, by considering how a variety of games of different styles effect its application. This is also how Milam and El Nasr continued their work (2010a), by reviewing a further 21 games on top their initial four (2010b).

REFERENCES

S. Björk and J. Holopainen. *Patterns in game design*. Chicago: Cengage Learning, 2005.

Epic Games (2009) Unreal Development Kit. [software] version July 2013. North Carolina, USA. Epic Games. Available from: https://www.unrealengine.com/products/udk/. Accessed: [Wednesday 4th December 2013].

J. Feil and M. Scattergood. *Beginning game level design*. Course Technology PTR, 2005.

D. Milam and M. S. El-Nasr. Design Patterns to Guide Player Movement in 3D Games. *Proceedings of the 5th ACM SIGGRAPH Symposium on Video Games*, Los Angeles, 25-29 July, Los Angeles, USA: ACM, 37-42, 2010a

D. Milam and M. S. El-Nasr. Analysis of Level Design 'Push & Pull' within 21 games. In: *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, California, 19-21 June, California, USA: ACM, 139-146, 2010b.

LUDOGRAPHY

2K Games (2007) *Bioshock*. [Console game: Xbox 360

343 Industries (2012) Halo 4. [Console game: Xbox 360] Washington, USA: Microsoft Game Studios.

Bungie (2007) Halo 3. [Console game: Xbox 360] Washington, USA: Microsoft Game Studios.

Bungie (2010) Halo Reach. [Console game: Xbox 360] Washington, USA: Microsoft Game Studios.

EA Digital Illusions (2011) Battlefield 3. [PC, Console Game] Redwood Shores California, USA: Electronic Arts.

EA Digital Illusions CE (2013) Battlefield 4. [PC, Console Game] Redwood Shores California, USA: Electronic Arts.

Epic Games (2007) *Unreal Tournament 3.* [PC, Console Game] North Carolina, USA: GT Interactive.

Epic Games (2013) *Gears of War Judgment.* [Console game: Xbox 360] Washington, USA: Microsoft Game Studios.

Infinity Ward (2007) *Call of Duty 4: Modern Warfare*. [Console, PC Game] California, USA: Activision.

Infinity Ward (2013) *Call of Duty: Ghosts*. [Console, PC Game] California, USA: Activision.

Valve Corporation (2007) *Team Fortress 2*. [game]. Washington, USA: Valve Corporation.

AUTHOR BIOGRAPHY

RICHARD LANNIGAN studied Games Computing at the University of Lincoln and graduated in 2014 with a first class degree. Has a strong interest in game design, with a specific focus on level design. Career goal is to become a lead level designer or creative director for a high profile company in the games industry, working on AAA titles.

PROCEDURAL GENERATION OF RACE TRACKS IN AN OPEN SOURCE RACING GAME

Jordan Blake and Grzegorz Cielniak
School of Computer Science,
University of Lincoln, UK
Email: jordan.blake@outlook.com, gcielniak@lincoln.ac.uk

KEYWORDS

Procedural generation, The Open Race Car Simulator, TORCS, track generation, open source, racing game

ABSTRACT

This paper presents a system for the procedural generation of race tracks, implemented in The Open Racing Car Simulator (TORCS). Race tracks that users perceive as fun are a key element of racing games. However, the process of hand-crafting a playable race track is time consuming. We create a system that procedurally generates new tracks that meet a given design specification and are perceived as fun by players. Assessment is conducted by quantitatively comparing generated tracks to the design specification (the default TORCS tracks), and by collecting user feedback on the level of fun and difficulty experienced with the generated tracks.

INTRODUCTION

Procedural content generation (PCG) in video games has been a point of interest for developers for almost as long as video game consoles have been in the home. A well-known early example of procedurally generated content in video games is *Elite* (Braben and Bell 1984), a space game for which many different solar systems are generated at run-time for the player. This allows every new game to widely vary, offering a vast increase in entertainment over static content. Furthermore, PCG simplifies the content creation process of video games and thus reduces the time consuming and intensive process of handcrafting in-game content (Yannakakis and Togelius 2011).

Recent research into procedurally generating race tracks has explored complex techniques to generate new tracks to exacting specifications (Togelius et al. 2007) (Cardamone et al. 2011). These methods are highly computationally expensive and do not provide players with the instant content – long wait times in video games breaks flow and interrupts the player's experience (Johnson and Wiles 2003). Therefore it is desirable to develop PCG methods that provide players fun and exciting content delivered in the shortest possible time. This paper presents a race track generation system integrated with the open source racing game TORCS (Wymann et al. 2013). The system uses a justifiably chosen model and procedurally generates the gameplay aspects of new race tracks. An example of the race track output is observable in *fig 2*. The track is analytically assessed for similarity to the model. A qualitative assessment of their fun and difficulty is performed through user testing against random tracks, and the time taken for generation is recorded. The paper concludes with discussion of future work.

TORCS

The Open Racing Car Simulator (TORCS) is a cross-platform open source racing car simulator, written in C/C++. TORCS tracks are loaded as an ordered list of individual segments. A segment is an individual part of the track that has a number of properties affecting how it appears in-game. The first property is the **type**, which defines the segment as either a straight (**str**) or a curve (**lft** or **rgt**). The proceeding properties then depend on the type of segment. It is these values that are procedurally generated for the creation of a new track.

Curved segments are essentially sections of ovals. They take the properties (visualised by *fig 1*):
arc – the amount of curve, in degrees
start radius – the distance from the start point to the centre of the curve
end radius - the distance from the start point to the centre of the curve

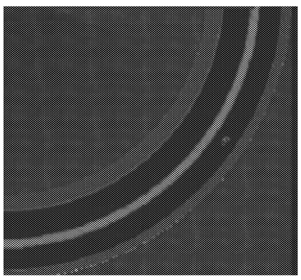


Figure 1: A Representation of a Curved Segment, Where the Start Radius (Green) and End Radius (Blue) are Both 100, and the Arc (Pink) has a Value of 90

RELATED WORK

TORCS is a popular playground for researchers who deal with the topic of racing games, and it has been used for existing research on procedural track generation. One example is *Interactive Evolution for the Procedural*

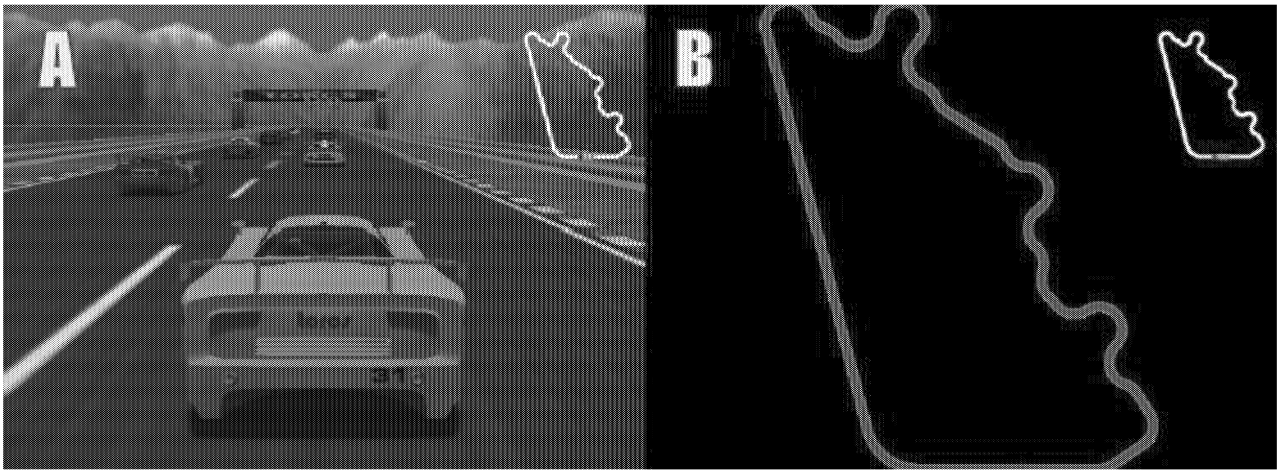


Figure 2: (A) Shows an In-Game Screenshot of TORCS Running Our PCG Track. (B) A Full Overview of the Generated Track

Generation of Tracks in a High-End Racing Game, which uses evolutionary algorithms to generate new tracks (Cardamone et al. 2011).

A system of evolutionary generation for race tracks is detailed by Togelius et al. (2007). It uses a model to create a track segment by segment, and tests all possible values for each segment of the track. The track model used is based on player data, and adjusts the generated tracks to suit their individual play styles. The result is a track that suits the input track model very well, though the video game used is simple and two dimensional with limited physics.

Cardamone extends Togelius' work, with implementation in TORCS. In their work, generation takes 20-30 minutes to generate 10 tracks. To avoid players having to wait in-game for tracks to generate, they have two separated aspects of their program: a web front end that a user interacts with to create the tracks, and a backend that deals with the time-consuming evolutionary computation. Users can queue tracks to be created and download them at a later time. There is also a bank of existing tracks that can be downloaded immediately to be played. The system provides generation of tracks with likeness to the model, but at the expense of time and with a large amount of computational power necessary.

TRACK MODEL & GENERATION

In order to quantify how fun a race track is and how difficulty is a factor, it must be understood how difficulty relates to fun. As mentioned in the previous section, existing PCG uses player data for their track models to create content suited to particular play styles. This differs from the design of commercial racing games, which almost universally have a number of pre-made tracks either crafted by a designer, or based on a real-world location. Players are not given tracks designed specifically for them, but are expected to adapt their play accordingly.

In the research for this paper, recognised definitions of fun were explored. A Theory of Fun (Koster 2013) states that fun as the mastery of a mental challenge, and Malone (1980) classifies fun as involving challenge, fantasy, and curiosity. By adapting tracks to player performance, there is a risk of removing the element of challenge, and if the model is predictable the curiosity is also compromised. Commercial racing games, with a variety of differing tracks, can invoke challenge through forcing the player to adapt themselves whilst also not compromising the integrity of curiosity. As a result of this, it is concluded that a track model without direct input of the player is desirable. Existing tracks proven to be fun could provide a worthwhile model to use.

Existing TORCS tracks are accessible for analysis as their data is presented in plain text. We use the popular and widely enjoyed tracks packaged with TORCS as our model and analyse the segment information. MATLAB (MathWorks 2013) is used for the analysis of data in this project. The existing data used to generate the model is analysed through histogram representations.

As TORCS is written in C/C++, this project's artefact is a C++ application. The C++ standard library (STL) contains many different mathematical distributions that generate random numbers in a predictable pattern based on input values. These distributions are quantitatively compared to the existing data through the use of MATLAB functions that quantifies the fit of a distribution to input data.

Despite TORCS being able to handle various different segment heights and advanced track elements such as banking, we focus on finding suitable lengths, arcs, and radiuses with the objective of the techniques being transferrable to other track attributes. To find the ideal values, the existing TORCS data was combined and analysed. MATLAB is used to generate histograms of the different attributes in our model (*fig 3* shows the arc

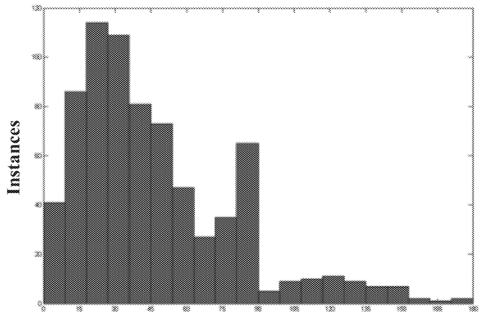


Figure 3: Arcs of all Tracks Packaged with TORCS

values). The histogram data is then visually compared to the data in different distributions in the C++ STL. Distributions provided with the C++ STL are compared ad hoc to see which provided the closest fit to the existing data. The Weibull distribution (*fig 4*), a continuous probability distribution, is chosen as the closest fit for all track segment properties, providing a close approximate fit to the values gathered from TORCS.

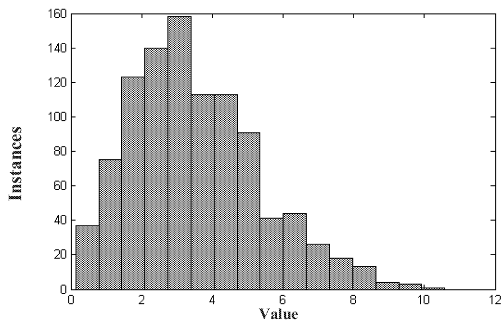


Figure 4: The C++ Weibull Distribution

The results from the C++ STL implementation of the Weibull distribution with seed value 0 shows a visually similar distribution to the model data in *fig 3*. The Weibull distribution has two values that affect its output – shape (2.0 in *fig 4*) that affects the curve, and scale (4.0 in *fig 4*) that affects the range. To determine which input values are suitable for the track generation, a MATLAB function – wblfit - returns estimated values for a Weibull distribution that matches input values provided. TORCS data was read into MATLAB and fit values were returned (*table 1*). The results of generation using these values are visible in *fig 5*.

Table 1: Results of the Weibull Fit Function with Input Data from TORCS

Attribute	Shape	Scale
Length	49.40	1.34
Arc	67.57	1.70
Radius	52.21	0.79

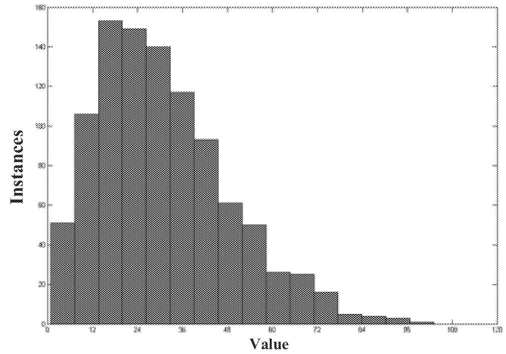


Figure 5: C++ Generation of Weibull Values from the Results of the Weibull Fit Function in *Table 1*

The result of the generation in *fig 5* is similar to that of the input data in *fig 3*. The resulting tracks are procedurally generated up to a track length value that is modifiable via a configuration text file. The resulting track using data formulated from this method is visible in *fig 6*, with the model accounting for the generation of the track from the green marker to the red marker.

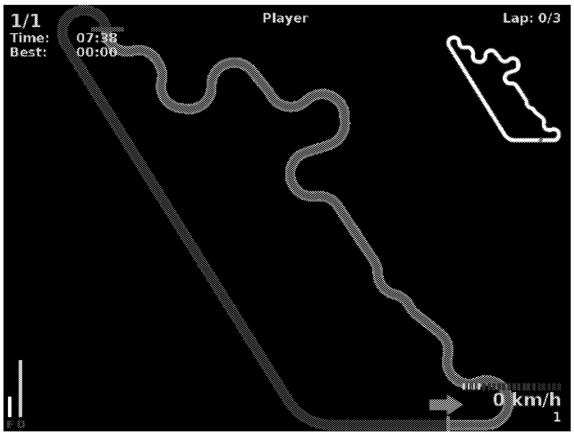


Figure 6: A Display of the Generated Track Segments, Starting at the Green Marker and Ending at the Red

Tracks in TORCS are required to be fully connected to function properly (with the final segment joining the first segment). If tracks are left open at both ends, the starting point for the track faces towards an open end hanging in mid-air (the crossing of which causes the game to immediately crash). This issue proved difficult to solve, and is resolved by performing a function to return to the starting point once the generation had ended. As a result, all tracks have a long straight section before returning to the start point (visible in the faded segments in *fig 6*). Ideally the entirety of the track would be procedurally generated, but there were difficulties in the implementation, so long straights remain to prevent the application crashing. There has been no recorded crash of the game caused by generated tracks, so the straight segment does serve its purpose.

RESULTS

To quantitatively assess the similarity of the PCG tracks against the packaged TORCS tracks that are used as the model, we use the same assessment strategy as used for their creation – the wblfit function in MATLAB (which generated the data in *table 1*). This shows how the data fits the expected distribution. With use of the fit function and by comparing the average values of each parameter, similarity between the model and the output can be quantified. *Table 2* shows attributes of TORCS tracks and ten PCG tracks quantified for comparison.

Table 2: A Comparison of the TORCS Values to Those of the Generated Tracks

	Mean	STD	Weibull Scale	Weibull Shape
TORCS lengths	85.80	78.83	49.40	1.34
PCG lengths	71.62	94.88	39.78	1.46
TORCS arcs	47.61	34.34	67.57	1.70
PCG arcs	59.37	37.27	60.68	2.22
TORCS start radius	171.08	283.66	52.21	0.80
PCG start radius	69.74	82.05	62.14	1.33

The results demonstrate strong similarities between the input data and the generated tracks. The PCG tracks show shorter lengths, but with a greater standard deviation that provides balance. They particularly have a nearly identical distribution shape. The compared arc results are similar to those of the lengths, with nearly identical standard deviations. The mean and Weibull values also share a likeness. The largest disparity is with the corner radiuses where TORCS tracks show an extreme mean and standard deviation. However, this is explained by observing from the raw input data that the input range is very wide, with the greatest input values appearing very few times (*fig 7*). The key similarity is that the shape parameter bears likeness to the original.

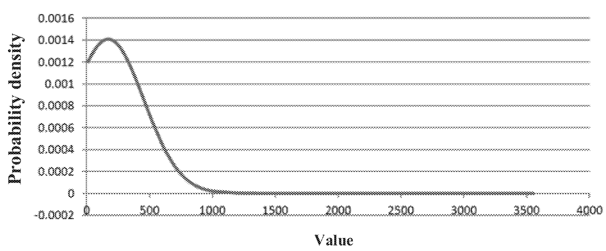


Figure 7: A Normal Distribution of the Start Radius of the Model with the Range of Values Along X

For qualitative analysis, user data was captured through a questionnaire with five participants, all of whom are avid gamers across a variety of genres. Users were asked to complete three different (to them, unlabelled) tracks:

- A PCG track using the method detailed in this paper
- A track that comes packaged with TORCS
- A pure random track that uses a uniform distribution for its values (*fig 8*)

The choice of a purely random track was to judge whether there is any qualitative difference in using a determined model versus a simple random number generator. In the questionnaire users were asked which track they enjoyed the most, which they found the most difficult, and whether the difficulty of the tracks increased the amount of fun they had.

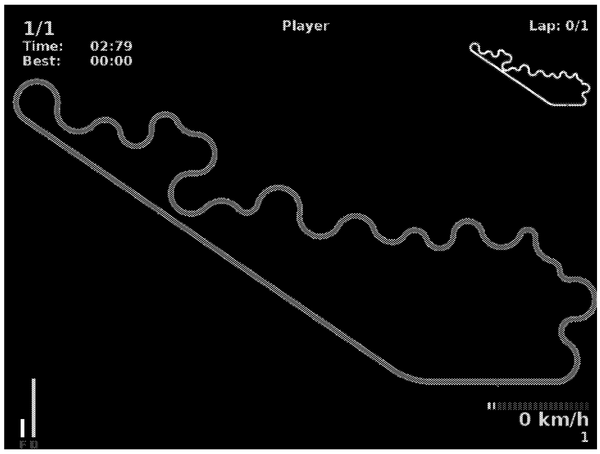


Figure 8: A Track Made with Random Generation (Uniform Distribution) for User Testing

Users reported that they found the pure random tracks to be too difficult, noting the frequency of "twisty" corners that caused them issues. The procedural tracks were almost unanimously preferred to the random tracks, though the TORCS track was received the most favourably. The random track was noted as the most difficult by 80% of participants, but 60% stated that the added difficulty did not increase their enjoyment.

An objective of this project was to assess how players of different skill levels perceived the difficulty of tracks in correlation to the fun they experienced. Due to the low sample size with all participants assessing themselves as being similarly skilled, this assessment was not possible.

As speed of generation is a key aim for this project, the time for track creation was recorded. For these tests an Intel i5-2500K @ 3.3 GHz and an Nvidia GTX 560Ti were used. The resulting times include generation of the XML file and the 3D file, which make the track accessible in-game. *Table 3* shows the time for 250 tracks, all of which were deemed playable (a track is invalid if it contains intersecting segments or does not have connecting ends). If a track is not classified as playable, the generation is retried.

Table 3: The Timing of the System over 250 Generations

Total time (ms)	Mean (ms)
198211	792.84

When tracks generate on a first attempt, they are generally completed within 0.5-0.8 seconds. Cardamone's research shows an output of 10 tracks in 20-30 minutes. Our system can generate 250 tracks in under four minutes, showing a significant increase in the speed of generation.

CONCLUSION

This system generates tracks to be played nearly instantly in TORCS at an average generation time of less than 0.8 seconds (*table 3*). Generated tracks are quantitatively similar to the input model and are received well by users.

This system shows a significantly faster generation time than previous work, generating tracks over 25 times the number of tracks in less than half the time. A fast system like this can be integrated with a video game without breaking the flow of the experience, removing the possibility of frustration a player may feel from long wait times. Other systems have advantages, providing a result closer to their models through the use of evolutionary computation. Though this project goes through as few processing stages as possible to save time, outputted tracks remain similar to the provided model.

Prior to user testing, it was expected that the unintentional long straight (prevalent on all tracks) would be poorly received, but it did not seem to greatly impact player's perceptions. However, it would still be optimal to create an implementation that can fully close tracks with every segment being part of the input model. User testing showed that a model-based system does provide greater playability than a track based on uniform random numbers. This particularly validates this project for further development. The testing regarding whether player skill affects reception to PCG content was not assessed, in part due to the low number of testers, all of whom had extremely similar backgrounds in racing games. Testing with users who can give a variety of different opinions would benefit future iterations.

In summary, although the system does not provide tracks that are of an equal quality to the handcrafted tracks provided with TORCS, the results show promise for future developments. The generation does show that a considered approach to generating a model is worthwhile and has the potential to provide fun tracks. The initial aim of a speedy generation is met, allowing for a user experience that is different every time, and with a fun factor that beats completely random generation.

Further work can add more elements to the model, possibly including more factors that TORCS supports but were not included in the project. This project focused solely on gameplay, but the principles could be applied to the aesthetic properties of tracks. Different approaches could be made to eliminate the straight line artefact that is unique to this implementation, making as much of the tracks (or ideally, the entirety of them) procedurally generated according to a model.

REFERENCES

Braben, D. & Bell, I., 1984. "Elite", Acornsoft.

Cardamone, L. Loiacono, D. & Lonzi, P.L. 2011. "Interactive Evolution for the Procedural Generation of Tracks in a High-End Racing Game". In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, pp. 395-402.

Johnson, D. & Wiles, J. 2003. Effective affective user interface design in games. Ergonomics, 46(13-14) 1332-1345.

Koster, R. 2013. *A Theory of Fun For Game Design.* O'Reilly Media, Inc.

Malone, T. W. 1980. "What makes things fun to learn? Heuristics for designing instructional computer games". In *Proceedings of the 3rd ARM SIGSMALL symposium and the first SIGPC symposium on Small systems*, pp. 162-169.

MathWorks, 2013. *MATLAB*

Togelius, J. Nardi, R.D. & Lucas, S.M. 2007. "Towards automatic personalised content creation for racing games". In *IEEE Symposium on Computational Intelligence & Games*, pp. 252-259.

Wymann, B. Espié, E. Guionneau, C. Dimitrakakis, E. Coulom, R. & Sumner, A. 2013 "TORCS: The Open Racing Car Simulator" v1.3.5.

Yannakakis, G.N. and Togelius, J. 2011. Experience-driven procedural content generation. *Affective Computing, IEEE Transactions on,* 2(3) 147-161.

BIOGRAPHIES

Jordan Blake is an undergraduate student at the University of Lincoln, UK. He is currently undertaking his MComp degree in Games Computing. His research interests include procedural content generation, AI, and robotics.

Dr. Grzegorz Cielniak is a Senior Lecturer at the School of Computer Science, University of Lincoln, UK. He obtained his Ph.D. in Computer Science from Orebro University, Sweden in 2007 and MSc in Robotics from Wroclaw University of Technology, Poland in 2000. His Ph.D. thesis addressed a problem of real-time people tracking for mobile robots. Dr. Cielniak was awarded a fellowship through the European Union Marie Curie Actions programme (FP6) during which he visited the Autonomous Intelligent Systems Laboratory in Freiburg, Germany. He co-authored more than 30 peer-reviewed publications (h-index: 11). His research interests in games technology include procedural content generation, physics simulation and AI.

STORYTELLING

Procedural Story Generation in Games

Kieran Wagg and Grzegorz Cielniak
School of Computer Science
University of Lincoln
E-mail: kieranw1992@yahoo.co.uk, gcielniak@lincoln.ac.uk

KEYWORDS
Procedural Content Generation, Storytelling in Games, Narrative, Interactive Fiction

ABSTRACT

This paper examines the viability of using procedural stories in video-games, and investigates how these stories compare to traditional storytelling methods. A system for generating such narratives is developed, implemented and tested. The result is a system capable of simulating the actions of multiple characters and procedurally generating stories users find interesting. The experiments evaluated the feasibility and quality of procedurally generated stories, using user experiences to understand how they perceive the resulting narratives.

INTRODUCTION

When it comes to developing video-games that provide entertaining and engaging experiences for players, a good 'replay value' is desired. This value is an arbitrary way of describing how enjoyable it is to play a video-game multiple times, with the goal being a constant level of enjoyment every time. For story-driven video-games, however, this is challenging, as the narrative contributes to the main appeal. This means, after experiencing it once, the story can become less interesting, as the player already knows what will happen.

To overcome this, developers are incorporating multiple branching narratives into such video-games, allowing players to take different paths every time. However, this still leaves the player with a finite number of play sessions before they have experienced all stories the game has to offer. The use of procedurally generated narratives helps solve this issue, as the amount of variation they provide is easier to establish than comparable pre-planned narratives. These stories present their own challenges, as the absence of the designer's planning means other measures are necessary to ensure the story is coherent.

This paper advances the research into procedurally generated narratives by developing and implementing a system capable of creating stories that users find interesting and engaging. These stories focus on using believable characters, and the interactions between them, to develop these narratives, with the generation system drawing from the existing game *Façade* (Mateas and Stern, 2005). User testing assesses the quality of the implemented generator and its stories, with both metric and questionnaire data evaluating the feasibility of using these procedural narratives to create engaging player experiences.

RELATED WORK

Story Generation

Procedural content generation is used in video-games for, amongst other things, producing purely random content, creating characters and quests, and generating content based on user actions (Togelius *et al*, 2011). This allows the developers to spend less time creating content and focus more on the development of the game itself. These approaches also benefit players by either providing unique experiences, or adapting the game to suit their preferences.

Research is also investigating procedurally generated narratives, often regarding the application of such stories to video-games. The *Carnegie Mellon Oz Project* (Carnegie Mellon University, 2001) developed a computer system to create 'interactive dramas'. To do this, they utilised a 'drama manager' to handle the generation of the story by organising its important moments, or 'plot points'. This drama manager can adapt the generated stories in response to the users' interactions, but is not dependent on these interactions, allowing the story to continue without them.

Additionally, in order to maintain the user's suspension of disbelief, 'believable characters' are used (Mateas, 1999). This is achieved by utilising the 'Eliza effect', an illusion in which a computer system seems more complex than it actually is (Bates, 1992; Wardrip-Fruin, 2009), to create characters capable of acting like people, instead of merely resembling them.

Façade (Mateas and Stern, 2005) demonstrates the potential for story generation systems to create interesting and enjoyable experiences for players. Narratives are generated using a 'beat sequencer' to select and organise the available plot points, creating a story that adapts to the players actions. The way *Façade* selects its beats is guided by a graphical representation of the desired tension values for the story at various points.

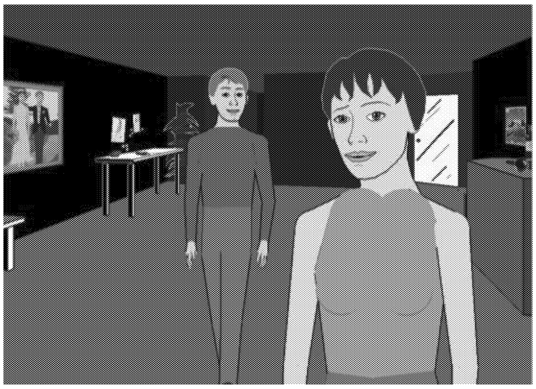


Figure 1: Screenshot of Façade

This 'tension graph' supplies the narrative's structure, providing a template for how it should progress. The current level of tension represents the story's state, with a function defining the desired level of tension at any time. The beat sequencer then selects plot points to match these target values as best as possible (Mateas and Stern, 2003). This allows the narratives to stay consistently satisfying, while maintaining some variety. These graphs are user-defined, with designers selecting target values that will result in the type of story they want the system to create.

Representing Characters

The Five Factor Model is a psychological model that defines the following traits as fundamental aspects of personality and behaviour: openness (to experience), conscientiousness, extraversion, agreeableness and neuroticism (Stangor, 2011, 601). This provides a solid foundation for representing characters as it can provide simple representations of complex personalities. Despite this, many psychologists acknowledge that this model is a flawed system for predicting human behaviour, as its traits are 'highly general' (Goldberg, 1981).

METHOD

To advance research concerning procedural storytelling, this paper proposes a solution that works within a video-game's Artificial Intelligence module. While always using the same structure, the implementation of this solution is dependent on the video-game's requirements, the engine used and the preferences of the developers.

Narratives are generated using two aspects of the story: its characters and their actions. Various models specify the capabilities of these areas, and a generation system utilises these models to build each narrative as the game progresses.

Generation System

The proposed story generation system treats stories as a sequence of actions between two points in time. Characters perform actions, and each action elicits a reaction from the other character involved (i.e. when character A talks to character B, character B reacts). The generator also uses these actions to create pre-scripted sequences, called events. These are then used as larger scale actions, with subsequent actions seeming like reactions to the event. By combining these events and actions, the generator can create narratives by organising events and simulating character actions between them (see Figure 2).

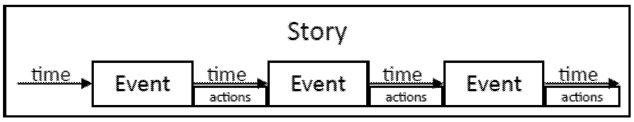


Figure 2: Composition of a Story

In order to arrange these events coherently, an 'event organiser' decides which event will happen every time a new event is needed. Each event has preconditions for when it can occur, called prerequisite and exclusive events, which allow the event organiser to ignore any invalid events. An event is only valid if its prerequisite events have already happened and its exclusive events have not.

To ensure each generated narrative is interesting for the user, the generator assigns the story a state. This state is updated with every action/reaction pair, allowing progress to be monitored. The generator ensures the narrative is 'interesting' by specifying a set of target values for the state, represented using a tension graph. When selecting the next event, the event organiser determines which events will change the story's state to match the current target best. This leaves a set of possible events that will result in an interesting story for the user.

Character Model

Characters are important for generating compelling narratives, as the stories are derived from their behaviour. While some aspects are only necessary for differentiating between characters, others affect their actions (and therefore, the story). Figure 3 shows how these parameters interact when characters choose their actions.

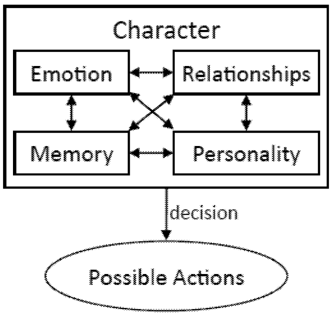


Figure 3: Interactions between Character Parameters When Making Decisions

The characters' personalities are the driving force behind their behaviour, as these values remain constant throughout the story. The others, however, change frequently, in order to adjust the characters' behaviour to suit the context. The implemented model uses five personality 'traits': kindness, confidence, extraversion, seriousness and aggression. These range from zero to one, with one meaning an excess of that trait and zero representing a lack of it.

The system uses a sliding scale of happiness to represent each character's emotional state, as this method is simple and directly correlates to the character's desired behaviour. Similarly, other characters' actions have straightforward effects, as receiving 'bad' actions make the character feel worse, while 'good' actions cheer them up.

Characters' relationships also use a sliding scale, with a value of one meaning they love the other character and zero meaning they hate them. Additionally, these values are independent of each other, such that one character's feelings for another may not be mutual, encouraging emergent behaviour.

Memories are stored as a list of all actions involving the character. These are used to determine the history between two characters, allowing a character's past actions to affect their current behaviour, as well as influencing other characters' actions towards them. However, these values are only used to improve the coherency of the characters' behaviour, with the other parameters having a larger influence over their decisions.

Action Models

Various models are used to control how actions interact with the story and its characters, as these comprise the bulk of the narrative and are crucial for making believable characters. The first model determines the probability a character will perform each type of action, allowing the character to choose their next action. To accomplish this, each action has a weighting corresponding to the values for the character's personality, emotional state and relationship with the target character (Table 1). The total weighting for each action is equal to the total value of each weight multiplied by the character's value for the corresponding trait.

Table 1: Action Dependency Table

	Personality					Emotional State	Relationship State
	Kindness	Confidence	Extraversion	Seriousness	Aggression		
Greet	0.25	0	0.25	-0.25	0	0.25	0
Compliment	0.166	0	0.166	0	0	0.166	0.5
High Five	0.166	0	0	-0.166	0	0.166	0.5
Hug	0	0.166	0	-0.166	0	0.166	0.5
Kiss	0	0.166	0.166	0	0	0.166	0.5
Ignore	-0.125	0	0	0.125	-0.125	-0.125	-0.5
Criticise	-0.166	0	0	0	-0.166	-0.166	-0.5
Slap	-0.125	0	0	0.125	0.125	-0.125	-0.5
Shout	0	0.166	0	0.166	0	-0.166	-0.5
Punch	0	0.166	0	0	0.166	-0.166	-0.5
Talk	0	0.166	0.166	0	0	0.166	0.5
Do Nothing	0	-0.25	-0.25	0	-0.25	-0.25	0

Following this, the total weighting is adjusted based on the two characters' history and the probable change in the story's state. The action's 'Global Tension Change' parameter (Table 2) is used to provide an estimated tension change for each action, as the actual change in tension depends on the other character's reaction. The estimated new value is compared to the current target tension, and the action's weighting adjusted based on how close this value is to the target. Once all possible actions have weights, the action with the highest weighting is chosen. If multiple actions are equally weighted, the character selects one of them at random (using a uniform distribution).

Table 2: Action Global Tension Change Values

Action Type	Global Tension Change
Greet	-0.010
Compliment	-0.020
High Five	-0.010
Hug	-0.030
Kiss	-0.015
Criticise	0.020
Slap	0.025
Shout	0.035
Punch	0.040
Talk	-0.010

Table 3: Action/Reaction Matchup Chart

Row = action Column = reaction	Greet	Compliment	High Five	Hug	Kiss	Criticise	Slap	Shout	Punch	Talk	Ignore	Do Nothing
Greet	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓
Compliment		✓	✓	✓	✓					✓	✓	✓
High Five			✓							✓	✓	✓
Hug				✓	✓					✓		✓
Kiss				✓	✓		✓			✓		✓
Criticise						✓	✓				✓	✓
Slap							✓	✓		✓		✓
Shout								✓	✓	✓		✓
Punch									✓	✓		✓
Talk		✓	✓	✓	✓	✓	✓			✓	✓	✓

The second model specifies which reactions are suitable for each action, preventing characters from reacting to an action inappropriately. For example, when a character attempts to high five them, a suitable reaction would be to high five them back, as opposed to punching them. This is complicated further by the fact that, of the twelve actions implemented, one never elicits a reaction (doing nothing) and one is used only as a reaction (ignoring someone).

Table 3 shows the list of possible actions (rows) and their valid reactions (columns), with a tick indicating a valid pair. For example, Greet/Hug (action/reaction) is a valid pairing, whereas Hug/Greet is not. This is because the Greet action is reserved for characters who have not yet met each other (i.e. interacted with each other). This therefore causes inconsistencies if, for example, a character hugs someone they already know, who then greets them.

When these actions and reactions resolve, they affect both the story's state and the characters involved. Each pair changes the story's tension level, the emotional state of the characters involved, and their relationships with each other. As the action/reaction pairs affect two different components of the story, two more models are used to detail these effects. One model is used to describe their effects on the characters, while the other describes their effects on the tension level. Overall, pairs of 'good' actions provide benefits, improving mood, increasing affection and reducing tension, while pairs of 'bad' actions have an inverse effect. Mixed pairs are more complicated, with the character who receives the 'good' action typically benefiting, while the other character suffers.

Implementation Details

The proposed system was implemented using C# and incorporated into a simple text-based game (see Figure 4). Data for the tension graph, character parameters and event details are stored in external text files, which are read in when the game starts. This allows the designer to create, edit or delete various components while preserving others.

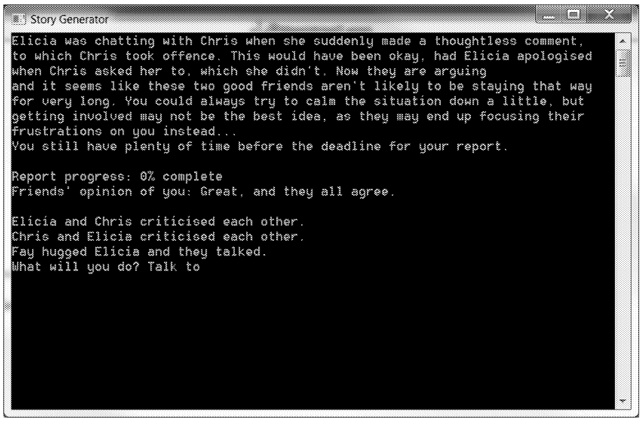


Figure 4: The Implemented System

EXPERIMENTS

Data Collection

Eight users tested the implemented system, with each user asked to play through three stories. Two stories were procedurally generated using two different tension graphs. This allowed the tests to determine which graph created the stories the users found the most interesting. The third was a 'control' story, using predefined sequences of events. This served as a way of confirming the system's ability to generate procedural narratives that were as interesting as predefined ones for users. It also assessed whether the users could tell when they had control over the stories' events. To reduce the impact of any external factors, the order in which users experienced the narratives varied, but all three were consistent for all users (i.e. all tests used the same control story and tension graphs).

During these tests, each stories' actual and target tension values were recorded throughout. Comparing these values for each story demonstrates how well the system was able to match the desired graph. The values' Root-Mean-Square-Distance (RMSD) provided a measure for their similarity. This measure was used to assess the system's capabilities because it is 'useful when comparing different models on the same set of data' (Hyndman and Koehler, 2006), which is the case for these values. The more similar the two sets of values (i.e. the smaller the RMSD), the better the system worked, with identical values meaning the system worked perfectly.

Furthermore, users completed questionnaires concerning their thoughts and opinions of the experienced stories. The questionnaire investigated users' opinions regarding coherency, believability, consistency, appeal, level of control, and duration using a Likert scale. The users were also asked which of the three stories they preferred.

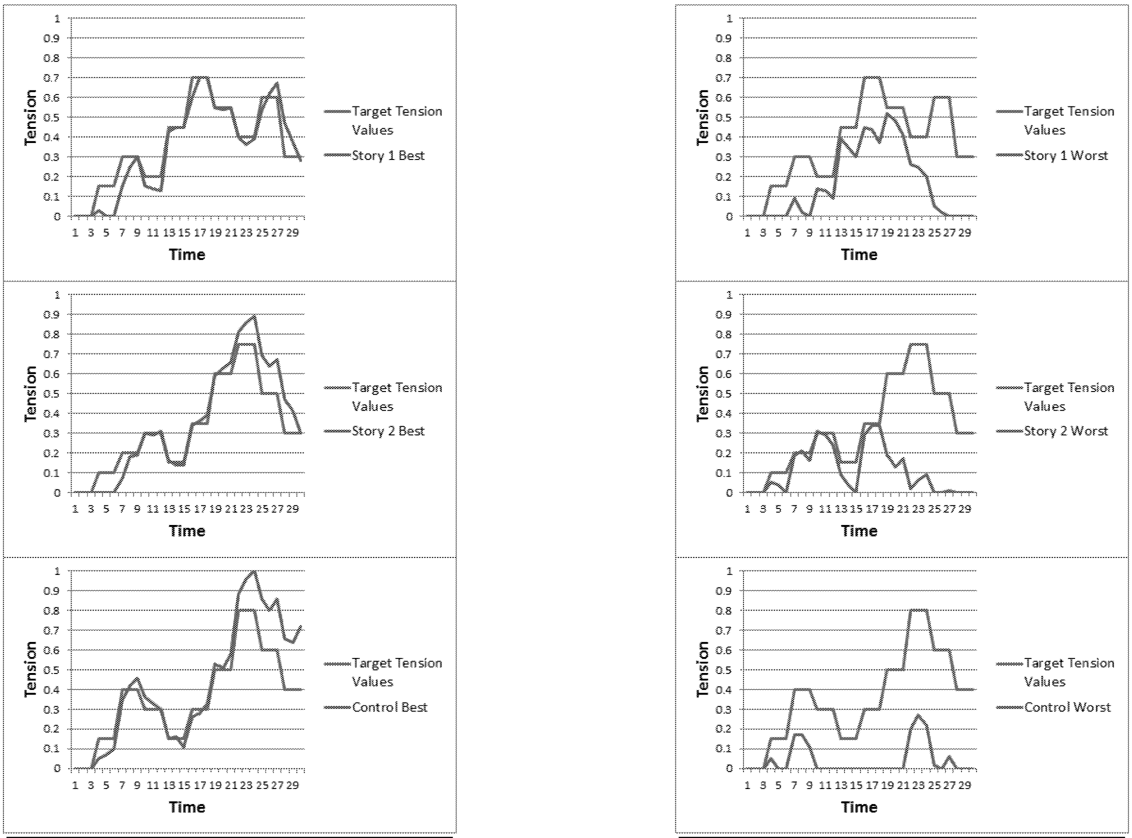
Results

Table 4 illustrates a summary of the questionnaire's results. In general, stories generated using the first tension graph received the highest ratings, and the majority of users preferred these narratives. Users awarded the lowest scores to the stories generated using the second tension graph, while the control story fared almost as well as the first tension graph's narratives. However, regarding the amount of control the users felt they had over the stories' events, both of the procedurally generated stories received far higher ratings than the control story. This suggests that it was clear when the users had a greater amount of influence over events. Despite this, all three narratives received low scores, implying that the users still felt that they did not have as much control as they would have liked. Overall, the results demonstrated the viability of procedural stories over the control, with the first tension graph generating narratives that users found the most satisfying. However, this does not mean that this is the optimal graph for generating stories, only that it is the best of the three tested. Furthermore, the 'optimal' graph would likely depend on the desired stories and available events.

Table 4: Questionnaire Results

Question	Story 1	Story 2	Control
Coherency	7.5	6.5	7
Believability	6.5	5.625	6.125
Consistency	6.5	6.25	6.375
Appeal	7.25	6	7
Level of Control	4.875	4.625	3.875
Duration		5.5	
Preference	50%	25%	25%

Figure 5 displays the best and worst matches for each story compared to their target values, along with the value of their respective RMSDs. These graphs show the system was able to match the values for the procedurally generated stories better than the control. This was because these narratives could adjust the tension by choosing appropriate events, instead of only using character actions for corrections. Story one also performed better in both instances than either of the others, while the control story was the worst in this regard.



	Story 1	Story 2	Control
RMSD	0.070875478	0.084695533	0.131237698

	Story 1	Story 2	Control
RMSD	0.25508822	0.321055551	0.367319116

Figure 5: Best (left) and Worst (right) Tension Metric Results

CONCLUSIONS

The implementation of the proposed system can generate a wide range of stories from the events it is given, providing a range of narratives comparable to that of existing branching systems. It is also possible to alter these narratives by changing which events the generator uses, meaning this one implementation can create many different kinds of stories, each varying as much as the provided events allow.

The analysis of the conducted research underlines the possibility for video-games to generate narratives that users find interesting and engaging. Further research can include the identification of the optimal parameters for generating these stories. Alternatively, further research could consider extending the effects of actions, such as, as one user suggested, indirectly affecting relationships (i.e. befriending someone's enemy makes you their enemy as well).

This research would be improved with a larger set of test data for analysis, as the limited range used here provides an equally restricted insight into the system's capabilities. This can be achieved by making the system available online, with users being able to test it and provide feedback at their leisure. Additionally, the values used in the models for the actions' effects are highly generic and would greatly benefit from a process of refinement, as the quality of the generated stories would improve by a comparable amount.

REFERENCES

Bates, J. (1992) *The nature of characters in interactive worlds and the Oz project*. School of Computer Science, Carnegie Mellon University Pittsburgh, PA.

Carnegie Mellon University (2001) *OZ Project Overview*. [online] Carnegie Mellon University. Available from http://www.cs.cmu.edu/afs/cs.cmu.edu/project/oz/web/overview2.html [Accessed 10 April 2014].

Goldberg, L. R. (1981) Language and individual differences: The search for universals in personality lexicons. In: Ladd Wheeler (ed.) *Review of Personality and Social Psychology: Volume 2*. California, USA: SAGE, 141-165.

Hyndman, R. J. and Koehler, A. B. (2006) Another Look at Measures of Forecast Accuracy. *International Journal of Forecasting*, 22(4) 679-688.

Mateas, M. (1999), An Oz-centric review of interactive drama and believable agents.

Mateas, M. & Stern, A. (2003), Façade: An experiment in building a fully-realized interactive drama. In: *Game Developers Conference, Game Design track*, 82.

Mateas, M. and Stern, A. (2005) *Façade*. [video game] Oregon, USA: Procedural Arts.

Stangor, C. (2011) *Introduction to Psychology*. Flat World Knowledge, L.L.C.

Togelius, J., Yannakakis, G. N., Stanley, K. O. and Browne, C. (2011) Search-based procedural content generation: A taxonomy and survey. *Computational Intelligence and AI in Games, IEEE Transactions on,* 3(3), 172-186.

Wardrip-Fruin, N. (2009) *Expressive Processing: Digital fictions, computer games, and software studies*, The MIT Press.

AUTHOR BIOGRAPHIES

KIERAN WAGG is a student from the University of Lincoln currently working towards a degree in (MComp) Games Computing. This is his first work to be published, and this conference will be the first he has attended. His research interests regarding games include procedural content generation, dynamic narratives and character representations.

DR. GRZEGORZ CIELNIAK is a Senior Lecturer at the School of Computer Science, University of Lincoln, UK. He obtained his Ph.D. in Computer Science from Orebro University, Sweden in 2007 and MSc in Robotics from Wroclaw University of Technology, Poland in 2000. His Ph.D. thesis addressed a problem of real-time people tracking for mobile robots. Dr. Cielniak was awarded a fellowship through the European Union Marie Curie Actions programme (FP6) during which he visited the Autonomous Intelligent Systems Laboratory in Freiburg, Germany. He co-authored more than 30 peer-reviewed publications (h-index: 11). His research interests in games technology include procedural content generation, physics simulation and AI.

A Digital Approach to Storytelling with MOGRE

Almir D.V. Santiago,
Paulo N.M. Sampaio

Graduate Program in Computing and Systems
Salvador University (UNIFACS)
Salvador, BA, Brazil

email: david.valente.santiago@gmail.com, pnms.funchal@gmail.com

Luis R.S. Fernandes

Exacts Sciences and Engineering Competence Centre
University of Madeira (UMa)
Funchal Madeira, Portugal

email: cs_painkiller@hotmail.com

Valéria Farinazzo Martins

College of Computing and Informatics
Presbitarian University Mackenzie
São Paulo, SP, Brazil

email: valeria.farinazzo@mackenzie.br

Keywords — 3D; OGRE; MOGRE; Virtual Reality; Authoring Tool; StoryTelling; MOGRE-ST

Abstract — Storytelling is a way to convey events using words, images and sounds, often by improvisation. Stories or narratives have been shared in all cultures as a mean to entertain, educate, for cultural preservation and as a way to transmit moral values. This paper presents a solution for interactive storytelling, targeted to children and juvenile public, called Mogre-Storytelling. This tool aims to be intuitive, providing different functionalities for the creation and customization of 3D scenarios, allowing the addition of 3D models from the Internet and even to enable the creation of virtual stories using multimedia elements such as speech bubbles, sounds and images. Mogre-Storytelling is a straightforward solution for young users who want to create stories with 3D scenarios and is also an alternative to Mogre/Ogre application developers, since this tool enables the rapid creation of customized scenarios and 3D graphics with a higher quality, and further exporting the created scenarios to other compatible tool. A case study was also carried out testing the utilization of Mogre-storytelling with 14 children and the results of the analysis are discussed in this paper.

I. INTRODUCTION

Stories or narratives have been shared in all cultures as a mean to entertain, educate, for cultural preservation and as a way to transmit moral values. In particular, storytelling is an effective and important educational tool for children. Traditional book storytelling enables multi-sensory experiences including speech (narration), vision (seeing the book) and touch (turning pages and pointing) (Zhou et al, 2004). 3D storytelling provides a richer multi-sensory experience for children given the possibility of interacting with the story objects and characters, navigating within the virtual scenarios with an enhanced feeling of presence.

Given the advantages of storytelling for educational purposes, a number of contributions in the literature demonstrate the development for children storytelling in order to promote free expression, creativity and fantasy play by engaging children as story authors (Gazotto et al 2010). These works outline the benefits associated with several paradigms of storytelling, such as collaborative authoring in large group or classroom settings (Cooper & Brna 2000, Di Blas et al 2009), Multimodal 3D storytelling environments for children (Wang et al 2007), Alice (Conway et al 2000), Avatar Storytellers (Avatar Storytellers 2014) and Zoo Burst (Zoo Burst 2014). In particular, the rendering engine OGRE (OGRE 2001) is one potential solution to provide the development of 3D interactive applications. Due its flexible features and library available, a variety of tools have been proposed to support the authoring and rendering of 3D scenarios, such as Ogitor (Ogitor, 2013), OGRE Editor Multi Scene Manager Project Environment (OGRE Editor, 2013), OGRE – MOGRE Editor (MOGRE Editor, 2013), OGRE-CL (Sampaio et al, 2013), etc. Although research in the direction of creating active tools for supporting interactive 3D storytelling seems to go in promising directions, real world contexts have as yet little benefited from these advances and new tools (Gazotto et al 2010).

This paper presents the main aspects about the development of a solution for interactive storytelling, targeted to children and juvenile public, called Mogre-Storytelling. This tool was conceived to be simple and easy to use, providing different functionalities for the creation and customization of 3D scenarios, allowing the addition of 3D models from the Internet and enabling the creation of virtual story using multimedia elements such as speech bubbles, sounds and images. Mogre-Storytelling was developed based on the graphical rendering engine MOgre which is a c# wrapper for OGRE (Object-Oriented Graphics Rendering Engine) (OGRE 2001).

OGRE/MOgre is a scenario oriented 3D engine, which makes available a library of classes (APIs) for the description of virtual worlds and objects in a high abstraction level and graphical quality.

The development of Mogre-Storytelling was carried out in collaboration with the educational services from the Madeira Whale´s Museum (Portugal). Mogre-Storytelling was proposed as a supplementary tool in order to allow young students that regularly visit this museum to demonstrate their knowledge acquired after the visit. Therefore, Mogre-Storytelling is one relevant tool to help unveiling environmental awareness among these young visitors.

This paper is organized as follows: Section II introduces some Mogre-Storytelling´s implementation issues; Section III features some functionalities of this tool through the presentation of some snapshots of the application; Section IV discusses a case study in order to assess the utilization of Mogre-Storytelling, and; Finally some conclusions and future perspectives of this work are presented in Section V.

II. MOGRE-ST: DESIGN AND IMPLEMENTATION ISSUES

In this section the main aspects about the development of the Mogre-Storytelling (Mogre-ST) are discussed. For this purpose, some of the main requirements for the proposed system are initially identified. After that, the modular architecture for the development of the system is also introduced. Finally some details about the development tools are presented.

A. Requirements

The requirements reflect the needs that a client wishes to be fulfilled by the application being developed (Oberg et al, 2010; Malan & Bredemeyer, 2009). These requirements serve as guidelines, and most of the times, they may represent objectives to be achieved in the project development. In this case, Mogre-Storytelling (Mogre-ST) will allow users to create their own 3D stories importing 3D objects from external libraries or pre-defined 3D scenarios from other compatible tools (Fernandes 2010, Sampaio et al 2013).

The following functional requirements were defined for Mogre-Storytelling:

- Opening and saving created storytelling projects.
- Creation and edition of 3D scenarios (modification of color and texture of the ground terrain, sky, density of fog, color of shadow and sun).
- Addition and removing objects (2D/3D and multimedia objects - audio, video, text and image) to/from the scenario.
- Manipulation of 2D/3D objects within the scenario (Positioning, rotation, size, cloning and elimination).
- Animation of the 2D/3D objects within the scenario (Positioning, rotation, resizing, showing/hiding objects, previewing animation and resizing animation).
- Importing pre-created scenarios from OGRE-CL (Fernandes 2010, Sampaio et al 2013) using the OGRE-ML format.
- Addition of new 3D objects to the Mogre-Storytelling objects library.

Once all the requirements and main functionalities of the system were identified, the architecture for the development of the system can be proposed.

B. Architecture

The architecture of a system aims at detailing all the software components, the externally visible properties, and the relations among these components. This architecture is a solid basis to the verification that all the initial requirements for the system were correctly taken into account (Bredemeyer & Malan 2004). Therefore, a modular architecture was proposed for the development of the system, as depicted in Figure 1.

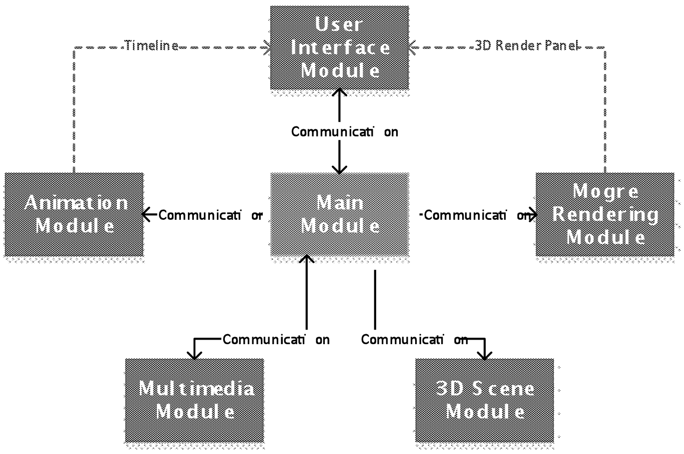


Fig. 1. Architecture of Mogre-Storytelling

Mogre-Storytelling is composed basically of six main modules (depicted in Fig. 1):

- *Main Module*: responsible for providing generic functionalities (Open new project, open existing project, save a project, etc.) to the main interface and to coordinate the execution of all other modules.
- *User Interface Module*: responsible for the management of the main application window and the respective user interaction (control buttons, events handling, 3D objects, multimedia objects, etc.).

- *Rendering Module*: composed of the SDK Mogre libraries (OGRE 2001), it contains the functions related to the Mogre initializing process for starting rendering the virtual scenario.

- *3D Scenario Module*: responsible for the management and customization of the 3D scenario (e.g. modification of textures for the ground and sky), opening and saving XML files for handling storytelling projects and handling 3D objects within the virtual scenario.

- *Animation Module:* responsible for the management and rendering of animation (timeline), such as adding/removing keyframes (which define the initial and final points within the temporal transition of an object during animation).

- *Multimedia Module:* allows the management and presentation of the multimedia components by the utilization of simple interface for the addition of text, image and sound.

C. Implementation Aspects

Mogre-Storytelling was implemented using C# which is adopted by the MOGRE library, and the XML language which was applied to describe all the descriptive languages for storytelling scenarios, animation, multimedia, etc. C# has been widely applied for the development of windows and web applications, XML-based services, etc.; and a number of development tools is also available.

III. MOGRE-ST FUNCTIONALITIES

This section introduces the main functionalities of the Mogre-Storytelling (Mogre-ST) application through the presentation of the main user interfaces (interactive windows and graphical components) of this tool. The Mogre-Storytelling interface is composed of different panels and menus each one providing distinct functionalities. Among the components of the Mogre-ST interface are: Main Menu, Objects Gallery, Scenario Configuration Panel, Mogre Rendering Panel, Mogre´s Main Menu Panel, Support Menu, Multimedia Panel and Timeline.

The Mogre´s *Main Menu* Panel allows the manipulation of all objects within a scenario, such as positioning, scaling, rotating, rolling, restarting (initial status), replicating and eliminating the objects. Fig. 2 illustrates Mogre´s Panel *Main Menu*.

The *Support Menu* provides options for the scenario configuration, objects management, multimedia, timeline, animation control (Play, Stop, etc.) and Full screen mode. In order to provide the user access to the scenario configuration panel, the "Scenario Configuration" button on the Support Menu has to be selected. After this selection a configuration panel is presented on the right side of the configuration window with four different tabs, as depicted in Fig. 3.

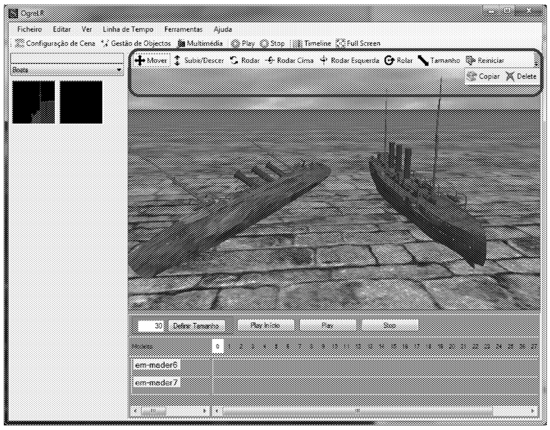


Fig. 2. Mogre´s Main Menu Panel

The *ground terrain* and *environment configuration* provides a list of material and textures available, presented to the user as snapshots. This list of materials and textures is previously registered in an XML file called *MaterialML*.

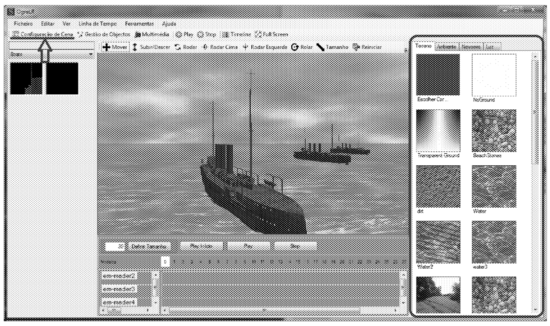


Fig. 3. Ground Terrain Customization

As illustrated in Fig. 3, the first tab allows the *configuration of the ground terrain*. This figure depicts the material assigned to the ground terrain as "sea". In order to modify the ground terrain, the user has to select a snapshot of his preference among all those presented on the panel. The first snapshot allows the user to select a plain color for his ground terrain instead of a texture.

The second tab allows the *configuration of the environment*, as depicted in Fig. 4. The scenario environment is related to the background image of the sky rendered on the virtual scenario. In this image, the sunset texture has been already selected. Also, in order to modify the background texture, the user has to select a snapshot from the list presented on the panel. The first snapshot also allows the user to select a plain color for his environment.

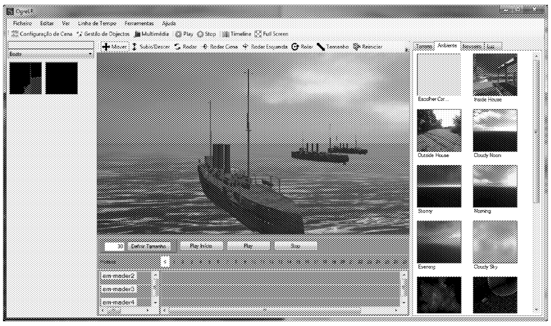


Fig. 4. Environment Customization

Fig. 5 illustrates the third tab, which is applied for *fog configuration*. In this image, we can also verify that a gray shade of fog has been applied. In order to increase or decrease the intensity of fog the user can apply the slider called “amount of fog”. The predefined color for the fog is white. In order to define another color the user can click on the button “Fog Color” and then choose another color. If the user wants to apply the same color to the fog and sky in order to improve realism in the scenario, he can also click on the button “Apply the same color to the sky”.

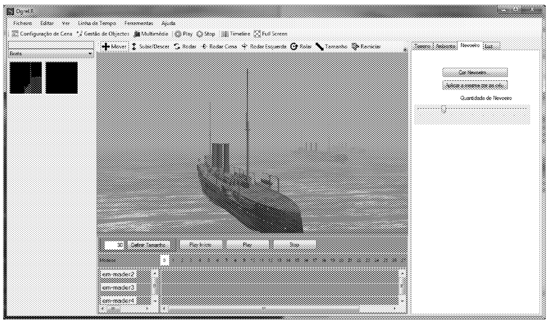


Fig. 5. Fog Customization

As an example of the combined utilization of fog and sky colors in order to improve the realism of the scenario, consider if the color of the fog is black, and if this color is also applied to the sky, it is possible to obtain a dark night scenario effect, as depicted in Fig. 6. Also, if the color of the fog and sky is blue marine, it is possible to obtain a bottom of the ocean effect.

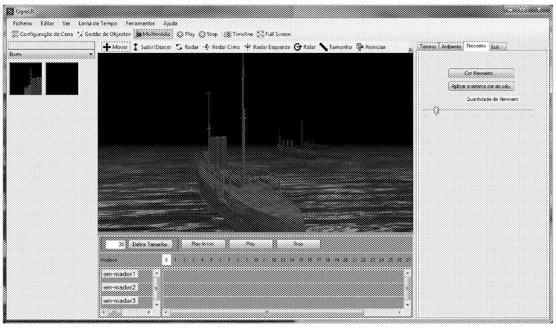


Fig. 6. Customization of fog with a black shade

Fig. 7 illustrates the fourth tab, which allows the *configuration of brightness*. This configuration determines the light effects allowing users to aesthetically customize colors. In this image the color of the sun was set to a dark green and the color of the shadow to dark blue providing a better aesthetic to the scenario’s color scheme. Note that the color of the sky remained orange as its original texture.

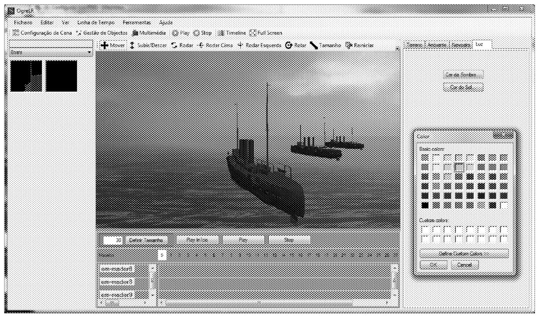


Fig. 7. Customization of Brightness

The *Multimedia Panel* allows the addition of audio, video, image and text within the scenario and their respective animation using the timeline. Fig. 8 illustrates a multimedia panel on the right side of the application window, which is presented when the user clicks on the “Multimedia” button. This panel allows the user to insert text, audio, image and video to his story.

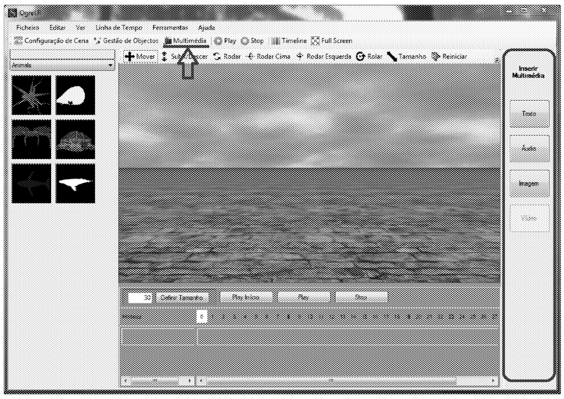
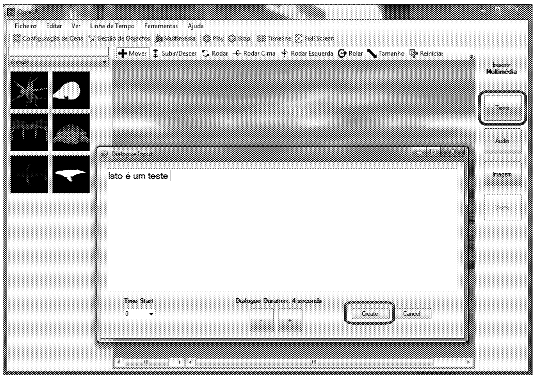
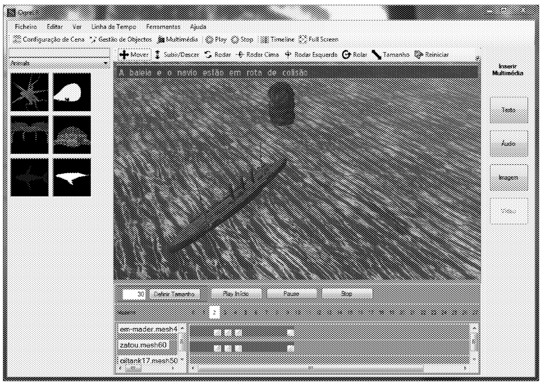


Fig. 8. Multimedia Panel

In order to insert a text to the story, the user has to select the “text” button, and a dialogue window will be presented so that he can write the desired text. After that, the user can also determine the instant the text will be presented and its respective presentation duration, as depicted in Fig. 9(a). The text can be visualized when the user starts the animation associated with the scenario. It is important to notice that the size of the text window will be automatically adjusted according to the size of the text. An illustration of a text presentation is depicted in Fig. 9(b).



(a)



(b)

Fig. 9. Add a text/dialogue into the story

In order to insert an audio sequence to the story the user has to click on the "audio" button on the right side of the application window. After that, a dialogue window will be presented so that the user can choose the audio file of his/her preference. The user can also determine the instant on the timeline the audio will start to be presented and its respective presentation duration, as depicted in Fig. 10. Similarly, the audio will be presented when the animation is triggered.

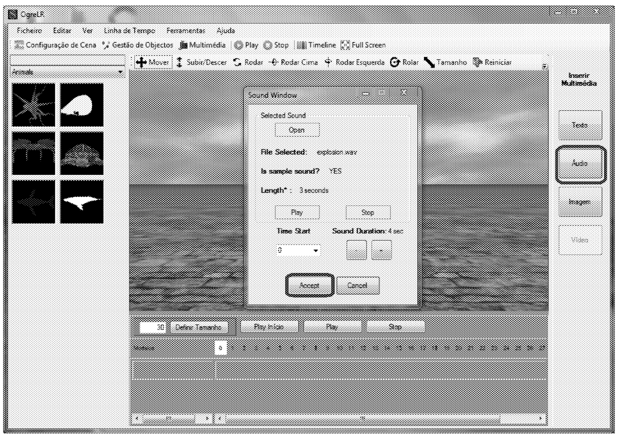
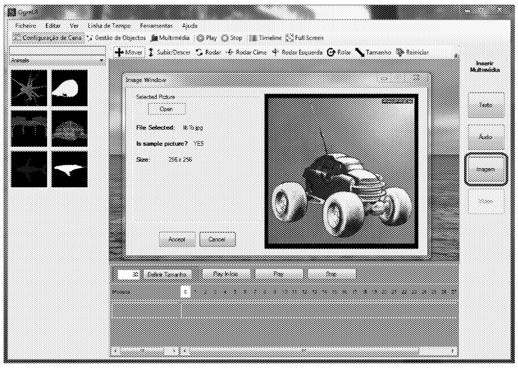
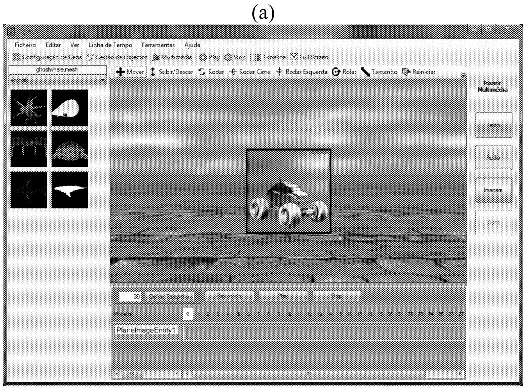


Fig. 10. Insert an audio sequence into the story

The user can also add a 2D image to the story. For this purpose, he must click on the "image" button on the right side of the application window. Similarly as discussed to text and audio, a dialogue window with snapshots of the image library will be presented to the user, and he can click on the desired image, as depicted in Fig. 11(a). For instance, consider the presentation of an image as illustrated in Fig. 11(b).



(a)



(b)

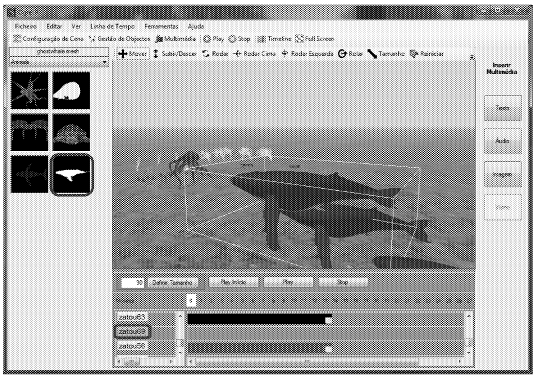
Fig. 11. Insert an image on the virtual scenario

The Timeline allows the temporal manipulation of the animation within the scenario. All the 3D and multimedia objects have a respective timeline. Whenever an object is added to the virtual scenario, its timeline is created with a related identifier, as depicted in Fig. 12(a).

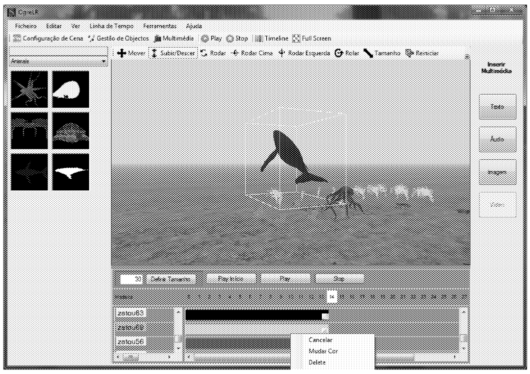
Each timeline is composed of some interactive buttons, for instance, to determine the duration of the animation, start the visualization of the animation from the current instant, start over the presentation and stop the presentation. At any time the user can click at any instant of the timeline to make the animation "skip" in time. Also, the timeline of any animated object within the story can be selected at any time allowing the user to verify the duration of the animation, in this moment the visualization camera spots the animated object presenting it as selected (Fig. 12a).

For instance, Fig. 12(b) illustrates the timeline for the objects called zatou63 and zatou56 whose both animation start at 0 seconds and finish at 13 seconds. Nevertheless, the object called zatou69 has no animation associated with it. The animation for each object within the application is determined by the transition between two different states (position, scale, etc.) for this object in two different time instants. The animation is then implemented by the application by the linear transition between these instants during the presentation of the story.

Fig. 12(b) also illustrates the animation of an object. In this figure, the animation evolves between the instants 0 and 14 seconds. For the configuration of this animation, the time instant 14 seconds was first selected, and the 3D object (whale) was rotated as to provide a rising effect. When the presentation is triggered, the whale spins upwards and slightly rises between the instants 0 and 14 seconds, shifting linearly between these instants.



(a)



(b)

Fig. 12. Timeline and animation of a recently added 3D object

After the implementation of the Mogre-Storytelling, it was important to assess its utilization with the development of some didactic activities with the target users. In the next section a case study is discussed.

IV. CASE STUDY

As introduced previously, the proposal of the Mogre-Storytelling tool aimed to support the didactic activities carried out at Madeira Whale Museum by the students visiting the museum. This section discusses the validation of the implemented tool with the proposal of some didactic activities carried out with these students, and the assessment of the results is also proposed.

A. Characterization of the Testing Group

Since the application was designed to target users in different ages, the development and validation of storytelling within educational context has an aggregated level of complexity requiring the implementation of several heterogeneous groups, coordination of schedule, and the previous definition of suitable curricula. In order to make this validation process as pragmatic as possible, the proposed activities were carried out for only one Madeira Whale´s Museum visiting class of 14 students ageing between 8 and 12 years old during one week. During this session the students applied Mogre-storytelling for 2 hours.

B. Evaluation Procedure - Presentation and Analysis of Data

In order to assess the functionalities and educational potential of the developed application, the students created a text with a thematic story so that they could build up afterwards a 3D story using Mogre-Storytelling. The creation of this text aimed at developing their creative and collaborative writing skills. Besides, the background story aimed at providing an environmental awareness concerning the marine biodiversity, which is one of the main goals of the Madeira Whale´s Museum visiting classes.

After having applied Mogre-Storytelling to create their piece of stories, the students were invited to fill out a questionnaire, so that they could register their main difficulties and overcoming strategies. The questionnaire was proposed aiming to assess the usability of the tool under six aspects: 1) Adaptation to the tool, 2) Configuration of the scenario environment, 3) Crowding the bottom of the ocean, 4) Animation, 5) Multimedia and 6) Finalization.

The *Adaptation to the tool* activity aimed at verifying how hard (or easy) it was for the students to get acquainted with the tool. The analysis of this task helped us to verify that most of the students only learned to use to tool after the teacher´s explanation (13 against one). That happened since most of the students waited the teacher to explain the utilization basics before starting to use the application. Nevertheless, there was only one 8 years old student that preferred to learn how to use the application by herself before the explanations. In order to asses this task the following metrics were analyzed: Explore the virtual environment; Navigate back and forth (Travelling within the 3D environment) and; Navigate upwards and downwards. By these results, it is possible to conclude that the students, after explanations or by self-learning, were able to navigate and understand the virtual scenario successfully.

In the *configuration of the scenario environment* activity, the students were asked to configure the virtual environment modifying the type of ground terrain, sky, amount of fog and color of lights and shadows. By the analysis of the results of this activity, it is possible to conclude that most of the students had no difficulties to modify the required parameters. Despite that, the students still needed some help with the first parameter (ground terrain) in order to find out how to modify it. Nevertheless, after having understood the modification mechanism, they were able to carry on with the following parameters.

The *crowding the bottom of the ocean* activity aimed at verifying how the users were able to apply 3D objects into the virtual scenario. For this purpose, the students were required to crowd the ocean with fishes, whales, shipwrecks, alga, among others. Later on it was also required the students to customize these objects by modifying their position, size, rotation angle and also cloning them. By the analysis of these tasks, it was possible to conclude that the students were able to easily add and manipulate the 3D objects inside the virtual environment. Similarly, they still needed some help with the first activities such as moving and cloning. Once they have learnt how to manipulate the objects they applied the methods to modify the size and rotation angle.

In the *animation* activity the students were required to produce an animation with the 3D objects within the established timeline. During the assessment of this task the following metrics were observed: (1) Finalization of the proposed animation tasks – in order to obtain the number of students that concluded the tasks with or without help, and; (2) Opinion of students about the usability of timeline – in order to understand how difficult it was for the students to apply timeline. By the answers provided by the users, we could verify that the students had problems to understand and apply animation. As expected, the students were able to perform the required tasks, although they had difficulties initially. Similarly to the previous tasks, once the teacher carefully explained the utilization of timeline the students were able to carry out remaining activity.

In the *multimedia* activity the goal was to assess the students´ ability to manipulate text, image, animation and sound. Similarly, it was possible to verify that the after the initial doubts and once the students understand the manipulation mechanisms, they were able to add and manipulate multimedia objects with no relevant difficulties. Among all, the manipulation of text represented an important initial drawback. This initial problem occurred either since students did not know which mechanism should be applied to add text or because they were not sure about which text they should write on. Once they had initial instructions, these problems were easily overcome.

The *finalization* activity aimed at providing a general assessment of Mogre-Storytelling by the students. Despite the difficulties experienced by some of the students, it was important to evaluate how this tool can be attractive and educational to the users. Therefore, the results indicated that most of the students considered the tool very entertaining and easy to use.

Although the proposed application received a very positive feedback from the students, we could verify that in some cases the difficulties experienced by the students were subjective, however, in other cases they were generalized. We could verify that the younger students between 8 and 9 years old performed the tasks in general with help either from the tutors or from the peers. It was clear their ability to modify the parameters and to add images to the virtual scenario (for instance). These students were interested to explore and experience the application selecting the available icons. Therefore, their interaction and modification were in general morphologic to the scenario, such as more or less objects, large versus smaller objects, color and ground terrain modifications. In the opposite, students between 10 and 12 years old provided a more functional interaction being more concerned with the animation of the objects.

V. CONCLUSIONS

This paper presented the main aspects about the development of a solution to the creation of 3D scenarios for storytelling. Mogre-Storytelling is an alternative for users who want to create stories with 3D scenarios and is also an option to application developers using Mogre/Ogre. This tool enables the rapid creation of customized scenarios and 3D graphics with a higher quality, while still allowing the integration of multimedia presentations.

Mogre-Storytelling was developed targeting users between 8 and 12 years old. Although this tool has been developed for a specific (young) group of users, older users also demonstrated interest to it due its dynamic features, interaction capabilities and high-quality graphics. Therefore, it was important to assess the utilization of this tool concerning usability, entertainment level, etc.

As for the validation process, most of the users needed initial instructions before the first utilization, in particular, concerning the localization of the available functionalities, mouse control and keyboard for environment navigation and visualization. After getting acquainted with the application environment, the users were able to navigate and create their own stories easily. In general, the students considered the application as entertaining and some of them even decided to keep exploring it after having finished the proposed activities. The results of the validation were encouraging concerning the education goals proposed. In particular, the feedback obtained from the users was helpful in order to correct and improve some functionalities of the application.

As for future works, more complex stories should be created in order to fully explore the functionalities of Mogre-Storytelling, and the interoperability among the existing OGRE/Mogre tools should be provided in order to optimize the creation of 3D Games and Virtual Interactive Applications.

ACKNOWLEDGEMENTS

We would like to give our special thanks to Prof. Silvia Carreira and to the Madeira Whale´s Museum for their support in the development and validation of Mogre-Storytelling.

REFERENCES

Conway, M., Audia, S., & Burnette, T. (2000). Alice: lessons learned from building a 3D system for novices. CHI '00 Proceedings of the SIGCHI conference on Human factors in computing systems.

Cooper, B.; Brna, P. (2000). *Classroom Conundrums: The Use of a Participant Design Methodology*. Educational Technology & Society 3(4), 121–153.

Di Blas, N., Garzotto, F., Paolini, P., and Sabiescu, A. 2009. *Digital Storytelling as a Whole-Class Learning Activity: Lessons from a Three-Years Project*. In Proceedings of the 2nd Joint international Conference on interactive Digital Storytelling: interactive Storytelling (Guimarães, Portugal, December 09 - 11, 2009). I. A. Iurgel, N. Zagalo, and P. Petta, Eds. Lecture Notes In Computer Science, vol. 5915. Springer-Verlag, Berlin, Heidelberg, 14-25.

Fernandes, D. (2010). Bases para a Geração Automática de Cenários Virtuais OGRE (In Portuguese). M.Sc. Dissertation in Informatics Engineering, University of Madeira, Funchal, Madeira, Portugal.

Garzotto, F.; Paolini, P.; Sabiescu, A. (2010). *Interactive Storytelling for Children*. In proceedings of IDC 2010, June 9–14, 2010, Barcelona, Spain.

Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., Kafai, Y. (2009). Scratch: Programming for everyone. Communications of the ACM, Communications of the ACM 52(11):60-67.

Sampaio, P.N.M.; Teixeira, D.J.O.; Fernandes, D.M. Creating 3D Scenarios with OGRE Creativity Labs. Published in Brazilian Computing Society ́s Journal on 3D Interactive Systems, Vol.4, number 1, ISSN: 2236-3297, 2013.

Wang, D.; Zhang, J.; Li, J.; Dai, G.; Lin, Q. (2007). *A Multimodal 3D Storytelling System for Chinese Children*. Technologies for E-Learning and Digital Entertainment. Lecture Notes in Computer Science Volume 4469, 2007, pp 511-526.

WEB REFERENCES

Avatar Storytellers - *A K-12 Immersive Education Curriculum & Virtual World for K-12*. Last visited in February 2014. http://www.avatarstorytellers.com

Bredemeyer, D.; Malan, R. (2004). "What It Takes to Be a Great Enterprise Architect." Cutter Consortium Enterprise Architecture Executive Report, Vol. 7, No. 8, 2004. from http://www.cutter.com/offers/greatarchitect.html.

OGRE – Open Source 3D Graphics Engine. (2001). Last visited in February 2014. http://www.ogre3d.org/

Ogitor SceneBuilder. Last visited in November 2013. http://www.ogitor.org/HomePage

OGRE Editor Multi Scene Manager Project Environment. Last visited in November 2013. http://www.youtube.com/watch?v=TU_Tc4EBLHQ

Oberg, Roger. Probasco, Leslee. & Ericsson, Maria(2010). Applying Requirements Management with Use Cases. Rational Software – Technical Paper TP505. On November 8th 2010, from http://www.wthreex.com/rup/papers/pdf/apprmuc.pdf

OGRE – MOGRE Editor. Last visited in November 2013. http://www.youtube.com/watch?v=xXJDvKzY1Us&feature=related

Malan, R.; Bredemeyer, D. (2009). Functional Requirements and Use Cases. Bredemeyer Consulting - White Paper. 8/3/01 On November 8th 2010, from http://www.bredemeyer.com/use_cases.htm

Zoo Burst – Digital Storytelling. Last visited in February 2014. http://www.zooburst.com

GAME
HARDWARE
PROGRAMMING

ACCELERATING GPU WORKLOAD SIMULATION USING MICROSOFT WARP

Eric Nilsson
Faculty of Computing
Blekinge Institute of Technology
37 179, Karlskrona
Sweden
E-mail: erne09@student.bth.se

KEYWORDS
System simulation, Software rasterization, WARP.

ABSTRACT

A convenient approach toward more transparent debugging and profiling of GPU-accelerated applications is to simulate GPU-bound workloads on CPUs. This approach is also applicable in situations where target hardware is simply not available, as is often the case with server-side applications, or would require too many system resources to initialize. However, when simulating GPU-bound workloads on CPUs, one may experience severe performance losses due to computational overhead. Consequently, the subject of this study is performance variations between three device drivers of the DirectX framework; using DirectCompute and the high speed software rasterizer WARP (Windows Advanced Rasterization Platform). The performance of WARP is compared to that of traditional GPU hardware acceleration and the standard driver for software rasterization - the Reference Device Driver. Experimental results show a major performance boost when compared to that of software rasterization using the reference device driver, indicating that performance losses traditionally obstructing simulation of throughput-oriented workloads on CPUs may be sufficiently amended by technologies, such as WARP, to the degree that simulation may be considered viable for extended use above and beyond that of current in-development utilities.

INTRODUCTION

When developing GPU-bound applications, careless additions to program execution, such as branching, register use, or memory accesses, may induce substantial performance obstacles due to massively parallelized instruction sets and architectural differences between on-chip hardware (see Performance Considerations by Kirk and Hwu (Kirk and Hwu 2010; ch. 6) for an analysis on the volatility of GPU performance). The increased utilization of complex GPU-kernels has brought forth the need of more extensive debugging- and profiling-options involving access of data that may be difficult to retrieve from hardware. Therefore, it may be desirable for developers to access data being computed on the graphics card - a possibility often limited in terms of GPGPU (General-Purpose computing on Graphics Processing Units) technologies, possibly due to architectural differences between chip manufacturers. A preferred solution to this problem is to simulate GPU workloads on CPUs (Microsoft 2013a), such as described by Kerr et al. (Hwu 2011; p. 416-419) concerning the implementation of the GPU Ocelot compilation framework (Georgia Institute of Technology 2013), often in exchange for substantial performance implications. Other reasons to simulate GPU-kernels may concern pre-silicon development - that is, development for hardware not yet existent, when hardware is busy, or otherwise unavailable (Microsoft 2013c).

This paper comprises an investigation into the performance of simulated GPGPU-kernels, in relation to hardware acceleration, by the means of analyzing several software rasterizers. Accordingly, the material concerns inquiry into the DirectCompute-framework on the Windows-platform, analyzing the performance of a GPGPU-kernel on-chip, using the DirectX standard software rasterizer, and utilizing the DirectX 11 addition Microsoft WARP which promises high-speed software rasterization (Microsoft 2013c). The study concludes that performance losses inflicted by software rasterization may be sufficiently amended for simulation of GPU-bound workloads to be considered viable for extended use, as originally proposed by Microsoft (Microsoft 2013c). Thus, this paper proposes use of WARP-like technologies if graphics hardware is unavailable, not sufficient, or busy, and to act as an extension of the reference device driver as means to verify hardware- or driver-bound errors for complex GPU programs. As such, the study concerns the fields of simulation and GPU-technologies, with the purpose of facilitating debugging and profiling of GPU-kernels, whilst maintaining acceptable performance. The remainder of this document presents the method and process to acquire the data used, the technologies using which it has been acquired, the results in and of their own, the conclusions based off the results and finally; personal reflections from the author with propositions of further study.

CONTRIBUTION

Object of Study

Due to the nature of the study, it is important that the experiment process is measurable and deterministic. For the purpose of the experiment, the data computed corresponds to a square matrix multiplication, since such an operation is highly parallelizable, as demonstrated by Kirk and Hwu (Kirk and Hwu 2010; ch. 3). Additionally, verifying the result of such an operation is trivial, and potential losses in computational precision may be easily established. The operation is therefore considered suitable for the purpose of the study. Furthermore, the matrix dimension 200×200 is selected for the respective matrices as it is big enough to execute efficiently on the GPU, but yet reasonably sized to keep simulated measurements comparable to those of the hardware accelerated reference case. Throughout this material, the compiled data is referred to as $AB = C$, and thus make out the Object of this experiment.

Method of Study

In order to establish the Object of Study, $AB = C$, the experiment utilizes DirectCompute HLSL kernels using which the result matrix is calculated, in its entirety, using some device driver, or Subject (see section Subjects of Study). The experiment is devised of the following approximate steps in order to compile the Object of Study $AB = C$:

1. Randomize two matrices A and B of desired datatype (single floating point or integer).
2. Establish the product-matrix $AB = Ref$. The resulting matrix will be used as a reference matrix to verify the final result.
3. Start a synchronized high-precision timer.
4. Dispatch the DirectCompute program calculating the product matrix $AB = C$.
5. Stop the timer once the kernel has finished execution.
6. Establish possible deviation between resulting matrix C and previously established matrix Ref.

The above process makes out the Method of this study.

Subjects of Study

The kernels described in section Method of Study are comprised of HLSL, compiled using Microsoft FXC, and executed using the Direct3D compute shader pipeline stage. The programs are run using three types of acceleration technologies with varying strategies of simulation. These are comprised of the following:

Hardware-Acceleration (GPU)

The execution of a DirectCompute-kernel on a graphics card. By utilizing a modern video card, hardware acceleration allows DirectX to utilize the throughput-oriented design of modern GPUs; maximizing FLOPS (FLoating-point Operations Per Second) capacity. Being the common case; the methodology involves no simulation. Thus, measurements collected using this device driver acts as a reference for the simulated Subjects.

Software Rasterization (CPU)

The simulation of a DirectCompute-kernel on the CPU using the DirectX reference device driver. The reference rasterizer supports every Direct3D feature and is intended to be used for debugging- and verification purposes. Hence, the reference device driver is only available on systems where the DirectX SDK is installed. As such, although the reference device driver does feature some optimization for CPU execution, the driver is not intended for outside-development use (Microsoft 2013a). As the reference device driver is designed for the purpose of framework functionality verification, rather than speed, one may expect poor performance.

Windows Advanced Rasterization Platform (CPU)

The simulated execution of a DirectCompute kernel using a specially devised software rasterizer developed for high performance and full framework conformance by Microsoft in the latest version of DirectX. Being based on the reference device driver, WARP utilizes thread pooling to distribute tasks efficiently on the CPU, along with grouping execution in batches for optimum performance. Furthermore, WARP uses JIT (Just-In-Time) compilation to generate SSE2 and SSE4.1 SIMD (Single Instruction, Multiple Data) instructions, in addition to x86-compatible native instructions to significantly speed up simulation of GPU-bound workloads (Microsoft 2013c). Having outperformed some low-end integrated GPUs, according to Microsoft, and the device driver being included in Windows 7 and 8 run-times, Microsoft recommends using the driver for retail applications such as casual games (Microsoft 2013c). Although Microsoft has presented some performance comparisons to hardware acceleration, there is no performance benchmark indicative of WARP performance in relation to the reference device driver - the device commonly used for debugging and verification purposes. As such, expectations on the performance of WARP is unclear but the driver is expected to perform faster than the reference device driver, from which it is derived, due to its promised optimizations and high-speed focus.

These DirectX device drivers make out the Subjects of this study.

Kernels

In addition to the Subject drivers mentioned in section Subjects of Study, two kernels with varying level of memory access optimization are examined. The programs are implemented in accordance to the CUDA matrix multiplication kernels as described by Kirk and Hwu (Kirk and Hwu 2010; p. 67, p. 87). Furthermore, aforementioned kernels accommodate both integer- and floating point precision. Descriptions of these kernels are presented below and complemented with source code under section APPENDIX.

Matrix mult. w. Thread Blocks
A kernel producing $AB = C$ from two given matrices; writing back C for further analysis. The program is executed with one thread for each element in the resulting square matrix, and likewise each produce a lone element of the product matrix. Execution is performed in blocks of 16×16 threads, since this was the block dimension, out of samples 8, 16, and 32, that performed optimally whilst hardware accelerated on the system described under section Equipment. This program will be referred to as the Basic Kernel throughout the material.

Matrix mult. w. Thread Blocks and Shared Memory
Similar to the previously described kernel, but optimized to utilize shared memory in order to reduce time-consuming access of global memory, as presented by Kirk and Hwu (Kirk and Hwu 2010; p. 77-93). Stratton et al. (Stratton et al. 2008; p. 1-3) instructs that the CUDA GPGPU-model may be applied onto multicore CPUs, including locality-wise execution of logical thread blocks (that is; all threads in a block limited to a single core), with the utilization of local- and shared-memory approximately corresponding to the L1- and L2-caches of a CPU core. Hence, the kernel is presented as a scenario due to the preconditions of WARP, stating that a kernel optimized for GPU-execution is likewise optimized for execution with WARP (Microsoft 2013c). Thus, we investigate a more optimized kernel to see whether or not this behavior may be replicated in the experiment in terms of memory access optimization. The program is referred to as the Tiled Kernel throughout this material.

Tools

The experiment process is subdivided into three major components, all of which are compiled using Visual C++ 2012. These tools are presented in detail below.

matrixgen
Denotes a utility developed to generate matrices of different dimension and precision. In addition to randomly generating matrices, matrixgen also compiles the reference matrix Ref; used to validate the product matrix returned from the subsequent DirectCompute dispatch. The utility is written in C++ and utilizes C++ AMP to generate and multiply matrices A and B into product matrix Ref. In order to produce random values in a C++ AMP program, the solution includes the random number generator library C++ AMP RNG. As the utility utilizes Microsoft C++ AMP-technology, the utility requires Windows 7 or later in order to run.

experiment
Making out the primary component of the study, experiment uses DirectCompute technology to compile the product matrix C from matrices A and B (previously generated by matrixgen). The application outputs data surrounding the execution of the kernels, such as execution time in milliseconds, to an intermediate file. The utility is written in C++, with its respective DirectCompute kernels devised in HLSL. Since the tool is developed using the Windows 8 SDK, Windows 8.0 or later is required to run the tool. Furthermore, the utility requires a DirectX 11.0- or DirectX 11.1-compatible graphics card in order to produce the hardware accelerated reference case.

analytics
A utility devised to compose data surrounding possible precisional deviation between matrices C and Ref. The tool, written in C++, compiles the minimum- and maximum deviation encountered, as well as to calculate the standard deviation of any precisional discrepancy in produced product matrices. In turn, the program outputs this information to an intermediate file.

These applications are, in turn, run as subprocesses in a script specifying the various configurations and number of times to run each program. This script, written in Python, compiles the assorted results of respective program and outputs a range of files suitably formatted for interpretation by Gnuplot. The source code manufactured for the sake of this study is freely available via an online Git repository (Nilsson 2013), along with a guide on how to compile and run the solution in order to replicate the experiment. Furthermore, the complete results presented in this material are also available for download and may be acquired for further analysis.

Equipment

The results presented in the paper are gathered from experiments performed on a system with the following specifications:

CPU Intel Q9550 Quad Core 2.83GHz
GPU ATI Radeon HD 5800
OS Windows 8.0

This system setup was selected for use as Microsoft claims that the WARP device performs best on mod-

Table 1: Average execution time in milliseconds of a 200×200 integer matrix multiplication and corresponding percentage decrease- and increases.

	BASIC	TILED	
HARD	1.16	0.24	−79.3%
SOFT	11610.02	9866.40	−15.0%
WARP	15.31	18.97	+23.9%
	−99.9%	−99.8%	

Table 2: Average execution time in milliseconds of a 200×200 floating point matrix multiplication and corresponding percentage decrease- and increases.

	BASIC	TILED	
HARD	0.77	0.22	−71.4%
SOFT	10247.03	10909.88	+6.5%
WARP	14.08	17.44	+23.9%
	−99.9%	−99.8%	

ern quad-core CPUs (Microsoft 2013c).

Process of Study

For the purpose of this study, the Object of Study, being 200×200 matrices of integer or floating point precision, is randomized with numbers between zero and ten. The product matrix of these matrices is subsequently computed 100 times for each configuration. In this way, the Basic Kernel is executed with each Subject of Study - being Hardware Acceleration, the reference device driver, and WARP - respectively, as is the Tiled Kernel. For each execution, data surrounding the dispatch time of each kernel (regardless of program initialization) is garnered along with precision-wise deviation data. This process is repeated for integer- and floating point precision, in order to establish if there is any performance-wise discrepancy betwixt them.

Results

Considering the average percentage decreases in execution time of 99.8 to 99.9% (presented in table 1 and 2); results indicate significant improvements in the performance of both kernels when using WARP, compared to that of the the reference device driver. These improvements, accounting for three orders of magnitude on average, induce an execution time spanning tenths of milliseconds; a considerable advance from that of roughly ten seconds. The measurements indicate that the reference device driver is 4 to 5 orders of magnitude slower than the hardware accelerated reference case, depending on whether or not the kernel utilizes shared memory to accelerate memory accesses on the video card. Meanwhile, the WARP device driver performs but 1 to 2 orders of magnitude worse than hardware acceleration, also depending on hardware shared memory utilization.
In terms of memory latency, table 1 demonstrates performance gains with the hardware accelerated Subject (when utilizing shared memory amongst blocks), with varying results for the other Subjects; either increasing or decreasing execution time (see table 2). As such, applicable to both integer- and floating point precision, the performance of WARP is impaired by the kernel utilizing shared memory. In the Microsoft guide on WARP (Microsoft 2013c), the author claims that an application tuned to run efficiently on hardware will run efficiently whilst simulated using WARP as well - and vice versa. However, the collected data rather indicate a roughly 20% percentual increase in execution time, independent of precision, even though the same kernel accelerates the hardware accelerated Subject by roughly 70%. This demonstrates that that the capabilities of WARP do not extend to memory latency optimization by lower-latency shared memory. The floating point scenario of this effect is visualized in figure 1.
Analysis indicates no loss in precision between reference- and product matrices. Neither is there any recorded divergence in results compiled by respective Subjects. In hindsight, this behavior may be expected as WARP conforms to the precision requirements of the Direct3D 10- and 10.1-specification (Microsoft 2013c). See the Microsoft guide on Floating-point Rules (Microsoft 2013b) for more information surrounding floating point precision in the Direct3D-framework. Accordingly, albeit not being explicitly developed for the purposes of precision, WARP performs equally as with the reference device driver. Whether or not this behavior is replicated for double floating point precision is uncertain.

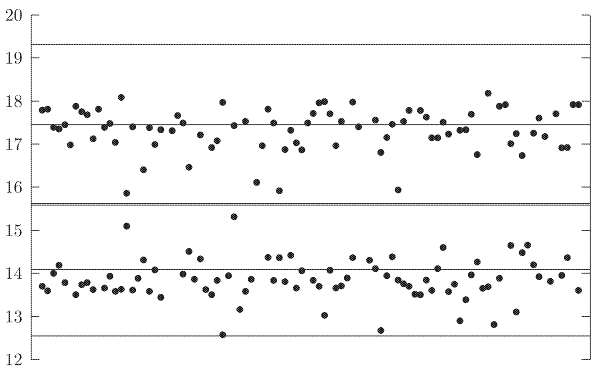


Figure 1: Execution time using WARP with floating point-precision for the Basic and Tiled kernels, indicated by their respective mean- and separated by standard deviation values. Samples outside of respective standard deviation are not presented in this visualization.

CONCLUSION

Based on the findings of the experiment described in this paper, using WARP to simulate the presented kernels has magnitudes greater performance than if one were to apply the reference device driver in the same manner. Hence, if one were to compare the execution of WARP and the reference device driver side-by-side, and assume the same area of application, WARP is superior in terms of execution time.

However, keeping in mind the major performance improvements offered by WARP, it is important to consider that the two may be appropriate for different purposes. The reference device driver is primarily proposed by Microsoft as an accurate debugging-/pre-silicon-development tool (e.g., for the purposes of driver verification). On the contrary, WARP is intended for use in a broader sense, such as to render graphics for casual games, in addition to error-profiling purposes (Microsoft 2013c). Accordingly, the WARP device is included in the Direct3D 11 runtime for outside-development use. This indicates potential for extended use above and beyond that of debugging- and profiling purposes, since the WARP device may be utilized where the reference device driver would otherwise be orders of magnitude too slow. Such an example would be the isolation of hardware or graphics driver errors in high-performance graphics applications - often encountered by video game developers.

In conclusion; this study proposes, pursuant to the established performance of WARP, that WARP-like technologies are feasible for use where the reference device driver is too slow, such as verification of advanced computer graphics, or when hardware is not available, such as the acceleration of graphics in virtual platforms.

Future Work

The performance improvements of WARP, in comparison to the reference device driver, calls for further inquiry into what the flaws of using WARP may be. Microsoft lists, amongst other limitations latency-oriented CPUs may have over throughput oriented GPU architectures, memory bandwidth as a potential bottleneck for WARP-type acceleration methods (Microsoft 2013c). In accordance to the suspicions raised by Microsoft, it may be beneficial to investigate the performance of WARP when such bottlenecks are stressed. Furthermore, the author suggests complementary elaboration into double floating point precision calculations, with the intent of examining whether or not the Subjects detailed in section Subjects of Study may have differentiating effects on computational precision.

Additionally, the use of more complex kernels in coagency with the WARP-driver should be examined in order to study the effects of more demanding simulation on CPUs.

REFERENCES

Hwu W.m.W., 2011. *GPU Computing Gems Jade Edition*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st ed. ISBN 0123859638, 9780123859631.

Kirk D.B. and Hwu W.m.W., 2010. *Programming Massively Parallel Processors: A Hands-on Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st ed. ISBN 0123814723, 9780123814722.

Stratton J.A.; Stone S.S.; and Hwu W.M.W., 2008. *Languages and Compilers for Parallel Computing*. Springer-Verlag, Berlin, Heidelberg, chap. MCUDA: An Efficient Implementation of CUDA Kernels for Multi-core CPUs. ISBN 978-3-540-89739-2, 16–30. doi:10.1007/978-3-540-89740-8_2.

WEB REFERENCES

Georgia Institute of Technology, 2013. *GPU Ocelot*. `http://b.gatech.edu/1a489L1`. Computer Architecture and Systems Lab, School of Electrical and Computer Engineering. Accessed: 2013-10-21.

Microsoft, 2013a. *DirectX Driver Types*. `http://bit.ly/1b63NiX`. Accessed: 2013-10-21.

Microsoft, 2013b. *Floating-Point Rules*. `http://bit.ly/1bOQnZu`. Accessed: 2013-11-16.

Microsoft, 2013c. *Windows Advanced Rasterization Platform (WARP) Guide*. `http://bit.ly/19hiwrZ`. Accessed: 2013-10-21.

Nilsson E., 2013. *DV2549 - Github*. `http://bit.ly/1c4wd20`. Accessed: 2013-10-21.

AUTHOR BIOGRAPHY

ERIC NILSSON conducted his master thesis in Computer Science on graphics acceleration in virtual platforms ('Paravirtualizing OpenGL ES in Simics') in collaboration with Intel Corporation to obtain his M.Sc.E. from Blekinge Institute of Technology, where he majors in Software Engineering. In addition to his interest in the sciences, he has a keen interest in horror games; an engagement he pursues to the best of his ability. Currently, Eric is involved with system simulation at Intel where he carries out his duties as a Software Engineer.

E-mail: erne09@student.bth.se

APPENDIX

Matrix mult. w. Blocks

```
#ifndef DV2549_FXS_MULTFLOATBASIC_FX
#define DV2549_FXS_MULTFLOATBASIC_FX

#include <CommonFloat.fx>

[ numthreads( BLOCK_SIZE, BLOCK_SIZE, 1 ) ]
void main(
    uint3 tIdx : SV_GroupThreadID,
    uint3 bIdx : SV_GroupID ) {
    const uint row = bIdx.y*BLOCK_SIZE+tIdx.y;
    const uint col = bIdx.x*BLOCK_SIZE+tIdx.x;
    if( row>=cRows || col>=cCols ) {
        return;
    }

    float sum = 0;
    for( uint i = 0; i<aRows; i++ ) {
        uint idxA = row*aRows+i;
        uint idxB = col+bRows*i;
        sum += mA[idxA]*mB[idxB];
    }
    mC[row*cRows+col] = sum;
}

#endif // DV2549_FXS_MULTFLOATBASIC_FX
```

Matrix mult. w. Blocks and Shared Memory

```
#ifndef DV2549_FXS_MULTFLOATTILE_H
#define DV2549_FXS_MULTFLOATTILE_H

#include <CommonFloat.fx>

groupshared float mAs[BLOCK_SIZE][BLOCK_SIZE];
groupshared float mBs[BLOCK_SIZE][BLOCK_SIZE];

[ numthreads( BLOCK_SIZE, BLOCK_SIZE, 1 ) ]
void main(
    uint3 tIdx : SV_GroupThreadID,
    uint3 bIdx : SV_GroupID ) {
    const uint row = bIdx.y*BLOCK_SIZE+tIdx.y;
    const uint col = bIdx.x*BLOCK_SIZE+tIdx.x;

    float sum = 0;
    const uint blocks = ceil(
        (float)aRows/(float)BLOCK_SIZE );
    for( uint i = 0; i<blocks; i++ ) {
        mAs[tIdx.y][tIdx.x] = mA[
            row*aRows+( i*BLOCK_SIZE+tIdx.x )];
        mBs[tIdx.y][tIdx.x] = mB[
            col+bRows*( i*BLOCK_SIZE+tIdx.y )];
        GroupMemoryBarrierWithGroupSync();

        for( uint j = 0; j<BLOCK_SIZE; j++ ) {
            sum +=
                mAs[tIdx.y][j]*mBs[j][tIdx.x];
        }
        GroupMemoryBarrierWithGroupSync();
    }
    if( row>=cRows || col>=cCols ) {
        return;
    }
    mC[row*cRows+col] = sum;
}

#endif // DV2549_FXS_MULTFLOATTILE_H
```

AUTHOR LISTING

AUTHOR LISTING