14TH INTERNATIONAL CONFERENCE

ON

INTELLIGENT GAMES AND SIMULATION

GAME-ON® 2013

EDITED BY

Anton Eliëns

and

Wim Lamotte

November 25-27, 2013

TheHotel

Brussels

BELGIUM

A Publication of EUROSIS-ETI

Cover Art: Divinity: Original Sin © 2013 Larian Inc. All rights reserved. Developed by Larian Studios. All company names, brand names, trademarks and logos are the property of their respective owners

14TH International Conference

on

Intelligent Games and Simulation

BRUSSELS, BELGIUM

NOVEMBER 25 - 27, 2013

Organised by

ETI

Sponsored by

EUROSIS

Co-Sponsored by

Binary Illusions

Ghent University

Larian Studios

University of Skövde

FLEGA

Incontrol Simulation Solutions

TheHotel

Brussels, Belgium

EXECUTIVE EDITOR

PHILIPPE GERIL
(BELGIUM)

EDITORS

General Conference Chair

Anton Eliëns
VU University Amsterdam/University of Twente
The Netherlands

General Program Chair

Wim Lamotte
University of Hasselt, Hasselt, Belgium

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

Facial, Avatar, NPC, 3D in Game Animation
Yoshihiro Okada, Kyushu University, Kasuga, Fukuoka, Japan
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
Daryl Charles, University of Ulster, Northern Ireland, United Kingdom
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
Sandy Louchart. Herriot-Watt University, Edinburgh, United Kingdom
Tshilidzi Marwala, University of Witwatersrand, Johannesburg, South-Africa
Sam Redfern, National University of Ireland, Galway, Ireland
Miguel Tsai, Ling Tung University, Taichung, Taiwan

INTERNATIONAL PROGRAMME COMMITTEE

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
Chris Darken, The MOVES Institute, Naval Postgraduate School, Monterey, USA
Andrzej Dzielinski, Warsaw University of Technology, Warsaw, Poland
Maja Pivec, FH JOANNEUM, University of Applied Sciences, Graz, Austria
Tina Wilson, The Open University, Milton Keynes, United Kingdom

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
Olli Leino, IT-University of Copenhagen, Copenhagen, Denmark
Sean Pickersgill, University of South Australia, Adelaide, Australia
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

INTERNATIONAL PROGRAMME COMMITTEE

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....

Mohamed Abbadi, Tilburg University, Tilburg, The Netherlands
Roberto Beauclair, Institute for Pure and Applied Maths., Rio de Janeiro, Brazil
Guiseppe Maggiore, Tilburg University, Tilburg, The Netherlands
Tony Manninen, Ludocraft Ltd., Oulu, Finland
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
Roger Smith, US Army, Orlando, USA

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®
2013

© 2013 EUROSIS-ETI

Responsibility for the accuracy of all statements in each peer-referenced paper rests solely with the author(s). Statements are not necessarily representative of nor endorsed by the European Multidisciplinary Society for Modelling and Simulation Technology. Permission is granted to photocopy portions of the publication for personal use and for the use of students providing credit is given to the conference and publication. Permission does not extend to other types of reproduction or to copying for incorporation into commercial advertising nor for any other profit-making purpose. Other publications are encouraged to include 300- to 500-word abstracts or excerpts from any paper contained in this book, provided credits are given to the author and the conference.

All author contact information provided in this Proceedings falls under the European Privacy Law and may not be used in any form, written or electronic, without the written permission of the author and the publisher.

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

GAMEON® is a registered trademark of the **European Technology Institute** under nr: 1061384-761314

EUROSIS-ETI Publication

ISBN: 978-9077381-80-9
EAN: 978-9077381-80-9

Preface

Dear Participants,

Introducing the 14th edition of this conference, I am happy to present to you the contents of the proceedings and to welcome all the participants of GAMEON'2013, November 25-27, 2013, to TheHotel, Brussels, Belgium.

Quoting Bernard Suits, we may characterize playing a game as a voluntary attempt to overcome unnecessary obstacles, and for all of you that have made the effort to be present, I am confident that, looking at the rich program of this edition, you will be amply rewarded for taking the trouble to be here.

I wish to express my gratitude to all contributers, those who have submitted papers, but also the organizers that have made the conference possible by their diligent work, and the sponsors supporting this event, with of course an explicit thanks to Philippe Geril, whom I personally consider to be the mastermind behind this all.

When I speak about playing games to my students, I often make the remark fun is highly overrated ... power, identity, excitement ... it's all in the game, and, as already alluded to above, the same holds for the game conference, the importance of which, in my view is as a social event, where we meet as researchers and scholars, make new connections, and deepen our knowledge. However, in my previous experiences of the GAME-ON conference, this has always be fun as well, and, again looking at the program, and the participants, it is easy to say enjoy yourselves, and, of course, the city of Brussels, including the Comics Museum, which would be worth a visit on its own.

For those not able to attend the conference in person, I'll see you another time, and hope that you enjoy the papers in these proceedings, which rightfully may be considered the fruit of our yearly gathering.

Brussels, 25 November 2013

Anton Eliëns
GAMEON'2013 General Conference Chair

CONTENTS

Preface ..... **IX**
Scientific Programme **1**
Author Listing **105**

GAME DESIGN

Lessons from Piracy: How pirates can save the Games Industry
Joseph Kehoe **5**

XIMPEL for Ethical Frameworks
S.V. Bhikharie and A. Eliëns **8**

GAME EVALUATION AND HEURISTICS

Evaluating Game Heuristics for measuring Player Experience
Björn Strååt, Magnus Johansson and Henrik Warpefelt **15**

Neural Network for Multitask Learning applied in Electronics Games
Alexandre De C Lunardi, Raniel Ferreira Correia, Alex F.V.Machado
and Esteban W.G. Clua **20**

IN-GAME CHARACTERS

Character Generation using Interactive Genetic Algorithm
Umair Azfar Khan and Yoshihiro Okada **31**

Avatars and the "Imitation Game" – Can Machines Smile?
Helena Barbas **36**

VIRTUAL WORLDS

Cooperative AI in Real-Time Strategy Games
Andreas Stiegler and Daniel Livingstone **45**

EmohawkVille: Towards Complex Dynamic Virtual Worlds
David Holaň, Jakub Gemrot and Martin Černy **52**

HTN or State Space - Who Should Do Planning in Your Game?
Martin Černy and Jakub Gemrot **59**

CONTENTS

CLASSROOM GAMING APPLICATIONS

A New Concept for Teaching AI using as Example Classics from Electronic Games
Laura L. Dias, Alex F. V. Machado, Tamillys P. C. Pinto, Flávia L. C. D. Andrade and Rafael R. Padovani **69**

A Serious Game Creation Project as Teaching Method for Geography
Merikki Lappi and Esa Lappi **76**

Why Children should program Computers without Computers
Euler Rithiele Q. Alvarenga and Alex F. Machado **80**

REAL WORLD APPLICATIONS

A monitoring tool to support remote caretaking of senior citizens
Aryan Firouzian and Henna Nissinen **89**

Serious gaming helps children view Highway Code from another perspective
Anthoula Alipasali and Anton Eliëns **94**

Applying the ALFIL Crowd Simulator to a Real Evacuation Incident: The News Divine Scenario
Cesar Garcia-Garcia, Victor Larios-Rosillo and Hervé Luga **99**

SCIENTIFIC PROGRAMME

GAME DESIGN

LESSONS FROM PIRACY: HOW PIRATES CAN SAVE THE GAMES INDUSTRY

Joseph Kehoe, Institute of Technology Carlow
email: joseph.kehoe@ITCarlow.ie

KEYWORDS

Methodology

ABSTRACT

Current game development industry practices are based on a hierarchical model much like the Hollywood film industry model. We propose a new model for the games industry based on the innovative governance techniques used by pirates in the 17th century.

INTRODUCTION

The computer game development industry has been continuously changing since its inception. This is witnessed by the number of new game platforms that are in development and this entails new architectures and new ways of interacting with the game. On top of this, and perhaps more importantly, delivery mechanisms have moved from real-world specialist shops to online stores. Revenue streams from games have moved on from those based on outright purchase to subscription, DRM licensing or free-to-play with in-game purchases. All this fragmentation creates problems and opportunities for game developers.

The dominant model of game development is one based on the "Hollywood Model". While there are similarities between game and film production we suggest that a more appropriate model is that of the historical pirate "industry". The historical data on piracy in the 17th century provides guidance for the games industry in terms of company strategies and new development processes. These techniques provided an alternative to the standard hierarchical model that was presumed to be the only way to successfully manage a large skilled crew. In fact, the more democratic methods used by the pirates generally outperformed the standard approach and allowed the pirates to run *one of the most sophisticated and successful criminal organisations in history* Leeson (2008).

It is our thesis that the techniques of pirate governance can be transferred to game development and will outperform the existing model of game development. Indeed we believe that the computer game development market is already heading in this direction.

THE HOLLYWOOD MODEL

Hollywood currently holds a central place in western culture. The Academy Award ceremonies are watched worldwide by an audience of hundreds of millions and its "stars" are feted worldwide. They have fame that game developers can only dream of.

Hollywood deals with stories told through the audio-visual medium. Films have been turned into games and vice versa although not always with success. The urge of game developers to imitate Hollywood appears to be natural. Hollywood is dominated by a small number of large studios and distributers. They control the releases that make it into your local cineplexes. This is the same situation in games with a small number of publishers and large game studios controlling the games we are likely to end up playing.

We can also see the similarities in output. A large number of the bestselling games have little or no original IP with sequels forming a large percentage of each years best selling games (the top ten bestselling games of 2012 were all sequels). These releases tend to rely on special effects and CGI with the story relegated to secondary status.

Both industries are male dominated and hierarchical. Studios exert a large degree of control over their employees and both industries see themselves as special (embodied with an "artistic" ability) while simultaneously wanting to be accepted into the mainstream. Game award ceremonies are copies of film awards. The games industry is already following the Hollywood model. In return for the perceived glamour we lose artistic integrity and independence. Hollywood films and AAA games are made purely for profit. In the Hollywood model Indy developers are locked out, lacking access to funding and distribution. This results in homogeneity of output, a pyramidical pay structure and total studio control.

THE PIRATE MODEL

The true and full history of piracy is too long a story to go into here but important aspects of their culture, I will argue, can be used as the basis of a model for game development. In particular I refer to the 16th and 17th century pirates and buccaneers of the Caribbean and Americas. Depending on your point of view the pirates operating in the Caribbean around the 17th century can

be viewed as *true revolutionaries, quasi-anaschists in an age of absolutism [or] heartless criminals, vicious individuals, who has rejected civilized society in favor of barbarism, greed and bloodlust.* Lane (1998)

What is true is that in an age dominated by a hierarchical stratified society they presented an alternative form of community. They rejected rule by birthright opting instead for an alternative democratic and meritocratic social contract. Their approach to self government was revolutionary at the time and is still considered so today by many people. The pirate approach to managing a campaign is embodied in their use of *Ship's Articles.* A number of these articles survive but the most complete is probably those of Captain Roberts Johnson (2002). Before each voyage or campaign a new set of Ship's Articles was drawn up and agreed. This laid out the command structure of the ship's company, how the profits were to be divided amongst the crew and the crew's rights and duties while on the campaign. At the end of each campaign profits were divided as per the agreed terms and the crew could choose whether to sign up for another voyage or not. These were revolutionary ideas in a time when a crew had few if any rights. Pirate crews had more rights, better conditions and a lower workload than anyone in the Merchant or Royal Navy at that time.

The most revolutionary and important part of the ships articles was the right to vote. These votes could take place at any time, baring during combat. They voted on, amongst other things, whether to engage with a ship and where the ship should sail to next. Leading directly from this was the election of the ships officers from amongst the crew. The officers could be impeached at any time during a campaign. Apart from being the military commander the captain was treated equally. He was given his own cabin but *every man, as the humour takes him, will use the plate and china, intrude into his apartments, swear at him, seize a part of his victuals and drink, if they like it, without his offering to find fault or contest it.* Johnson (2002)

One of the effects of this was the elimination of a separate officer class. In the merchant and military navies at the it was almost unheard of for a common crewman to rise through the ranks to become an officer. There was a distinct lack of hierarchy on a pirate ship and many of the officer posts found on other ships were dispensed with. The posts that were kept were held by crewmen voted most competent by their peers and the powers held by each "officer" were strictly circumscribed. The most important post was that of quartermaster who acted as a counterbalance to the captain.

Profits from a campaign were distributed equitably among the crew by the quartermaster. Crew who were officers got a larger share than general crew but the differences were not that large. Individual crew members could make more money from one pirate campaign than from a lifetime of service in the navy. Commercial ships profits, on the other hand, went to the ships owners whose only interest in the conditions of the sailors was their effect on profits.

Finally pirates operated the first workplace compensation scheme. Crew who were injured during service were given an extra lump sum at the voyages end. Compensation was generous and different amounts were given depending on the severity of the injury.

The results of their novel approach to managing crews manifested itself in a number of ways. Firstly pirates enjoyed much better conditions than their honest compatriots. Sailors had very few rights, endured long hours of dangerous work with little rest and low pay. A voyage that might make the fortune of the ships owner would have no effect on the livelihood of any of its crew, except perhaps the captain. Jumping ship was a common occurrence with some captains even resorting to refusing all shore leave to their crew when they reached port. In fact, many crew did jump ship, often to join pirate ships. The reason mutiny was treated so seriously was because of the very real fear that the crew would no longer accept the unquestioned and harsh authority of the officers. Pirate crews risked hanging but the rewards were such that this was considered an acceptable risk by many sailors.

Merchant ships kept crew numbers to an absolute minimum in order to save costs. This increased the risk to the crew members and lead to a perpetual crunch time amongst the crew. Cordingly (1999) states that pirate crews were much larger so there was less work for each individual to do. Contrast the fact that the navy had to employ press gangs with the fact that pirate ships were oversubscribed.

Pirate crews had what we call "buy-in". They were not hired hands but part owners involved in decision making and profit sharing. They fought harder than merchant crews precisely because they had more at stake. Once a merchant ship was taken its sailors could join the pirates and get a share of the profits from their plundered cargo. A small crew of highly motivated pirates could take over much larger ships of less motivated merchant seamen.

Another side effect of the pirate profit sharing scheme was that profits made by the pirates stayed in the Caribbean while profits made by merchants went back to investors. The profits that pirates made allowed for the development of a vibrant local economy that supported and protected the pirates. These factors led to the success of piracy in the Caribbean and the Americas for many years.

ADAPTING THE PIRATE MODEL

The current approach to game development was put in place for a number of good reasons. First there was the large capital investment required to set up and equip a game studio. Specialist software such as game engines and 3D modelling tools soon cost more than most in-

dividuals (and many companies) could afford. For this reason game studios required substantial injections of capital from outside investment companies who tend to be risk averse.

Second was the dominance of proprietary consoles. Releasing a game on a console required the permission of the console manufacturers who acted as gatekeepers. Developing for a console required the backing of specialised game publishers. These publishers exert, in many cases, complete control over the game studio not the originators of the IP. A developer of commercial software can rely on extra sources of income on top of the sale price of the package such as training and a yearly maintenance fee. They can be fairly sure that most of their existing customer base will buy the next release of their software thus ensuring a growing customer base for each version. A game on the other hand is a one shot sale at a fixed price point with no training or maintenance required and no tie-in of existing customers. Each new original IP game release starts with an installed user base of zero, a risky proposition which helps explain the preponderance of sequels.

When we suggest that the pirate model is followed we are referring to the adoption of the pirate management techniques. The current approach to games development is based on a number of assumptions that are no longer true. Firstly, there is the assumption that a large capital investment is required to set up a game studio. Many of the tools that used to be prohibitively expensive, such as top tier games engines, are now available either at no cost or under terms favourable to start-up companies. Entry into the market place no longer requires large capital investment.

Secondly, distribution channels are opening up. As the grip of the console manufacturers lessen it is becoming easier to get games published. Microsoft, for example, now allows independent games companies to release games on XBox live. Games written in HTML5 can be released and accessed on all platforms. Virtual storefronts such as Steam, Google Play and the App Store are becoming the normal way of delivering games to the public. This combination of easily accessible distribution channels and lower capital costs is reducing the influence of gatekeepers.

Even setting aside these changes criticism of the large game studios is growing. The lack of originality of the games produced by large game studios is bemoaned. Studios prefer to produce sequels rather than original IP. Shareholder owned companies do not make games to express artistic vision but to produce quarterly profits.

Under the pirate model control over the final release and development process would be retained internally. Equal voting rights means that developers have input into all decisions. Key management positions would be filled by ballot ensuring that all levels of management remain part of and answerable to the team. Every developer would get an agreed share of the profits. These companies would be agile, able to change tack at any time and would exist only for an agreed objective. It seems likely that it would also reduce the problem of long working hours due to misjudged schedules or underemployment of developers. This would be a radical departure from current approaches but would keep control over the game completely in-house and under developer control. The risks associated with developing original IP would be outweighed by the potential rewards that would accrue to the development team.

CONCLUSION

The pirate approach to project management has already proven its worth. Running a large sailing ship requires iron discipline, tight teamwork between a variety of highly skilled crewmen, and the ability to make instant command decisions. Pirates proved that a self governing, democratic, approach was not only possible but able to outperform the autocratic hierarchical model.

Game development is similar in that a missed deadline can break a game and many disparate highly skilled people must work together. At present the developers do not see the profits of their work and often cannot make the original IP they want. The current approach to game development has resulted in a lack of original IP being released, high workloads and burnout among computer developers and an unfair sharing of profits. Software development companies can learn much from the pirate approach. Pirates used agile methods similar to those used in today, Beck and Andres (2005), Shore (2007), but they extended their usage to incorporate the entire organisation. It is time to regard pirates as exemplars of what is possible when an industry shakes off its shackles and embraces the ultimate agile charter - the "Ship's Articles".

REFERENCES

K. Beck and C. Andres. *Extreme Programming Explained: Embrace Change.* Addison-Wesley, 2005.

D. Cordingly. *Life among the Pirates: The Romance and the Reality.* Abacus, 1999.

C. C. Johnson. *A General History of the Robberies and Murders of the most notorious Pirates.* Conway Maritime Press, 2002.

K. E. Lane. *Blood and Silver: A History of Piracy in the Caribbean and Central America.* Signal Books, 1998.

P. Leeson. An-arrgh-chy: The law and economics of pirate organisation. *Journal of Political Economy*, 2008.

J. Shore. *The Art of Agile Development.* O'Reilly, 2007.

XIMPEL FOR ETHICAL FRAMEWORKS

S.V. Bhikharie and A. Eliëns
Business Web & Media
Department of Computer Science
Faculty of Sciences, VU University
De Boelelaan 1081
1081 HV Amsterdam
The Netherlands
E-mail: svbhikha@few.vu.nl & eliens@cs.vu.nl

KEYWORDS
ximpel, ethics, interactive video

ABSTRACT

Videogames provide us with possibilities to think and reflect about moral dilemmas in an applied and interactive way. Ethical frameworks for videogames can be similarly applied to XIMPEL, an interactive media platform. Through the exploration of two examples we abstract the relevant properties and combine them with the knowledge about ethical frameworks for videogames to define a template that can be used to apply ethical frameworks to XIMPEL. This is practically achieved by properly structuring a XIMPEL playlist with a compelling narrative.

INTRODUCTION

We are confronted with moral dilemmas in the real world in places like our neighbourhood and workplace and in virtual worlds in places like online communities and videogames. Videogames in specific are interesting tools for exploring dilemmas, because they provide us with possibilities to think and reflect about choices in an applied and interactive way.

(Zagal, 2009) proposes that a videogame can be seen as an ethical framework if it couples the evaluation of in-game actions with the narrative framework that contextualizes them. He concludes that a game is ethically notable if the ethical framework in a game is both discernible and consistent and that the dilemma is actually a moral one and that the player is the one facing it.

Similarly, we can apply the concept of ethical frameworks to other digital interactive media. In this paper we explore how ethical frameworks can be applied for XIMPEL, the eXtensible Interactive Media Player for Entertainment and Learning.

WHY XIMPEL FOR ETHICAL FRAMEWORKS?

XIMPEL is a platform for creating interactive media productions. It resolves the tension between immediate attention and convenient exploration by allowing viral clips to act as an interface to a rich repository of other (interactive) video material, as well as other interactive applications including casual mini-games and geographical maps (like Bing or Google Maps). In other words, XIMPEL allows for interactive storytelling with an immersive navigational structure. On www.ximpel.org you can find tutorials, examples and downloads to get started.

Unlike videogames, XIMPEL aims to be accessible to a wide audience as an authoring tool, since it allows for the creation of an interactive story through an XML-based playlist, offering a standardized and well-known way to define your own interactive experience. With the release of version 3.0 of XIMPEL, YouTube video support has been included to make it even easier to add video content to your interactive production, directly streaming from the popular video portal. Just knowing a YouTube id is enough to make your first interactive creation using the online playlist editor at http://ximpel.few.vu.nl/tester/.

In short, XIMPEL provides a convenient and accessible way to create ethical frameworks. In fact, since its inception in 2007, many XIMPEL applications have been created (see http://ximpel.few.vu.nl/showcase.html for a selection), with quite a few dealing with moral dilemmas. In the next sections two applications are discussed, which have an ethical framework encoded into them.

XIMPEL EXAMPLE: CLIMATE PROFILER

The Climate Profiler, developed by the XIMPEL team, is an application that determines how important the climate is for the player through a series of choices. Each choice affects one or more of the 3 p-values: people, planet and profit.

Depending on the choices you make, you get to see a specific narrative path. After the player reaches the end of a path, the state of the world is shown with an image that contains a positive or negative visual representations of each p-value.

The player can also view the path that was taken and see how each choice made affected the p-values.

By giving feedback on the choices made, the player is encouraged to reflect upon the choices made and replay and explore other paths of the Climate Profiler.



Figure 1: Climate Profiler with 1 of the Possible 8 End Images

Furthermore, certain choices made are not necessarily good or bad, but are presented as dilemmas to create conflict for the player. For example, the choice between nature and luxury is not necessarily one that players need to make in real life, but in the context of the Climate Profiler, the player is challenged to think about the trade-offs being made and how it affects the ethical system.

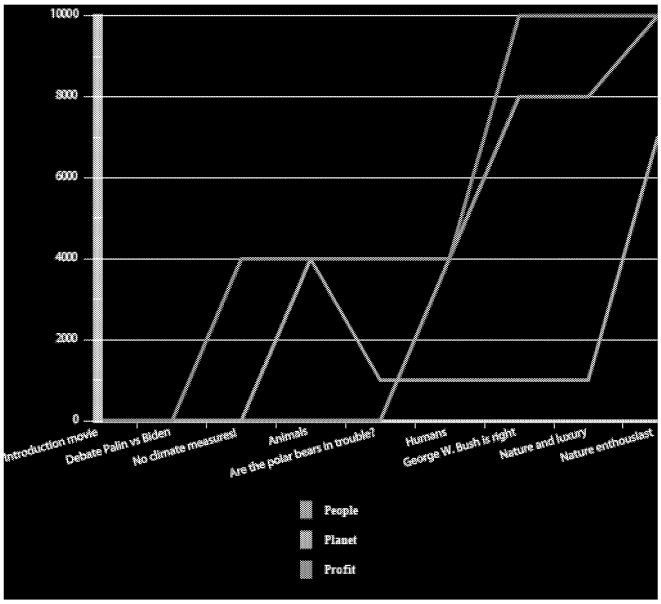


Figure 2: The Climate Profiler Graph Shows the Player How Each choice Affected the P-Values

XIMPEL EXAMPLE: HEALTHY EATING

The Healthy Eating application1, developed by students Frederieke Nieuwland en Jorien van Ginkel at VU University, deals with ethics on a personal level, namely eating habits.

The application is structured in a linear way. After a short videoclip, the player is asked to make a choice concerning food. Each choice influences your health score. After 6 choice segments, the player views the ending message, which is followed by an evaluation of their health score.

The dilemmas presented are quite lighthearted of nature, like choosing between eating or skipping breakfast. Furthermore, the videoclips used are quite humorous. Therefore, when the player makes a choice that has an unexpected outcome, not only does it grab your attention, but it makes players reflect on their choices.



Figure 3: Healthy Eating – Choice Between Two Pastries

Because the goal of this application is to create more awareness about eating habits, it creates a positive environment to do so. It never forces it ethics on the players, but cleverly shows how certain eating choices affect them.

ETHICAL FRAMEWORK TEMPLATE

Both the Climate Profiler and Healthy Eating give us a valuable insight in how to use XIMPEL to create ethical frameworks. The Climate Profiler does this by showing how a choice affects the outcome, both in a visual way with the end image and with a detailed graph after playing. The Healthy Eating application uses a more personal approach that tries to engage and surprise the player.

What both have in common is that they present the player with immediate feedback after a choice is made. Another important aspect is the narrative. The player is giving a certain context that makes it clear why choices have to be made. Both applications do this by means of an introduction video that sets the tone for the coming dilemmas.

If we abstract these properties from the examples and combine this with the knowledge from (Zagal, 2009), we can define a template that allows us to effectively use XIMPEL for ethical frameworks:

1. The player is made familiar with the ethical system through a narrative.

2. The player should get proper feedback when making a choice in a dilemma.

The question that now remains is how can we translate these two rules in a concrete way to XIMPEL?

APPLYING THE ETHICAL TEMPLATE

The template we defined in the previous section can be applied concretely to the XIMPEL playlist, which is the heart of an application.

We start out by specifying an introduction videoclip. This clip introduces the main narrative and context we as creators want the player to explore.

Next up is adding dilemmas with choices and proper feedback. The dilemma can be introduced through a videoclip. The choice moment itself can be a short looping video or an image. The possible choices are marked with overlays: visual clickable markers over specific sections of the application screen. Note that it is also possible to give a player a limited amount of time to make a choice. If no active choice is made, the player can be forced to go along with another path. This creates additional tension and could possibly enrich the moral dilemma, especially if you want the player to act quickly.

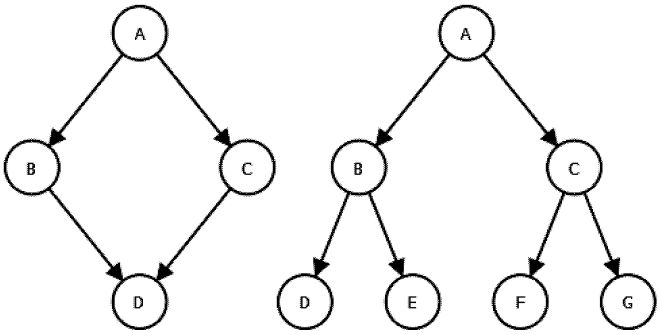


Figure 4: XIMPEL Story Graph Structures

Depending on the (non-)choice made, the player gets feedback in the form of a videoclip that shows the consequence(s) of their action. It is up to the creator to determine what happens afterwards: the story graph for the narrative can reflow into one path or branch out into multiple paths (figure 4).

Figure 5 lists a sample of the introduction of a playlist applying the ethical template. The *subject* is the main building block in a XIMPEL playlist. Each subject always has a unique id, a description and a list of one ore more *media* elements. Available media types are local *video, picture and youtube* type.

The first subject contains the optional leadsto property. This property is used to link one subject to another. In this case, after playback of the local video intro.mp4, the next subject that will be loaded is “dilemma1”. This subject is our first dilemma and displays a picture still with a *canvas*, containing two clickable *overlays.*

```
<ximpel>
  <subject id="introduction"
    leadsto="dilemma1">
    <description>
      Introduction
    </description>
    <media>
      <video file="intro.mp4"/>
    </media>
  </subject>
  <subject id="dilemma1">
    <description>
      Dilemma 1
    </description>
    <media>
      <picture file="1.png">
        <canvas>
          <overlay x="114" y="120"
            width="120" height="342"
            leadsto="1a"/>
          <overlay x="585" y="124"
            width="100" height="340"
            leadsto="1b"/>
        </canvas>
      </picture>
    </media>
  </subject>
  <subject id="1a"
    leadsto="dilemma2">
    <description>
      Dilemma choice 1a
    </description>
    <score value="-10"/>
    <media>
      <video file="d1a.mp4"/>
    </media>
  </subject>
  <subject id="1b"
    leadsto="dilemma2">
    <description>
      Dilemma choice 1b
    </description>
    <score value="5"/>
    <media>
      <video file="d1b.mp4"/>
    </media>
  </subject>
  <subject id="dilemma2">
    <description>
      Dilemma 2
    </description>
    <media>
      ...
    </media>
  </subject>
</ximpel>
```

Figure 5: Sample of XIMPEL Playlist Structure for Ethical Frameworks

Overlays can be customized using various properties, with the basic ones being x, y, width and height. Each overlay links to a specific subject ("1a" and "1b"), one for each of our available choices.

After making a choice, either subject "1a" or "1b" is loaded and a *score* value is added to the player's current score. Note that a score value can be both positive or negative, which could possibly lead to certain choices canceling each other out.

After playback of the media of a the chosen branch, both subjects redirect the player to the same next subject, "dilemma2" (partially listed). The graph structure of the playlist in figure 5 equals that of figure 4a.

PRACTICAL CONSIDERATIONS FOR APPLYING THE ETHICAL TEMPLATE

As can be seen from the sample playlist in figure 5, the amount of subjects grows quite fast depending on the number of choices for each dilemma. When creating a more branched out graph structure like in figure 4b, having multiple links to one subject makes creating a playlist more manageable in terms of playlist authoring and gathering of necessary media elements.

Furthermore, by having a limited amount of end points, a narrative can be closed off in a natural way working towards a specific ending and lets the player achieve the same goal in multiple ways.

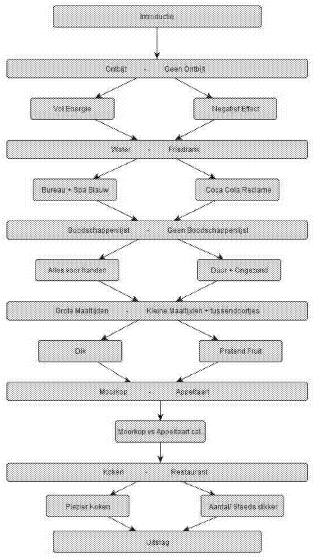


Figure 6: Example of Story Graph for the Healthy Eating Application

Independently of the chosen branching structure, XIMPEL provides a default scoring mechanism to give weight to your choices. This makes it possible to evaluate the player after finishing a path. With some additional (Actionscript 3) programming, it is possible to keep track and evaluate multiple score values. This allows for more depth and precision in keeping track of decisions. Likewise, it allows for a custom way to provide feedback to the player, for instance with an image like in the Climate Profiler (figure 1).

We can repeat the process of creating dilemmas and choices with overlayed videos/images as many times as needed. However, in doing so we also need to make sure that the narrative properly supports the ethical framework. If we do not enforce this throughout the whole narrative, the interactive experience becomes disjointed and the dilemmas become meaningless. One practical way to tackle the issue of structuring dilemmas is by creating a story graph, which provides a blueprint for the XIMPEL playlist you will be building (figure 6).

When the player finally reaches an end point of a path, some form of reflection should be offered, whether this is in the form of an ending videoclip, or by using the scoring mechanism to inform players about how well they did. This is essential for creating a meaningful experience within the ethical framework.

CONCLUSIONS

By looking at two notable examples of XIMPEL applications with a solid ethical framework, we have defined an ethical framework template, using insights from ethical frameworks within a videogame context. This template can be practically applied to XIMPEL by creating a playlist according to the principles:

1. The player is made familiar with the ethical system through a narrative.
2. The player should get proper feedback when making a choice in a dilemma.

Practical considerations like playlist size, overall narrative structure and score evaluation are important as well.

FUTURE WORK

XIMPEL is an ongoing effort and new ways to interact with different media types are currently being explored:

- Geographical maps. This allows the player to use the map as an interface for accessing other media types.
- Minigames. These games be anything, from classics like memory to platform games or even more complex like 3D games. The challenge here is to find minigames that work well together with other types of media.

Other research topics include (but not limited to) alternative interfaces like XIMPEL TV and Pixel Video and generating video summaries.

REFERENCES

Zagal, J.P. 2009. "Ethically Notable Videogames: Moral Dilemmas and Gameplay." In *Breaking New Ground: Innovation in Games, Play, Practice and Theory* (DiGRA, London, September 2009).

GAME EVALUATION AND HEURISTICS

EVALUATING GAME HEURISTICS FOR MEASURING PLAYER EXPERIENCE

Björn Stråat
Magnus Johansson
Henrik Warpefelt
Stockholm University, Department of Computer and Systems Sciences/DSV,
Forum 100,
SE-164 40 Kista, Sweden
+46-709723222
bjor-str@dsv.su.se
magnus@dsv.su.se
hw@dsv.su.se

KEYWORDS
Games, Immersion, Usability, Heuristics

ABSTRACT

Usability evaluation tools in the form of heuristic lists can be very helpful in software development. In recent years, game researchers have developed heuristics aimed at special design issues within computer game design. This article examines if these game heuristics are able to capture and evaluate softer values of computer game interaction, based on the flow, challenges and immersion of game play. The authors suggest that these softer values should be called *Gameworld Interaction*. An analysis of the existing game heuristics was made, resulting in a Net Heuristic list, which was then tested on a single game. The result showed both a possible gap in the heuristics, and that even though the game design did adhere to the heuristics, problems still occurred.

INTRODUCTION

There are several ways to analyze and evaluate productivity software from a usability perspective. By involving users in tests or by doing expert or heuristic evaluations, software producers can optimize the usability and user experience off their products. This process is well known in most areas of computer software development. However, these methods were not applied in the computer game industry for a long time (Isbister & Schaffer, 2008), mostly because the available evaluation tools were not constructed to measure computer game software.

Part of the entertainment in playing games is that they are challenging; that is how they are designed. Without challenge a game quickly becomes uninteresting. Productivity software on the other hand, such as a word processor, should contain no designed challenges at all. Since most evaluation methods are developed to find (and eliminate) user challenges, it is clear to see why games need specially designed tools. Evaluating a game using generic Human Computer Interaction (HCI) -evaluation tools is unfair to the game. A deliberate game challenge may be flagged as a severe error from a pure usability perspective, while it may add value to the game experience from an entertainment and game challenge perspective.

Lately, in order to address this issue, game researchers are developing heuristic lists for evaluation of games. These lists are based on the productivity software standard Heuristics (Nielsen, 1994) combined with rules-of-thumb (Barwood and Falstein, 2006) from the game industry. A number of studies have been made, and several methods are in use today by the game design industry (Federoff, 2002; Laitinen, 2006; Desurvire et al, 2004; Pinelle et al, 2008; Desurvire and Wiberg, 2009).

All these methods are well-defined and based on solid methods. However, what they all have in common is that they are fairly unspecific and sometimes more aimed at good game design than good game usability. Game usability, just like usability in productivity software, adheres to the interaction of the artifact.

The authors of this article would like to define two areas of computer game interaction; what the players do within the world of the game to actually play it (walk around, interact with items, solve riddles etc.), termed *Gameworld interaction (GI)* by the authors, and what the players do with the game settings to set it up for use (setting up controls, sound, saving progress, graphics etc), termed *Supporting interaction (SI)*.

The authors examine the heuristic lists from the contemporary literature, with the purpose of finding heuristic principles with explicit focus on usability in GI. A list of findings is put together and tested on a single game.

A heuristic evaluation can be used during game development to find usability issues at an early phase of the production, thereby minimizing the costs for creating good game experiences. It can also be used as a validation tool in mid process to verify Gameworld usability in line with the developer's intent. For game research purposes, the heuristic evaluation may serve as a comparison tool between different games.

BACKGROUND

Why Work With Heuristics?

Heuristics can be viewed as a list of "dos and don'ts" that can be applied directly by game developers. Short development cycles require fast and easy ways to measure usability in user interfaces (Shaffer, 2008). Heuristic evaluations do not take very long time, and do not require

several test subjects. Just a few evaluators can find a lot of problems in a two hour session. The evaluators do not need to be usability experts as long as they critically examine the system in question, using a predefined set of rules. Heuristic evaluations have more impact if they are performed early and repeatedly during a design process.

Gameworld and Support Interaction

Rollings and Adams (2003) describe the action of saving the progress in a game as one of the most immersive-breaking activities that occur in a game. This would indicate a natural subdivision between the actual game-play (things the players do, see, expect and experience in the game world) and other things the players can or must do with the game (such as changing the graphical settings, saving, reloading, etc.). By dividing game-play from other activities a player can do, we introduce a natural division between two types of player activities where we can study the challenges of the game-play (Gameworld interaction, GI) in isolation from setting up the game system (Supporting interaction, SI).

Challenges, Immersion and flow

Jonas Linderoth (2011) describes how the challenges in a game can be expressed as either a search for things players can do in the game, called *exploratory challenges*, or in the execution of actions, called *performatory challenges*. For example, deciding what move to make in a chess game would challenge the player's exploratory skills; what can be done to improve the chance for victory or get out of a threatening situation? Most people can move the pieces physically, but the decision behind the move must first be analyzed by carefully exploring the options and consequences. On the other hand, anyone who has seen a game of tennis knows that the ball shall be struck with the racquet, over the net. Here, the challenge lies in performing the task; not every player manages to win a game of tennis, even if they know exactly what to do. Linderoth uses the affordance concept (as defined by JJ Gibson (1977) and later Norman (1998)) to describe the ability to find and execute actions - the world around us affords using, based on our abilities. If a performatory or exploratory challenge is not properly afforded, we will not be able to act on it and instead have a performatory or exploratory restriction, which in essence would indicate a game specific usability issue.
Based on this reasoning, it is clear that any challenge of a game should appear in the game-play and not in manipulating the game software. Regardless of whether the challenge lies in finding out what to do next or in performing a task, the design of the game must afford all possible options to the player. Juul (2010) states: even if a game is easy to use in terms of manipulating the game world, it still can be difficult in terms of challenging game events. Juul refers to Beaudouin-Lafon (2000) and concludes:

> *"Interfaces are interaction instruments that mediate between the user and the domain objects the users want to act on"*

The game interface could as described in the quote be seen as an *interaction instrument* designed to make the game possible to play. Therefore, when evaluating a game, using HCI evaluation tools, the evaluator must distinguish between a deliberate game challenge, and genuine HCI issues. From the point of view of GI and SI, regular HCI issues are easier to spot in the SI-interface of the game, while GI issues might be considered part of the challenge design. Further, it is important to reflect upon what would be a usability problem in one system is part of the appeal in another. The purpose of using a game is not to set a diverse set of properties, but to play the game even though controlling these diverse settings are an integral part of the ongoing game experience.

Immersion and Flow
Immersion is the way a player feels connected to the activity of playing the game or "*the sensation of being surrounded by a completely other environment […] that takes over all of our attention…*" (Page,1999).

Ermi and Mäyrä (2007) suggest three different types of immersion:

- Sensory-based immersion relates to the audiovisual execution of games. Games are often aesthetically appealing and can capture the attention of even inexperienced players
- Challenge-based immersion is based on the interaction with the game challenges. If balanced correctly, a game challenge is not too easy to perform (which would make it dull) nor to difficult (which would make it frustrating). Challenges can be related to motor skills or mental skills, depending on game type.
- Imaginative immersion allows the player to use their imagination, sympathize with game characters and feel involved in the storyline.

.
When the level of immersion is just right a user will achieve flow. Either case of immersion can create a sense of flow, according to the definition by Mihalyi Csikszentmihalyi (2009). The concept is described as "*A state of mind when a player experiences a high degree of focus and enjoyment*".
If the game offers a good challenge, which keeps the player challenged and without elements of frustration or boredom, the player may reach a certain state of mind. The player stops thinking about time and space and accepts the game as part of reality. As the game goes on, the challenges get tougher, but keep in line with the growing skill of the player, preventing both boredom and frustration.

Gameworld Interaction in This Study
The expectation a player has on a game is dependent on the immersive powers of the game. The exposition of the game world contributes to immersion as it puts the player in a certain mood and awakens certain expectations.
The game world experience comes from interaction with the world, when the player meets the challenges of the world and successfully overcomes them. Challenges in the game world should be intentional, and help the player to reach the goals of the game, as well as being all part of the designers vision. Any unintentional challenge that is not part of the goal, can be seen as an intrusion.

The authors have decided to call the interaction, expectations and experiences within the world, i.e. playing the game, *Gameworld Interaction*, (GI) to make a stronger heuristic evaluation ground. It differs from other definitions of gameplay since its purpose is to define heuristically evaluational activities.

To define if a heuristic principle from the related research can be used as a measure of Gameworld Interaction, it should:

- Adhere to the gameplay or game world
- Adhere to in-game challenges
- Not allow for immersion break
- Not allow for break of flow

RELATED RESEARCH

Melissa Federoff (2002) was first to create a set of heuristics for games, in cooperation with a game developer team. Federoff collected heuristics from the 400 project list (Barwood and Fahlstein, 2006) 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. This became the very first set of heuristics aimed solely at the game industry.

Sauli Laitinen used Nielsen's heuristics to evaluate a computer game under production (Laitinen, 2006). 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, Caplan and Toth (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 is based on Federoff's work (2002) and the 400 Project Rule List (Barwood and Fahlstein, 2006). The HEP list carries a strong focus on game design issues and learnability of the game (Isbister and Schaffer, 2008), and is widely used by game designers.

Pinelle, Wong and Stach (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 (2009) 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.

METHOD

From the heuristic lists mentioned in the related research section, we derived a Net Heuristic list, a Supporting Heuristic list and a Meta list.
The Net Heuristic list contains heuristic principles that match the four criteria for measures of Gameworld Interaction. The Supporting Heuristic list contains principles necessary when designing game menus for settings etc. The Meta list contains principles of such a high abstraction or universal wisdom, that they are not really applicable, e.g. "make the game fun to replay". Only the Net Heuristic list will be presented in this article.

The PLAY heuristic list (Desurvire & Wiberg) is the most recent, and also widely used by the industry. It was used as a basis for comparison with the other lists.
First three researchers separately analyzed the heuristic lists from Desurvire (HEP) , Federoff and Pinelle & Wong, one list at a time. All entries in the lists were categorized as either *GI* or not. Any principles that would not fit into the *GI* category were omitted - e.g. a rule about allowing the player to turn of the game whenever she wants was considered to be a supporting feature, while a rule about reasonable behavior for computer controlled units was considered a *GI* principle.

When all the lists were examined, the selected principles from the different lists were compared. Any duplicate was omitted. The three researchers did this separately, and then discussed, compared and analyzed their lists. This left a list of principles that finally was compared to the PLAY list. If any duplicates were found, the PLAY version would take priority, and the other version would be pruned.

The final list – the Net Heuristic - was then used to perform a pilot test on a computer game.

The pilot test
Two researchers examined the same game separately. A two hour session was played in a first person role playing action game, Tomb Raider 2013. Resulting notes from the researchers were compared and analyzed. Only in-game situations were considered. Issues with setting up the game or maintenance such as character skills, inventory upgrades etc were not considered. All issues were written down, explaining when they occurred, if they were recurring and what effect they had on the gaming experience. The list of issues was then analyzed using the Net Heuristic list.
If an issue found in the game could be identified in the Net Heuristic list, it was considered a mistake in the game design and no further notice was given. If it could not be identified in the Net Heuristic list, a second search was made in the omitted heuristics. If it couldn't be found there either, a gap in the tool was noted. Finally, if an issue adhered to principles from the Net Heuristic list but still was breaking immersion or flow, it was also considered a gap in the tool.

RESULT

The complete Net Heuristic as follows:

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).

Results from the game evaluation test

A total of 12 issues were found. Of these, 10 could be considered to be resulting from designers not adhering to the heuristics. One issue did follow the principles of several heuristics but still fell short, creating a breach in flow and immersion:

Some events are based on quick response key pressing (QTE – quick time event); like wrestling free from an enemy requires mashing of left- and right-move key, followed by other key instructions. This is shown as an overlay UI, which either is hard to see the first times (leading to death for the avatar) or takes focus from the event itself. This makes for poorer immersion. On this specific account, we can apply the following PLAY rules: "*The players experience the user interface/HUD as a part of the game*", "*Players should be given context sensitive help while playing so that they are not stuck and need to rely on a manual for help*" and "*game utilizes visceral, audio and visual content to further the players' immersion in the game*". However, even though these heuristics have been properly met, the situation is still confusing, misleading and disruptive to immersion, rendering this a gap in the tool.

The last issue could not be found in either of the heuristic lists, and was causing breaches of immersion:

The graphics in the scenes invites to interaction (barrels are for jumping on etc) but no such interaction is possible. This issue is frequent throughout the game. It cannot be clearly mapped against any of the heuristics in the Net Heuristic list, nor the supportlist.

Table 1: Relation between GI Criteria and Net Heuristics

Criteria	Pinelle et al.	HEP	Play
Game play or game world	1	1-4	1- 9
In-game challenges	1	1,2	1,5-6,8
No immersion break	1	1-4	1-5, 7, 9
No break of flow	1	1,2	1,5,6, 8

ANALYSIS AND DISCUSSION

As can be seen in table 1, the criterion "Game play or game world" matched all the heuristics. On one hand, this might make this criterion redundant, but on the other hand, it may be wise to make this criterion the only mandatory one, since any heuristic that does NOT fall under this rule should not be considered further.

We found that several of the heuristic principles from the material covered a large part of the Gameworld issues. The list put together can be used as a short tool for easy evaluations. The results from the game evaluation showed that the existing heuristic principles can be refined and

extrapolated. Even though the game designers follow the heuristic principles, they still may fall short.

We also feel that the articles strongest contribution is the definition of the concept of *Gameworld interaction*. It allows for a more specified view on games and game design, and in this particular instance it allows for specified heuristics.

We have shown that the existing heuristics do cover a large amount of the needs for computer game evaluations, but also that combined the heuristics can be used to pin-point softer values of game experience. The heuristics are written at a high level of abstraction, which might make them either to obvious or unnecessary. Most need more and deeper exploration to be fully functional. With she Net List as a base, future work on the heuristics would be able to provide more concrete advice to game designers, as opposed to the current heuristics that are very abstract.

No alternative heuristics to overcome any issues found are suggested. The article is an attempt to conceptualize and frame the *Gameworld Interaction.* As a natural next step, the authors intend to cooperate with the game industry to examine the concept, and to further evaluate the heuristic principles. More tests on more games are of course needed to establish a firmer ground. More research will lead to new heuristic tools which can help evaluate games from this perspective.

REFERENCES

Beaudoin-Lafon, M (2000). "Instrumental Interaction: An Interaction Model for Designing Post – WIMP User Interfaces." *CHI 2000 proceedings*, page 448

Norman, D.A. (1999) "Affordances, conventions and design". in Interactions, vol. 6, no. 3, pp. 38-43.

Csikszentmihalyi, M. (2009). "Creativity: Flow and the Psychology of Discovery and Invention". HarperCollins e-books.

Desurvire, H., Caplan, M., & Toth, J. A. (2004, April). "Using heuristics to evaluate the playability of games." In *CHI'04 extended abstracts on Human factors in computing systems* (pp. 1509-1512). ACM.

Desurvire, H., & Wiberg, C. (2009). "Game usability heuristics (play) for evaluating and designing better games: The next iteration." In *Online Communities and Social Computing* (pp. 557-566). Springer Berlin Heidelberg.

Ermi, L., & Mäyrä, F. (2007). "Fundamental components of the gameplay experience: Analysing immersion." *Worlds in Play: International Perspectives on Digital Games Research. New York: Peter Lang Publishers*, 37-53.

Federoff, M. A. (2002). "Heuristics and usability guidelines for the creation and evaluation of fun in video games" (Doctoral dissertation, Indiana University).

Gibson, J. J. (1977). "The theory of affordance", *in Shaw, R. E. and Bransford, J. (eds), Perceiving, acting and knowing, Toward an ecological psychology*. LEA: Hillsdale, NJ

Isbister, K., & Schaffer, N. (2008). "Game Usability: Advice from the experts for advancing the player experience." CRC PressI Llc. 79-80.

Isbister, K., & Schaffer, N. (2008). "Game Usability: Advice from the experts for advancing the player experience." CRC PressI Llc. 81-82.

Juul, J (2010). "A Casual Revolution: reinventing video games and their players." The MIT Press

Laitinen, S. (2006). "Do usability expert evaluation and test provide novel and useful data for game development." *Journal of usability studies*, *2*(1), 64-75.

Linderoth, J (2011) "Beyond the digital divide: An ecological approach on gameplay." *Proc of DiGRA 2011*

Page, B. (1999). "Hamlet on the Holodeck: The Future of Narrative in Cyberspace." *MFS Modern Fiction Studies*, *45*(2), 553-556.

Pinelle, D., Wong, N., & Stach, T. (2008, April). "Heuristic evaluation for games: usability principles for video game design." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1453-1462). ACM.

Rollings, A., & Adams, E. (2003). "Andrew Rollings and Ernest Adams on game design." New Riders Pub.

WEB REFERENCES

Barwood, H. Falstein, N. (2006) The 400 Project Rule List
Online: http://www.finitearts.com/Pages/400page.html
Last accessed: 20130510

GAME REFERENCES

Tomb Raider
Square Enix, Crystal Development
Online: http://www.tombraider2013.com/

NEURAL NETWORK FOR MULTITASK LEARNING APPLIED IN ELECTRONICS GAMES

Alexandre de C. Lunardi, Raniel Ferreira Correa, Alex F. V. Machado
Instituto Federal de Educação Tecnológica do Sudeste de Minas Gerais, Departamento de Computação, Brazil.

Esteban W. G. Clua
Universidade Federal Fluminense, Instituto de Computação, Brazil.

KEYWORDS

Multitask Learning; Transfer Learning; Neural Network.

ABSTRACT

Transfer of Learning between tasks may play an important role in Machine Learning, leveraging the knowledge generated by the training signals of a task in other similar tasks. One of the ways that can be employed to transfer learning among these tasks is by Multitask Learning. In this article, we will quote some ways to use the Multitask Learning, emphasizing the use of the same combined with artificial Neural Networks, where a network is capable of learning multiple tasks in parallel, reducing the time that a task can be completed. With this, we demonstrate that in addition to increase learning in a single application, the knowledge acquired can be availed by a neural network similar in other applications. In our tests, we use real data to control the games Pong and Air Hockey through a single neural network.

INTRODUCTION

One of the areas that inspired the Artificial Intelligence (AI) in the search of the building of a cognitive system is Psychology. An example is the term Transfer Learning (TL) that was quoted initially by the philosopher John Locke (Hergenhahn and Olson 2005) and later defined by the psychologist Edward Lee Thorndike through Learning Theory (Perkins and Salomon 1992). He demonstrated that learning is incremental and automatic (Hergenhahn and Olson 2005). For Thorndike, transfer occurs when something learned in one context is applied in another related context (Perkins and Salomon 1992). In AI, more specifically in the area of Machine Learning (ML), Transfer Learning concerns the application of knowledge acquired from one task to another, related or not (Pan and Yang 2010).

Among of the techniques of Transfer Learning, we can highlight Multitask Transfer Learning, a form of inductive learning. The purpose of the MTL is to improve the performance in generalizations, using the information of specific domains contained in the related tasks. In other words, TL treats as learning between any kinds of different tasks and MTL is a subdivision of the TL where tasks must have similar characteristics. As result, it makes tasks use shared representations and training signals to serve as an inductive bias (Caruana 1997), creating a healthy and intelligent system. An example shown by Caruana (Caruana 1997) proposes that a network trained to recognize types of doors (single or double) in images collected with a robot-mounted color camera, which has trained more than one task at the same time, is better than multiple networks with only one exit.

Another important factor is the fortification of the transfer of learning using Neural Network (Caruana 1997), which is one concept based on the observation of human systems, in this case, the brain itself. Most of the studies in transfer learning have used neural networks efficiently, as in (Caruana 1997) and (Baxter 2000).

As Caruana (Caruana 1997) proposes the recognition of the doors with MTL, we present the use of Neural Networks for Multitask Learning in games, analyzing two classic games with simple neural networks and we propose a multitask learning technique with a neural network for the control of the two games, this is, the two sets using the same network.

The Overview section shows the main techniques that will be employed in the work, being fundamental to the understanding of it, since there are few studies with MTL and Neural Networks, for example. In the Neural Network Modeling section we present the games that will be used along with the respective neural network topologies and a hybrid neural network, able to run both games, transferring learning from one game to another.

Tests and Results section, we show the performance of the three neural networks, and propose the use of a function described by Caruana (Caruana 1997), able to improve the transfer of tasks between the perceptron Network Hybrid. Discussion section presents the advantages and disadvantages of using this type of neural network. In addition, we discuss a possible evolution to more complex problems, such as games highest levels, with many having boots using TL. In the section Conclusions and Future Works present the results found in testing, describing a possible system that is able to control one or more games at the same time.

OVERVIEW

In this section we present the main definitions used during the development of the article. These concepts are essential to understanding the next sections, since, as it is a theme

(Transfer Learning) with few research, each topic is crucial in this work.

Transfer Learning

The main definition of Transfer Learning, according to (Pan and Yang 2010), says that given a domain source (DS) and a task source (TS) learning, a domain target (DT) and a target task (TT), the transfer aims to help the function (f(T)) of DT learning task using the knowledge acquired of TS and DS, where DS is different from DT or TS is different from TT.

Common machine learning algorithms traditionally address isolated tasks. Transfer Learning attempts to change this by developing methods to transfer knowledge learned in one or more source tasks and use it to improve learning in a related target task. Transfer learning is a machine learning with an additional source of information apart from the standard training data: knowledge from one or more related tasks (Torrey and Shavlik 2009).

Inductive Transfer Learning

According to the definition of Transfer Learning, the Inductive Transfer Learning is based on a transmission of knowledge a source domain and task to target task, however, only the tasks (source and target) are necessarily different (Pan e Yang, 2010) .

In Inductive Transfer Learning, the training domains are known, while the test domains are hidden during training (Torrey and Shavlik 2009). However, the two domains (training and testing) are also necessary to induce a function of intended destination.

Multitask Transfer Learning

The concept of Multitask is right next to the Inductive Transfer. MTL is an inductive transfer method that uses the domain specific information contained in the training signals of related tasks. The difference is that multitasking uses parallel tasks and can learn about various tasks and to transfer knowledge to others in the same environment (Caruana 1997).

MTL is based on the notion that tasks can serve as sources of mutual inductive bias to another task. An inductive bias is, for example, which causes in a student to prefer some cases over/to others; this is a way to select a task in relation to each other.

MTL uses the information contained in the signal training tasks related to it to benefit various tasks. When the training data contains the teaching signal for more than one task, it is easy to note that from the point of view of any task, other tasks may serve as a bias. If this bias exists, the induction should be biased to prefer hypotheses that are useful in various tasks. (Baxter 2000)

MTL aims to develop a machine learning system that is able to retain acquired knowledge to use in the future learning for a lifetime. In our work, we consider similar tasks, but with substantially different characteristics.

Artificial Neural Networks

A neural network is a massively parallel distributed processor that consists of simple processing units, which have the natural propensity for storing experiential knowledge and making it available for use (Haykin 2001). It resembles the brain in two respects:

- knowledge is acquired by the network from its environment through a learning process;
- connection strengths between neurons, known as synaptic weights, are used to store the acquired knowledge.

Neural networks are composed of nodes or units connected by oriented links. A link from unit *j* to unit *i* serves to propagate the activation a*j* from *j* to *i*. Each link also has numeric weight W*j* with *i* associated, which determines the intensity and the sign of the connection. Each unit *i* first calculates a weighted sum of its inputs. Then an activation function *g* is applied in this sum to derive output (Stuart and Peter 2003).

With the network output defined, the output found is compared with a desired output. Thus, the difference between them has an *error signal* or simply an *error* (Haykin 2001).

Algorithms for MTL

The bayesian approach (Krunoslav 2005) is the predictive distribution for the target tasks for a new test case of multitask learning for neural network, given the entry to the event, as well as the outputs and inputs for the training cases (Krunoslav 2005). This distribution is obtained by integration of the prediction model with the posterior distribution of the parameters of the multitask architecture made by neural networks.

This means that bayesian networks attempt to improve the accuracy and/or the speed of formation of some or of all the tasks (Krunoslav 2005).

Lawrence and Platt (Lawrence and Platt 2004) proposed an algorithm known as MT-IVM, which is based on Gaussian Processes (GP), to handle the Multitask Learning case. MT-IVM tries to learn parameters of a Gaussian Process over multiple tasks by sharing the same GP prior.

In this work, we chose another technique: Artificial Neural Networks. Among all the techniques and algorithms mentioned above, the use of neural networks can be noticed, along with MTL, as the most widely used technique, as in its architecture, a neural network can have all their inputs and outputs connected, easing therefore that a task can help other similar task. In addition, among the main items found for Multitask Learning ((Caruana 1997) and (Schrum and Miikkulainen 2011)) using artificial neural networks as the main learning technique.

The concept of Neural Network Learning for Multitask relates to the use of a network with multiple tasks (outputs) that are entirely connected to a hidden layer shared. The

Backpropagation algorithm is execute in parallel with the outputs in the MTL network. With the outputs from a common hidden layer found, it is possible that the internal representations that appear in the hidden layer of a task are used for other tasks. (Baxter 2000)

MTL explores different types of relationships among tasks, but MTL networks do not say how tasks are related. For this reason, Backpropagation networks perform a limited kind of unsupervised learning in the hidden layer of learned characteristics for different tasks (different outputs). The details about unsupervised learning occur and how it works are still not well understood (Baxter 2000).

For example, in a Multitask Neural Network with two modes, each one consisting of two outputs, the network always knows which of the two tasks is performing, and takes the appropriate outputs for each task (Schrum and Miikkulainen 2011).

Electronic Games Involving MTL

The use of Multitask Learning in electronics games is very vague. The study (Schrum and Miikkulainen 2011) demonstrated the use of the technique multitask in games where NPC (Non-Player Character) executing more than one task. This work used to control the MTL of NPC, which differs much of our research. At most there is another great reference which used MTL in electronics games, making it difficult to review what already exists in the area.

In Transfer Learning, find more applications in games. Banerjee and Stone (Banerjee and Stone 2007) presents a reinforcement learning game player that can interact with a General Game Playing system and transfer knowledge in one game to expedite learning in many other games. Sharma (Sharma et al. 2007) present a multi-layered architecture named Case-Based Reinforcement Learner (CARL). It uses a novel combination of Case-Based Reasoning (CBR) and Reinforcement Learning (RL) to achieve transfer while playing against the Game AI across a variety of scenarios in MadRTS™, a commercial Real Time Strategy game.

NEURAL NETWORK MODELING

In this section, we present the games, show in Figure 1, that we use for the tests, describing the main idea of each one. We are also show the modeled neural networks for each type of game. All the networks were trained with backpropagation and the first game modeling was done according to the configuration of a neural network for the game Pong (Macri 2011). The tests were made with various neurons in the hidden layer in order to determine the best configuration for the neural network (number of neurons in the hidden layer).

The Pong Game

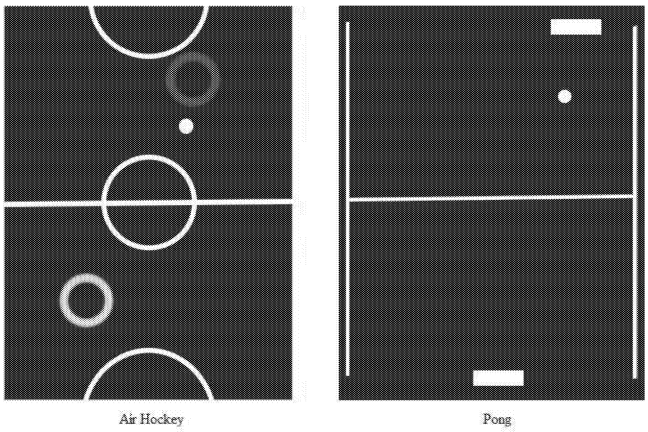


Figure 1: The left image is owned by the Air Hockey game. The right is the classic game Pong.

Pong (marketed as PONG) is one of the earliest arcade video games; it is a tennis sports game featuring simple two-dimensional graphics. While other arcade video games such as Computer Space came before it, Pong was one of the first video games to reach mainstream popularity. Pong quickly became a success and is the first commercially successful video game, which led to the start of the video game industry.

The player controls a paddle (horizontal bar) in the game by moving it horizontally at the bottom of the screen, and competes against the computer or another player that controls a second paddle on the upper side. Players use their paddles to hit the ball and send it to the other side. The ball speed increases each time it is hit and restarts the speed if one of the players does not hit the ball. The objective is to make more points than your opponent, so that the opponent cannot return the ball to the other side (Kent 2001).

Network modeling for Pong

For the game Pong, a network with three, five or ten neurons in the hidden layer was set up, in other words, three neural networks, according to the Figure 2 that presents the configuration for the network with five or more neurons in the hidden layer. In both networks, we use the following inputs, according to the network neural proposal for the same game in (Macri 2011):

- Y Direction of ball, Dy;
- X Direction of ball, Dx;
- X Position of ball, Bx;
- Y Position of ball, By;
- Initial Position x the padle, Pix.

With inputs defined, the neural network is triggered at the instant when the ball leaving the paddle of the opponent in order to calculate the position (Px) of the paddle controlled by the computer.

The Air Hockey Game

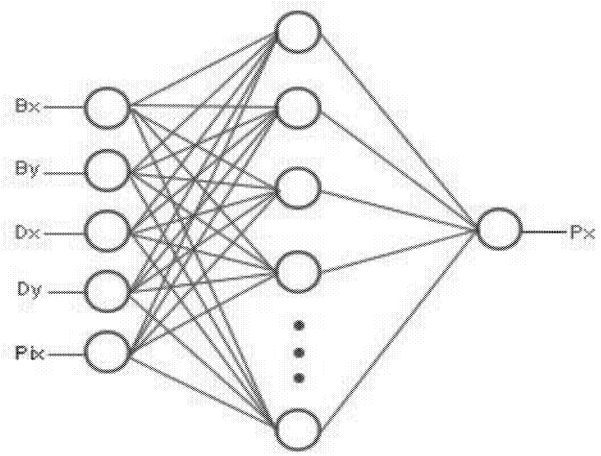


Figure 2: Neural Network for Pong with 5 or more neurons in hidden layer, where *Px* is the final position of the paddle.

Air hockey is a game based on ice hockey and similar to Pong. The player controls a mallet in the game by moving it to all sides, and competes against the computer or another player that controls a second mallet on the opposite side.

Players use their mallets to hit the disc and send it to the other side. The disc speed increases each time it is returned and restarts the speed if one of the players does not hit the disc. The goal is to make more points than your opponent, so that the opponent cannot return the ball to the other side.

The final position shall be to where the paddle should move to collide with the little ball. For this game the increase of speed of the ball is disregarded.

Neural Network modeling for Air Hockey

As in Pong, neural networks are set up for the learning game, however, we modeled the three networks with two outputs each (X and Y position of the paddle), because the paddle in the Air Hockey game have horizontal movement, which is not present in Pong. The hidden layer were formed by three, six or twelve neurons, according to Figure 3, shown with a configuration that allows six or more neurons.

For this game was proposed a network with six inputs:

- X direction of ball, Dx;
- Y direction of ball, Dy;
- X position of ball, Bx;
- Y position of ball, By;
- Initial position of the paddle x, Pix;
- Initial position of the paddle y, Piy.

The use of the speed of the ball was disregarded, as well as the speed of the player's paddle. With inputs defined, the neural network is triggered at the instant when the ball leaving the paddle of the opponent in order to calculate the position (Px and Py) of the paddle controlled by the computer.

Neural Network modeling for Multitask Learning

As in Pong and Air Hockey, three neural networks for learning the game were set up, however, we modeled the

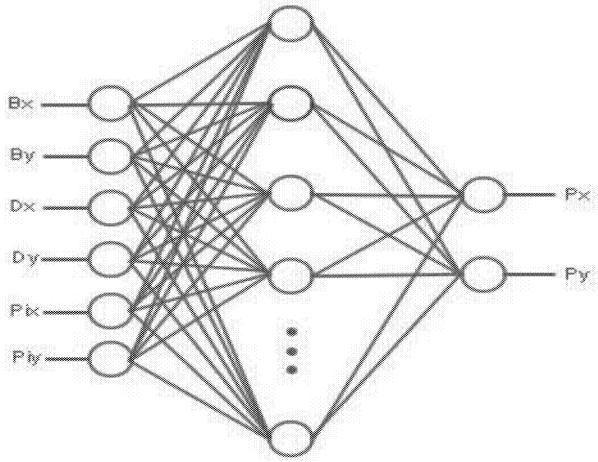


Figure 3: Neural Network for Air Hockey, where *Px* and Py are the final position *x* and *y* directions of the paddle, respectively.

three neural networks with three outputs combined with the Pong and Air Hockey networks (X and Y position of the Air Hockey paddle and X position of the Pong paddle), and the hidden layer is formed by three, six or twelve neurons, as figure 4 (this figure shows a network with six or more neurons).

This network with six inputs (the same six inputs of Air Hockey) executes all tasks of Pong and Air Hockey. Being the difference between the first two networks above only input Piy present in neural network of Air Hockey; we consider a network with the inputs of Air Hockey is able to control the two games.

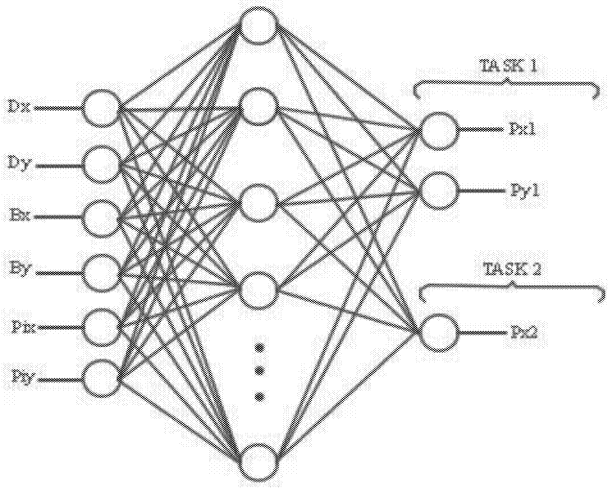


Figure 4: Neural Network for Pong (Task 2) and Air Hockey (Task 1) Games using Multitask Learning. In Task 1, *Px* and Py are the finals positions *x* and *y* directions of the paddle, respectively. In Task 2, *Px* is the final position *x*

The proposed architecture was:

- X direction of ball, Dx;
- Y direction of ball, Dy;
- X position of ball, Bx;
- Y position of ball, By;
- Initial position of the paddle x, Pix;
- Initial position of the paddle y, Piy.

As in the previous two neural networks, the hybrid network is triggered after leaving the paddle ball opponent to calculate the output for both executions Pong as the Air Hockey.

The idea is that the data learned in one of the games may help the other, linking the tasks and generating outputs which satisfy the two sets.

EXPERIMENTS AND RESULTS

This chapter show the results obtained in the three neural networks, in relation to the training of networks with 20 instances for each network. With the results, we compare the outputs of a neural network for each game with the results of hybrid neural network.

In our tests, as in Caruana (Caruana 1997), we define the maximum error as 0.1, as a parameter to stop the implementation of the proposed neural networks. After this limit is reached (or not), keep the number of cycles required to obtain this error. In some cases, where the network has not converged, the limit of the cycles was 10000. The Pong network was modeled and implemented in the C# language, a simple, modern, general-purpose and object-oriented (Ecma International 2012), using the library AForge.NET (AForge.NET 2013), used in works such as Zheng (Zheng et al., 2011), we have analyzed the learning considering twenty instances for the initial training. From this, we create a neural network similar to the Air Hockey with the same number of errors, cycles and instances of the network created for Pong. From these two networks, we implemented the third network modeled with the two games data, with ten instances of each game. For all the networks the learning rate was 0.4 and the teacher was 0.1. For each neural network we create data in a simulated environment, marking and noting the positions, i.e., the information for each entry of the networks necessary for training. For each network, both for the networks of each game, as for the hybrid neural network tested with twenty different instances, with data coming from the reality of a running game.

Three main applications were developed with the implemented networks:

- A neural network to X output of Pong;
- A neural network to the X and Y outputs of the Air Hockey;
- A neural network with all outputs (tasks) for the two games.

Each network has been performed 12 times with the same number of hidden layers in order to find a mean number of cycles. At the end of these 12 executions, only the number of hidden layers of the network has been changed, according to the description of the neural networks modeling, totaling 36 executions.

Pong Experiments

For training the Pong network, the data were collected from drawings similar to the original game in order to provide real positions, according to the proposed model. Therefore, we ensure that the positions found and used in the training instances to be close to reality.

The assumed values for the positions and directions of the *X* axis of the set (in accordance with the modeled network are the inputs *Dx, Bx* and *Pix* and the output *Px*) ranging from 0.0 to 0.6. The values of the *Y* axis (*Dy* and *By*) ranging between 0.05 and 0.95.

For this network, we used a set of 20 instances in order to train it with 3, 5 and 10 neurons in the hidden layer. When we run 12 training in each network, taking into account the maximum error previously described, we recorded the worst, best and mean cycles required for the network learning, as shown in Figure 5.

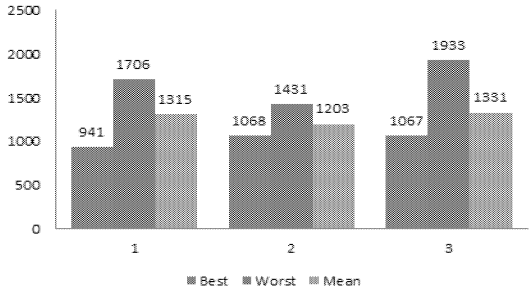


Figure 5: Results of training each neural network for Pong. The vertical axis corresponds to the number of cycles in a network. The index 1 corresponds to the network with three neurons in the hidden layer. The index 2 corresponds to the network with five neurons in the hidden layer. The index 3 corresponds to the network with ten neurons in the hidden layer.

Air Hockey Experiments

Similarly to Pong, for training the network of Air Hockey, tests were made with data collected from similar drawings to the original game in order to provide real positions, according to the proposed model.

The assumed values for the positions and directions of the *X* axis of the set (in accordance with the modeled network are the inputs *Dx, Bx,* and *Pix* and the output *Px*) ranging from 0.0 to 0.6. The values of the *Y* axis (the inputs *Dy, By,* and *Piy* and the output *Py*) ranging between 0.05 and 0.95.

For this network was the same number of instances of Pong (20 instances) with the purpose of training the networks with 3, 6 and 12 neurons in the hidden layer.

When execute 12 training in each network, also considering the errors of 0.1, we note the worst and the best average of cycles required for the learning network, in accordance with Figure 6.

Multitask Learning Experiments

To network with MTL (hybrid), we use 10 instances of the network for Pong and 10 of the network for Air Hockey. As in the other tests executed 12 times each training in order to store the worst, the best and the mean of the results.

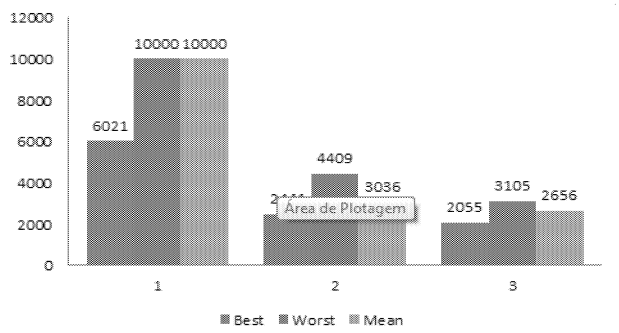


Figure 6: Results of training each neural network for Air Hockey. The vertical axis corresponds to the number of cycles in a network. The index 1 corresponds to the network with three neurons in the hidden layer. The index 2 corresponds to the network with six neurons in the hidden layer. The index 3 corresponds to the network with twelve neurons in the hidden layer. The limit of the cycles is 10000.

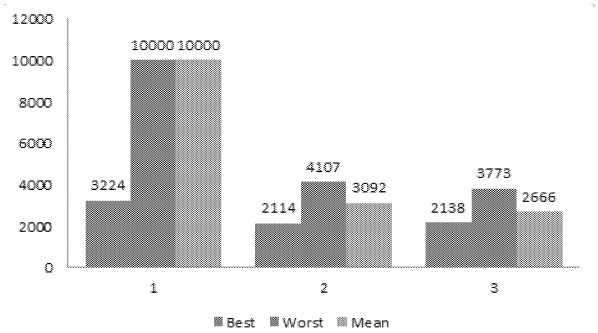


Figure 7: Results of training each neural network with MTL. The vertical axis corresponds to the number of cycles in a network. The index 1 corresponds to the neural network with three neurons in the hidden layer. The index 2 corresponds to the network with six neurons in the hidden layer. The index 3 corresponds to the network with twelve neurons in the hidden layer. The limit of the cycles is 10000.

How the values of the instances used were of known networks, the values *X* also vary from 0.0 to 0.6 and *Y* axis values range from 0.05 to 0.95.

Another detail concerns the inputs and outputs of the network. How instances do not meet all the inputs and outputs of the MTL neural network (the network of Pong has at least input and the two previous networks do not have three outputs), we assume that the value to 0.0 complete data. This was done in order to report that this information will not be used, i.e., 0.0 is a value that overrides an entry in the network, even being used on networks like position.

For example, we can consider that an instance of Air Hockey which uses a variation of the paddle in the *Y* direction, if used as the learning Pong consider the position *Piy* to 0.0 as a way to indicate that this axis does not vary. Similarly, the output of the network with MTL, if used for air hockey, for example, to find an exit 0.0 in the task 2 (see Figure 5 - Pong) should not use this data, i.e., does not influence the learning with this output.

We could notice that, if the level of abstraction is not well defined, the neural network delays to converge. This happens due to negative transfer from one task to another, which will be explained in the next section of this chapter. The error takes into account the implementation of the network satisfies both games at the same time.

The results of executions of hybrid neural network are in Figure 7.

Results with MTL for the games

To find the related tasks and optimize learning between tasks, we can use Peaks Functions (Caruana 1997) on the hybrid neural network. Each function has the following form:

P001 - IF (A> 1/2) THEN B, ELSE C. (1)

…

P006 – IF (F>1/2) THEN A, ELSE B.

Considering the six tasks (P001 .. P006) and the six hidden layers (A .. F), we have the relation among the tasks, defining the most important task within a hidden layer. From this, we have analyzed the defined Peaks Functions (1) to verify if the relation among the hidden sharing unit was positive.

In relation to the set tasks, the learning will be faster and, according to the tests made by Caruana (Caruana 1997) and Schrum and Miikkulainen (Schurman and Miikkulainen 2011), the number of cycles will be shorter.

With the weights initialized randomly, the results presented for MTL were acceptable, however they aren´t better than those found by the Pong neural network. Regarding the network of Air Hockey, some results were better or very similar. However, if we consider that only running a neural network where it is able to satisfy both games at the same time, (test with a network with three neurons in the hidden layer, another with six and another with twelve) present acceptable results, we can say that, even though the number of cycles is greater, the use of this network in the two games is feasible. Thus, we consider that the network only serves as a kind of system able to generate results for more than one game.

Without using levels of abstraction to relate tasks, we can say that, if we use the trained weights considering the importance of a *Z* task to another W task (for example, a task of Pong and another of Hockey) in a neural network with MTL, the result is better than the use of a network for each task and better than the use of a type of network architecture for each game. Thus, we can use a network that serves for both games and, if two users are playing simultaneously, only one network would be able to control the two games.

The ability of a neural network to learn and transfer between related tasks can be proven in Caruana (Caruana 1997) and Schrum and Miikkulainen (Schurman and Miikkulainen 2011). Some proposed mechanisms can help neural networks generalize hidden layers for different tasks, such as the amplification of statistical data, feature

selection, eavesdropping and representation bias (Caruana 1997). From these different mechanisms, in order to find different relationships among tasks, we should find how the tasks are related. In backpropagation nets, firstly, we use a supervised learning, performing a limited type of unsupervised learning on the characteristics of the hidden layers learned from different tasks. The problem is how this unsupervised learning happens and how it works, which are not well known (Caruana 1997). A network with MTL always knows which tasks are performing and chooses the outputs appropriately (Schurman and Miikkulainen 2011).

DISCUSSIONS

In this chapter, we present some issues on the evolution of research in relation to other ways of using MTL, along with the advantages and other points that need to be cited as a result of the use of the TL.

Evolution of Search Applied in Games

The search may be a step to show that the deeper in this area may be possible to use MTL in more complex problems, for example, all games of the same mechanics. For example: the games MOBA (Multiplayer Online Battle Arena)(Mittani 2013), also known as action real-time strategy (ARTS), is a sub-genre of the real-time strategy (RTS) genre, in which two teams often do players compete with each other in discrete games, with each player controlling a single character through an RTS-style interface. The games have a famous style, where a single AI could include them all, reducing development time for their own AI that are relatively similar, save time testing of AI, and even training could be reduced, taking into account that now there would be a previous learning through the transfer of learning. But of course it there is a charge, a default of for use of an engine (proposed in the Conclusion topic after) able to control multiple games. Specifically MOBA style games there several similar parameters to be used:

- positions of the mini map;
- routes more security;
- strategies gank;
- to build better items.

To build Bots (Virtual Robots) great, is very complicated, but with the thought that in the future this research can arrive at engine can, for example, through these parameters shown above, provide an intelligence Bots great not only alone but also strategies between them of the same team. The research provides a vision of a future in the sense of artificial intelligence to games platforms, if we deepen this area little studied.
Despite the games being simple, as well as modeling, precisely because of the lack of similar work, possible upgrades can be easily detached. The use of more inputs and outputs of the neural network, for both games, for others applications, with a model that uses more related tasks and tasks distinct, like a sports game for transferring an RTS game.

Advantages of Transfer Learning in Games

According to (Caruana 1997), a task can benefit from the information contained in the knowledge of other tasks trained. This happens due to the training tasks in parallel. The complete detail of what is being learned for all jobs available for all, because all tasks are being learned at the same time. The tasks usually benefit each other, what a linear sequence cannot capture. For example, if a task is learned sequentially before the second task, the second task cannot help task 1. This not only reduces performance on a task, but also can reduce the ability of a task to help task 2.

For tests performed in games industry, with a game learning from other tasks, reduce cost of software testing. This is because if a game with another learns that has been tested, it becomes easier to analyze the same. The only test to be done is to transfer between tasks reached a level considered optimal learning among them.

Another gain is the evolution of the AI opponents in games. Considering a game that has a great AI and other game use the knowledge of some of the first tasks, it is concluded that this new game AI is very good. This because each task only learn the necessary. Also, note that the leveling between games is better. With games transferring knowledge between games, the trend is occurring between the leveling difficulties between them.

Negative Transfer among tasks

We have to pay special attention to negative transfer of multitask learning that occurs when knowledge of a task contributes to a distortion of another task or, at worst, makes the task regress, instead of providing an evolution of the level of learning. Therefore, an important part of multitask learning is the modeling of levels of abstraction of the tasks, in other words, an input that transfers a knowledge negatively may have its level of abstraction reduced (Pan and Yang 2010).

According to the proposed model, levels of weights abstraction were not used, due to the difficulty found in the implementation of this technique, resulting in a negative transfer by increasing the number of cycles. How the tasks are similar, the increase was not high, as can be seen in Figure 7.

One way to resolve the negative transfer is to adapt the weights according to their importance, optimizing the result as mentioned in the previous index. Another way of softening is to apply multitask learning in similar tasks.

CONCLUSIONS AND FUTURE WORKS

With the tests and results obtained, the transfer among similar tasks is possible and reduces the cost for some tasks that are executed in parallel. Thus, a Neural Network covering various tasks can be used in several similar games, creating a single engine.

In the proposed work we have obtained relevant results, but we have also found some difficulties that will be studied to be resolved. The first is about levels of abstraction in transfer learning for games, seeking a formula or an ideal weight to use for each input, which would reduce the negative transfer degree. There is also another step about the modeling of multitask learning with neural networks in games, underscoring an efficient and simple way of modeling a network. Due its complexity, modeling is very difficult to be built, taking into view that is a generic way of several neural networks.

The idea that many neural network related can run many games is satisfactory and we have the construction of an engine of platform in mind: The Onemass Engine, which serves for all types of games, using distinct transfer, where the neural network will work with different tasks. Thus, the engine will perform all roles of AI without each game having its own algorithms, in other words, a global and generic engine that could transfer learning from one game to another.

As this distant transfer happens, more techniques of Psychology will be employed in Artificial Intelligence, in order to approximate increasingly the machines of human reasoning, taking into view that it is a branch that studies the man: a perfect cognitive system. With this, we will be closer, through an application in the area of electronic games, to reaching a field that still hasn´t been achieved in Computer Science: the Strong AI (Stuart and Peter 2003).

ACKNOWLEDGEMENTS

The authors acknowledge the Empresa Júnior da Computação (EMCOMP) and 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 for the financial support guaranteed for the accomplishment of this work.

REFERENCES

Banerjee B. and P. Stone. 2007. “General Game Learning using Knowledge Transfer”. Department of Computer Sciences, The University of Texas at Austin; Austin, TX 78712.

Baxter, J. 2000. “A Model of Inductive Bias Learning”, Journal of Artificial Intelligence Research.

Caruana, R. 1997. “Multitask Learning”, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213. 1997, Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

C# Language Specification (4th ed.). 2012. Ecma International. June 2006. Retrieved January 26.

Haykin, S. S. 2001. “Redes Neurais”. 2-ed, pg 28.

Hergenhahn, B.R. and M. H. Olson. 2005. “An Introduction to the Theories of Learning”, Pearson Education, ISBN 978-81-317-2056-1.

Kent, S.. 2001 “Ultimate History of Video Games”. [S.l.]: Three Rivers Press. 40–43 p. ISBN 0761536434

Krunoslav, K. 2005. “Multitask Learning for Bayesian Neural Networks”, Graduate Department of Computer Science University of Toronto.

Lawrence, N. and J. Platt. 2004. “Learning to learn with the informative vector machine,” in Proceedings of the twenty-first international conference on Machine learning. ACM, p. 65.

Macri, D. 2011. “An Introduction to Neural Networks with an Application to Games”, Intel Corporation.

Pan, S. J. and Q. Yang. 2010. “A Survey on Transfer Learning”, IEEE Transactions on Knowledge and Data Engineering, Vol.22, No.10, October.

Perkins, D. N. and G. Salomon. 1992, “Transfer of Learning”, Contribuition for International Encyclopedia of Education, Second Edition. Oxford, England: Pergamon Press.

Schrum, J. and R. Miikkulainen. 2011. “Evolving Multimodal Networks for Multitask Games”, Department of Computer Science, University of Texas at Austin, Austin, TX-78712-USA.

Sharma M., M. Holmes, J. Santamaria, A. Irani, C. Isbell and A. Ram. 2007. “Transfer Learning in Real-Time Strategy Games Using Hybrid CBR/RL”. College of Computing, Georgia Institute of Technology. IJCAI 2007.

Stuart, R. and N. Peter. 2003. “Artificial Intelligence: A Modern Approach”, 2ª ed. Upper Saddle River, New Jersey: Prentice Hall.

Torrey, L. and J. Shavlik. 2009. “Transfer Learning”, Appears in the Handbook of Research on Machine Learning Applications, published by IGI Global.

Zheng D., Y. Wang, Y. Deng, A. Yu and W. Li. 2011. “A Software Framework for Optimization of Process Parameters in Material Production”. Applied Mechanics and Materials (Volume. 101 - 102).

WEB REFERENCES

Library AForge.NET - Computer Vision, Artificial Intelligence, Robotics www.aforgenet.com.

"Dota 2: A History Lesson". The Mittani. 22 July 2013. Retrieved 24 August 2013. http://themittani.com.

BIOGRAPHIES

ALEXANDRE DE C. LUNARDI 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).

RANIEL F. CORREA 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).

ESTEBAN W. G. CLUA is Professor at Universidade Federal Fluminense and general coordinator of the UFF Medialab. He is PhD in Informatics from PUC-Rio. He is one of the founders of SBGames, SBC, Director of the IGDA Rio Academy, responsible for research and academy in the area of digital entertainment in the country and a pioneer in the field of scientific research in games and digital entertainment.

IN-GAME CHARACTERS

CHARACTER GENERATION USING INTERACTIVE GENETIC ALGORITHM

[1]Umair Azfar Khan and [2]Yoshihiro Okada
[1]Graduate School of Information Science and Electrical Engineering,
[2]Kyushu University Library,
Kyushu University, Fukuoka, Japan
umairazfar@gmail.com

KEYWORDS:
Character creation, genetic algorithm, evolution, story.

ABSTRACT

A story is a collection of characters where they interact with each other and perform different actions. The actions done by the characters govern their orientation towards being good, bad or impartial. It is important that the correct character is placed in the proper role to make the story believable. In a general Role-Playing Game, the story and characters remain the same which reduces the replay-ability of the game. We need a system by which the creation of these characters must adhere to the roles as given in the story. Interactive Genetic Algorithms are used to define the orientation of the characters and also the physical appearance. As a result, the roles in a story are filled by characters that fit a certain physical and mental criteria. With time, the generated characters start to adhere to the choices made by the users while keeping a certain degree of randomness to keep the stories fresh.

INTRODUCTION

When we talk about Digital Interactive Storytelling, we are normally concerned with the creation of a believable narration for a computer generated story. However, this research is centred around the creation of believable animated stories that are dependent on the characters and the actions performed by them.

In our past research (Khan and Okada 2012) we used a user's vocal input for the generation of environment and objects. With our system, we hoped to generate a believable story in real-time; but in order to have an effective story, we need to have characters that leave an impression on the user and at the same time, we want the user to play an active role while going through the story. One of the ways of doing this is to allow character customization, which is quite common in games these days.

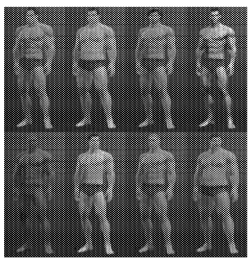


Figure 1: Characters that can be created in the famous game Saints Row: The Third

Many games allow the players to construct an avatar through a complicated menu. The users are able to create very detailed avatars using the menu driven system examples of which can be shown in Figure 1, where different types of avatars have been created by using the menu driven system of the popular game Saints Row: The Third.

Characters in a story must always show something similar to the user's imagination to be believable. A good character performing evil actions and similarly a physically weak character cannot perform acts that require immense physical strength.

Egri (1960) describes a character as a sum of physiological, sociological, and psychological qualities. Based upon this he describes a checklist of various aspects that will influence behaviour of the character which are as follows:

- Physiology (e.g., gender, age, height, weight, appearance, distinct, and physique);
- Psychology (e.g., moral standards, goals, obsessions, intelligence);
- Sociology (e.g., occupation, education, enemies, family life, friends, and hobbies).

For character generation, we make use of Interactive Genetic Algorithm with elements of a Knapsack problem to define the *Psychology* of the character. We also use another Genetic Algorithm with sigma selection to define the *Physiology* of the character.

RELATED WORK

There has been research in creating characters in different applications. Some research has only been towards defining the physical characteristics, while other has been towards creating intelligent agents. (Zavala 2010) has used Genetic Algorithms for creating a heterogeneous humanoid crowd population but it is limited to creating humanoids with physical characteristics. The Distance Function has been used to find best fit individuals. In the current research, for the definition of the physical characteristics of the individuals we use a similar system, but rely on the sigma selection to choose a character's physical attributes.

Early work by (Fujiwara and Sawai 1999) demonstrates a method in which a 3D surface with simple facial features is created from a 2D surface by breeding geometry.

(Albin and Howard 2009) present an approach which exploits the power of Evolutionary Algorithms (EAs), and

provide illustrative examples of how their methods may be realised within the context of surface-based model geometry in order to create a complete avatar.

(Ventrella 2000) who is also the chief author of the patent: Method and apparatus for creating and customizing avatars using genetic paradigm refers to a genetic approach to avatar design and customisation and how it is possible to breed using sexual reproduction.

(Fairclough 2004) follows the approach of having an omniscient agent that is present in a simulated environment and influences the characters that populate it. The OPIATE system dynamically casts characters into eight of the nine available roles, with the ninth role taken up by the user. The nine roles fall into the following categories as defined by (Propp 1968): Hero, Villain, Mediator, Donor, Helper, False Hero, Princess, King and Family. This casting is based on the actions done by the characters on each other. As a result the allocation of roles is dynamic.

(Lankoski and Björk 2008) propose a character-driven game design method, which uses ideas from dramatic character design to include gameplay into the design process. Possible and Impossible actions are given to characters in order to achieve their goals which affect the overall gameplay.

(MacNamee 2004) has proposed creating a reactive environment for the characters where it creates new paths at the spur of the moment. Such an approach might be useful when user interaction comes into play. If the user chooses to change the attributes of a character, the reaction might end up changing the entire story or maybe parts of it.

CHARACTER GENERATION

The complete life cycle of character generation involves the allocation of the mental and physical attributes using Genetic Algorithm. The user involvement is necessary for the final selection of the character. The first time the story is created, the characters will be randomly generated and the user will have the option to manipulate them. After the user completes the story and restarts the game, the characters that were selected the last time will be used as parents for the next generation. This character creation process is shown in Figure 2:

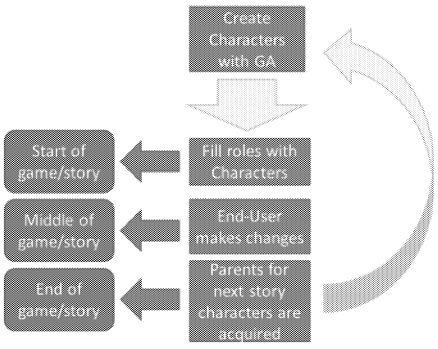


Figure 2: Complete Characters Creation Cycle

(Lebowitz 1984) says that coming up with an optimal combination of traits is difficult and uses a heuristic approach which works by first selecting a stereotype that has the minimal total discrepancy from the traits specified. Then another stereotype is picked which brings the values of the traits closer to the desired combination. Although this approach is quite sound but it is exhaustive and needs refinement which we believe can be achieved by Genetic Algorithm.

Our system requires two different Genetic Algorithm implementations to completely define a character physically and mentally. In this section, we will first define a Simple Genetic Algorithm and then introduce our implementation of the Genetic Algorithms for defining a character's orientation and physical representation.

Genetic Algorithm

The Genetic Algorithm is a technique used to find a close approximate to the optimal solutions. Genetic Algorithms have been used for solving problems where finding the optimal solution can take a long time to be calculated. Genetic Algorithms are able to provide optimal or near-optimal solutions under a wide range of selection pressure (Goldberg 1993). The convergence of the Genetic Algorithm is dependent on the selection pressure which is the degree to which the better individuals are favoured. If the selection pressure is too low, the convergence rate will be slow and the Genetic Algorithm will take unnecessarily long to reach an optimal solution (Miller and Goldberg 1995).

A Genetic Algorithm is a two-step process: On the original population, selection is applied to create a new population, in which some of the members are repeated, thus creating the intermediate population. Then crossover and mutation are applied to the intermediate population to create the next population. The process of going from the current population to the next population constitutes one generation in the execution of a genetic algorithm. (Goldberg 1989) refers to this basic implementation as a Simple Genetic Algorithm (SGA).

For our character creation process, the acquired character must display some degree of randomness. GA aims to optimise the solution by repeated iterations. In order to create compelling stories that adhere to a certain norm, we do not need characters that are optimal as will be explained shortly. A character that loosely fits the requirement of a role is good enough for a story. As a result, we are searching for a solution that can be acquired by just looking at the local maxima/minima and we do not need to further optimize the results. For this reason, we are using Elite selection for both of our Genetic Algorithm implementations. Elite selection quickly converges to a solution which reduces the number of iterations in a GA. Since we also want to keep the solutions a bit random, elite selection provides a fair amount of randomness to keep things fresh.

Genetic Algorithm with Knapsack Problem

According to (Propp 1968) the number of actions that a character can perform needs to be limited. Based on the complexity of the story, this number can vary and it is best

left to the story creator to decide as to what this number should be.

A character is a collection of limited number of actions that a person can perform. There have been people who have performed extraordinary feats under extreme circumstances, but there have also been many people who were unable to perform even under the most favorable of conditions. Since actions speak louder than words, we concluded that the orientation of a character towards good or bad is dependent on the actions that a person can perform. For this reason, we defined a number of actions that can be taken to be good or bad and associated a numerical value to those actions (Khan and Okada 2012).

For our research purposes, we kept the total number of actions to be 20 although this number can change based on the complexity of a story. These 20 actions are distributed into 10 good actions and 10 bad actions. For our Genetic Algorithm a chromosome can contain a total of 10 actions, which is a combination of good and bad actions. We have taken a slight detour from the way the chromosomes of a Genetic Algorithm are made. For our chromosomes, the order in which the actions appear holds no importance. What actions it does contain is what determines the fitness of the chromosome in the population.

The genes which represent actions are numbered from 0 to N-1, where N is the total number of genes, which in this case are 20. Genes from 0 – 9 are taken to be good actions and the genes from 10 – 19 are taken to be bad actions. The numbers reflect the intensity of the action. The lower numbers represent good actions and the higher numbers represent bad actions. The sum of these actions decides whether the character is a good character or a bad character. For example, a good character that has all the good actions will have a combined gene sum of 45. Any randomly created character that has a sum equal to or closer to this value can be taken as a good character. This is where the Knapsack Problem comes in as we want to find the right collection of genes that produces the closest solution.

Based on the requirement of characters within a story, the different characters will have different gene sum and thus have different orientations. As a result, the number of ways that sum can be achieved is increased with the number of genes present in the gene pool.

As shown in Figure 3, the genetic algorithm is being used to find a chromosome, the sum of whose actions is roughly equal to a certain value.

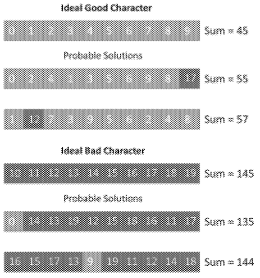


Figure 3: Probable Solutions through GA

Thus, from a gene pool of N actions, chromosomes are created which have N–N/2 actions each as genes. According to the requirement of the characters, the GA tries to find the chromosomes that are closer to the solution. After every crossover, it is checked if a gene has been repeated or not. If there is a repetition, the repeated gene is replaced by another gene from the gene pool. For mutation, a random gene is picked from the chromosome and it is replaced by another gene from the gene pool. By following this approach repeatedly, we arrive at a chromosome whose sum of actions is closest to the desired character. This process is best described in Figure 4.

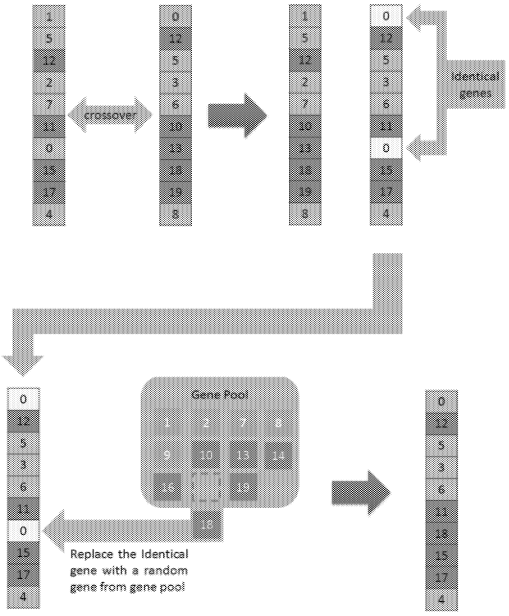


Figure 4: Complete Crossover Operation

The fitness function that is being used for the GA to work is quite simple at the moment. Even though it is producing good results, we will be looking into better ways to achieve these results as this only covers the initial generation and selection of characters when a story is started. There are two more constraints that we have placed on this system. Firstly, we decided that every action must have a different numerical value. Secondly we have not created any relationships between the actions themselves. Normally it is the combination of actions that lead a person to perform a third action. So a new action can only be unlocked if a person has the propensity of performing the first two actions. Such details would indeed make the character selection process much reformed but it will also add more complexity to the overall system.

Genetic Algorithm with Sigma Selection

For the physical makeup of the characters, we need to make a selection that is based on user choice. Since we want the user to have a wide variety of physical attributes to choose from, the number of combinations that can be created are enormous. As a result, the usage of Genetic Algorithm seemed logical as it helps in converging to a solution quickly when the number of combinations is great.

For our research purposes, we are using the Unreal Development Toolkit (UDK). UDK is already being used for a large number of educational projects and since it is a game development tool, it seemed to be the best choice for developing our game/story generation system. UDK provides the creation of modular character objects, called modular pawns. The 3D character is created by joining parts together which are then governed by a skeleton. As a result, we can place any model on top of the skeleton and it will move according to the bone structure of the skeleton. This can best be explained by Figure 5:

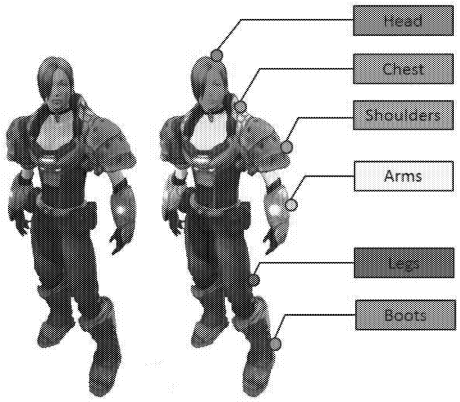


Figure 5: Modular Pawn

Each module represents one important part of the body. User selection will play a major role in this selection. As the user selections will start to fall within a certain range after a while. The mean of their choices will represent the optimal solution while the standard deviation will represent the range in which the selections were made. We need to choose a model that is close to the mean and falls within the standard deviation range. Any model within this range is acceptable as we are not trying to find the optimal candidate. To make this possible, we will couple the elite selection with sigma selection to retrieve our candidate. Sigma selection will be used to find the acceptable model within a module and elite selection will be used to find the candidate whose modules deviate the least bit from the optimal solution.

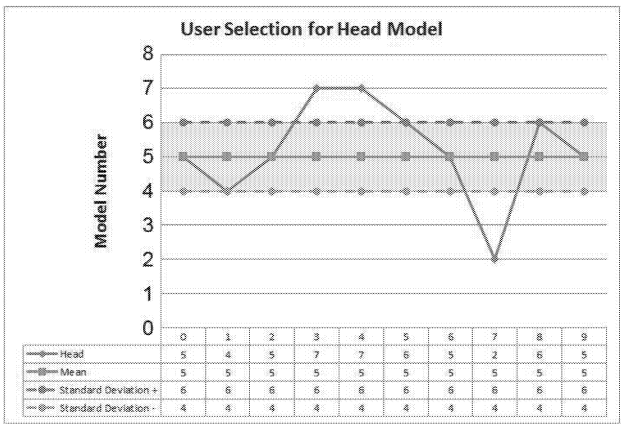


Figure 6: User Selection for Head Module

In Figure 6, we see that among a range of 10 models, the users have made their selections. If the algorithm returns the number 5, it will be the most optimal candidate among all the models. However, the model number 4 and 6 are equally good as they fall within the standard deviation range. The same process will be applied to rest of the modules to get all the acceptable model numbers to define a character. Hence, the chromosome that is created will contain all the model numbers that fall within the standard deviation range. The best chromosome in the population will be the one whose modules deviate the least bit from the mean values.

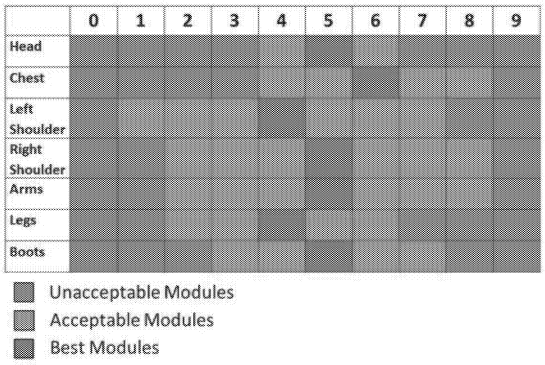


Figure 7: Module Selection

In Figure 7, we can see that the model numbers that fall within the green range are accepted to be part of a chromosome. The farther they are from the "blue" values, the more unfit they are. The chromosome whose modules are the closest to the blue values is selected as the best chromosome in the selection. These figures only show 7 modules with only 10 models each to choose from; this is only for demonstration purposes. We have used this method on 10 modules with 50 models to choose from which increases the total number of possible combinations to 50^{10}. There can be any number of modules and their respective models.

IMPLEMENTATION

Unreal Engine 3.0 has already been used in the creation of several AAA titles such as Gears of War, Batman' Arkham Asylum and Unreal 3. Unreal Development Kit (UDK) is free to use for any educational software and was therefore an ideal choice for creating our system.

All the DLLs that contain the code for Genetic Algorithms are created using C++ in Visual Studio 2010 and are used by the UDK using unrealscript. The user interface is created using Autodesk® Scaleform® which leverages the power of the Adobe® Flash® tool set. Whenever the game is started, the GAs check the past characters that were present in the previously played game and uses them as parents for the next population. With this information, a new character is created that is closer to the choices made by the users previously. This can be best shown in Figure 8, where a character and its attributes are being displayed.

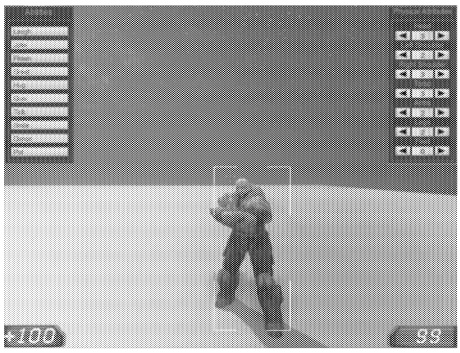


Figure 8: Character Creation Interface

As the user brings the crosshair on the character, its mental abilities are displayed in the 'Abilities' menu to the left and its physical characteristics are displayed in the 'Physical Abilities' menu to the right. Both the menus are interactive and if the user does not like any of the characteristics, they can be changed by clicking on the menu.

RESULTS

A total of eight students in our lab tested the system and then filled out a questionnaire. The questionnaire asked them about the utility of the user interface as well as their preferences when it comes to playing Role Playing Games (RPGs). Most of the users had previously played RPGs among other types of games. It was surprising to note that the genre of choice was Action RPGs rather than Turn Based RPGs since the genre of choice among the Japanese users has remained Turn Based RPGs for a long time. This might be a changing trend and it points towards making the game more action oriented.

Almost every user wanted the option to customize their own avatar and they also wanted to view their own character as they played the game. They also thought that the number of body parts that could be manipulated was not enough and they would like the option to customize them further. This is an indication that character customization plays a large part in a game and story narration. People like to put effort in the creation of a character and the also want to view the created character while playing the game. The users were however divided when it came to selecting the Non-Playable Characters (NPCs) in a game.

Lastly, everyone agreed that the more customization we have in an RPG the more enjoyable the experience would be.

CONCLUSION

In order to define the characters of a story in an RPG, we have made use of specialized Genetic Algorithms. The creation of the characters is a two-step process, the first being the allocation of mental while the second being the definition of physical attributes for defining the character's physical appearance. The created character represents the user's thought process as the physiology and psychology is based on the choices that the user made the last time the system was used.

REFERENCES

Albin-Clark, A., and Howard, T. 2009. Automatically generating virtual humans using evolutionary algorithms. In TPCG, Eurographics Association, W. Tang and J. P. Collomosse, Eds., 61–64.

Egri, L. The Art of Dramatic Writing. Simon &
Schuster, New York. 1960.

Fairclough, C., Story Games and the OPIATE System, Ph.D. dissertation, University of Dublin - Trinity College, 2004.

Fujiwara, Y. and Sawai, H., Evolutionary computation applied to mesh optimization for a 3-D facial image, IEEE Transactions on Evolutionary Computation, Vol. 3 No. 2, 1999.

Goldberg, D. "*Genetic Algorithms in Search, Optimization and Machine learning"*. Reading, M A: Addison-Wesley, 1989.

Goldberg, D. E., Deb, K., & Thierens, D. Toward a better understanding of missing in genetic algorithms. *Journal of the Society of Instrument and Control Engineers, 32(1)*, 10-16, 1993.

Miller, B. L. and Goldberg, D. E. Genetic algorithms, tournament selection, and the effects of noise. IlliGAL Report 95006, Illinois Genetic Algorithms Laboratory, Department of General Engineering, University of Illinois, Urbana-Champaign, 117 Transportation Building, 104 South Mathews Avenue, Urbana, IL 61801, USA, July 1995.

Khan. U. A, Okada, Y: 3D Terrain Generation and Texture Manipulation by Voice Input, Proc. of GameOn-Asia 2012, pp. 71-75, 2012.

Khan. U. A, Okada, Y: Evolving story and character generation for role-playing games, Proc of the workshop at SIGGRAPH ASIA 2012, pp 59-64, Nov, 26-27, 2012.

Lankoski, P., Björk, S. Character-Driven Game Design: Characters, Conflict, and Gameplay. In GDTW 2008 Conference Proceedings, Liverpool, UK, 2008.

Lebowitz, M., Creating characters in a story-telling universe, Poetics, Volume 13, Issue 3, Pages 171-194, June 1984.

MacNamee, B., Proactive Persistent Agents: Using Situational Intelligence to Create Support Characters in Character-centric computer games, PhD. Thesis, Computer Science Sept., TCD 2004.

Miller, B. L. and David E. Goldberg. Genetic algorithms, tournament selection, and the effects of noise. IlliGAL Report 95006, Illinois Genetic Algorithms Laboratory, Department of General Engineering, University of Illinois, Urbana-Champaign, 117 Transportation Building, 104 South Mathews Avenue, Urbana, IL 61801, USA, July 1995.

Propp, V. Morphology of Folktale. University of Texas Press, 1968.

Ventrella, J., Avatar Physics and Genetics. In Proceedings of the Second International Conference on Virtual Worlds (VW '00), Jean-Claude Heudin (Ed.). Springer-Verlag, London, UK, UK, 107-118, 2000.

Ventrella J. J., Harvey W. D., Werther B. M.: US patent 6545682: Method and apparatus for creating and customizing avatars using genetic paradigm, April 2003.

Zavala, M., Luga, H., Larios, V., "*Virtual Reality Advanced Editor For Crowded Worlds Simulations*". International Conference on Computer Graphics and Artificial Intelligence, Athens Greece, 28-29 May 2010 (3IA 2010).

AVATARS AND THE "IMITATION GAME" - CAN MACHINES SMILE?

Helena Barbas
CESNOVA - http://cesnova.fcsh.unl.pt/
Faculdade de Ciências Sociais e Humanas – Universidade Nova de Lisboa
Av. de Berna, 26-C
1069-061 Lisboa - Portugal
E-mail: hebarbas@fcsh.unl.pt

KEYWORDS
Digital Humanities, character studies, avatars, facial animation, Turing

ABSTRACT

Considering Alan Turing's challenge in «Computing Machinery and Intelligence» (1950) – can machines play the «imitation game»? – it is proposed that the requirements of the Turing test are already implicitly being used for checking the credibility of virtual characters and avatars. Like characters, Avatars aim to visually express emotions (the exterior signs of the existence of feeling) and its creators have to resort to emotion codes. Traditional arts have profusely contributed for this field and, together with the science of anatomy, shaped the grounds for current Facial Action Coding System (FACS) and their databases. However, FACS researchers have to improve their «instruction tables» so that the machines will be able, in a near future, to be programmed to carry out the operation of recognizing human expressions (face and body) and classify them adequately. For the moment, the reproductions have to resort to the copy of real life expressions, and the present smile of avatars comes from mirroring their human users.

INTRODUCTION

The Turing test, being the final verisimilitude barrier in the humanization of machines, is already being used implicitly as an assessment rule when it comes to visual representations of human realistic figures, and credibility of virtual characters and avatars. These latter are already being evaluated in terms of the «imitation game». But human emotion's virtual representations and their databases are still in an embryonic state. New software and larger databases, complete enough, together with a global cooperation, will be needed; and new developments are in order to reach the goal of creating the best avatars and synthetic characters, and make them express emotions in a way that can engage (or fool) human users. In order to do so, the relationship between physical external expression and inner feeling will have to be better cleared and explained.

Trying to systematize the larger picture of the issues relating to virtual characters – Avatars, personas – this paper will focus on some of the turning points in a path that is far from complete or faultless. It will address only non-invasive studies, and try to explore the state of the art concerning external face movements as metaphors for feelings (mechanical/electrical research as Duchenne's, brain studies and invasive procedures will be addressed in future work).

The word Avatar is new, but it has been invested with the meaning of its other synonyms; the theorizations on character can superimpose on it, and contribute to enrich a future avatar taxonomy, and vice-versa. Being the fate of Avatars to visually express emotions – the exterior signs of the existence of a conscience, a soul, a feeling – its creators will have to resort to an emotion code. The traditional arts (painting, sculpture, and mainly theatre) have profusely contributed with studies on human emotions representation. Together with the science of anatomy, they have shaped the foundations for the current Facial Action Coding System (FACS) and their databases. These could avail themselves from other research areas, i.e., as Forensic Anthropology, to create a global human expression all-inclusive catalogue.

The last section will focus on how, at this moment, Avatars are expressing emotions. Mainly their creators resort to facial recognition tools, and real life capture of individual features. Some new techniques, like 3D scans, are contributing to a better quality in emotion representation, and allowing the exploration of close-up practices, but are still relying on all kind of human actors to produce a smile.

THE IMITATION GAME

Alan Turing, in his ground-breaking paper «Computing Machinery and Intelligence» (Turing 1950), opens up with a provocative question: «Can Machines Think?»; and immediately reformulates it into a wider and more inclusive problem: can machines play the «imitation game»? The «imitation game» became known as the «Turing Test»: it experiments a computer's ability to exhibit intelligent behaviour equivalent to, or indistinguishable from, that of a human. This year's *Loebner Prize*, in honour of Alan Turing [http://www.paulmckevitt.com/loebner2013/], was proof that machines cannot yet fool judges into thinking that they were speaking with a real life person. The Turing test is aimed only at written conversations, but it is tacitly in mind as an assessment rule when it comes to visual presentations of human beings, and to verisimilitude of virtual characters and avatars. These too are being evaluated in terms of the «imitation game».

Turing did not address the question of emotion representation, but there are two moments where his vision anticipates some of the problems being faced now. One has to do with the needlessness to emulate the human: «No

engineer or chemist claims to be able to produce a material which is indistinguishable from the human skin. It is possible that at some time this might be done, but even supposing this invention available we should feel there was little point in trying to make a "thinking machine" more human by dressing it up in such artificial flesh.» (p.2); the second shows the procedure for any attempt to replicate human behaviour: «If one wants to make a machine mimic the behaviour of the human computer [man as a machine] in some complex operation one has to ask him how it is done, and then translate the answer into the form of an instruction table. Constructing instruction tables is usually described as "programming". To "programme a machine to carry out the operation A" means to put the appropriate instruction table into the machine so that it will do A.» (p.5). The obvious conclusion is that, with enough storage capacity [available now], all machines can emulate whatever the programmer is able to design. And this means not only to create software and databases large and complete enough to enfold all possible human emotion representations, as well as, with time, to create the best avatars and synthetic characters, and make them express emotions in a way that can engage (or fool) human users.

AVATAR, CHARACTER, PERSONA

The word "avatar" has its roots in the Sanskrit "avatarana" (descent), in Hinduism referring to a bodily manifestation of a deity when coming to Earth in order to aid humans. Its original meaning has slowly downgraded. In his book *Biographical and Critical Essays* (1856) the poet David Masson connected the word Avatar with the notion of "ethos" alluding to the 'Scottish Influence in British Literature' (Masson 1856); in *The Life of John Milton* (Masson 1859), the term reoccurs at least four times, the last one referring to a person: «As we regard Spenser's genius […] we must be glad that the avatar of Donne […] was so brief and partial».

The first appearance of the term Avatar denominating a digital human representation showed up in the 1985 computer role-playing game *Ultima IV: Quest of the Avatar* (Apple II). In this game users tried to achieve the honorific designation of Avatar in the world of Sosaria, which implied conquering "eight virtues". From *Ultima V: Warriors of Destiny!* onwards, it became a synonym for the user's digital persona. In primitive gaming, the term was also used in *Habitat* (Lucasfilm, Ltd./LucasArts), a first large scale commercial multi-user RPG (Commodore 64). The players were represented by Avatars:



Figure 1: *Ultima IV: Quest of the Avatar* (1985)

In 1992 Neal Stephenson recasted the term in his science fiction novel, *Snow Crash*. In it computer users were able to enter a virtual 3D world (Metaverse) where each was represented by an avatar, a human-like simulation made up of only pixels.

Commonly defining a computer (animated) character representing a real life person, a concept or an artificial entity, the term avatar became interchangeable with the older notions of mask, persona and character. Character classification is an historical as well as theoretical problem, and so will be any avatar taxonomy. Meanwhile, whatever the theory, they have in common the fact that character is defined either from an exterior standpoint (action, drama, gestures, behaviour) or from an internal/psychological perspective (language, emotions, stream of consciousness, *qualia*). In Aristotle's *Poetics* (1447.a), character is defined by its own actions, and the meaning of action is "drama": the outcome of the movement of a body in space and time. Initially associated with theatre, it enriches itself with new considerations when entering the fields of interactive narrative, multimedia and games. The main objective of this action was (and still is) to awaken emotions in the audience; its main goal to encourage (a then medicinal) *catharsis*. The situations presented on stage sought to create empathy between the heroes and the common man; the latter could safely, from his spectator seat, experience the tragic events. Other requisite pervading all character studies is the need for verisimilitude: to win at the «imitation game».

Posterior studies can be summarized in E. M. Forster interpretation of Henry James literary terminology: characters can be "flat" and/or "round"; and he also emphasizes that these are not real people: «they are like real people. Characters' lives are different from real lives, and common activities such as sleeping and eating occupy little space in novels…» (Forster 2005) – or, for what matters, in virtual worlds and games.

EARLY EMOTIONAL CODES

In opposition to metaphors and *qualia* in Literature, to visually represent emotions – the exterior signs of the existence of a conscience, a soul, feelings – any creator has to resort to a code. In Classical Rhetoric the problem was solved because Poetics provided clear instructions for this purpose. In medieval drama characters were reduced to types (the villain, the hero) who also obeyed quite strict (non-written) rules (the colour of clothing, how to enter the stage – the bad ones from the left, good ones from the right, etc.). All this knowledge has been successively condensed in a catalogue of behaviours, Cesare Ripa's *Iconography* (1593), which aspires to be a kind of encyclopaedia of each physical expression corresponding to each of the several human emotions or feelings, to be used by painters and sculptors.

Following Andreas Vesalius, *On the Fabric of the Human Body* (1543) the greatest sculptors of the Renaissance like Leonardo da Vinci (1452-1519) or Michelangelo (1475–1564), are known to have themselves dissected corpses in order to study anatomy. In case of the latter, even so, he had difficulty in representing the female form (anatomically,

some of his sculptures – as *Night* – and paintings present a male torso, to which breasts were added).

In all arts – all artists of all times have the need to represent the human figure as their concern and, at least in European tradition, the urge to humanize that figure by making it externally convey emotions. All artists have sketch books and draw all the time. Some of these representations were summarized and recorded in syllabus for students or followers. As an example, the book *Methode pour apprendre à dessiner les Passions… par Monsieur Le Brun, premier peintre du Roi…* (1702) can be referred which exhibits the author's scientific knowledge of anatomy, exploring it to practice the art of resemblance to a point where the affections, and all their representations, almost become excessive:



Figure 2: *Terror* by Charles LeBrun (1702)

This wish to pedagogically compile all the possible or necessary expressions of emotion has a gap in what concerns theatrical theories and practices (Vicentini 2012). Acting took time to freed itself from oratory: «the study and complex taxonomies of the passions and increasingly subtle forms of expression, from the voice and face to the body and limbs» so that «a conception of the actor's art as a discrete form, with its own specific properties, was able to develop, and at various levels including the doctrinal». The treatises evolve into descriptive works of psychological nature (Archer 1888), de par with others more empirical and duly illustrated (Anon. 1807), here representing the passions of joy, fear, pity, hatred, wonder, grief, anger, scorn, jealousy and love:



Figure 3: *The Theatrical speaker – or an elucidation of the Whole Science of Acting* (1807)

Before Darwin there are numerous other scholars studying the subject of human emotion and its expressions. In the case of artists, the best examples come from sculpture, which continues the par with anatomical medicine. The consolidation of the relationship between the arts and medical science converge in the writings of the Swiss poet and physiognomist Johann Kaspar Lavater (1741-1801). As his mentors and inspiration are the works of the Italian (also playwright) Giovanni Battista Della Porta (Della Porta 1586) and the English author Thomas Browne (1605-1682).

Lavater's book on physiognomy (Lavater 1775), with 68 plates etched by Thomas Holloway (1748-1827) represents a turning point. The word Physiognomy (from the Greek *physis*/"nature" and *gnomon*/"judge" or "interpreter") is the assessment of a person's character, or personality, from his outer appearance, especially the skull and the face. The human features become the symbol of psychological assets, moral qualities or flaws.

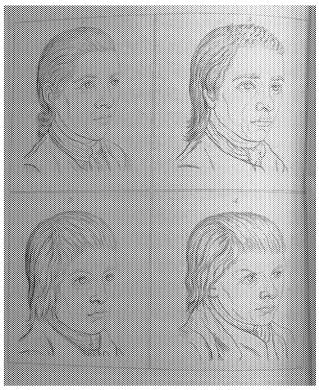


Figure 4: Portrait of two brothers, Lavater (1975), III, p.157

Lavater's works and images, his studies of the human face, were used broadly by painters and psychologists – the first to better portray the emotions, the latter to diagnose misbehaviour in humans by appearance and features.

The Austrian physician Franz Joseph Gall (1758-1828) establishes a direct link between the morphology of the skull and human character – all well explained in the title of his book: *The functions of the brain and those of each of its parts with observations on the ability to recognize the instincts, propensities, talents, or the moral and intellectual dispositions of men and animals, by the configuration of their brains and their heads* – launching the «scientific Phrenology»: an anatomic study and justification of character.

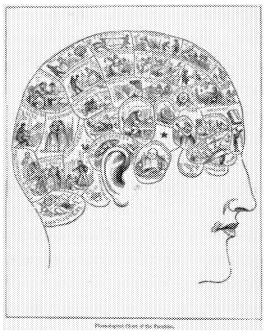


Figure 5: Phrenologic Chart of the Faculties

The success of Phrenology is undisputable: Lavater and Gall are going to be paired and studied together in a book full of images and exercises (Ysabeau 1909), some giving examples of racial characteristic differences. For his part, taking advantage of the recently invented art of photography, the Italian criminologist Cesare Lombroso (1835–1909) establishes the use of facial recognition for forensic or criminal purposes. His contemporaries' the English Pre-Raphaelites also resort to photography as preliminary study for some of their paintings. The main predicament with these perspectives is the idea that behaviour can be read – literally – as a sign of an interior way of feeling, or pathology.

In *The expression of the emotions in man and animals* (1872) Darwin demonstrates the obvious deficiencies of the method of physically decoding emotions: «Although most of our expressive actions are innate or instinctive, as is admitted by everyone, it is a different question whether we have any instinctive power of recognizing them». And adds: «M. Lemoine argues that, if man possessed an innate knowledge of expression, authors and artists would not have found it so difficult, as is notoriously the case, to describe and depict the characteristic signs of each particular state of mind.» and proposes that faces be studied from an anatomical perspective, from the inner working of the face muscles, which had good fortune and has been used for some of the present researchers.

TAXONOMIES OF EMOTIONS

To clearly state what emotions are, how many, and what kind of emotions are being expressed, recorded and analysed, it would be necessary to have a previous map of the human mind and its functions as background. Albeit the recent development in cognitive sciences, we are still very far from dominating the "computational architecture of a mind" (Sloman Croucher 1981). However, thirty years of systematic research have contributed to the assemblage of considerable human expression/emotion databases, and, through statistical accumulation – and (future) machine learning – there might be some hope of progress towards a reliable 'catalogue'.

Facial Action Coding System (FACS) – and its flaws

FACS aim to objectively describe and measure facial expressions and movements. Based on an anatomical analysis of facial action, it offers an advanced method for describing facial movements, and relate them to emotions. A Facial Expression is defined as visible manifestation of the affective state, cognitive activity, intention, personality and psychopathology of a person. Movements of individual facial muscles are encoded, as well as slight different instant changes in facial appearance (micro-expressions).

In 1978, Paul Ekman [http://www.paulekman.com/] launched his Facial Action Coding System, which is still the most widely used method available and followed. In a Darwinian approach (subsequently inspired by Guillaume-Benjamin Duchenne - 1806-1875), Ekman created an "atlas of emotions" with more than ten thousand facial expressions. He had a corpus of six basic emotions (Anger, Disgust, Fear, Happiness, Sadness, Surprise) which, in the nineties, was enlarged to encompass some more (Amusement, Contempt, Contentment, Embarrassment, Excitement, Guilt, Pride in achievement, Relief, Satisfaction, Sensory pleasure, Shame). For his tests he used Pictures of Facial Affect (POFA), consisting of a hundred and ten black and white images of Caucasian actors portraying the six universal emotions plus neutral expressions. It was studied how the contraction of each facial muscle, both singly and in unison with other muscles, changed the appearance of the faces. Each change was called an Action Unit (AU). His first enquiry aimed to evaluate emotion recognition rates in normal and psychiatric populations around the world.

His work is considered seminal on the specific biological correlates of his "precise emotions", demonstrating their universality and discreteness. Recently he forwarded the concept of "micro-expression" (an idea made known and popular through the TV serial *Lie to me*). Presently Ekman offers on-line training tools and courses, useful "from animation to counter-terrorism", promising the apprentice the ability to become proficient in detecting «micro-expressions in one hour».

Due to subjectivity and time consuming issues, FACS have been established as computer automated systems that detect faces in videos, extracts their geometrical features, and then produce temporal profiles of each facial movement.

Ekman work has been followed and criticized. The main problem is that the facial expression analysis system estimates the measurement of actions and also classifies the actions. Empirically it is possible that all the human facial physical movements might be registered someday, but the coding, the relationship between physical expression and emotion will surely be more complex.

First, physical expression only contributes in 55% for the understanding of the message (Verma Kumar 2013). Some other issues have come up related to the quality of retrieval. FACS ignore face asymmetry and, not only all human faces are asymmetric (as every classical artist knew), but asymmetry of expression «helps to convey complex emotional feelings such as conflicting and/or hidden emotions due to social conventions» (Ahn et al. 2013). It does not yet contemplate body position and gestures, and «posture also affects the perception of emotion» (Normoyle et al. 2013). Also, there is not yet an agreement as to the nomenclature: kinds, names and types of emotions to be analysed. Some have been discarded, as Laughter, one of the conventionally primary ones, and the most difficult to understand (Freud *dixit*). Recent studies on laughter (Ruch et al. 2013), or on the human smile (Freitas-Magalhães 2006), are being carried out mainly by psychologists, both resorting to pictorial representations.

The other is a double question, and concerns the fact that the expression of facial emotions is culturally determined, (Jacka et al. 2012), and there are different assessments of emotions output mainly in European and Asian countries. The code for emotion representation has also evolved in time: we now might not recognize the expression (callisthenics) in a painting, or sculpture, that were surely evident for their contemporaries. The acting theories and manuals for aspiring actors have progressed and changed

with the fashions, and evolved in space – the expression of emotions can vary from country to country, what makes them also nurtured by education.

Concerning the tests themselves, the great majority of researchers resort to professional theatre or cinema actors, rehearsed to prepare their images. Actors, whatever the media, are the great fakers of emotions' expression, they exhibit them in every drama, every night on stage, without actually experiencing them; they exercise and exhibit them, some to perfection (the best actors are the most sincere in showing the emotions they are not feeling). This issue seems scientifically precarious, because the image of the emotions evaluated has been purified, mediatized; something similar happens when researchers resort to painting or staged photographic images.

In the technical component part, the majority of the present databases are composed of 2D scans, and will have to be upgraded to comply or accept 3D scans.

Empirically it will be possible to create a global database of human face expressions, (following the example of The Human Genome Project) including all races, ages and times, all disciplines. But the emotional meaning attributed to each one of those expressions, the coding, cannot rely upon present human external hapazardous readings.

Forensic Anthropology – Contributions

The construction of such a global database could avail itself from other practices. Forensic facial approximation, or craniofacial reconstruction, involve «building a likeness of the head and face on the skull of an unidentified individual, with the aim that public broadcast of the likeness will trigger recognition in those who knew the person in life.» (Hayes et al. 2005).

Virtual approximations typically encompass scanning the skull, producing a grid image, and then applying soft-tissue depths at appropriate points to achieve the representation of a face. It can be applied in many multimedia services such as forensic medicine, archaeology, face animation etc. (Duan et. al. 3013). 3D scanning methods more directly and properly reflect the shape relationship between the skull and the face and, avoiding the sculptor manual intervention, it can better contribute to machine learning (Xu et al. 2013).

Experiences have been made with Egyptian mummies (i.e. Tutankhamun, *National Geographic* cover in 2005) or other famous people, like Johan Sebastian Bach's original skull [http://www.conradaskland.com/blog/2008/03/modern-reconstruction-of-js-bachs-face/]:

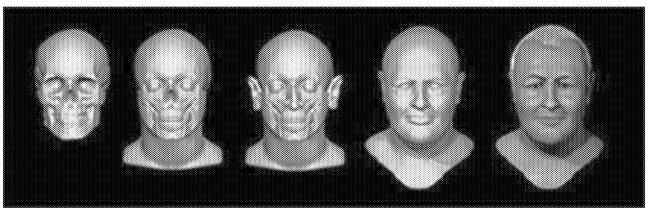


Figure 6: Johan Sebastian Bach's cranial reconstruction

HOW AVATARS ARE EXPRESSING EMOTIONS

Software to create fictional characters has evolved attaining some perfection, faster in exploring quantity than quality. *Monsters Inc. I* (2001, Pixar) boasted about its technical achievements, namely: «the breakthrough depiction of fur and hair, which has the shadowing, density, lighting, and movement consistent with the real thing. This is seen to best advantage with Sulley's feathery blue-green and purple spotted coat that includes nearly 3 million individual hairs, and with Boo's hair and pigtails». (VV.AA. 2001). The great software design achievement was the program allowing to multiply to perpetuity the representation of one hair and its movements.

Microsoft launched an testing platform called *Personas* (2002), now available as *Persona* through Mariner [https://www.marinersoftware.com/products/persona/]. It's a database of characters, pigeonholed into archetypes, a background which would signal each one more particular motivations and allow to predict their behaviour. The software put forward several functions: create, interact and learn – a step further than the options existing in *Second life*.

Still functional and quite interesting is *ARTNATOMY* (2005) [http://www.artnatomia.net/uk/artnatomiaIng.html], a facial expression learning tool authored by the Spanish designer artist Victoria Contreras Flores, in cooperation with anatomy professor Carlos Plasencia Climent. It is intended to «facilitate the teaching and learning of the anatomical and biomechanic foundation of facial expression anatomy». It shows a complete list of the face muscles, their graphic location and their functions, as well as their combined interrelation forwarding a series of corresponding emotions.

The reproduction of muscle movements triggered by inner feelings has to resort to the copy of real life expressions, and the present humanity of avatars come from mirroring their human user.

James Cameron's film *Avatar* (2009) could not be ignored, in particular because he announced that the making of this movie had to wait for the delivery of the technical developments he wanted to use.

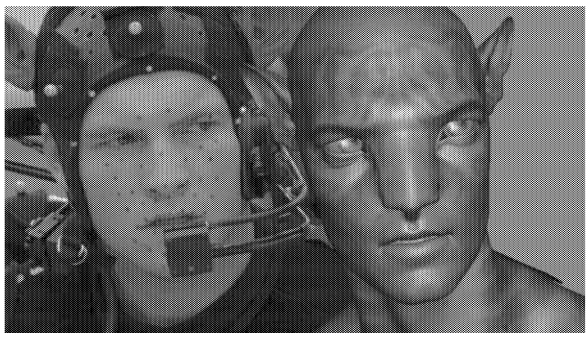


Figure 7: *Avatar* (2009) by James Cameroun

Concerning his characters, he resorted to photorealistic computer-generated images; innovations include an improved method of capturing facial expressions: actors had to wear individually made skull caps fitted with a tiny camera and microphone positioned in front of their faces. The

information collected was then transmitted to computers and duly processed.

In the field of facial recognition technology, and real life capture, the Microsoft application *Avatar Kinect* (2011) [http://www.youtube.com/watch?feature=player_detailpage&v=jb4bzomm9iU] announces that: «now your Avatar can Smile, Frown, Nod and Speak like you». The virtual image will reproduce the user, and she will be able to control its movements and expressions from her living room.

A step forward has been achieved with the game *L.A. Noire* (2012, Bondi/Rockstar Games). Technically it used *MotionScan* (Depth Analysis) platform; the system captures the views from 32 cameras arrayed as stereo pairs to register the face of the actors, without having them wear any special suit. The video stream is processed in a computer using Stereo Vision techniques to produce a 3D model for each frame. This technical advance in the representation of faces and emotions, has allowed the recourse to cinematic close-up practices.

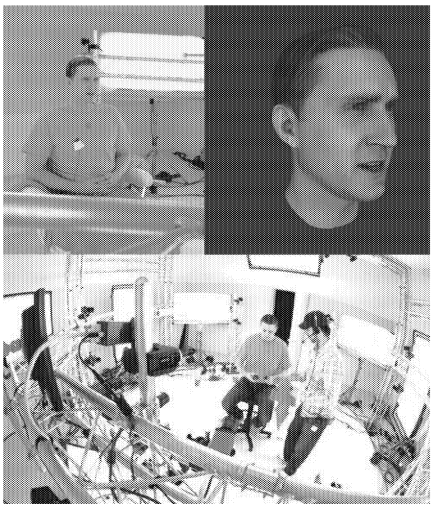


Figure 8: MotionScan in *L.A. Noire* (2012)

The plots are crime stories, happening in Los Angeles, in 1947, with the expected characters of 'film noir'. Playability consists in accompanying the work of the detective, and helping him to solve the case. But the user has to find the killer by interrogating the suspects: «When questioning any persons of interest, you will be challenged to determine the validity of their statements», analysing their emotions to decide if they are lying or not, if they are guilty or not. The expressions to be read – and their meaning – are coded in the game *Official Guide*: «***Topic:*** *Stolen auto courier*: Cole demands Jean to tell him where she picked up the car, but she claims she can't remember. She gives you a look similar to the one you doubted before. *Correct Assessment: Doubt*» (Bogenn Barba 2012).

Face recognition software is being used to create new, external and different fictional characters and avatars. The incompleteness of the state of the art, the irregularity of representation of expressions, the impossibility of being sure when coding the represented emotion – even with 3D scans – in films, or videogames, is not so critical if applied only to stop motion and avatars for ludic purposes. But can become very serious if used on humans with any kind of evaluation intention, like lie detecting, the study of psychological reactions or "counter-terrorism".

CONCLUSION

From the evolution of the avatar word and meaning, it was established the Avatars' relationship with characters, as more traditional types of fictional representations of the human. Both these delineations (ancient and modern) require credibility and authenticity (verisimilitude), namely in what concerns the expression of emotions. A taxonomy for synthetic characters will have to go beyond the literary ones, and include visual representations (as in painting, sculpture and theatre).

Characters depict emotions. In order for avatars to duly express emotions it will be necessary a catalogue, and a proper code for their representation. The creators have profited from the existing FACS databases. These are yet faulty, and cannot clearly state what emotions are, how many, and what kind are being expressed, recorded and analysed. The necessary background, a map of the human mind and its functions, is still in the making. Like in old Phrenology, FACS maintain the idea that behaviour can be read – literally – as a sign of an inner way of feeling, or pathology. Scientifically, the corpus is poor and in constant change, and the majority of the tests have resorted to mediatized images (actors, paintings).

Empirically it is possible that all the human facial physical movements might be registered some day in a global database of human face expressions, (i.e. after the example of The Human Genome Project); but the coding, the relationship between physical expression and emotion, are surely more complex than the ones being used at present. Following Turing's proposal, the researchers of human face expressions need to know how this complex operation of human behavior works in order to translate the answer into the form of an instruction table. The present «instruction tables» – databases and software – need yet to be improved so that, in a near future, machines will be able to be programmed to carry out the operation of recognizing human expressions (face and body) and classify them adequately. Meanwhile, Avatars are (still) cheating at the «imitation game» because they only smile with the help of human actors and users.

REFERENCES

Anon. 1807. *The Theatrical speaker – or an elucidation of the Whole Science of Acting*. J. Smeeton, London.

Ahn, J.; Gobron, S.; Thalmann, D.; Boulic, R.; 2013. "Asymmetric facial expressions: revealing richer emotions for embodied conversational agents". *Comp. Anim. Virtual Worlds*. doi: 10.1002/cav.1539

Archer, W.; 1888. *Masks or Faces? – A Study in the Psychology of Acting*. Longman, Greens & Co., London & New York.

Aristotle. 2000. *Arte Poética*. Trans. A. M. Valente. Fund. Calouste Gulbenkian. Lisbon.

Bogenn, T.; Barba, R.; 2012. *Bradygames*, Official DLC E-Guide, Vol.1 (p.9) - http://www.rockstargames.com/lanoire/files/LANoireGuide-TheNakedCity-ASlipOfTheTongue.pdf

Browne, Th.; 1643. *Religio Medici.* 1904. *The Religion of a Doctor, Hydriotaphia, and the Letter to a Friend.* Vol. I, Charles Sayle (Ed.) Grant Richards, London. http://ebooks.adelaide.edu.au/b/browne/thomas/religio/contents.html

Darwin, Ch.; 1872. *The expression of the emotions in man and animals*. Murray. London, 358; 360-61.

Della-Porta, G. B.; 1586. *De humana physiognomonia libri IIII*. Vico Equense. Italy. http://hos.ou.edu/galleries//16thCentury/DellaPorta/1586/

Du, S.; Martinez, A. M.; 2013. "Wait, are you sad or angry? Large exposure time differences required for the categorization of facial expressions of emotion". *Vision – a Journal of scientific research on biological vision.* http://www.journalofvision.org/content/13/4/13.full

Duan, F.; Yang, S.; Huang, D.; Hu, Y.; Wu, Z.; 2013. "Craniofacial reconstruction based on multi-linear subspace analysi". *Mingquan Zhou Multimedia Tools and Applications.* January.

Forster, E. M. 2005. *Aspects of the Novel*. Penguin Books, Harmmonds., (3rd.), 61.

Freitas-Magalhães. A. (2006). *The psychology of human smile*. University Fernando Pessoa Press. Porto.

Gall, F.-J.; 1822-1825. *Sur les fonctions du cerveau et sur celles de chacune de ses parties. avec des observations sur la possibilité de reconnaitre les instincts, les penchans, les talens, ou les dispositions morales et intellectuelles des hommes et des animaux, par la configuration de leur cerveau et de leur tête*. 6 vols., J. B. Baillière, Paris.

Hayes, S.; Taylor, R.; Paterson, A.; 2005. "Forensic Facial Approximation: An Overview of Current Methods Used at The Victorian Institute of Forensic Medicine/Victoria Police Criminal Identification Squad".*The Journal of Forensic Odonto-Stomatology*, Vol.23 No.2, December, 45-50.

Jacka, R. E.; Garrod, O. G. B.; Yub, H.; Caldarac, R.; Schynsb, Ph. G.; 2012. "Facial expressions of emotion are not culturally universal". *Proceedings of the National Academy of Sciences.* James L. McClelland (ed.), Stanford University, Stanford, CA. http://www.pnas.org

Le Brun, Ch.; 1702. *Methode pour apprendre à dessiner les Passions… par Monsieur Le Brun, premier peintre du Roi…* Chez François van der Plaats. Amsterdam.

Lavater, J. K.; 1775. *Physiognomische Fragmente, zur Beförderung der Menschenkenntniss und Menschenliebe*. Gott schuf den Menschen sich zum Bilde. Erster Versuch. Leipzig & Winterthur.

Masson, D.; 1856. *Biographical and Critical Essays – chiefly on English Poets*. MacMillan & Co., Cambridge. 406-409.

Masson, D.; 1859. *The Life of John Milton narrated in connection with the Political, Ecclesiastical and Literary History of his time.* Vol. I - 1608-1639. Gould and Lincoln, Boston. 193, 198, 846, 878.

Normoyle, A.; Liu, F.; Kapadia; Mubbasir, B.; Norman I.; Jorgy, S.; 2013. "The Effect of Posture and Dynamics on the Perception of Emotion". http://cg.cis.upenn.edu/images/normoyle13Emotions.pdf

Ripa, Cesare. 1987. *Iconologia*. (*Iconologia overo Descrittione dell'Imagini universali* 1593) Trans. J. Barja, Y. Barja, R.M. Sánchez-Elvira, F. Garcia Romero.Vol. I-II. Akal. Madrid.

Ruch, W.; Hofmann, J.; Platt, T.; 2013. "Laughter studied from pictorial representations – and photographs; Investigating facial features of four types of laughter in historic illustrations". *European Journal of Humour Research* 1(1) 99-118 http://www.europeanjournalofhumour.org

Sloman, A.; Croucher, M.; 1981. "Why Robots will have Emotions". *Proceedings IJCAI*, Vancouver

Stephenson, N.; 2011. *Snow Crash*. Penguin Books. Harmonds. (3rd.)

Turing, A.; 1950. "Computer Machinery and Intelligence". *Mind*, No.59, 433-460. http://www.loebner.net/Prizef/TuringArticle.html

Verma, A.; Kumar, L.. 2013. "A Comprehensive Survey on Human Facial Expression Detection". *International Journal of Image Processing* (IJIP), Volume (7): Issue (2) http://cscjournals.org/csc/manuscript/Journals/IJIP/volume7/Issue2/IJIP-746.pdf

Vicentini, Cl.; 2012. *Theory of Acting - From Antiquity to the Eighteenth Century*. Trans. R. Bates and A. Weston. Marsilio & Acting Archives. Naples. 13 http://www.actingarchives.unior.it/Public/Articoli/1b1d5403-ed73-40f8-b072-b22763c6982b/Allegati/Claudio%20Vicentini_Theory%20of%20Acting.pdf

Xu, S.; Ma, G.; Meng, W.; Zhang, X.; 2013. "Statistical learning based facial animation". Journal of Zhejiang University SCIENCE C, Volume 14, Issue 7, July. 542-550

Ysabeau, A. 1909. *Lavater et Gall – Physiognomy and Phrenology rendues Intelligibles pour tout le monde*. Garnier-Frères. Paris.

VV.AA. 2001 (?) "Monsters Inc. Production notes". Pixar. http://cinema.com/articles/724/monsters-inc-production-notes.phtml

AUTHOR BIOGRAPHY

HELENA BARBAS (1951) is Professor-Lecturer of the Department of Portuguese Studies – Faculdade de Ciências Sociais e Humanas, Universidade Nova de Lisboa (F.C.S.H. – U.N.L.). She holds a MA (1990) and a PhD (1998) in Portuguese Studies – Comparative Literature – Literature and the Arts. in 2005. She gained her "Habilitation" (2008) in Literature and Cyberarts.

Helena authored books and several essays, translated theoretical texts, novels, drama and poetry. She has collaborated with diverse newspapers, magazines and with national TV cultural programs.

Since 2003 Helena has centered her research in the field of Digital Humanities: human-machine interaction, serious games, e-learning, web.2, and interactive digital narrative.

She was a member of the InStory team (2005-2007) – best Portuguese web mobile project 2006. She prepared a project on serious games, PlatoMundi, aiming to introduce e-learning and ethical issues in game playing; she is developing Numina – Project José Leite de Vasconcellos (1858-1941).

In 2011 she received the (2nd.) SANTANDER Award for the Internationalization of the F.C.S.H. Scientific Production 2010. Presently she is a full member of CesNova.

Homepage: http://www.helenabarbas.net

VIRTUAL WORLDS

COOPERATIVE AI IN REAL-TIME STRATEGY GAMES

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

Daniel Livingstone
University of the West of Scotland
Paisley Campus
PA1 2BE Paisley, Scotland
E-mail: daniel.livingstone@uws.ac.uk

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

ABSTRACT

After a review of the current state-of-the-art of cooperative AI in commercial Real-Time Strategy games, we identify a number of weaknesses and limitations in current approaches to cooperative AI. We argue that the creation of AI players better able to communicate and cooperate with human players should enrich the genre, making new types of game play possible and breaking away from currently common scenarios where the human player has to protect an AI which is incapable of coordinating or cooperating in planning or implementing high level strategies. We demonstrate a prototype implemented in StarCraft II, and present a preliminary evaluation showing that our AI is capable of choosing actions appropriately according to both an agreed strategy and in reaction to the actions taken by its human ally.

INTRODUCTION

Real-Time Strategy (RTS) games are an interesting application for game-AI research, offering rich challenges to both human and AI players (Buro 2004; Yildirim and Berg Stene 2008), usually requiring significant strategic and tactical planning as well as multi-tasking. There are many RTS games each with their individual problems and game mechanics. Some focus on building and maintaining complex economies alongside strategic and tactical combat, while others skip economic management and instead focus solely on tactical unit control in combat.
While AI for RTS games has been widely studied, one aspect less well studied - and less well provided for in commercial RTS games - is cooperative AI, artificial intelligence where the AI takes on the role of another player working alongside the human player and cooperating in devising and executing strategies and tactics to win a joint victory over other (human or AI) opponents.
In this paper we present the current State of the Art in cooperative AI in RTS games based on our own review of currently available and older but notable commercial titles. We then present our own work in developing a cooperative AI for Starcraft II, demonstrating how the AI is able to use simple communication commands to negotiate strategy with the human player and select units to be built accordingly. A preliminary evaluation is presented which highlights how the AI plays differently according to the agreed strategy and in response to the human player's actions.
Finally, we will argue that improved cooperative AI in RTS games presents an opportunity not just for academic research relevant to real world problems, but that it offers new possibilities for enhancing and enriching the range of game-play experiences available in the RTS and related genres.

COOPERATIVE AI IN STRATEGY GAMES: STATE OF THE ART

Creating Artificial Intelligence players for real-time strategy games that are competitive against human players is an important subject for both research and industry, with commercial games such as StarCraft often being used as a testbed (cf. Weber et al. 2010; Togelius et al. 2010; Churchill and Buro 2011). Other related research focuses on aspects like believability (Lidén 2002; Livingstone 2006) or machine-learning approaches (Weber 2009; Hsieh 2008). Recently StarCraft II, one of the world's most successful eSport stratregy games, introduced an AI league with the latest expansion Heart of the Swarm, where human players or teams can challenge AIs of various difficulties and play styles in a tournament format similar to the human-vs-human eSport ladder. Yet, while it is quite common to find AI players as opponents in strategy games - with most including some form of single player campaign - it is rare to find games where a human player and an AI cooperate to achieve a goal. We surveyed 23 real time strategy games, mostly with release dates between 2000 and 2013, but also including key popular and genre-defining older titles like Command and Conquer (1995) or StarCraft (1998). From this survey we identified three basic categories of human-AI cooperation.
The first, and most common, approach is having an AI ally which plays by different game rules to the human player and is highly dependent on either external events or direct orders issued by the human player. In this category there is generally no bidirectional communication between player and AI, and the majority of examples feature no human-AI communication at all. Instances of this category of ally can be found in the single-player campaigns of many titles. For example, the final StarCraft II: Heart of the Swarm mission "The Reckoning" features a Terran ally producing units while ignoring their normal technology requirements. The AI-ally attacks with all its produced units in given intervals. Such timers are examples of an external event influencing the AI ally and are often used for storytelling reasons. In this case, part of the mission objectives for the Human player is to make sure that a hero unit belonging to the AI ally survives.

We can include in this category some games where there is a very limited form of bidirectional communication. Again examples can be found in single player campaigns, particularly in more recent games. Consider a scenario where the human player has to escort a convoy controlled by its AI ally. The convoy might have to prepare before rolling out. Once the convoy is ready to set off a message will be sent to the player to allow them to start the convoy. The AI is clearly playing to modified rules to fulfill a specific role, and the limited communication is heavily scripted and pre-determined.

Our review discovered examples that show that this class of cooperation covers the majority of RTS games, single player campaigns in particular, such as some missions of the single player campaigns of Warhammer 40k: Dawn of War, Warcraft III, Star Trek: Armada or Command and Conquer: Generals. One of the likely reasons for the prevalence of this type of cooperation is that it allows the creation of more varied single player missions that rely on more straightforward programming and scripting and does not require the development of a true player-level cooperative AI. Such controllability of the AI players is an important aspect for game developers, to allow the AIs to make plot-based decisions rather than game-mechanics-based decisions, like leaving a back entrance undefended.

The second class of cooperation also lacks bidirectional communication, but this time the AI is playing by the same rules as its human ally. A good example of such an approach is the single player campaign of Command and Conquer: Red Alert 3. The campaign was tailored for two human players playing together. When there is just one human player, the ally will be controlled by AI. Obviously, this ally has to overcome the same challenges and plays by the same rules as the missing human player would have. Yet, there is no way to communicate with the AI ally, which leads to many situations where the single human player has to 'babysit' their ally. Having the player take care of the AI commonly occurs in the first category where it is generally part of the design - here it is a result of poor AI and a lack of human-AI communication. This category of cooperation is also found in the multiplayer modes of many RTS games, where it is possible to team-up with an AI player, with similar problems frequently occurring. In this category the AI player usually plays by the same rules (though often 'cheating' to a certain extent, perhaps having perfect knowledge of the game map or being able to issue thousands of commands per second), but as before there is no bidirectional human-AI communication. Some games, such as the original Age of Empires, allow unidirectional communication from the human to the player where a limited number of messages could be sent by typing chat commands. While intended to support human-human cooperation in a time when high-speed internet was uncommon, the AI allies in the game also attempt to follow these short commands. There were 35 chat commands, but some of them were ignored by the AI as they were included for taunting, like "Blame it on your firewall!". Examples that were taken into the account by the AI are "Gold please" or "Attack now". Other titles with similar approaches include World in Conflict or Anno 1404 or the multiplayer mode of many real time strategy games that allows replacing a human by an AI player, for example the multiplayer modes of StarCraft II, Homeworld 2 or Supreme Commander.

The final category of cooperation would offer bidirectional communication between AI and human and have both playing by the same rules. This kind of cooperation is very rare and attempts to make the AI a true ally instead of a partner that needs protection as found in the previous categories. One example of this would be Sins of a Solar Empire. This game offers an unusual mission system that allows players to hand out quests to other players, like "destroy X amount of economy buildings of player Y", where Y might be a third player, an enemy in common. The quests can be parameterized and offer simple bidirectional communication as both human and AI players create such missions for their allies. Working with these, it is possible to establish some basic strategies, such as "Let's attack economy first". This is also one of the very few examples of communication and coordination in games where the AI player can lead the process and is not limited to responding to a human offer or order. Yet, this approach is still limited as these missions offer only methods to express the immediate tactical goals, and it is not possible to agree on long-term strategies like "I will build the fleet, you will supply me with resources". Similar communication and coordination can also be found in different titles such as Civilization V, where, due to their turn-based nature, the time pressure on communication is less severe and dialogues with parameterized statements can be more easily used.

This last category offers some very interesting opportunities for strategy games. It seems to be the only way solving the baby-sitting problem of human-AI cooperation in multiplayer game modes and cooperative campaigns like the Command and Conquer: Red Alert 3 example above. The reason for this kind of cooperation almost never being used in single player campaigns could also be a consequence of game designers awareness of and desire to limit the cases where the player has to care for the AI. If more effective solutions could be found for enabling human-AI cooperation in RTS games, this could lead to newer innovations such as new game modes and battle types such as campaigns where you learn and train together with AI allies, creating your own individual AI companions.

HUMAN-COMPUTER COMMUNICATION AND COLLABORATION

To enable strategic cooperation between an AI and a human player, it is first necessary to create some means to allow them to communicate. As well as providing means for players to control their own units and issue direct commands, consideration needs to be made over how the player can interact with computer allies.

There is a range of current and recent research studying communication between human players (Drachen and Smith 2008; Dabbish et al. 2008; Wadley et al. 2007) , and it is clear that the quality, clarity and success of communication has a large impact on success and failure in team based online gaming. We have also conducted our own review of player to player communication channels in contemporary real-time strategy games (Stiegler and Livingstone 2013a).

Human players often establish a certain vocabulary to quickly and precisely describe certain strategies or game elements and situations. Established names for certain rush strategies in the StarCraft II eSports scene are, for example, a "six-pool" or a "four-gate". Yet, such symbolic terms are only valid in certain communities and meaningless or even misleading in other contexts. A "six-pool", for example, means building a Zerg Spawning Pool once the player controls six workers, while "four-gate" means building 4 Protoss Stargates as soon as possible. That's easy to confuse and only meaningful once the terms were defined. Players may also refer to a wide range of more general strategic and tactical concepts ranging from detail unit commands to very abstract strategies. Obviously, the semantic understanding and communication capabilities of an RTS AI player will be limited compared to its human partner.

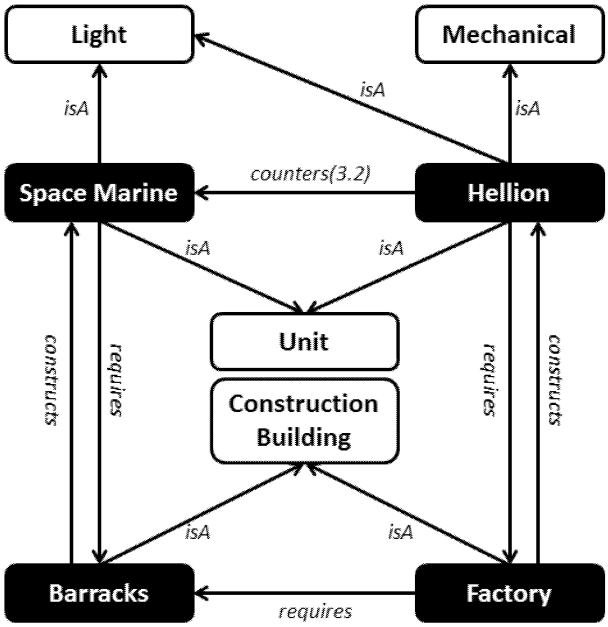
IMPLEMENTATION

Our current implementation is an AI module for StarCraft II dealing with high-level decisions, like which units or buildings to build, when to invest in economy and which overall strategy to pursue, using communication to agree on a strategy with a human ally. The AI implementation in our approach uses a utility-system where the utility values are calculated by inferring on a semantic structure describing the game mechanics of StarCraft II (Stiegler and Livingstone 2013b). Having such a semantic representation of the game mechanics available at run-time allows the AI to translate information achieved through communication with their human allies directly into new nodes, relations or parameters associated with the structure. This is particularly useful, as human players will usually refer to relatively abstract semantic constructs rather than defined unit types. A human player might, for example, indicate a need for "anti-air units". A semantic mapping of what units are capable of shooting air units and in which situations which of these are effective in the current game situation helps the AI to understand such information.

At game start, the AI is initialized with a predefined, handcrafted semantic structure describing the game mechanics of StarCraft II. The semantic structure contains unit classes that are present in the game, like the Space Marine, and more abstract classes added to describe the game mechanics. Nodes represented in the game are shown as black boxes in Figure 1, while abstract classes are shown as white boxes. The semantic structure also defines a couple of relations, of which some, like the very important "counters" relation, are parameterized and achieved through a series of automated tests. Figure 1 shows a simplified excerpt of the semantic structure.

As the game progresses, the AI adds nodes, relations or parameters to the structure to represent new knowledge gathered, either through observation or communication with an ally. Important parameter associated with nodes in the structure are, for example, its own unit counts or the observed unit counts of enemy players. The semantic structure will also represent each player with a single node. Communication with a human ally might add new nodes or relations, for example to express that the human ally wants to focus on air units, which would add a "prefers" relation from the node representing the human ally to an abstract "air unit" node.

The utility system defines several option-classes, like "build a unit" or "advance in techtree". Each of these option-classes is a function inferring information from the semantic structure. The "build a Space Marine" option in the utility system, for example, being part of the "build a unit" option-class, would check "counters" relations from observed (or expected) hostile unit classes to the Space Marine node. Using the semantic structure in Figure 1, the utility of a Space Marine will be small if the AI observed several hostile Hellions.



Figures 1: Semantic structure excerpt

Within each AI cycle, there are further operations executed besides the inference functions for the option-classes. These functions serve the purpose to refine the semantic structure. An example of such a refinement function would be unit-count propagation to deal with uncertainty. Again using the example from Figure 1, if the AI observed a Factory in the past, it can increase an "expected unit count" over time, walking down "requires" relations that end in the factory. Three minutes after a Factory was observed, the "expected unit count" for the Hellion might have reached a value of 4.102, making the AI believe that the enemy could have at least 4 Hellions by now. These "expected unit counts" can then be taken into account by the inference functions.

The semantic structure can also be used by the communication module, for example to answer abstract human questions like "Are you going to build units?" or to formulate more abstract statements itself like "I will go teching".

The AI implementation does not use any of the StarCraft II hard coded AI APIs. Instead, it interfaces the same API the user interface of a human player would. It also obeys the same visibility constraints, like fog of war.

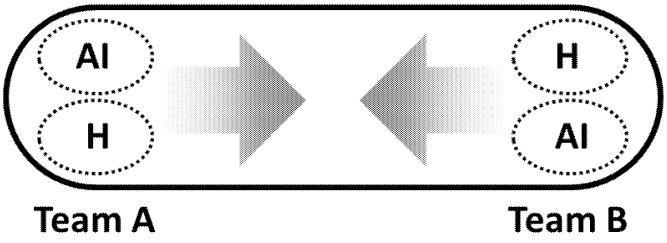
The set-up procedure for a game using our decision module would thus be:

1. Create a semantic structure representing the game mechanics decided to be relevant for AI reasoning. Similar representations are often already used during game development, for example for balancing or gameplay analysis.
2. Identify what actions the AI module should handle and classify them in option classes which follow similar reasoning. Write inference scripts on the semantic structure for each option class.
3. Identify which ongoing refinement actions are required and write scripts working on the semantic structure for them.
4. Describe the human-AI communication interface within the semantic structure and introduce respective relations and nodes.

Note that this is an iterative process and it will probably require several loops through steps 1-4, as each step might introduce changes and adjustments to the rest of the semantic structure.

TEST ENVIRONMENT

As we wish to test our high-level decision module rather than tactical combat, the simplified game-mode used for our approach automates micro-management, so neither human nor AI players can control their units in combat. Instead, they produce their army which automatically attacks the enemy in waves. Players can, however, still position their units within their attack force and make high-level decisions for each unit, like deciding whether a fast unit should run ahead of the main force or escort it. Every human player is teamed with an AI player. The armies of a human player and their AI ally will attack together. Also, while both have separate resources, they share technology tree progress and upgrades, offering additional opportunities for cooperation, like one player completely focusing on upgrading while their ally supplies the army. The costs to advance in economy were increased, as it is no longer possible to make explicit raids on the enemy workers, due to units no longer being controllable in combat. This plays an important role in the StarCraft II metagame. The test scenario is similar to popular "Tug of War" minigames.



Figures 2: Schematic View on the Test Scenario

Figure 2 shows a schematic overview on the test scenarios. The AI base is located next to the base of its human ally. The armies of both teams walk across a simple battlefield and extra resources are awarded to the team controlling at least half of the battlefield, to mimic the benefits of map-control in a full StarCraft II game.

Communication between Human and AI players takes place by issuing commands of a small, defined vocabulary, where both Human and AI players express their ally what they'd like them to do and what they are currently planning to do themselves. Low-level decisions are communicated by indicating which unit type a player wants to focus on. This is done in the "next-goal panel", where small icons for all the available units can be selected, representing a certain unit choice. Note that these statements just serve the purpose of communication and don't involve any obligations to a player. A player could tell their AI ally that they will be producing Space Marines, and hence the AI might want to produce something else instead. As the AI and human are on the same side, the AI trust the human player. The "next-goal panel" also shows the human player which unit their AI ally wants to produce next.

Strategy-level decisions are communicated in the "strategy panel". Here, players can tell their ally what kind of strategy they want them to pursue, while the AIs do the same. These recommendations for their allies are expressed by players by indicating one of four high-level strategies:

- Force: Focus on producing units as fast as possible.
- Tech: Focus on acquiring the best units or upgrades, even if that means investing time and resources at the moment.
- Support: Focus on units and upgrades that support by battleforce or unlock research options for me.
- Free: I don't have any recommendations. Do whatever you think is best.



Figures 3: The strategy panel (bottom) and the next-goal panel (top right)

There are several options for cooperative strategies. The most straightforward one would involve both players distributing their income equally towards unit production, research and economy and both fielding an army of similar value. The game mechanics offer several synergies that can be exploited to increase the effectiveness of the combined army. A single medivac, for example, might be quite weak, but if your ally already produced a couple of infantry units, the single medivac can be a powerful addition due to its healing capabilities. Cooperation could also involve covering several areas of gameplay by either of the two players. For example one player focusing on infantry, while the other player focuses on mechanized units or aircraft. The main tool

for communication for such scenarios is the next-goal panel, where players indicate which unit type they want to focus on. This allows their allies to predict that such units will be present in the future and chose their further steps respectively.

Further cooperative strategies involve more intense communication. These strategies involve shifting not just the produced unit types between the human and the AI, but also what aspects of the gameplay they will cover. An example would be one player focusing entirely on army production, while their ally focuses entirely on performing upgrades and research. These strategies have some benefits as it avoids producing buildings twice. The research player, for example, might not need production buildings at all. Obviously these strategies require more communication and cooperation between the players, as their actions are no longer self-sufficient, unlike the approach mentioned above. The researching player, to stay with the example, has to trust their ally to produce an army. Both players will therefore have to agree which technology path they want to follow, to avoid producing units for which they don't have powerful upgrades yet or researching upgrades for units which aren't planned to be produced. The main tool for communication is also the next-goal panel, to let the ally know which upgrades or units are next on the chopping block. It will also involve agreement through the strategy panel, as both players have to agree on the strategy and the role they will take in it.

A similar family of cooperative strategies involve one player buying time for their ally. In these strategies, one player tries to build many units with some important upgrades and research in early game stages in order to keep their opponents busy. In the meantime, their ally skips military production and research but instead invests most - if not all - resources into economy to have large funds available in later game stages. These cooperative strategies offer benefits in later game stages, where the economy player can field expensive units or upgrades and will therefore hopefully out-tech the opponents, while the early-game player might fall behind in terms of unit power once the opponent team is producing better units. The main tools for communication would be the strategy panel for players to agree on the overall strategy and their role in it. The next-goal panel would probably be less important, as players are still independent in their general unit choice. Yet, this strategy can be combined with focusing on different aspects of the game mechanics, like the earlygame player focusing on cheap but numerous ground units and the late-game player supplying late-game anti-air units. In such a strategy, the next-goal panel will also become important.

There are two important limits regarding AI communication and cooperation in our approach. The first one being that the AI is not able to compensate for situations where the human player incorrectly sets the "next-goal panel", whether they are lying on purpose or because they just forgot to update the panel. In this case, the AI will still expect that the human player is focusing on the production of the indicated unit, although the human is doing something completely different. The AI will, however, notice that the choice of the human might be something very ineffective in the current situation. In these cases, the AI will slowly start to ignore the human statement until it is next updated.

The second limitation originates from the highly limited communication channels. In some scenarios, there are multiple and very different strategies, that would all fall under one high-level strategy category. A support-strategy for infantry units might involve building medics, or it might involve building Siege tanks to cover the infantry. The human and AI player have no channel at the moment to agree on such fine-granular strategies and the AI would act as it sees fit. This could be improved, however, by adding additional complexity to the "next-goal panel", to increase the number of options for indicating short to mid-term goals and strategies.

EVALUATION

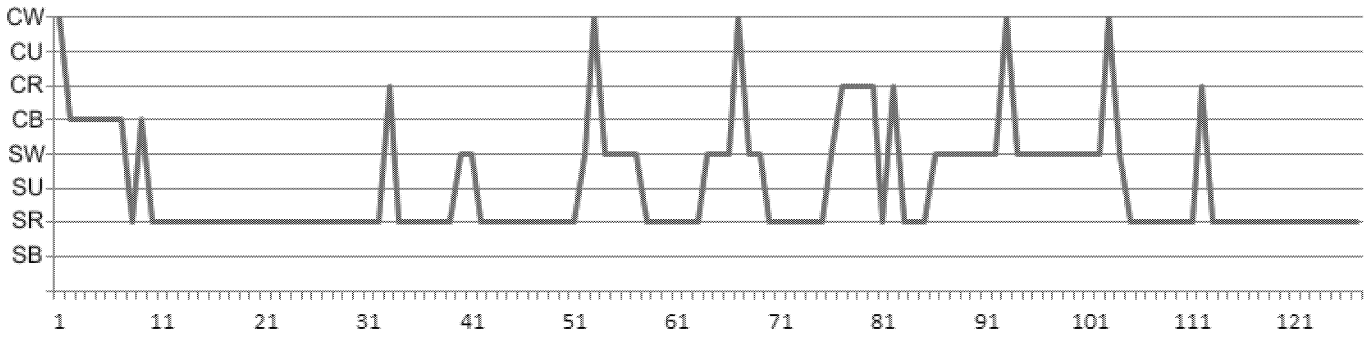
Evaluating AI in strategy games, or computer games in general, involves taking several requirements into account. Typically this would involve processing performance, memory footprint, believability, usability and much more. For our initial evaluations we consider only whether the behaviour of the AI is appropriate for a particular strategy agreed with a human player and according to the current conditions in game.

To do this a number of games were played in which the AI action for each cycle was logged. A cycle is the step within the AI runtime where the AI decides which actions to execute next. Currently a cycle lasts 5 seconds, allowing the AI to change its current course of action every 5 seconds.

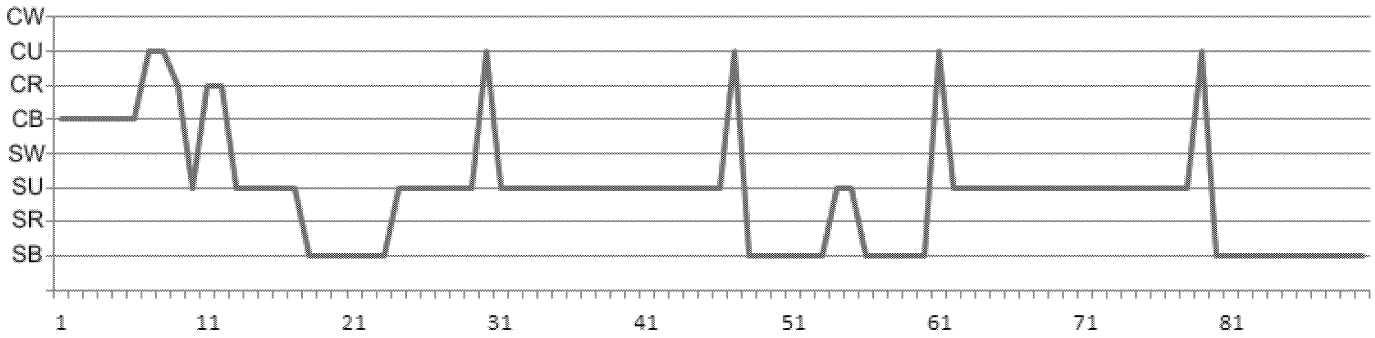
The AI was teamed with the same human tester, who was told what kind of playstyle they should follow and what preferred strategy they should communicate to the AI. The human-AI team played against a single opponent AI, also using our implementation, which received resources equal to those of the human-AI team. The goal of this first evaluation was to identify if the AI-ally was actually adapting its play style to actions of the human player as well as to the agreed strategies. As long as it fits to the predetermined game style, the human players were also encouraged to follow recommendations of the AI ally. The Figures 4-7 illustrate some examples of the evaluation results. In each cycle, the action of the AI was coded into one of eight possible categories, made up of four possible construction actions and four possible resource saving actions. In terms of gameplay, there is no difference between the four save actions, but recording the current goal of the AI when saving resources greatly helps understand and decode what the AI is attempting to do at any one time.

- CW / SW: Constructing workers / Saving resources for workers
- CU / SU: Constructing units / Saving resources for units
- CR / SR: Constructing research items / Saving for research items
- CB / SB: Constructing buildings / Saving for buildings

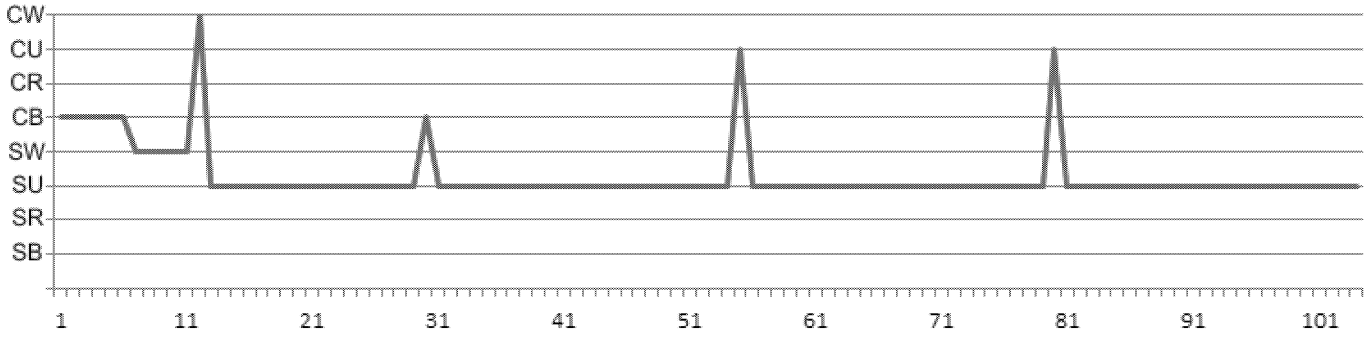
The Figures 4-7 below show a few selected examples of AI actions for games with a range of pairings of human player strategies matched with different AI strategies. In total 17 games were played, and the results below are considered representative of typical games.



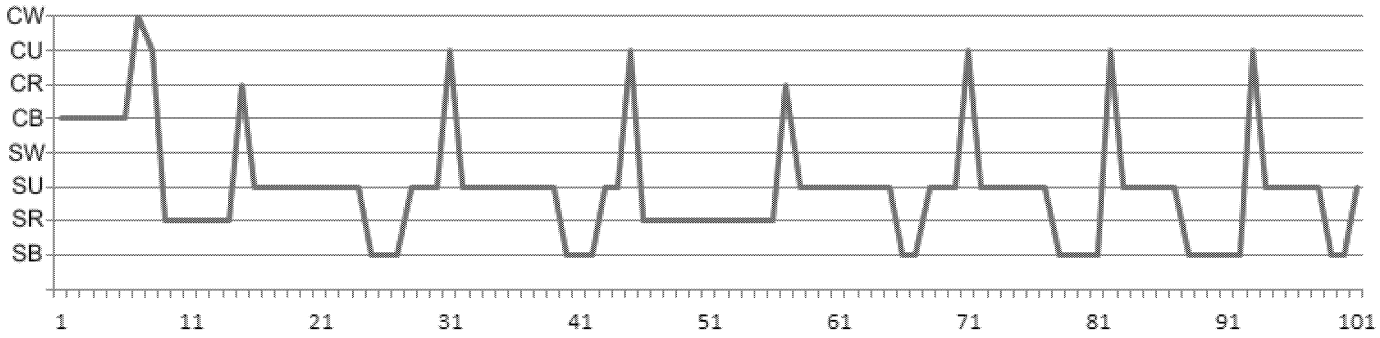
Figures 4: AI State graph for Human playing Force and AI playing Support



Figures 5: AI State graph for Human playing Support and AI playing Force



Figures 6: AI State graph for Human playing Force and AI playing Tech



Figures 7: AI State graph for Human playing Tech and AI playing Force

From a closer inspection of these results, we came to the conclusion that the AI is adapting its play style to both the agreed strategies and the actions of the human player. Figure 5, for example, shows an AI that spent most its time in performing research or saving resources to do so. Figure 7, just like Figure 4, shows an AI that was told to support the player, shows a different behavior. While also performing some research, it also invested into military units. This is due to its human ally not building up military forces but instead trying to fulfil technology requirements to achieve late-game units, so the AI had to compensate and adjusted its strategy accordingly. In Figure 4, the human player opted for supplying a large army, leaving more room for the AI to pursue its research.

Differences between Figure 4 and 5 shows the effect of AI playstyle if it is told to solve a different task. In this case the human and AI player roles have been switched. Figure 5, where the AI was providing the army, rather than being supportive, clearly shows an AI focusing on unit production rather than research.

Figure 6 is an interesting example, showing an AI that tried to bring high-end technologies to the field. The human player, however, is producing some military units, but tries to unlock the research requirements to help the AI. As illustrated in Figure 6, the AI built only two units, but both of them were high-tech. It did not spent time constructing or saving for buildings to unlock technology requirements, besides the production buildings it had to have. This is a quite sophisticated cooperative strategy that requires bidirectional communication and agreeing on long-term strategies.

Although not explicitly measured, the test games indicated that the AI is actually following reasonable strategies and offers reasonable recommendations. We are currently working on turing-like tests to further test the believability of the AI and to see if this approach feels less like babysitting to human players.

CONCLUSIONS

We have demonstrated through a review of the current state-of-the-art in commercial RTS games that human-AI cooperation is currently not fully exploited, and tends to be used in quite limited forms and commonly sees the player having to take a protector role to look after the AI. A richer range of game experiences for both single-player and multi-player RTS are possible if human-AI cooperation can be improved.

Accordingly, we have implemented a prototype in StarCraft II. Using a simple graphical communication panel to coordinate high level strategies, AI players can fulfill the part of an equal team-mate with a human ally. The AI does not blindly follow a set strategy, a common weakness in existing games, but also responds to the actions and situation of their ally. The AI was tested across a number of play sessions and was observed to behave appropriately under a range of conditions according to its current strategy.

Our prototype will be released on Blizzard Arcade to allow others to test the system. Future work will aim to both increase the richness of the communication possible to allow short, mid and long term strategies to be shared, potentially to include cooperation in micro-management of units, and to more rigorously evaluate the effectiveness and believability of the AI.

REFERENCES

Buro, M. 2004. “Call for AI Research in RTS Games”. In *AAAI Workshop on Challenges in Game AI*, 139-141.

Churchill D. and Buro M. 2011. “Build order optimization in StarCraft”. In *Proceedings of AIIDE*, 14-19.

Dabbish, L.; Kraut, R.; and Patton, J. 2012. “Communication and commitment in an online game team”. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 879-888.

Drachen, A. and Smith, J.H. 2008. “Player talk - the functions of communication in multplayer role-playing games”. In *Computers in Entertainment*, vol. 6, no. 4.

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*, 3106-3111.

Lidén, L. 2002. “Artificial Stupidity: The Art of Intentional Mistakes”. In Rabin, S. (ed.) *AI Game Programming Wisdom*, 41-48.

Livingstone, D. 2006. “Turing's test and believable AI in games”. In *Computers in Entertainment*, vol. 4, no. 1.

Weber, B.G. 2009. “A data mining approach to strategy prediction”. In *IEEE Symposium on Computational Intelligence and Games*, 140-147.

Stiegler, A. and Livingstone, D. 2013. “Cooperative Artificial Intelligence in Games”. In *Computing and Information Systems Journal*, School of Computing, University of the West of Scotland, in press.

Stiegler, A. and Livingstone, D. 2013. “AI and human player cooperation in RTS games”. In *Proceedings of the Foundations of Digital Games*, 449-450.

Togelius, J.; Preuss, M.; Beume, N.; Wessing, S.; Hagelback, J.; and Yannakakis, G. 2010. “Multiobjective exploration of the StarCraft map space”. In *IEEE Symposium on Computational Intelligence and Games*, 265-272.

Wadley, G.; Gibbs, M.; and Benda, P. 2007. “Speaking in character: using voice-over-IP to communicate within MMORPGs”. In *Proceedings of the 4th Australasian Conference on Interactive Entertainment*, 24.

Weber, B.G.; Mateas, M.; and Jhala, A. 2010. “Applying goal-driven autonomy to StarCraft”. In *Proceedings of the Sixth Conference on Artificial Intelligence and Interactive Digital Entertainment*.

Yildirim, S. and Berg Stene, S. 2008. “A survey on the need and use of AI in game agents”. In *Proceedings of the 2008 Spring simulation multiconference*, 124-131.

EmohawkVille: Towards Complex Dynamic Virtual Worlds

David Holaň, Jakub Gemrot, Martin Černý
Faculty of Mathematics and Physics, Charles University in Prague
Malostranské náměstí 25, 118 00, Prague 1, Czech Republic
E-mail: david.p.holan@gmail.com, gemrot@ksvi.mff.cuni.cz, cerny.m@gmail.com

KEYWORDS
Interactive 3D Virtual Worlds, Intelligent Virtual Agents, Action Selection

ABSTRACT

Several recent successful computer games feature a large 3D open world where the player has a high degree of freedom and may roam freely through the environment. While graphical fidelity of contemporary open worlds is stunning, the behavior of non-player characters (NPCs) is kept rather simple. This is an opportunity for the academia to develop novel techniques and tools that would allow for easier creation of complex behaviors that are interactively believable — i. e. that remain believable despite unpredictable world changes stemming from player's actions. However, state-of-the-art techniques — both in academia and industry — are usually deployed only in 3D game environments with low complexity — i.e., the number of NPC actions is kept small. We think that if this complexity is raised, the state-of-the-art techniques will be challenged and will have to be improved if not transformed totally. However, no research-friendly complex 3D virtual world is publicly available. In this paper, we present EmohawkVille: an open-source first-person 3D virtual world that allows for simulation of a day-to-day life of NPCs and compare it with various environments already available. We discuss general issues arising during developing behaviors for the EmohawkVille domain. We also present a qualitative case study that confirms suitability of EmohawkVille for experiments concerning NPC behaviors.

INTRODUCTION

Many contemporary computer games take a great effort to achieve a high level of believability of their virtual worlds. This is especially true for games with large open world, where the user is free to discover the environment on his own and is relatively unconstrained by the game. Due to time and resource constraints and to the difficulties neccessarily involved in behavior design, non-player characters (NPCs) display complex behaviors only during crucial game events. In between, the NPC behaviors are rather schematic. The sole exception to this rule is the combat behavior, which is usually well designed and reasonably complex. But even recent and succesful open-world games such as Skyrim (Bethesda Game Studios 2011) or Red Dead Redemption (Rockstar Games 2010) have resorted to severely limited non-combat NPC behaviors.

The reason is that going beyond this simple behavior and still maintaining the suspension of disbelief introduces significant difficulties to the NPC behavior authoring. The designer cannot just script a peasant to go to a field and perform a harvesting animation — the player would expect the harvest to progress and eventually end with an empty field. The progress of the harvest cannot be scripted linearly because if a player action causes the peasant to be delayed on his way to the field or not to arrive at all, the harvest should not continue.

Even a very simple task, such as the harvest scenario, needs to take into an account many possible obstacles, if it is to become beliavable. What if the player steals the scythe from the peasant? What if a fight starts near the field? How about seasonal changes? Even navigating properly from a village to the field might be a challenge if there are dynamic obstacles, e.g. a chariot or band of villains. If such possibilites are not taken into account, the NPCs are easy for the player to "break" and may provide even worse illusion of a real world than rather static NPCs.

The harvesting scenario is just a single example. For a truly alive open world, dozens of different and often much more complex scenarios are needed, which implies that the world needs to be equipped with a rich ontology of items and actions NPCs (as well as the player) can perform.

As the world ontology grows, the number of meaningful NPC action sequences increases and the behavior complexity rises. Not only the means-ends analysis becomes more demanding, new problems emerge such as transitional behaviors, joint behaviors, behaviors ordering or behaviors interleaving (Plch 2009). At the same time, game studios usually cannot afford to let an expert programmer design such day-to-day behaviors, because that would be cost-prohibitive. Most of the NPC design is usually carried out with the aid of some visual tool by scripters with little programming experience.

At this place, academia could provide action selection mechanisms (ASM) and accompanying tools that would

help inexpert scripters to create complex behaviors that are interactively believable, that is, behaviors that sustain their believability under non-determinism brought by the player. However, most of the academic research is carried out in environments that either have simple ontologies or are static or discrete. Games on the other hand are dynamic, multi-agent environments that can be for all practical purposes considered continuous in both time and space. There are languages and techniques that can be applied to such worlds either from the multiagent community or the field of robotics or automated planning. However, to our knowledge, there is currently no 3D virtual world publicly available that would provide rich ontology for NPCs out of the box. This means that in this particular problem area, academia is one step behind the industry — we do not even have an environment to work with.

In this paper, we present our extension of the Pogamut 3 platform (Gemrot et al. 2009) called EmohawkVille[1], the first step towards an open-sourced complex simulation of NPC everyday lifes in 3D virtual world. We believe that creating a fully working, accessible and polished environment fosters academic progress. The large amount of research work evaluated on our first environment — Pogamut for Unreal Tournament 2004 — supports this view. We have also exerted great effort to make Pogamut and EmohawkVille a mature tool and we have resolved a large part of the technical issues encountered during NPC behavior development (sensors and actuators interface, navigation and pathfinding, character animation support, etc.).

The structure of the paper is as follows. Firstly, we go through available virtual worlds. Secondly, we present our newly created virtual world EmohawkVille together with a library that allows to create NPC behaviors in Java. Thirdly, we present a concrete implementation of complex behavior in EmohawkVille and report on an user-case study of the library with four programmers. Finally we discuss the suitability of the current version of EmohawkVille for academical research.

RELATED WORK

To experiment practically with NPC behaviors means to find a suitable virtual world first. Firstly, the world should be extensible with new items, game mechanics etc. as NPC behavior experiments typically require unique settings. Secondly, the world should support human players which is especially important for interactive storytelling or immersive projects. Thirdly, it should be possible to code the NPC behavior in a standard programming languages (such as C++, C#, Java etc.) either directly or via a foreign function interface. It is impractical to create behaviors in a sandboxed scripting language that is native to the chosen virtual world (e.g. Papyrus of Elder Scrolls V: Skyrim) as the availability of 3rd party libraries (e.g. Prolog support) for such languages is severly limited. Fourthly, the world mechanics should be easy to understand even for people with little gaming experience. Fifthly, we are not concerned with virtual worlds that put an NPC into a role of some disembodied entity such as a commander of an army in the world of StarCtaft: Brood War — the ASMs for such worlds are fundamentally different from behaviors of emboddied NPCs. Sixthly, price of the software should be considered. Finally, we are omitting raw frameworks like Unity, Unreal Development Kit or CryEngine, or even more basic 3D game engines such as Ogre or Irrlicht, as it requires considerable amount of time to build a virtual world from scratch.

We continue with the list of virtual worlds of interest and conclude the section with comparison of their traits.

ADAPT. ADAPT (Shoulson et al. 2013) is a platform for designing and authoring human characters. It mainly focuses on animations so interactivity is not sufficiently detailed. Furthermore, the tool relies on Unity Pro (Unity Technologies 2005), which is not available for free.

CAROSA. CAROSA (Allbeck 2010) models behaviors of virtual agents and seems to be focused on crowd simulations. Actions are defined via Parametrized Action Representation (PAR). The agent behavior is specified by its schedule, groups and roles, i.e., a data-driven approach. Unfortunately the tool seems to be unavailable outside the parent institution.

Bethesda games. Recent role-playing games by Bethesda Softworks, such as The Elders Scrolls V: Skyrim, feature somewhat detailed interaction. However, the scripting language Papyrus does not provide foreign function interface and offers considerably less expressive power than modern programming languages.

MMOs. There is a long tradition of character control software targeting popular MMO worlds, such as SecondLife or World of Warcraft. Unfortunately, usage of such software is usually against the terms of service of the respective game. MMO games also predominantly use a client-server architecture with the server application being unavailable.

The Pogamut platform. The Pogamut platform aims at providing Java interfaces for various games. The most prominent is the Unreal Tournament 2004 (UT2004) module (Gemrot et al. 2009). UT2004 is a first-person shooter (FPS) and as a such only actions related to combar are available.

The Pogamut platform also provides bindings for a custom city-like environment for Unreal Engine 2: Runtime developed for StoryFactory (Bída et al. 2011) project. Our EmohawkVille uses many assets originally made for StoryFactory. While UE2 is available for free, its license effectively prohibits creation of virtual worlds supporting human players.

[1]Can be downloaded at `http://pogamut.cuni.cz/main/tiki-index.php?page=EmohawkVille`

Table 1: Table of interesting features of various virtual worlds. EV stands for EmohawkVille. Availability is listed with respect to academic and educational use.

Virtual World	EV	UT2004	UE2	ADAPT	CAROSA	Skyrim	CTF
World Complexity	High	Average	Empty	Empty	High	High	Low
Dynamicity	High	Average	N/A	N/A	High	High	Average
Availability	Free	Paid	Free	Paid	Restricted	Paid	Free
Extensible	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Setting	Everyday Life	Armed Combat	N/A	N/A	Everyday Life	Fantasy	N/A
NPC's role	Civilian	Soldier	N/A	N/A	Civilian	NPC	N/A
NPC's Language	Java	Java	Java	C#	PAR, Tables	Sandboxed Script	Python
Human player	Yes	Yes	Yes	No	No	Yes	No

The relevant properties of the aforementioned worlds are shown in Table 1. Besides EmohawkVille, no virtual world possesses all the qualities we seek.

EMOHAWKVILLE

EmohawkVille is a FPS virtual world with detailed interactive elements of day-to-day life. For example, an agent can pick up a cucumber, put it on a chopping board and then start slicing it with a kitchen knife. EmohawkVille is based on Unreal Development Kit (UDK) (Epic Games Inc. 2009) and thus is capable of displaying the world in state-of-the-art graphics. UDK is free for educational and non-commercial use and EmohawkVille itself is available under GPLv3.

Architecture overview

Designing an environment for AI evaluation always (explicitly or implicitly) involves two parts: the actual mechanics that govern the virtual world and the interface the AI uses to perceive the state of the world and issue actions. In EmohawkVille the world mechanics are implemented in UnrealScript - a proprietary language deployed with the UDK toolkit. A low-level interface to the world is exposed through TCP/IP communication with a derivation of the GameBots protocol (Bída et al. 2012). The Pogamut platform provides a high-level Java interface for writing the actual AI and takes care of many common tasks (navigation, caching sensory data, etc.). Both the UDK and the Java part have been designed with possible further extensions in mind and the basic NPC support is separated from the model of the general EmohawkVille ontology which is in turn separated from the implementation of the specific mechanics for our cooking scenario (explained later). The UDK part also fully supports interaction with a human user through the UDK visual client.

Ontology

In this section we explain various aspect of the EmohawkVille ontology: it starts with general concepts and continues with a brief introduction to specific objects, actions and object-to-object interactions for our first EmohawkVille scenario.

We have decided to describe the ontology in great detail, as these "details" such as action durations or action interruption make an NPC behavior specification difficult in practice.

General Concepts

This section deals with generic concepts that govern EmohawkVille: avatar, action, simulation, process, item and container. A simplified class diagram is given in Figure 1.

Avatar. An avatar is a physical representation of an NPC or a human player in a virtual world.

Action. Actions performed by avatars either directly alter state of one or more objects or start a process that represents a durative action (see below). For example, a player or an NPC can set the power of a stove plate. EmohawkVille can be extended by providing custom character animation for any action.

Simulations and processes Simulations represent spontaneous changes of the world state such as a physical process taking place.

A process is a simulation of a long running action performed by an avatar. A process can be stopped by the interrupt action, e.g., an NPC may stop chopping the cucumber as soon as it chopped enough cucumber slices or because the water starts boiling in the pot.

Item. Items are lifeless objects that are meaningful for interaction. An item in EmohawkVille can either be directly present in the environment or it can be contained in an inventory of a possessor. Items can be dynamically created and destroyed. For example, the result of

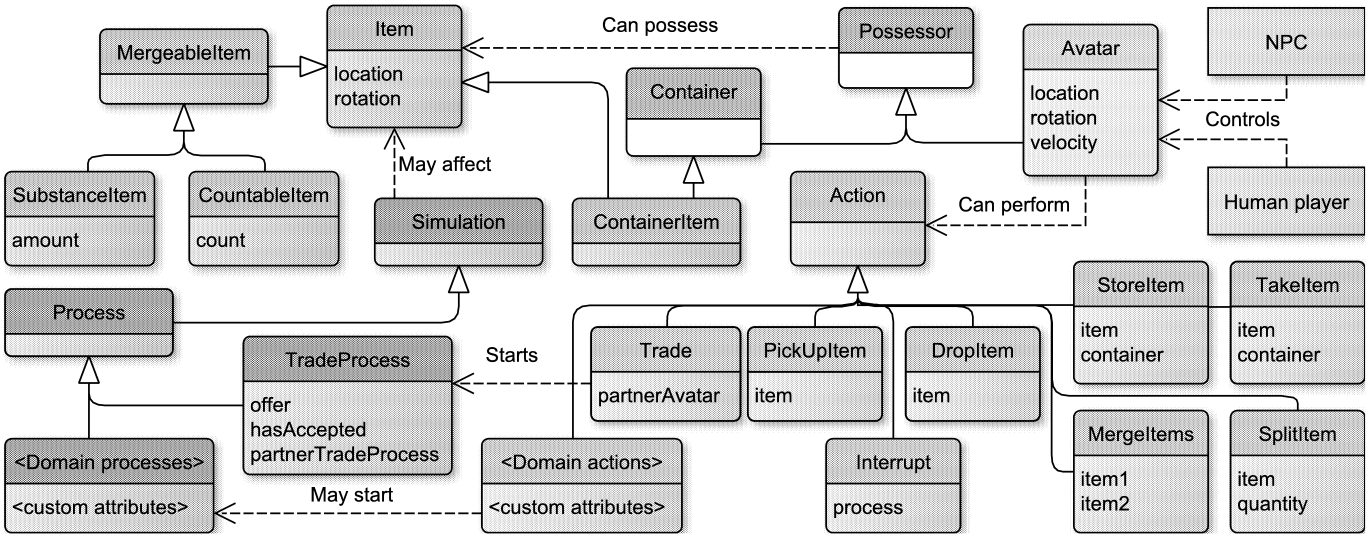


Figure 1: Class diagram of EmohawkVille's most important general concepts.

chopping a cucumber is the destruction of the cucumber and the creation of cucumber slices.

Possessor and container. A possessor is an entity that can own items. Possesors are either avatars or containers. In contrast to avatars that are active in the world, containers are passive objects (a pot, a wardrobe, etc.). Containers may constrain the amount/type of objects they store.

There are actions to pick up items, trade items with another avater, drop items in the environment or store them in a container.

Specific items and actions

The overall theme of EmohawkVille so far is cooking.

At this moment, EmohawkVille features 20 item types and the stove. Interaction is provided by 14 actions, of which nine are instant and four initiate a respective process, e.g. chopping a vegetable or stirring the broth. All actions can be performed both by an NPC and by a human player.

The central complexity of the NPC behavior stems from the simulation of cooking. Some ingredients can be boiled, some fried. Water evaporates from pots and ingredients may burn or char if not stirred or flipped. Concrete details are not relevant for the paper but can be found in (Holaň 2013).

CASE STUDY

To confirm that the EmohawkVille platform is suitable for NPC behavior experiments we performed a case study with four human subjects. The motivation of the evaluation was to 1) estimate overall usability of the platform for experiments with NPC behaviors and 2) to gain some insight on impacts of the increased complexity of the virtual world. Moreover, the evaluation helped us with identifying what parts of the virtual world and the Pogamut module are particularly hard to understand.

Empirical studies concerning usability of frameworks for NPC behavior development are scarce; we followed the methodology from our previous research (Gemrot et al. 2012). We designed 5 hours long controlled experiment with human subjects. Subjects were asked to solve two assignments — program NPC behavior for cooking tasks. Measurement of the subjects' performance during the task, questionnaires and structured interviews were used to gather both quantitative and qualitative data about the code subjects have written and their opinions about the EmohawkVille framework.

The crucial part of an experiment design are the tasks human subjects have to complete during the experiment. To design experiment tasks with an appropriate level of difficulty we firstly implemented a complex NPC for EmohawkVille ourselves called the Chef NPC. Insights gained from this implementation were then used to design tasks for the qualitative case study.

The Chef NPC

The Chef NPC was designed to use all items and actions the EmohawkVille world offers. It can prepare food consisting of multiple ingredients created by various cooking procedures.

Also the Chef can react to sabotage or help in an intelligent, albeit emotion-less, way, e.g., if the user puts water to the pot the Chef will not do it again. Additionally, if the Chef is unable to find required ingredients it may request them from other (human) players.

The action selection mechanism of the Chef NPC exploits a hierarchical goal-driven architecture similar in principle to behavior trees (Champandard 2007). The tasks of the chef NPC have the following functionality,

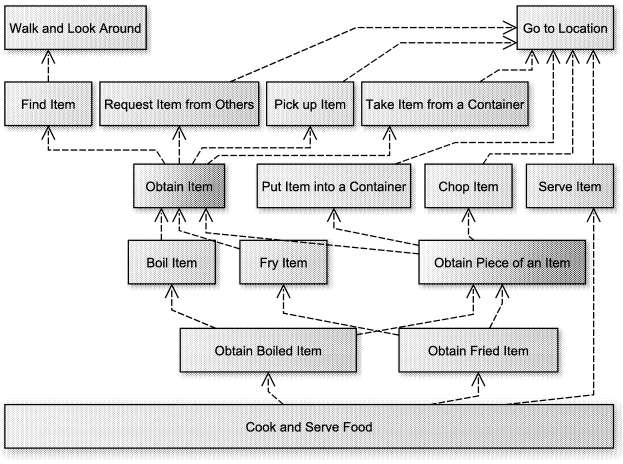


Figure 2: Dependency diagram of the implemented tasks of the Chef NPC. Yellow tasks were left in the NPC template for the case-study. Orange tasks were used as assignments for subjects of the case-study.

1) tell whether the task was finished with the distinction of success or failure to achieve the task goal, 2) execute logic for current frame that progresses the NPC towards fullfiling the task goal, 3) abort the task.

In every time step the agent executes its main task logic which in turn may execute one or more subtask logics. In the case of a static environment a task would simply execute a series of actions or subtasks in order. To achieve reactivity, the subtask that is closest to the end of the series and has its preconditions fulfilled is found in every logic iteration and executed. Thus the agent is both opportunistic and fault tolerant: it does not try to achieve subgoals that are already met and upon a hostile change in the environment it tracks back to the last unfulfilled subgoal. For example, if the NPC is to acquire cucumber slices and such slices already exist, reactive implementation can skip steps of collecting a whole cucumber and slicing it.

The examples of tasks used by our Chef are: acquire a certain item, go to a certain place, do a certain task (e.g. boil water or chop a cucumber), etc. See Figure 2 for the whole task hierarchy.

The implementation of all the tasks took about 10 man-days (80 man-hours) and was done by the creator of the Pogamut EmohawkVille module (the first author of the paper). This hints us that the desired behavior, although relatively simple at first glance is in fact fairly complex as even with high-level tools it tooks a non-trivial amount of time to implement. While the initial sketch of the behaviors was done quickly, the majority of the time was spent in determining possible points of failure, debugging and testing the behaviors. For example the relatively basic task of obtaining an item decomposes into four subtasks: 1) if the NPC can directly pickup the item, pick it up, 2) if the NPC knows a location of the item, move to that location, 3) if the NPC does not know of the item location search for the item 4) if the item cannot be found anywhere in the environment, find another avatar that possesses the item and try to trade it. Subtasks 2-4 cannot be directly executed in the environment and decompose further. Notice that this is a lower-level task (Figure 2). This shows that it is the complexity of the tasks and not user-unfriendliness of the platform that caused the 10 man-days long work and that better tools and techniques are needed for actual deployment of complex NPC behaviors in games. Note that we have just described a single scenario that is not backed, for instance, by buying ingredients or cleaning the kitchen up afterwards. We furthermore back this arguments with our experiences working on an AAA game project and that to our knowledge there is no larger scale commercial game that would involve NPCs with non-combat behaviors comparable in complexity to tasks on third level of our hierarchy not to mention the higher levels.

Experiment Setting

The experiment started with a 45 minutes long introductory presentation, which covered the mechanics of the EmohawkVille virtual world, the Pogamut module and the goal-driven architecture. The evaluation consisted of two assignments where the subjects were asked to implement a particular task in an NPC template based on the Chef NPC. Throughout the evaluation, the subjects answered appropriate questionnaires. All subjects had considerable previous experience with writing NPC behaviors for the Pogamut UT2004 module, and as a result, they could compare it with the Pogamut EmohawkVille module. The evaluation took 5 hours to complete (including the introductory presentation).

Assignments

As the implementation of the Chef NPC took days to complete, we had to scale it down for the subjects. We decided to create an NPC project template that was based on the Chef NPC code, but with empty implementation of "Obtain item" and "Obtain a piece of an item" tasks, which we asked subjects to implement. Tasks that were used in the original task implementation as sub-tasks were pointed out in the assignment.

The challenge of assignments lay in 1) learning how to use sub-tasks, 2) designing task stages, 3) implementing stage decision logic, 4) implementing the code executing each stage.

The "Obtain item" assignment was chosen for its relative simplicity. The task was to collect an item matching a given predicate.

The "Obtain a piece of an item" assignment was chosen as a representative of a complex task. The task required an NPC not only to produce slices obtained by chopping appropriate vegetable but also to satisfy chopping pre-

conditions (obtaining knife, gathering vegetable, finding a chopping board) and chop as many vegetables according to the number of slices required by the task parameter. For reliable execution, the task decomposes into 9 stages, none of which can be left out.

Hypothesis
We had two hypothesis. 1) The EmohawkVille with its Pogamut module is not hard to understand and use. 2) It is particularly hard to write robust NPC behaviors in a complex environment.
The first task was meant to produce data for the former hypothesis, the second task was meant to produce data for the latter hypothesis.

Subjects
We recruited four subjects for the evaluation. The first two subjects were students who finished the course on human-like agents (S1 and S2) and therefore had a considerable experience with development of NPC behaviors for the Pogamut UT2004 module. The other two subjects were researchers who developed parts of the Pogamut platform itself (R1 and R2).

Questionnaires
Every subject received six questionnaires throughout the evaluation. Questionnaires contained both questions with 5-point Likert scale (1 - "best/most", 5 - "worst,least") as well as open-ended questions. Questions were designed to 1) assess programming experience and skills of the subject, 2) check whether subjects understood the assignment and determine quality of their solution, 3) investigate usability of EmohawkVille and its Pogamut module, 4) report on the difficulty of the assignment.

Results and Discussion

All subjects understood the assignments well; The question "Have you understood the assignment?" was always answered by the option 1 or 2.
Researchers required considerably less time to complete the first assignment (Table 2). This can be explained by greater experience with Java, libraries and associated design patterns. The times to complete second assignment were similar for all subjects. One possible explanation is that first two subjects learnt the relevant skills during the first assignment, which would support Hypothesis 1.
Let us take a look at subjective difficulties of the first assignment (Table 3). The responses showed that the EmohawkVille Pogamut module is not sufficiently intuitive yet. Subjects reported on the confusing concept of observations and memorizations. This does not support our Hypothesis 1. On the other hand, the mechanics of the virtual world were found to be intuitive by all subjects, which in turn supports it.

Table 2: Times subjects estimated and actual times required by subjects to complete the first two assignments.

Assignment	**1**		**2**	
	est.	**actual**	**est.**	**actual**
R1	30m	47m	60m	91m
R2	25m	35m	45m	96m
S1	45m	62m	45m	95m
S2	20m	95m	60m	99m

Table 3: Reported difficulties of the first assignment, 1 - very easy, 5 - very hard. Subjects were asked what they think about difficulty of 1) the assignment (Assgn), 2) understanding the EmohawkVille Pogamut module (Module), 3) understanding the NPC template (Tmpl), 4) understanding the virtual world mechanics (Mechs), 5) formulating the NPC behavior (Form).

	Difficulty of				
	Assgn	**Module**	**Tmpl**	**Mechs**	**Form**
R1	2	4	2	1	3
R2	3	3	2	2	2
S1	2	4	2	2	2
S2	4	3	4	2	4

For the second assignment (Table 4), the first question was changed to be relative: "What do you think about the difficulty of the assignment in comparison with the first assignment?" (1 - much easier, 5 - much harder). Researchers reported increase in difficulty while students considered the assignment to be of the same difficulty or even less difficult. It became clear from open-ended questions, that this was because students had not realized the full extent of the task. Researchers were more aware about weaknesses in the code they have produced while students did not notice that some of their behaviors could fail. This partially supports our Hypothesis 2. It also provides ground for the research of tools suitability, as less-experienced user were not aware of problems with behaviors they produced; such as is the situation within computer game industry, where NPC behaviors are typically scripted by junior programmers as senior programmers are working on the game engine core.

CONCLUSION

In this paper we have presented the EmohawkVille virtual world. The case-study has found the platform to be suitable for experimenting with complex NPC behaviors.

Table 4: Reported difficulties of the second assignment, 1 - very easy, 5 - very hard. Subjects were asked what they think about difficulty of 1) the second assignment relatively to the first assignment (Assgn), 2) understanding the EmohawkVille Pogamut module (Module), 3) understanding the NPC template (Tmpl), 4) understanding the virtual world mechanics (Mechs), 5) formulating the NPC behavior (Form).

	Difficulty of				
	Assgn	**Module**	**Tmpl**	**Mechs**	**Form**
R1	5	3	1	1	4
R2	4	3	2	2	2
S1	3	4	3	2	2
S2	2	3	3	3	2

We have argued that the academia should focus on designing novel tools and techniques that would ease the burden of designing complex NPCs behavior, but such research is infeasible due to the unavailability of open-sourced 3D virtual environment. We have shown that EmohawkVille overcomes this obstacle.

As a future work we plan to gamify the environment by introducing a challenge inspired by the MasterChef TV show. Subjects will be challenged to create a team of cooperating NPCs that cook the maximum number of specified dishes within a given timeframe. During this experiment we aim to evaluate multiple behavior specification tools and guidelines to determine, which are the best suited for a practical application. We believe that testing tools on realistic tasks is a vital yet often underestimated part of the academic research concerning development of computer games.

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 (GAČR), by student grants GA UK No. 655012/2012/A-INF/MFF (70%) and 559813/2013/A-INF/MFF (30%). This research is partially supported by SVV project number 267 314.

REFERENCES

Allbeck J.M., 2010. "CAROSA: a tool for authoring NPCs". In *Proceedings of the Third international conference on Motion in games*. Springer-Verlag, Berlin, Heidelberg, MIG'10, 182–193.

Bethesda Game Studios, 2011. "Elder Scrolls V: Skyrim". `http://www.elderscrolls.com/`. Last checked: 2013-09-13.

Bída M.; Brom C.; Popelová M.; and Kadlec R., 2011. "StoryFactory — a tool for scripting machinimas in unreal engine 2 and UDK". In *Proceedings of the 4th international conference on Interactive Digital Storytelling*. Springer-Verlag, Berlin, Heidelberg, ICIDS'11, 334–337.

Bída M.; Černý M.; Gemrot J.; and Brom C., 2012. "Evolution of gamebots project". In *Proceedings of the 11th international conference on Entertainment Computing*. Springer-Verlag, Berlin, Heidelberg, ICEC'12, 397–400.

Champandard A., 2007. "Understanding behavior trees". *AIGameDevcom*. URL `http://aigamedev.com/open/article/bt-overview/`.

Epic Games Inc., 2009. "Unreal Development Kit documentation". `http://www.unrealengine.com/udk/documentation/`. Last checked: 2013-07-01.

Gemrot J.; Brom C.; Bryson J.; and Bída M., 2012. "How to compare usability of techniques for the specification of virtual agents' behavior? an experimental pilot study with human subjects". In *Proceedings of the 2011 international conference on Agents for Educational Games and Simulations*. Springer-Verlag, Berlin, Heidelberg, AEGS'11, 38–62.

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. "Agents for Games and Simulations". Springer-Verlag, Berlin, Heidelberg, chap. Pogamut 3 Can Assist Developers in Building AI (Not Only) for Their Videogame Agents. 1–15.

Holaň D., 2013. "Items and actions in 3D virtual environment Emohawkville". Faculty of Mathematics and Physics, Charles University in Prague. Master Thesis.

Plch T., 2009. *Action selection for an animat*. Master's thesis, Charles University in Prague.

Rockstar Games, 2010. "Read Dead Redemption". `http://www.rockstargames.com/reddeadredemption/`. Last checked: 2013-09-13.

Shoulson A.; Marshak N.; Kapadia M.; and Badler N.I., 2013. "ADAPT: the agent development and prototyping testbed". In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. ACM, New York, NY, USA, I3D '13, 9–18.

Unity Technologies, 2005. "Unity documentation". `http://docs.unity3d.com/Documentation/Manual/index.html`. Last checked: 2013-07-02.

HTN or State Space - Who Should Do Planning in Your Game?

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

KEYWORDS
Action Selection, Planning, HTN, Comparison

ABSTRACT

There is an ongoing discussion in the game AI community, whether to use AI planning for controlling non-player characters (NPCs) in computer games. Recent years have seen implementations of both state space and hierarchical task network (HTN) planning in AAA game titles, each having specific advantages and disadvantages. This paper is concerned with the performance aspect of both technologies and proposes a general methodology for comparing performance of action selection mechanisms. Two case studies comparing NPCs controlled by the JSHOP2 HTN planner and SGPlan 6 and Metric-FF state space planners in a game-like competitive environment are presented and their results are statistically analyzed. First environment is a puzzle-like scenario with minor combat element and strong competition for limited resources. Second environment features two pairs of cooperating agents in a combat scenario inspired by the Gears of War series. It is concluded that in puzzle-like domains HTN planning is inferior to state space planning, but for the second domain, the results are not conclusive.

INTRODUCTION

A large number of contemporary games consist of a dynamic, continuous real-time world inhabited by the player and other non-player characters (NPCs). This is especially true for first person shooters (FPS) and role playing games (RPG) which represent a significant share of the game market. Many serious games also have similar properties. Since graphical representation of game worlds has become very close to reality, the perceived intelligence of the NPCs is starting to be of vital importance to maintain believability.

One of the cornerstones of AI design for NPCs in games is the action selection problem — what to do next? The traditional game industry answer has been to use either behaviour trees (Champandard 2007) or finite state machines (Fu and Houlette-Stottler 2004), providing essentially fully preprogrammed responses to various external stimuli. But this is slowly changing, the need for smarter AI and lower design complexity has raised demand for alternative approaches, especially various forms of AI planning. Both STRIPS-style (Fikes and Nilsson 1972) and HTN planning (Ghallab et al. 2004) have already been implemented in multiple AAA game titles (Champandard 2013).

There is a major lesson learnt during the process of adapting AI planning as proposed by academia to controlling NPCs in computer games: while planning is considered a mature technology from the academic point of view, its application to NPC control is far from straightforward and it is often not an appropriate technique. On the other hand, from the number of succesful implementations it seems that planning turned out to be beneficial for game AI — at least in some scenarios. So our first question is: can we determine and classify scenarios where planning is — in some sense — better than reactive techniques? And can we find out when HTN planning is more suitable than STRIPS-style planning?

To answer these questions, it is neccessary to first formulate what "better" actually means. In context of action selection mechanisms (ASMs) for NPCs there are at least three points of view relevant to game AI: first is the *performance perspective* — how good is the AI at achieving its goals in the environment? Or, since computing power is a scarce resource for game AI, what is the tradeoff between performance and resources needed? Second is the *designer perspective* — how difficult is it to design AI using the technique, how much time and knowledge or skills are needed? Third is the *entertainment perspective* — whether it is actually fun to play a game with such AI?

In this paper we propose a general methodology for comparing various ASMs from the performance perspective, but we also briefly touch the designer perspective. The entertainment perspective has been left out, because the perceived entertainment is to a large extent influenced by many factors other than the AI itself which imposes large methodological complications to the research.

To demonstrate the methodology, we present two case studies evaluating HTN and STRIPS-style planning in game-like domains. One of the scenarios is more puzzle-like with indirect competition, while the other is a highly dynamic combat situation incorporating multibody coordination.

The rest of the paper is organized as follows: first, the

two planning formalisms studied are presented, then related work is discussed and the methodology for our experiments is given. Next, the individual case studies are described and their results are discussed. The paper ends with conclusions drawn from the results and discussion of future work.

PLANNING FORMALISMS

In this section we briefly introduce the two AI planning approaches that we have investigated: STRIPS-style planning and HTN planning.

STRIPS-style Planning

STRIPS-style planning defines a planning domain in terms of states and actions. Actions are defined by their preconditions and effects. Preconditions specify states in which the action is applicable. Effects describe how the world state changes once the action is performed. A planning problem consists of a domain, an initial state and a goal condition. A solution to the problem is a sequence of actions that is applicable to the initial state (all preconditions of actions are met at the time of their execution) and results in a state that meets the goal condition.
In the most widely used planning formalism - PDDL (Fox and Long 2003), the state consists of a truth assignment to a fixed set of logical atoms. However such a representation is often unneccessarily large as many pairs of atoms are mutually exclusive (e.g. `at(home)` and `at(garden)`). For this reason many PDDL planners internally represent state by values of a set of discrete state variables whch are automatically inferred from the original domain definition (e. g. Metric-FF (Hoffmann 2003)).
There are multiple ways to solve STRIPS-style planning problems. Currently, the fastest planners perform *state-space planning* — a heuristic search over the state space starting from the initial state. A nice property of this approach is that complete state information is always available for checking action preconditions and thus even complicated preconditions can be easily handled.
The most known planner used in games, GOAP (Orkin 2006) works with state-variables and performs a state-space search.

HTN Planning

HTN planning takes a rather different approach. A HTN planning domain consists of two types of tasks: primitive tasks, which correspond to actions in STRIPS-style planning and composite tasks. A composite task corresponds to a higher-level action and provides several decompositions — recipes how to accomplish the task with lower-level actions. Decompositions may be accompanied with conditions for their applicability. A planning problem in HTN is to gradually decompose a given task into a (partially ordered) sequence of primitive tasks such that all the decomposition conditions are held and the resulting action sequence is applicable to a given initial state.

RELATED WORK

AI planning has been involved in multiple commercial games. For a thorough review of both STRIPS-style nad HTN planning in commercial games see (Champandard 2013).
Some comparison studies of planning and reactive techniques have been made (Hoang et al. 2005, Long 2007, Cartier 2011). In general, planning techniques are found to be useful from the performance perspective if the environment is not too fast paced. In rapidly changing environments the results are less conclusive. Also only one of the studies involved statistical analysis of the results to distinguish between possible random coincidence and "real" difference.
A simple comparison of PDDL planners and reactive techniques in various dynamic conditions in a 3D game-like environment has been performed in our previous work (Cerny et al. 2013). It showed that, unsurprisingly, planning is advantageous only when the environment is small or it is not very dynamic or if it is very hostile. However the work tries to pinpoint the exact transition points and has let me test a statistical methodology to evaluate similar experiments.
We are not aware of any direct comparison of HTN and STRIPS-style planning.
Some research has also been done in comparing usability of agent programming languages to develop virtual agents (Gemrot et al. 2013). The methodology devised in the paper and previous work of the authors is inspirative for comparing algorithms from the designer point of view.

COMPARISON METHODOLOGY

In this section we present a general methodology for comparing ASMs. We mostly focus on comparisons from the performance perspective, but a short discussion of designer persepctive evaluation is also given.
For the purpose of computer games, it does not make much sense to evaluate the ASM by solving isolated problems. Thus we propose to incrementally create a broad set of different game-like environments, primarily multi-agent, to test the ASMs. Our previous comparison was done in full-fledged 3D game engine (Cerny et al. 2013), but creating complex scenarios in 3D continuous environments showed to be very time consuming. Thus for further research we have decided to simplify the environments to 2D simulations with game-inspired mechanics but running in discrete space and time. After sound results have been established for those simpler

environments a confirmative evaluation should be done in fully 3D and continuous environments to prove that the results do generalize.

The agents carry out actions in the environment and are given a real-valued reward signal at every time step. The performance of the agents is measured in two ways: the total reward accumulated during the simulation and the rank — whether it was better than other agents. Those data will be statistically rigorously analysed to determine significance of perceived differences.

Environment simulation is fully dynamic, i. e., it never waits for the agent to make a decision. One of the interesting questions is how does the performance of various algorithms change with varying speed of the simulation — that is, with different amount of processing power available to the agents. In the first part of the research, only fully observable domains will be considered, but some experiments with partial observability will also be run.

The action selection algorithms are wrapped up in *controllers*. A controller is responsible for running the algorithm, performing meta-reasoning about the deliberation process (e. g. interrupting a planner if the world has changed signicifantly since the start of planning) and gathering the results of the algorithm run.

Every environment has multiple *representations*. A representation is an interface between an environment and a controller. There may easily be multiple different representations (e. g., on a different level of abstraction) of the same environment for the same controller. The representation provides the controller with input data for the action selection algorithm and also provides information for controller's meta reasoning (e. g., what are the possible planning goals and what are their priorites, has the world changed considerably in last x steps, ...). The representation also translates high-level actions from the controlling algorithm (if any) into simple reactive plans that can be executed without deliberation (e. g., move to next room, shoot while the enemy is visible, ...). Thus the algorithms effectively do not need to be aware of the actual environment. The representations need to be carefully designed not to create significant advantage for any particular controller by translating its response into more powerful reactive plans.

An important part of the project output is a freely available codebase of environments and controllers [1]so that anybody can a) rerun the experiments and reproduce the results, b) test a new algorithm against any of the already connected, provided it can handle data in one of the supported formats and c) add a new environment to test the algorithms against as long as relevant representations are added.

The evaluation from a designer perspective is methodologically more complicated. Reliable results are very difficult to obtain without user testing (Gemrot et al. 2013). Due to high associated costs, user testing was not performed as a part of this study and was left for future work. However, personal experience from developing individual representations are an indicator of the actual difficulties

[1]The work-in-progress codebase is available at `http://code.google.com/p/aiste/`

CASE STUDIES

We have performed two sets of experiments to compare HTN planning using JSHOP2 (Ilghami and Nau 2003) and two PDDL planners: SGPlan 6 (Hsu and Wah 2008) and Metric-FF (Hoffmann 2003).

For the purpose of our experiments we have modified JSHOP2 code slightly. We have added a simplistic branch and bound optimization scheme to the original code and performed some technical adjustments for better integration in our experimental platform. JSHOP2 was chosen because it is in widespread academic use and has been repeatedly used in connection with game AI. JSHOP2 is written in Java.

SGPlan 6 and Metric-FF were chosen because they are in frequent academic use and have scored well at past International Planning Competition (IPC). Both planners use the PDDL language (Fox and Long 2003) as their input. The winner of the latest IPC (held in 2011), LAMA 2011 was based on the Fast Downward platform, which has shown severe obstacles for real-time planning in our previous study (Cerny et al. 2013) and would require different environment representation for efficient planning under severe time constraints, which we could not provide.

Both environments and all supporting code was written in the Java language. All experiments were run on a dedicated computing server with two AMD Opteron 2431 processors (6 cores each, 2.4GHz, 64bit) and 32GB RAM. Experiments were run in parallel while ensuring that each environment and each planner instance can occupy a full processor core without competing with each other.

Case Study I - Spy vs. Spy

The first domain is inspired by the 1984 computer game SpyVsSpy (Anon. author 2012). There are two agents moving on a grid of rooms. The rooms may contain one or several items, traps, trap removers and weapons. There are multiple types of items, traps and trap removers. The agents have the same goal: gather one instance of each of the item types and reach a goal destination. However, if there is a trap in a room, it has to be disarmed with a corresponding trap remover, otherwise an agent trying to pick up anything in this room dies.

An agent carrying a weapon may attack its opponent if they are in the same room. Such an attack has a 30% probability of killing the other agent. Weapons are

for single use only. Whenever an agent dies, all its belongings are "left on the floor". Agents are rewarded -50 points for death, 150 for reaching the goal with all items and -1 for each step of the simulation. The simulation ends when one of the agents reaches the goal or after a timeout.
Maps with 15, 30 and 70 rooms were randomly generated, 15 instances per size. The generation process ensured, that in the static case (i. e. if only one agent is present in the environment) the goal state was reachable for any starting position. Simulations were run at three different speeds - 100, 500 and 1000ms delay between steps. For each map and simulation speed, every pair of controllers was run twice. For the second run, the initial starting positions were switched to ensure fair comparison. This made for a total of 810 experiments.

Results
Let us first examine mean reward generated by each controller. There are two ways to examine reward. First is to examine the mean reward over all experiment runs, second is to calculate the *win percentage* that is the percentage of runs where the controller gathered greater reward than its opponent. The results for each controller pair are summarized in Table 1. We see that SGPlan is better than JSHOP2 and JSHOP2 is in turn better than Metric-FF and that the differences are huge. Indeed, the null hypothesis that two controllers are equally likely to gather greater reward in a single run has p-value $< 10^{-15}$ (tested using multiple comparisons of means with Tukey contrasts (Hothorn et al. 2008) over an ANOVA fit with a first order generalized linear model).
It is of interest, that when averaging over runs in a particular map size or a particular step delay the ordering is exactly the same and mean rewards and win percentage are roughly similar (less than 10% difference). Also the statistical significance of the differences is still high ($p < 10^{-5}$ in all cases).
From the designer perspective JSHOP2 was, in our personal experience, much more difficult to handle than PDDL planners. Developing PDDL representation for SGPlan and Metric-FF was very straightforward - in a discretized and nearly deterministic environment it is simple to describe individual actions. Developing HTN representation for JSHOP2 on the other hand was much more difficult. A direct decomposition resulted in performance inferior by a large margin to both PDDL planners. Involving domain-specific heuristics for ordering possible variable bindings and external A* path finding were neccessary for JSHOP2 to be competitive.
Domain-specific heuristics allowed JSHOP2 to find relatively good plans, but the planning time was still too high (432ms for maps of size 15, 647ms for maps of size 30 and 493ms for maps of size 70). The numbers should however be interpreted carefully: only succesful planning runs (those that found a plan) are included in the statistic, because there are often plenty unsuccesful runs which finish very quickly. However, succesful planning runs do not include runs which were terminated by the end of the simulation or radical change in the environment state and those interrupted runs tend to be longer than average (they ran long enough to be interrupted). This explains the decrease in succesful planning time for JSHOP2 in the largest domains. See the online appendix to this paper (Cerny 2013) for full planning statistics data.
The percentage of steps idle (steps agent had no plan and couldn't do anything) is a good indicator of planning performance. Metric-FF had huge issues even in the smallest domain (87% idle) while the other two planners had oscillated around 30% steps idle. The variance of JSHOP2 planning times was remarkably high (sd over 79% of the mean planning time for all map sizes), while SGPlan 6 has relatively stable performance (sd below 20% of the mean planning time for all map sizes).

Discussion
In Spy vs. Spy scenario, PDDL planning is more beneficial than HTN planning with JSHOP2. PDDL domain was easier to design than the corresponding JSHOP2 representation, yet with a correct planner, better performance was achieved. This is most likely due to relatively static and puzzle-like nature of the environment, but implementation details of JSHOP2 and the fact that it is in Java could have also played a role.

Case Study II - Cover Game

The second environment is called Cover Game. It is a combat scenario inspired by the Gears of War game series (Epic Games Inc. 2013) and also takes elements of gameplay from XCOM: Enemy Unknown (Firaxis Games 2013). In the scenario two pairs of cooperating agents fight each other with ranged weapons on a square grid map with lots of cover spots. Each pair of agents is controlled by a single controller, issuing actions for both bodies. In each turn, the agent may either move up to three squares, shoot at a visible enemy, suppress a visible enemy with fire (at most once per 2 turns) and enter full cover mode, if already in cover (full cover mode significantly increases cover bonus until the agent moves or fires). The probability of a succesful shooting is influenced by the distance to the target and its cover status. Agents that are a target of supression suffer a large aim penalty, but are not hurt. The damage done by a single shot is non-deterministic, on average 3 hits are neccesary to eradicate a target. A killed agent is respawned at one of the predefined locations. The team is rewarded +1 point for each kill and -1 for each death (i.e. it is a zero-sum game) and the game ends after a specific amount of turns. The agents have unlimited ammo and start to heal slowly if they are not hit for two turns.

Table 1: Overall Results for Spy vs. Spy. for every controller pair. The win percentage and mean reward along with standard deviation (in brackets) is reported.

Controller	Opponent	% Win	Reward	
			Controller	Opponent
SGPlan 6	Metric-FF	92	113.10 (46)	-24.45 (47)
SGPlan 6	JSHOP2	86	92.91 (70)	-21.13 (53)
JSHOP2	Metric-FF	87	92.94 (56)	-28.73 (64)

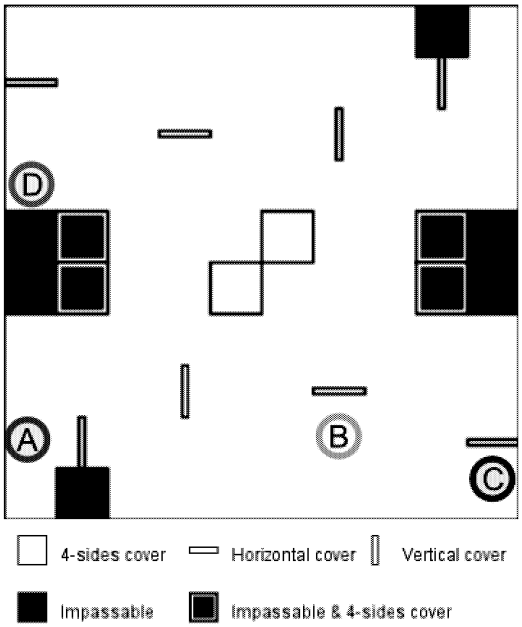


Figure 1: A sample scenario in Cover Game. 4-sided cover spots shield all four neighbouring tiles, while horizontal and vertical cover spots shield only the neighbouring tiles in the correct direction. Cover spots provide 180° shelter, e.g. agent B (yellow) is not covered from agent A (red), but agent C (blue) is. Agent D (green) cannot shoot at any other agent, neither can any other agent shoot at him because the line of fire is blocked by an impassable square.

Three manually created maps of different sizes were tested. The smallest one (10x10 tiles, called "Simple") is shown in Figure 1. There was also a larger map (22x20 tiles, called "Irregular", because it is not symmetric) and a very large map (27x37 tiles) inspired by the "Security" multiplayer map from Gears of War II.

Several different representations of the environment were developed. Two variants of low-level PDDL representation, where single action in the plan corresponds directly to a single action in the environment were tried. One forced the planner to alternate actions for one agent with actions for the other agent (this will be referred to as "LowAlt") and one which did not ("Low"). The idea was that forcing the planner to alternate would make it more difficult to find a goal, but the plan should complete faster, because of maximal parallelism.

A similarly low-level JSHOP2 representation was not tested in the final runs because even representation capturing only movement and finding cover had difficulties finding a single plan in a small environment within several seconds. Instead more high-level representations for both JSHOP2 and PDDL were created where the actions correspond to assigning "roles" to individual agents. The set of roles were the same for both representations and the domains were designed so that very similar role assignments are found by both representations. PDDL planners were thus tested with three representations. Since we are mostly interested with comparing techniques, matches between two instance of the same controller with different representations were excluded.

For each pair of controllers and representations, 4 rounds 200 simulation steps each were run at 100, 500, 1000 and 2000ms per step, resulting in total of 720 experiments.

Results

Results were analyzed with the same methods as in Case Study I, so refer there for details. A significant amount of draws were met, so the summary of draw percentages was added. The results are presented in Table 2. The message is clear, all of the role representations are significantly (all $p < 10^{-4}$) better than the low-level ones, regardless of controller. The differences between controllers with the same representation level are however insignificant (all $p > 0.7$).

The same pattern of results, although not always statistically significant, is repeated in all map sizes and step delays. This spares us strong conclusions, except for the fact that low-level representations are not practical in highly dynamic and non-deterministic domains.

However, looking at planning statistics reveals some interesting points. Most notably JSHOP2 plans very quickly (below 10ms). It was not enough to give him statistically significant performance advantage probably because the PDDL planners planned only slightly above 100ms which was the shortest step delay. The most likely cause is that PDDL planners are started in a separate system process and thus had the overhead of starting the process, while JSHOP2 ran in the same virtual machine as did the environment. See the online appendix to this paper (Cerny 2013) for actual numbers.

Table 2: Overall Results for CoverGame for every controller pair. The win and draw percentages and mean reward along with standard deviation (in brackets) are reported.

Controller	Opponent	% Win	% Draw	Reward	Significant
SGPlan 6 - Roles	Metric-FF - Low	79	6	4.42 (5)	yes
SGPlan 6 - Roles	Metric-FF - LowAlt	69	6	3.92 (6)	yes
SGPlan 6 - Roles	Metric-FF - Roles	50	6	0.56 (6)	no
SGPlan 6 - Roles	JSHOP2 - Roles	48	10	0.33 (5)	no
JSHOP2 - Roles	Metric-FF - Low	79	10	5.21 (6)	yes
JSHOP2 - Roles	SGPlan 6 - Low	77	12	4.65 (5)	yes
JSHOP2 - Roles	Metric-FF - LowAlt	75	12	4.12 (4)	yes
JSHOP2 - Roles	SGPlan6 - LowAlt	71	19	4.29 (5)	yes
JSHOP2 - Roles	Metric-FF - Roles	48	10	0.27 (5)	no
Metric-FF - Roles	SGPlan 6 - Low	77	8	5.54 (7)	yes
Metric-FF - Roles	SGPlan6 - LowAlt	77	12	4.58 (6)	yes
SGPlan 6 - Low	Metric-FF - Low	42	21	-0.06 (4)	no
SGPlan 6 - Low	Metric-FF - LowAlt	41	29	0.37 (4)	no
SGPlan6 - LowAlt	Metric-FF - LowAlt	38	29	-0.08 (3)	no
SGPlan6 - LowAlt	Metric-FF - Low	35	38	0.58 (4)	no

From our personal experience, implementing the high-level role representations for both PDDL and HTN planners was of roughly the same difficulty. But the design of the low level domain was quite cumbersome, as the mechanics of the domain such as probability to hit are not easily represented as logical atoms.

DISCUSSION

The experiment could not determine, whether HTN or PDDL planning is better, because the level of representation seemed to be the sole important factor. An interesting point is that the abstract domains with roles were only slightly more complicated than a simple behavior tree, but planners still needed a lot of time to evaluate it. For PDDL planners it is impossible to determine whether this was simply an overhead of starting a new system process, but JSHOP2 yields itself to more detailed analysis. 5ms is still a lot for a behavior tree evaluation, and translating the problem instance to JSHOP2 formalism is not included in this time. Also no string operations are performed during the translation or inside JSHOP2, as all atoms are encoded as integers. Profiling the JSHOP2 code showed, that over 90% of the time is spent working with variable bindings - although the domain would work exactly the same if variable binding changed to parameter passing. This could be of importance in practical HTN applications — if variable binding is not strictly neccessary, do not use it.

CONCLUSIONS AND FUTURE WORK

We have outlined a methodology for comparing action selection mechanisms and performed two case studies followin the methodology. The first case study showed than in puzzle-like environments, PDDL planning seems to be better than HTN planning. Results for highly dynamic CoverGame domain were not conclusive due to low statistical significance. But still the methodology showed to be feasible to apply and provide statistically supported results about various action selection mechanisms.

We have also noted that variable binding is an important performance factor in JSHOP2, even if the domain is so simple, that it is not needed. This should be noted for practical deployments of HTN planning.

An interesting point is that performance of both HTN and PDDL planners seemed to suffer similar penalty when scaled to larger domains or when under more time pressure.

As for designer perspective, our personal experience has shown that in a puzzle-like world, designing a HTN implementation requires much more effort than its PDDL counterpart.

Next steps consist of both extending the set of environments and connecting other action selection mechanisms such as reinforcement learning implementations and off-the-shelf probabilistic planners to the environments already designed. More thorough experimentation with the two domains presented in this paper should also yield further insights into HTN and PDDL planning applicability.

As a part of the research an open-source toolkit for

evaluating various techniques in game-like scenarios is being created. The toolkit will allow researchers from diverse fields to easily apply a novel technique to game-like domains and measure its performance therein. Such toolkit would foster interdisciplinary communication and has the potential to bring a unifying view on AI evaluation in games across the whole field.

From our previous experience in evaluating techniques from the designer perspective, the most promising approach for future work is to conduct user-studies as a part of labs or examination of students of relevant AI courses. The students will try to implement representations of the same environment for the individual algorithms (e.g. a PDDL domain description, a feature set for reinforcement learning). Then it will be measured how succesful agents are with those representations and the students will be asked to both quantitatively and qualitatively describe their desginer experience.

ACKNOWLEDGEMENTS

This work was partially supported by the student research grant GA UK 559813/2013/A-INF/MFF, by the SVV project number 267 314 and by the Czech Science Foundation under the contract P103/10/1287 (GAČR).

REFERENCES

Anon. author, 2012. "Spy vs. Spy (1984 video game)". URL `http://en.wikipedia.org/wiki/Spy_vs._Spy_%28computer_game%29`.

Cartier J.F., 2011. *Étude Comparative des Planificateurs Appliqués au Domaine des Jeux-Vidéos*. Master's thesis, Université de Montréal, Québec, Canada.

Cerny M., 2013. "HTN or State Space - Who Should Do Planning in Your Game?: Online appendix". URL `http://popelka.ms.mff.cuni.cz/~cerny/pddl_vs_htn_online_appendix.pdf`.

Cerny M.; Bartak R.; Brom C.; and Gemrot J., 2013. "Reactive and Planning Agents in Dynamic Game Environments: An Experimental Study". In *Proceedings of the 5th International Conference on Agents and Artificial Intelligence*.

Champandard A., 2007. "Understanding behavior trees". *AIGameDevcom*. URL `http://aigamedev.com/open/article/bt-overview/`.

Champandard A., 2013. "Planning in Games: An Overview and Lessons Learned". *AIGameDevcom*. URL `http://aigamedev.com/open/review/planning-in-games/`.

Epic Games Inc., 2013. "Gears of War". URL `http://gearsofwar.xbox.com`.

Fikes R. and Nilsson N., 1972. "STRIPS: A new approach to the application of theorem proving to problem solving". *Artificial intelligence*, 2, no. 3-4, 189–208.

Firaxis Games, 2013. "XCOM: Enemy Unknown". URL `http://www.xcom.com/enemyunknown/`.

Fox M. and Long D., 2003. "PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains". *Journal of Artificial Intelligence Research*, 20, 61–124.

Fu D. and Houlette-Stottler R., 2004. "The ultimate guide to FSMs in games". In *AI Game Programming Wisdom II*, Charles River Media. 283–302.

Gemrot J.; Hlávka Z.; and Brom C., 2013. "Does high-level behavior specification tool make production of virtual agent behaviors better?". In *Cognitive Agents for Virtual Environments*, Springer, vol. LNCS 7764. 167–183.

Ghallab M.; Nau D.; and Traverso P., 2004. *Automated Planning: Theory and Practice*, Morgan Kaufmann, chap. Hierarchical Task Network Planning. 229–252.

Hoang H.; Lee-Urban S.; and Muñoz-Avila H., 2005. "Hierarchical plan representations for encoding strategic game AI". In *Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment Conference*. 63–68.

Hoffmann J., 2003. "The Metric-FF Planning System: Translating "Ignoring Delete Lists" to Numeric State Variables". *Articial Intelligence*, 20, 291–341.

Hothorn T.; Bretz F.; and Westfall P., 2008. "Simultaneous inference in general parametric models". *Biometrical Journal*, 50, no. 3, 346–363.

Hsu C.W. and Wah B.W., 2008. "The SGPlan planning system in IPC-6". *Proceedings of the Sixth International Planning Competition*, 5–7.

Ilghami O. and Nau D.S., 2003. "A general approach to synthesize problem-specific planners". Tech. rep., University of Maryland Institute for Advanced Computer Studies.

Long E., 2007. *Enhanced NPC behaviour using goal oriented action planning*. Master's thesis, School of Computing and Advanced Technologies, University of Abertay Dundee, Dundee, UK.

Orkin J., 2006. "Three states and a plan: the AI of FEAR". In *Game Developers Conference*. vol. 2006, 1–18.

CLASSROOM GAMING APPLICATIONS

A NEW CONCEPT FOR TEACHING AI USING AS EXAMPLE CLASSICS FROM ELECTRONIC GAMES

Laura L. Dias, Alex F. V. Machado, Tamillys P. C. Pinto, Flávia L. C. D. Andrade, Rafael R. Padovani
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, {tamillyspantuza| rafael.rpadovani} @gmail.com and
flavia.dutra@ifsudestemg.edu.br

KEYWORDS
Artificial intelligence, data mining, decision trees, eletronic games, teaching tool.

ABSTRACT

There is a problem related to the introduction of Artificial Intelligence in computing courses, where often the student cannot fully absorb its functioning and object of study. Therefore, this paper proposes a practical tool to demonstrate the operation of Artificial Intelligence, showing the machine learning through data mining and generation of decision trees applied in electronic game simulations that were inspired by the most established game styles (like platform, puzzle and racing games) in order to facilitate its understanding by the results obtained. For this purpose, adverse situations were built in which the characters of the games would be submitted.

INTRODUCTION

The existing hypotheses of learning worry about the relationship between the object to be understood and the means necessary to understand it, with an emphasis on research about how students obtain, process and apply information.(Netto 1987)

According to (Nascimento et al. 2007), the learning of the adult learner is different from the one assigned to children and adolescents, as the adult seeks to transfer experiences in his everyday lives and real relationships experienced by him. The adult has a higher perception of his knowledge gaps and is quickly able to assimilate that ignores information or is unable to understand it, and can see how this can be detrimental to him. For this reason the adult learner is able to find ways to solve these problems. But universities continue to treat teaching an adult the same way that early childhood education is treated with the same teaching methods, which leads to the following question: Are the applications of pedagogical colleges considering the needs of adult education?

In another way, considering also that the teaching Artificial Intelligence (AI) suffers from the problem of the concept of intelligence is not always well defined or understood. Although many people have confidence that recognize intelligent behavior when he sees it, it is unlikely that someone can be specific enough to define the intelligence of a machine evaluated as intelligent and in parallel abstracting the complexity of a human being.(Luger 2004)

This requires a more efficient tool for approaching such discipline. However, the best known teaching model, which is based on lectures and written assessments, is still the most used, however this formal model where the learner only receives the information of the teacher is not the most viable since it doesn´t reach the full potential that each student needs to be able to transfer it to other areas of computing.(Scarton and Aluisio 2011)

It is seen that it is the duty of teaching in schools can make the understanding and perception of the activities students perform possible, providing an opportunity for them to be able to explain and argue accurately. It is believed that a method that approximates the real examples to learning process is an enabler for such a task, contributing to a clearer and more objective teaching organization viewed not only in theory but also in practical applications.(Freschi and Ramos 2009)

Given this, this paper presents a new approach to practical ways of applications in AI, using a teaching tool in video games, in a Course of Computer Science.

APPROACH TO THE PROBLEM

In his work (Bizzo 1998), the author says that in the last ten years many studies have been made all over the world, over the entire load of knowledge that students bring to schools before the formal knowledge acquired in the classroom itself. This knowledge has been termed "spontaneous knowledge, misconceptions or alternative conceptions." Then (Santos 2005) discusses several methods used to measure the real extent that students have about the environment in which they live before the teachers start teaching formal stage. More specifically, in regards to teaching science, "the maintenance of daily learning in students` concepts, after a certain period of school education has been recognized as it was difficult to obtain scientific concepts." The work of (Mortimer 2000) says that obtaining new knowledge does not mean abandoning previous conceptions, but understanding how they are

contextualized in their applications. The culmination of learning then becomes a conceptual profile evolved from the development of new phases of the profile and understanding where each situation is applied.

Based on these concepts, the intention of this paper is to create a collection of practical examples of the daily student to demonstrate how the AI works in electronic games, how the reaction of algorithms is associated with data mining and generation of trees, and how this may facilitate the understanding and teaching of Artificial Intelligence, reinforcing the burden of knowledge that students have brought from previous experiences .A model for application in electronic games that involves different environments and characters will be proposed. Each simulation has gone through the processes described in the concepts of AI algorithms demonstrating the machine learning through algorithms like APRIORI, M5P, J48 and others to verify their operation. All proposals presented used common data instances for classical games in the middle of video games, thus making it more common and easier to understand the tool development.
In the next sessions, all the steps proposed in each application made will be described. The construction of each one was based on AI knowledge and its usability in building games.

SOME CONCEPTS OF ARTIFICIAL INTELLIGENCE INVOLVED

This article includes relevant concepts within the area of AI that solidify the basis of this research, so important topics will be specified within this subject to enter the experiment. These being: Intelligent Agents, Machine Learning and Learning with Reinforcement.

Intelligent Agents

The mode of transmitting information that must be executed by the agent is passing him details about the program and letting it run. The attitude of the agent is to follow all the provided information, and only that. If a new situation occurs, the agent does not know how to treat it. Thus, the function is to pass on to him all that must be done in a general way and not for specific cases. It can be associated measures of performance or utilities in each situation faced as an evaluation technique. It is seen that the higher the index value will be the better its behavior will be. The function of the agent is to maintain the highest possible levels of usefulness without being told how to do it.(Rudowsky 2004)

Thus the agent may act in a particular environment in order to make autonomous decisions interpreting the situations that are captured by solving problems and determining his own actions.

According to (Nareyek 2002), computer games have grown in complexity since their inception, and more traditional approaches cannot be applied to them directly. Thus it can be said that Autonomous and Planning Agents become more suitable for such applications. However it should be noted how these tools may be appropriate in order to build electronic games that can be applied in a dynamic and real-time, and even in resource and invalid knowledge or incomplete.

This paper presents an approach for intelligent agents in applications for classic games and well-known dynamics.

Machine Learning

According to (Mitchell 1997), in learning machine, the computer performs tasks based on empirical data, of which the classification is one the most studied and it consists of mapping an instance represented by a set of attributes in a class, in the same manner that is represented in digital games where the technique of adaptation of the game to the player's actions can be utilized or to determine solutions to problems that are difficult to predict in the development process. While the game happens the instructor that is controlled by the computer can draw a profile of the human player based on their actions throughout a game session, so that you can minimize your actions and make the game more exciting and not so predictable.

The greatest challenge known in machine learning in games is the definition of the weights in evaluation functions. Programmers often apply to games routines that evaluate strong features, such as (amount of enemies, the territory size, available weapons, etc.) what is unknown is how to define the importance of each action and joining them appropriately, taking coherent meaning to the situation. The most commonly used mode is to linearly combine these parameters in order to obtain a weighted sum of the weights that train a classifier to reveal if the action set in a particular situation will gain, lose or tie and it isn´t often enough.(Furnkranz 2007)

Reinforcement Learning

According to the definition of (Campbell et al. 2009) in their work the reinforcement learning is a subset of machine learning in which a computer learns to solve a problem efficiently by interpreting information about its environment and its experiences. It is considered effective for converging optimal solutions most of the time, but with a great deal of attempts.

Their work (McPortland and Gallagher 2011) describes that over the past two decades, research lines focused on reinforcement learning have become promising, especially for robotics and multi-agent systems, in addition to classic games like backgammon and a certain lack of AI research to more modern games, such as first person shooter (FPS) with agents acting in complex environments.

A criticism by (Björnsson et al. 2004) in respect of this technique is related to the lack of efficiency, since many

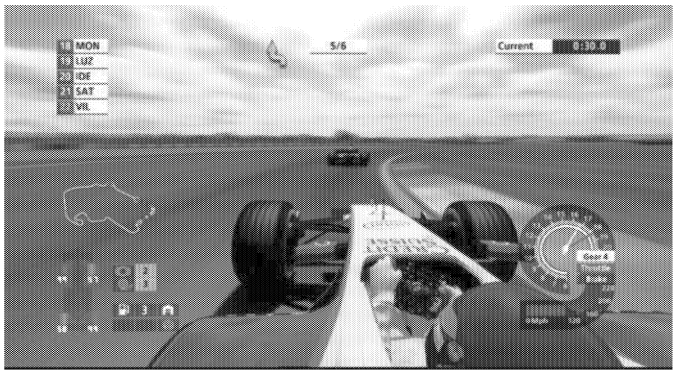
attempts are required before convergence of the solution, which makes it inapplicable for online gaming, for example, where is required that the answer is fast, even generating a great solution, many trials over time is a luxury we cannot afford.

Used Technologies

Based on the concepts studied in previous sessions and in order to show them in the practice the work has been developed using the algorithms APRIORI, J48 E M5P that were executed in Weka 3 to generate decision trees and make data mining.
The next sections describe these practical experiments that are the foundation of instances of work to the construction of the pedagogical tool.

A PROPOSED MODEL FOR RACING GAMES

In this session we developed a racing environment (Formula 1), in which notable features were evaluated for certain situations that could happen during the game. Those situations involved the players and they were also influenced by the environment. Following that, a list of parameters that were evaluated will be described. Such as the speed of the car, tire temperature, wear and so on. Was used as a parameter for the chances of skidding car, these were transformed into quantitative values to measure the importance of them for instance.



Figures 1: Racing Games (Sony Computer Entertainment 1996)

For data selection and evaluation of quantitative basis, M5P algorithm was used, thus generating the knowledge needed for machine learning and arriving at the following result:

- The larger the tire wear, the higher the chances of skidding will be.
- If there is little wear on the tires to cause the skidding, the track should be with the high humidity level and the speed of the car must be high.

It was used in the work, an assessment with qualitative parameters mentioned above, working with the J48 algorithm, generating similar rules of the quantitative basis.

Working with data mining, important conclusions were applied to Apriori algorithm to generate in the machine a relative idea of how much a user might be addicted to the running game due to the way he plays and conducts his moves.

Two Points of View for the Same Kind of Game

A second racing environment (Formula 1) was also developed, where similar characteristics to the previous section were evaluated, however, it was emphasized in other types of experience during the game. Situations that could pass during the game involved, besides the environment and the car, the possibility of the player to choose whether or not for a particular car model. So as before was also made a list of parameters, this time with the addition of some values such as acceleration and aerodynamics of the car that would be more specific values for each model, as well as climate environment. Because of this the changed classification parameter for which the player chooses certain car.

For each parameter, the intensity with which he applied in the game was passed. Using J48 algorithm was generated for a rule decision tree based on the consumption of tires within a qualitative scale of high, medium and low.

Working with data mining using the Apriori algorithm, the evaluation of how the player would behave towards certain choices was allowed. For this reason, we used the same parameters as before.

The rules generated by the Apriori algorithm,on the parameters passed as information, were satisfactory and reached the following conclusions:

- To:<medium acceleration>, <Low speed>, <rainy climate>, < medium speed track>, aggressiveness may be medium.
- To:<balanced aerodynamics>, <medium experimence>, the fuel consumption will be medium.
- To:<medium acceleration> and <medium fuel consumption>, <medium acceleration> and <soft suspension>, <medium fuel consumption> and <soft suspension>, aggressiveness may be medium.
- To:<moderate aerodynamics>, the acceleration may be medium.

Following the rules generated, the player will have higher chances of success in this type of situation during a game.

All the quantitative attributes for the game described, and their ratings in certain situations described were made for the

quantitative values 0, 1 and 2 to determine the choose of the kind of car for build the quantitative database.

Following is the table with the results generated by the algorithm using M5P instances cited above:

Table 1: Results Generated by M5P I

Dataset	carro_simulation
Numeric attributes	speed aerodynamics acceleration consumption/l suspension tire_wear tire_type climate lane lane_temperature take_car
Instances	11
Rule generated	tire_wear <= 1.5 : LM1 (6/0%) tire_wear > 1.5 : LM2 (5/0%) LM num: 1 take_car = -0.5 * tire_wear + 1.3571 LM num: 2 take_car = -0.525 * tire_wear + 1.125 + 0.2062
Number of Rules	2

In Conclusion, this application is based on the results of each algorithm.

A PROPOSAL FOR MODELING PLATFORM GAMES

A Focus on Experimental Mario World(Platform)

In this session, a simulation of a 2D platform game, similar to the situations in the game (Mario World) was developed. This experiment took into account not only the characteristics present in the environment, but also the characteristics of the actions taken by the character during the game. For that, parameters were used such as accessories of the character, size of the hole being overtaken by him extra help (Yoshi) and others. Was used as a parameter for the action that the character must perform these were transformed into quantitative values to measure the importance of them for instance.



Figures 2: Platform Games (Nintendo 1991)

The actions to be taken by the character are such as walking and jumping, walk, jump on the enemy and jump, run and jump and others.

The algorithm J48 using the parameters described above leading to a decision tree, a part of the tree is described below:

- If < small size hole > or <medium size hole> and <enemies on the hole = 0>
  - Action of character1: will be reasonable;
  - Action of character2: will be great;
  - Action of character 4: will be great;

- If <small size hole > or <medium size hole> and < enemies on the hole = 2>
  - Action of character 1: will be very bad;
  - Action of character 2: will be very bad;
  - Action of character 4: will be very bad;

For the data mining using the Apriori algorithm, whether the extra help (Yoshi) is significant or not to keep the character alive was taken into consideration. And in 80% of cases, extra support is important to avoid death of the character, reaching the following conclusion: if the character has an accessory and it is still in possession of (Yoshi) its chances of death are very small.

In this style of game, the same process which was previously presented in this article was submitted. The quantitative database was created and it was applied to the algorithm M5P, to be obtained knowledge of the weights of each action in the game and in the decision tree. All quantitative attributes of this modality and their ratings in certain situations are described as like before, using quantitative values to build the quantitative database.

The table with the results generated by the algorithm using M5P instances cited previously is as follows:

Table 3: Results Generated by M5P II

Dataset	mario_simulation
Numeric attributes	bonus hole_size hole_enemies yoshi dies
Instances	23
Rule generated	hole_size <= 1.5 : LM1 (15/40.86%) hole_size > 1.5 : LM2 (8/0%) LM num: 1 dies = 0.0813 * bonus + 0.2331 * hole_size + 0.1053 * hole_enemies - 0.1173 LM num: 2 dies = 0.106 * bonus + 0.304 * hole_size + 0.2062
Number of Rules	2

In conclusion, this application is based on the results of each algorithm.

A FOCUS ON EXPERIMENTAL WORMS(PUZZLE)

This session will be presented with a game environment very close to the game (Worms), considering the weapons that the character can use during the game and the obstacles he must overcome. For this experiment parameters were used such as wind, number of enemies, and away from others, which forms classified as a parameter using the weapon that the character should wear.



Figures 3: Puzzle Games (Team17 et al. 1995)

This should be chosen among the following options: <bazooka >, < grenade > and < rifle>

Using J48 algorithm, the following decision tree was generated:

- If <wind = favorable >
  - weapon: bazooka;

- If <wind = unfavorable > and <number of enemies = 1>
  - obstacle= thin: rifle;
  - obstacle= medium: rifle;
  - obstacle = thick: grenade;

- If <wind = unfavorable > and <number of enemie= several>
  - weapon: grenade;

In the process of data mining using the Apriori algorithm, the generated rules suggest that, if the wind was favorable, the character had a high level of life and there was only one enemy, the weapon which should be used is a bazooka. If the situation was several enemies and a close distance, the weapon which should be is the grenade. The algorithm suggested effective weapons for the situations described.

A PROPOSAL FOR MODELING FOR GAMES WITH SPECIFIC ASPECTS

In this session, an experiment of games where physical factors of the environment would act directly on the character was conducted, and it would pass through many situations generated by the scenery and the environment would be extremely influential during the game. For that, the parameters used were kind of terrain, wind speed character, angle and others.



Figures 4: Games With Specific Aspects (Naughty Dog 1996)

And as a classifying parameter, this was used:

- The character does or does not balance;

For the following parameters and instances provided, the J48 algorithm generated the following decision tree:

- If the < circular moviment = 90°>
  - The character: balances;
- If the <circular moviment = 180°>
  - The character: balances;
- If the <circular moviment= 360°>
  - The character: does not balance;

In data mining, setting some parameters, the behavior that the character should take in certain situations was evaluated and whether or not he would win the game. For this, new parameters were passed to the algorithm Apriori and it generated the rules described below:

- To: <jumps = 1>, <unstable terrain>, <medium spring>, the time will be insufficient;
- To: <speed = 0>, <strong wind>, the character will not probably win the game.
- To: <medium spring>, the terrain may be unstable.
- To: <unstable ground >, the spring may be medium.

The rules obtained give the player greater insight into their actions and demonstrate when the probabilities of failure are high.

ANALYSIS

Through this paper, a more didactic and intuitive form of teaching was tried to present for the content of the AI subject. By explaining these concepts using real content citing examples of our daily lives, students understand the purpose of each of them in an abstract way, but not exactly their functioning and logic, so they cannot apply them to some computational experience. In the case of the examples of games in which events are linear and pre-defined, the student understands what is being explained and, in addition, he can apply the concepts as intelligent agent, machine learning, and many others in systems developed for any area. These concepts are clearly visible by analyzing games, making this exemplification clearer and more objective.

All the tests taken in the classroom with students in the AI subjects of Computer Science courses were very satisfactory and the idea was very well received. They noticed that students could quickly understand the concepts and, at the end, they could set examples and forms of application of the contents demonstrated in several other areas.

CONCLUSION

According to the papers presented, it was possible to obtain parameters for a better understanding of the operation and modeling of AI in some classic games in the community and behold their potential for this integration in teaching programs of the subject in computer science courses.

This paper described some basic concepts of the subject that formed the basis for the understanding of experiments where the algorithms were run and generated results in both qualitative and quantitative instances that fed the data of the study.

The first study regarding the gaming platform of Mario World style by the J48 algorithm generated decision trees that showed that it is possible to classify the player's actions based on the characteristics of the environment and those ones developed by the characters throughout the game qualitatively.

The same happened in the experiments of puzzle games to Worms style and last on the physical aspects, which, given the conditions of the environment, the algorithm generated decision trees that serve as the parameter to the next player's actions, and in the case of this paper, it also served for teaching purposes.

FUTURE WORK

In this paper, a tool for application and exemplification of important parts of the AI subject in arcade classics was developed. This tool is still a new idea, but it is surprising by its usability and ease to understand.

The use of this tool was exemplified in this paper through demonstrations and theoretical analyses.

As a result of this research, a proposal is being prepared for the application of this methodology discussed in the paper to prove the improvement of student learning and skills in problem solving or even the creation of applications in the area discussed here. And it also intends to carry out the implementation of games and build their algorithms and source codes, which now enables the automation of the various analyzes presented in this article . In future work is of interest to develop the games presented here using the same techniques and knowledge of AI , and apply practical research on the reaction of the students before the constructed method was implemented for the purpose of real applications show significant results in their buildings, and satisfying experiences in teaching and educational learning .

REFERENCES

Bizzo, N. 1998. Ciências: fácil ou difícil? São Paulo: Ática, 1998.

Björnsson, Y.; Hafsteinsson, V.; Jóhannsson, Á.; Jónsson, E. 2004. Efficient Use of Reinforcement Learning in a Computer Game. In *Proceedings of the International Conference on Computer Games: Artificial Intelligence, Design and Education*, pp. 379-383.

Campbell, B.; Deo, P.C.; Hanganu, L. 2009. Muhlstein. Reinforcement Learning in Graphical Games. Rutgers University, New Jersey Governor's School of Engineering and Technology.

Freschi, M. and Ramos, M. G. 2009. Unidade de Aprendizagem: um processo em construção que possibilita o trânsito entre senso comum e conhecimento científico. *REEC: Revista electrónica de enseñanza de las ciencias*. Vol. 8, Nº. 1.

Furnkranz, J. 2007. Recent Advances in Machine Learning and Game Playing. *Oesterreichische Gesellschaft fuer Artificial Intelligence(OGAI Journal)*. Darmstadt.

Luger, G. F. 2004. Inteligência Artificial: Estruturas e Estratégias para a Solução de Problemas Complexos. 4. ed. Porto Alegre: Bookman, 774 p. ISBN.

McPortland, M. and Gallagher, M. 2011. Reinforcement Learning in First Person Shooter Games. In *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 1, March.

Mitchell, T. M., 1997. Machine Learning, McGraw-Hill.

Mortimer, E. F. 2000. Linguagem e formação de conceitos no ensino de ciências, UFMG. Coleção Aprender, Belo Horizonte.

Nareyek, A. 2002. Intelligent Agents for Computer Games. In *Marsland, T. A., and Frank, I. (eds.), Computers and Games, Second International Conference, CG 2000*, Springer LNCS 2063, 414-422.

Nascimento, E.; Nascimento, S.; Ferreira, S.P.A. 2007. Ensino universitário : estratégias e metodologias para a aprendizagem. Centro de Educação, UFPE. Trabalho de Conclusão de Curso. (Graduação em Pedagogia) - Universidade Federal de Pernambuco - UFPE. Orientador: Sandra Patrícia Ataíde Ferreira.

Naughty Dog. *Crash Bandicoot*. 1996. http://www.naughtydog.com/

Netto, P.S. 1987. A Aprendizagem como processamento da informação. In *S.P.Netto (Org.), Psicologia da Aprendizagem e do Ensino* (pp.79-109). São Paulo, SP:EPU.

Nintendo. *Super Mario World*. 1991. http://www.nintendo.com/countryselector

Rudowsky, I. 2004. Intelligent Agents. Proceedings of the Americas Conference on Information Systems, New York, August.

Santos, S. 2005. Para geneticistas e educadores: o conhecimento cotidiano sobre herança biológica. São Paulo: Annablume. Fapesp. Sociedade Brasileira de Genética.

Scarton C. E. and Aluisio S. 2011. O uso do MERLOT por Alunos de Teoria da Computação para a Criação de Materiais de Ensino-Aprendizagem. In: *XIX Workshop sobre Educação em Computação* (WEI - 2011), Natal - RN. XXXI Congresso da Sociedade Brasileira de Computação (CSBC - 2011).

Sony Computer Entertainment. *Fórmula 1*. 1996. http://www.scei.co.jp/

Team17, Microprose. *Worms.* 1995. http://www.team17.com/

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.

TAMILLYS PANTUZA COELHO PINTO 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.

FLÁVIA LUCIANA CAMPOS DUTRA ANDRADE has been teaching English for fifteen years. She has experience in English as a Second language(ESL) and English for Specific Purpose (ESP). She is graduated at Universidade Federal de Viçosa and she works at IF SUDESTE-MG, Campus Rio Pomba. She belongs to the Education Department and she has been teaching for various courses, including Bachelor of Computer Science.

RAFAEL RODRIGUES PADOVANI is graduate computer science by the IFSUDESTE-MG. Rio Pomba. He was a research at the group of Prof. Alex Fernandes da Veiga Machado, His main research interests are games.

A SERIOUS GAME CREATION PROJECT AS TEACHING METHOD FOR GEOGRAPHY

Merikki Lappi
Esa Lappi
Paivola School of Mathematics
Päivölän opisto
Päiväläntie 52
37770 Tarttila
Finland
E-mail: merikki.lappi@paivola.fi

KEYWORDS
Serious gaming, educational gaming, game creation, modeling, cooperative learning, problem based learning.

ABSTRACT

The Päivölä School of Mathematics has started a gaming project, in which the students learn Geography according to Finnish curriculum by creating sub models for an educational serious game that could be used by decision makers. The game is intended to give insight into the future challenges in humanitarian disaster relief operations. It will be created in cooperation with the international data farming community.

The game will simulate for example weather, food production, transportation, demographics and infrastructure during a global climate change, where the decision maker is trying to limit humanitarian disaster that might occur after extraordinary weather conditions.

The model is intended to be used in future Somalia type nations as well as to give insight into Finnish famine during the years 1867-1868.

During the project, the students create the sub models for the simulation game. The gaming project has connections with geophysics and geography, Finnish history, social studies, mathematical modeling, software engineering, paper writing, English language and teamwork.

Each student has his/her own area of expertise in accordance with cooperative learning methodology. The use of a real life problem is problem based learning.

INTRODUCTION

The use of educational games is an issue in the gaming industry. Playing a game can be used in training basic skills and understanding phenomena. (Tüzün et al 2009)

Creation of a game in order to teach software engineering or mathematics is not new, see e.g. (Cagiltay 2007), (Dell, 2011), but to use a game creation project in teaching is novel in Finnish teaching environment.

In this paper we present an ongoing game creation project, which is used to teach upper secondary school geography. Other topics are Finnish history, mathematical modeling, software engineering, paper writing, English language and teamwork. Educational framework is cooperative problem based learning. (Hmelo-Silver 2004)

The group of students

The Päivölä School of Mathematics (Päivölän kansanopiston matematiikkalinja) is a boarding school for mathematically talented students, which are selected using a weekend long incoming tests. They do the Finnish upper secondary school in two years with special studies in computing science and mathematics. Upper secondary school takes normally three years. The study program is described by (Hornyák 2011).

The students also work for Nokia 12 hours a week with different types of software engineering projects. Basic studies in university computer science are available. A short presentation of the teaching of computing science in Päivölä was published in EISTA 2009. (Lappi 2009)

The students of Päivölä have an outstanding record in science competitions, mathematics, computing science and European young scientist contests.

For example this year 3 out of 6 members in Finnish team in The International Mathematical Olympiad (IMO) were from Päivölä as well as one of two Finnish science projects selected representing Finland in the EUCYS – European Union Contest for Young Scientists in Prague. Three students participated this year in Intel ISEF science fair. The school has therefore a long experience in science projects.

This year the time allocated for science projects is used for the project described in this paper. Many of the projects over the years have been about computing and modeling. Therefore we decided we could try something more organized and more goal oriented.

The serious gaming project

This year we designed a novel serious gaming project, which combines geophysics, geography, some Finnish History and international cooperation into a serious gaming project. The creation of the game by the students is used as teaching method. Even the product can be used as an educational game and hopefully also in serious gaming.

The need for a serious game arose from the international data farming community. In the 26th what if? Workshop June 2-7, 2013 the team 4 outbrief had a topic "Climate Change and Humanitarian Assistance". The goal was to plan humanitarian assistance using data farming approach. This would need modelling, design of experiments and high performance computing.

The layout of the project

The project consists of several sub models that are combined to one model where the models use each other's results as parameters. The individual projects can reach different levels.

The first level for a sub model is a statistical distribution of possible outcomes. It gives for example the average temperatures in the area and its distribution. This level does require intensive reading in the chosen field but no advanced modeling or programming skills.

The second level is a simplified model, which adjusts the data according to global variables.

The third level is a simulation model, which gets global variables as input, and creates parameters for other models.

The fourth level is a simulation model which has interaction with other models in addition to global variables. The model uses input from the other models and creates output to be used by them.

Each student or a group of students chooses a field of interest and produces a full science paper that can take part in the Finnish Young Scientist competition (Tutki-kokeile-kehitä in Finnish). The paper can be on any of the levels above.

THE FINNISH CURRICULUM AS FRAMEWORK

In Finland the general topics for studies in each subject are given in *National Core Curriculum for General Upper Secondary Schools 2003*. (Finnish National Board of Education, 2003) The national core curriculum gives general guidelines for each subject and specific topics for each course (38 times 45 minutes). The local authorities and teachers have a lot of freedom in choosing how the topics are taught and learned.

The national core curriculum is used as the basis for creation of sub models of the serious gaming project. The project supports studies in several subjects like science and English language and ICT, but the topic modelled covers a major part of the curriculum in geography.

In geography there are overall goals and detailed structure of four courses: 1. physical geography, 2. culture geography, 3. geographical hazards, and 4. regional studies.

In this study we get the topics for each sub model from the selected parts of core curriculum. The citations from the National core curriculum in the following are written in *italic* with modeling focus for gaming in **bold**.

In modeling cycle adapted from Data Farming (Horne, 1999), (Horne, 2010) the interdepencies should be studied and the modeling process itself forces the students to think about the most important topics in each phenomena and interactions between different topics.

The preliminary team plan

The preliminary subject proposals are based on the geography curriculum. The students can make their individual choices and even choose another discipline like education or economy.

Teams #1 & 2 Climate

The idea is not to create new climate change models, but use the data from existing models in order to get forecasts and distributions for future climate and weather, and understand the weather mechanisms that caused famine in Finland in 1867 and today in Africa.

"The atmosphere in motion
*• **The structure and significance of the atmosphere**;*
*• winds and their origins; planetary and **regional winds and local winds**.*

The hydrosphere in motion
• the water cycle in nature;
*• the origins and **distribution of rain**;*
• seawater movements and their significance.

Weather and climate
• the weather and weather forecasting;
*• **temperature and climatic zones;***
*• **climate change**."*

Team #3 Food

The team links the weather to food production. Extraordinary cold weather froze the Finnish crops causing the 1867 disaster. The minimum level is a model combining the rain and heat conditions to food production, especially limits for serious hunger.

"The Earth's vegetation zones
• location and description of vegetation zones;
*• **conditions for and effects of human activity within different vegetation zones**."*

Primary production and the environment
*• **food production and supply**; sustainable agriculture and fishery;*
• different forms of agriculture;

Team #4 Economy (local and global) and team #5 Logistics

Transportation capabilities and costs for the overall game are the expected results of this topic. These teams combine extreme weather conditions with infrastructure. For example floods can destroy roads, bridges and airfields.

In 1867 the ice covered Baltic Sea made it impossible to ship food to Finland. The country could not have been reached by food even if there had been money to buy it. (Turpeinen, 1986) The global economy affects the ability to support a country in crises.

"Industry and energy
*• **raw materials and sources of energy;***
*• **location of industry**;*
• principles of sustainable industry and energy economy.

Mobility and interaction
*• **transport systems**;*
• tourism and its significance in different regions;
*• **world trade**;"*

Team #6 Population and culture

The team covers topics related to population structure, culture and administration.

"Population and settlement
*• **population trends and population growth;***
*• **regional distribution of settlements, migration trends and urbanisation**;*
*• **cultures and cultural changes** as well as minority cultures."*

Team 7 The map

Creates the game map.

Together the topics cover a whole that is meant to give the students insight into

"Natural hazards and hazard areas
*• risks related to outer space, endogenous hazards, **storms, floods**, drought and biological hazards;*
• preparation for natural hazards.

*Environmental hazards and hazard areas related to the **interdependence between human beings and nature***
*• hazards related to the use of natural resources: energy issues and supply of natural resources, **availability of clean water, erosion and desertification, climate change**, pollution and reduction in biodiversity;*

Human hazards and hazard areas
*• population growth and **hunger**, urbanisation, hazards related to globalisation, social and **political tensions, wars, movement of refugees and increasing social inequality**;*
*• **opportunities to control conflicts**."*

As the Päivölä students concentrate their focus on geography, the cooperation with North Atlantic Treaty Organization (NATO) Science and Technology Organization (STO) NATO STO MSG-124 workgroup and international Data Farming community contributes with the models for conflicts and conflict control.

The overall game requires a team which combines the sub models into a working software environment. This links the game map and the individual models. Global models are not directly linked with the map. (Figure 1).

Due to the challenge of such subproject, we have as an option to use old Päivölä students to join the team and possibly use it as a part of their university studies in computer science.

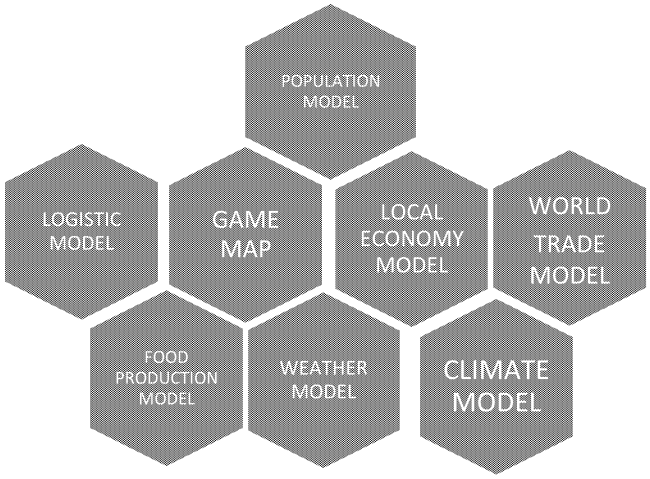


Figure 1: The model structure.

STEPS OF THE GAMING PROJECT – STUDENT'S POINT OF VIEW

1 Initial phase: research in the chosen field
2 Methology classes on datafarming and modeling
3 Modeling
4 Seminar: the students present their results to an audience consisting of fellow students, younger students and former students of which several have their PhD's in a related field and get feed back
5 Writing
6 Data farming workshop, with a playable version of the models
The time frame is from early October 2013 to January 2014.

STEPS OF THE GAMING PROJECT – TEACHER'S POINT OF VIEW

1 Initial phase: preliminary topics, background research
2 Introducing the topics and the methology to the students and motivating the students in their chosen topics
3 Individual tutoring
4 Planning and organizing the methology classes with emphasis on the needs of the specific groups
5 Supervising the work and helping out where needed
6 Organizing the seminar
7 Instructing the paper writing
8 Supporting the final stages of finishing the project in time

Time frame from mid June 2013 to January 2014

DISCUSSION

The multidisciplinary approach where we combine content with mathematics and computing is meant to motivate in an effective way students with different major interests. A real life problem solving task may make the tasks more motivating. The deeper insight into phenomena motivates the use of models and gaming as teaching methods. At the same time the students learn the methology in a problem based setting.

The humanitarian relief operation gaming project can be directly linked with Finnish core curriculum. Because a Finnish teacher has rather free hands in her teaching work, combining the Geographic and traditional science studies of Päivölä School of Mathematics can be done. So we have time resources available for the gaming project.

The data farming process was created for modeling and simulation purposes, but can be used for gaming purposes, also. From our experience from earlier modeling and simulation studies, the data farming loop of loops can be used for the guideline for student subprojects.

A real problem and work that might actually be used later is more motivating for the student. The international cooperation with actual goals gives extra boost to the work. When the whole group is working for the same project, the group cohesion is probably stronger.

The project with many teams without the same topic could lead each team only to focuse on their own part of the project. Due the interaction between the models the other team's work becomes familiar to the other students as well.

We have identified some risks in this type of teaching. First, all software projects can fail within the time frame of school year. The failure can be in the overall model or in each sub group. Thus each workgroup should start with the simple statistical data, and advance to the more complex models and probably some sub teams should stay on the first level of modeling. The process itself will help them learn geography, so most of the learning is achieved anyway even without completely successful overall model.

Some students can see the project as too challenging, and actually leave all the work to best programmers or a whole team can feel desperate and quit. To prevent our students from quitting and failing the presence and guidance by the teachers is available during the process and project is cut in smaller steps. The boarding school environment with teachers living in the same building as the students gives a good environment for guidance and encouragement.

The project requires good basic ICT and math skills of both students and teachers. If these requirements are met the type of instruction can be transferred to different environments and topics.

REFERENCES

Cagiltay N. E. 2007 "Teaching software engineering by means of computer-game development: Challenges and opportunities" *British Journal of Educational Technology* Volume 38, Issue 3, 405–415.

Hmelo-Silver, C E 2004. "Problem-Based Learning: What and How Do Students Learn? *Educational Psychology Review* September 2004, Volume 16, " 3, 235-266

Horne, G. "Maneuver Warfare Distillations: Essence Not Verisimilitude." In *Proceedings of the 1999 Winter Simulation Conference*

Horne, G. 2010. Summary of Data Farming. in *Proceedings 4th International Sandis Workshop*. Ed. Hämäläinen, J. Defence Forces Technical Research Centre Publications 23. 8-9

Hornyák B. 2011. "The mathematics programme of the Päivölä School" In *International Horizons of Talent Support, I* J.G. Gyori (Ed.) Hungary, 58-68

Lappi, M. 2009. "Cooperation Between a school with High Ability Students and a Technology Company." (presentation) *In Proceedins of 7th International Conference on Education and Information Systems, Technologies and Applications.* Orlando. http://www.iiis.org/CDs2009/CD2009SCI/EISTA2009/PapersPdf/E720SE.pdf

Finnish National Board of Education. *National Core Curriculum for General Upper Secondary Schools*. Regulation 33/011/2003

Turpeinen, Oiva; Nälkä vai tauti tappoi? Kauhunvuodet 1866–1868, Suomen historiallinen seura, 1986 (in Finnish)

Tüzün H, Yılmaz-Soylu M, Karakusb T, Inal Y, Kızılkaya G. "The effects of computer games on primary school students' achievement and motivation in geography learning" *Computers & Education* 52 (2009) 68–77

WEB REFERENCES

http://www.gamequarium.com/lessonplan.htm
Lesson Plan for Student-Created Games Project Written by Diana Dell, Gamequarium.com diana@gamequarium.com

AUTHOR BIOGRAPHIES

MERIKKI LAPPI was born in 1970. She studied first Geophysics at University of Helsinki but got her B.A in Nordic languages 2004. She worked for University of Helsinki Department of Geophysics during the years 1990 and 1991, worked as a part time simulation assistant in a fire protection engineering company Turvallisuusarviointi TA Oy from 1993 to 1997.

She has been working as full time teacher in upper secondary school level first in Maunulan yhteiskoulu (junior and upper secondary school) 1996-1997 and in Päivölän kansanopiston matematiikkalinja (Päivölä school of mathematics) and Valkeakoski Upper secondary school since 1997. She has been the leader of the mathematics program since 1999.

Merikki Lappi was given Nokia educational award year 2002, Academy of Finland teachers' award of Science competition Viksu year 2005, and Technology Industries of Finland Centennial Foundation award for mathematics teacher in 2007.

E-mail: Merikki.Lappi@paivola.fi.

WHY CHILDREN SHOULD PROGRAM COMPUTERS WITHOUT COMPUTERS

Euler Rithiele Q. Alvarenga
Department of Computer Science
IFSUDESTEMG, Minas Gerais, Brazil
E-mail: euler.sistemas@gmail.com

Alex F. Machado
Department of Computer Science
IFSUDESTEMG, Minas Gerais, Brazil
E-mail: alexcataguases@hotmail.com

KEYWORDS
children's learning, logic programming, teaching, kid games, learning techniques, psychology, algorithm, child

ABSTRACT

The pursuit of knowledge ever present in contemporary society goes through new motivations, currently the world of technology is increasingly interested in good professionals and has viewed favorably the teaching of "Be-a-ba" codes for children. Electronic games can have great importance in teaching, making learning courseware and intuitive, encouraging critical thinking and not letting them become just mere users in the digital age. Through techniques that stimulate curiosity and facilitates handling, any layman on the subject of computers may start "playing" with their little codes. The purpose of this paper is to discuss theories of knowledge bases correlating child with learning computational logic through recreational resources. The goal of a "programmer mind" early, not necessarily say what the child will be when it grows up as a professional, but the benefits to the child throughout their life will add to this learning.

INTRODUCTION

The teaching of computer programming and computational logic has been widely discussed in recent years. We made great efforts to develop special languages to support the first steps in education programming as an "easy start" for beginners (Mendelson, Green, & Brna, 1990) and even for the older. Steve Jobs said in his last interview, "I think everybody in this country should learn to program a computer, because it teaches you to think." With the significant expansion of new hardware technologies and open-source software, which has received investments from platforms like Arduino, Raspberry Pi, several universities and organizations like Code.org, Codecademy among others aim an acceleration in learning with greater facility for beginners.

Summarized in preparing little students for a world of more competitive, programming should be part of the curriculum of the students early, perhaps not in a compulsive way, because not all schools have the ability to do it, but at least as an interesting option, children taught not only how but also why.

"Do not tell them that computer science is so complicated," said Carlos Bueno, author of Laurem Ipsum (the subject of study of this work), learning these techniques is not about trace a certain future, set for kids. We would like to preserve the spirit of play and fun combined with learning, a child who doesn't know how to play can therefore be an adult who can not think.

Making a brief cultural observation, we see that both the Brazilian culture as in other countries, it is common for parents to encourage their children to practice activities complementary to school, they are: learning another language, ballet, physical activities such as swimming, soccer between other. We know that the ultimate goal is not necessarily turn them into diplomats or even olympic athletes, but it is fundamentally important that they do so, the world of new possibilities that will follow improvements in levels of social, health, even great names in the sport. Learn more about the codes do not necessarily make them "nerds", the importance of this study is through aggregated knowledge since early.

THE EDUCATION REVOLUTION

Forbes magazine notes that countries like Estonia, located in northern Europe, have invested in teaching programming to first graders. Across the country, children learn to write code. According to the Estonian government, the purpose and form relationships with people smarter with technology, computers and internet, and not just developers build applications for the future. "We dare, but do not know what will happen," said project coordinator Ave Lauringson ProgeTiiger.

Some researchers of this new trend care about some philosophical questions about it - why teaching programming receive special treatment? What is the reason children can not learn to read or calculate spectrograms outputs voltage suspension bridges? Another factor would be vocational, these children have open minds to think about software structures? Some of us are more have a greater ability to write poetry, or understanding the softer structures, such as psychology and economics. We are creating a problem making programming something special?

The ability to code allows people to "write" new kinds of things - interactive stories, games, animations, and simulations. The power of this learning is not limited to computers, extends to various segments of knowledge, there is a huge range of reasons to learn to program. Recent interest in the area sparked the rise of sites such as: Codecademy.com, Code.org that focus on the market already large and continues to grow rapidly. Companies such Google are investing in open source projects. The

company has funded fifteen thousand computers Raspberry-Pi, in partnership with six British educational institutions. Co-founder of Google, Eric Schmidt announced the move during a visit to Cambridge, where he teaches a class for children how to code.

"Guarantees such as this show us that such companies as Google not willing to wait for the government or someone else to solve the problems we are discussing, but I want to help them confront themselves."
Education has been revolutionized with these new strategies, in Brazil, schools already work with new methods, classes are used as a "bonus" to the curriculum. Robotics labs surrounded by parts and having computers connected to the Internet, "class" would not be the most appropriate term to define, students learn through play and are accompanied by teachers who act as partners in the research project, directing and answering questions. With this approach students learn to be empowered to make decisions and move on.

DEFINING A CHILD

The United Nations defines child as a human being below the age of eighteen. Child is biologically defined between the periods from birth to puberty.
We study the child in the light of psychology, emphasizing his childhood as a period of learning and new possibilities, an initial phase where human knowledge can be shaped through directions taken, aimed at cognitive development and social later.

The origin of the Latin word Childhood *infantia* refers to the individual who is not yet able to speak. This inability, regards to early childhood extending up to seven years. Clearly we can not define a child by his chronological age. Some authors establish that childhood has a generic meaning and like any other stage of human, to be a child is highly related to the social transformations of the medium in which it is embedded, defined (KHULMANN., 1998).

The Childhood defined by Piaget, Vytgosky, is in the range of the first years of life at twelve years old, which are developed stages of development, and despite being an analysis somewhat artificial, is constantly used as an analytical tool for explaining processes and characteristics on child development.

THE THEORY ABOUT CHILDHOOD'S THINKING

When we are trying to understand how the brain processes the information it imputed through in which we live, we seek to know how this information is stored permanently in our subconscious. Great theorists of psychology, as Piaget, Vygotsky concluded in behavioral and neurophysiological studies that the central nervous system only processes what arouses attention. According to the psychologist, pay attention because we understand that is because what is being presented has meaning and represents something new. (PIAGET, 1989).

Henri Wallon French physician and psychologist (1879-1962) advocated the concept of syncretism, for him, kids think merging the concept of imagination and reality (Salla, 2013), children observe the reality around them in a manner different from that used by adults. Children sometimes are caught saying curious things that do not make sense for the older ones, mix reality with fantasy. For Wallon children think in complementary pairs, think of an object by referencing it to another to justify their thinking.
The foundation of teaching children encode computer programs is the way they see the world, we can see that despite the possibility of understanding the students learn computer science as well as gain experience in writing computer programs. Arguably children learn through play, exploration and experience, awakening creativity techniques and communication.

For Vigostski (VIGOSTSKI, 2000) "[...] the concept is not an isolated training, fossilized and unchanging, but an active part of the intellectual process constantly at the service of communication, understanding and solution problems. "from this point, we can draw a parallel in the teaching of new concepts of learning, introducing the concept of logical thinking, solidifying early in the new little cognitive processes. Making training more gradual and evolutionary concepts, adding the "imaginary world" child, characterized by syncretic thinking triggering for future scientific concepts.
Cognitive processes are divided into four parts according to Piaget, he divides knowledge in the words learning and development. Target of our survey, children as young as seven years are shown to have the first contact with the codes and computational logic. The cognitive stages or transitional stages postulated by Piaget are (REIS, MARCIA, 2001).

The importance of the use of didactic ludic for this construction that introduces the teaching of computer programming for children makes it important to use this tool for a start easier and exciting. (piaget, 1989) stated that social interactions are of fundamental importance for the child to develop a mathematical-logical thinking.
In current times the children are presented with a multitude of things playful, sound-emitting toys, cartoons that "interact" with the viewer, online games that dispense tutoring adults (RUBIO, JULIANA, 2002). In this "ecosystem" as broad interactions and information [...] to discoveries, weaving relationships, organize thinking, logical reasoning, lie and be located spatially. Set up this way an initial framework of logical-mathematical references that require other, which can be enlarged. Are manifestations of skills, learning processes arising from informal, individual and cooperative relationship of the child in different environments and situations of different nature, for which there is no planning and control. (RCNEI, 1998).

USING CONSTRUCTIVISM AS A TOOL

Constructivism part of the belief that knowledge is not something that is done, finished, but a process in constant construction and creation. Thus, knowledge is established through action, the development and generation of a learning product that is the connection of being with the physical and social context in which they live, with the symbols produced by the individual and the universe of interactions experienced in society.
This construction is performed by action and not by gifts given previously to the subject, in the constitution of the genes or the environment in which he grew up. Thus, this method assumes that it is from the attitude that establishing the mind and consciousness, as well as our thoughts.

Constructivism is structured in the making, in writing que brings learning, nothing is finished or is genetically loaded it comes knowledge. The individual enhances creating relationships with the social and physical environment in Which They Live. The philosophical thinking Which Allows us to learn in other words doing and Relating without it being a method or a primer of steps to follow.
The game Promotes the child's action, Suggesting que challenges make it move in a process of cognitive construction through self-regulation. Self-regulation is one of the three forms of balancing and Occurs in integrations and differentiations between subsystems (parts) and all. This means que before a child is a challenge to structure able to Recognize and react to disruptive disorder modifying Their actions.

For Piaget the number is a mental structure that the child constructs. It is the synthesis of two types of relations: the order and inclusion. Ordinal and cardinal numbers are united in their constructions of logical standpoint. For this synthesis can take place before the child is required to build the concepts of classification and seriation. The first refers to sort objects according to the similarities, and the second has its ordering by the differences.

EDUCATIONAL TOOLS FOR TEACHING LOGIC

Second (Gardner 1985) each individual child in our case, is not endowed with the same skill set, so is the responsibility of the educator (he being parent or teacher) to see how the child behaves every form of knowledge presented to her and direct it to the most appropriate teaching tool.
For Vygotsky, the playful has great influence on the development of various areas of the child. Through games they are stimulated to new discoveries, curiosity, provides the development of language and concentration. (Vygostky, 1989). In the design of Piaget, the games consist a simple functional assimilation, an exercise of individual actions already learned generating even a feeling of pleasure in playful action itself and the control over the shares. Therefore, the games play dual role: to consolidate the schemes already formed and give pleasure or emotional balance to the child (Piaget cited Faria, 1995).

Basically we can divide the children into two groups:

• To those who can not read.
• Children who can read, but needing help to understand.

In both groups it is possible to teach basic learning computer programming and logical-mathematical. Observing these initial restrictions, but giving freedom to mix methods, where it is important that the child understands what is being taught and not necessarily maintaining a segregation of children by age as critical (PAPERT, 1980), we should give him the children will be able to control their learning process.

The logical-mathematical knowledge consists in coordinating relations and in the process of training and acquisition of the concept of number, the child goes through stages of mental construction.
The elements of paramount importance involved in the development of a logical-mathematical mind are:
• Classification, ordination;
• Logical sequence;
• Count (in different bases if possible);
• Inclusion of class;
• Intersection of Class;

How to present a logical-mathematical approach to such a child who still can not read? It would be possible to teach binary search or sort for a three year old? It seems absurd to talk about it, but even before reading this article already could have done it better or have had the opportunity to use this computational technique. Note:

I'm thinking of an animal and you have to guess what it is. You can do a linear search through the animal kingdom and find the answer. It is a duck? No. Is it a dog? No. Is it a fish? No. You can get some considerable acceleration knowing that I like dinosaurs and start with those I like, considering that the number of dinosaurs is small.
An improved algorithm is still looking for questions that allow you to eliminate entire kingdoms and phyla, no matter what the answer is. It's bigger than a duck? Does breathing air? Does it have fins? If you do it right, you will find the answer in about log (N) time, no matter the size of N.

Even within the fun in learning something new, launch use hand reading as a form of playful learning, awakening the imagination and complete immersion in an unfamiliar world. Paulo Freire describing the importance of the act of reading said that language and reality are linked dynamically, not only by the fact that it embodies the concepts expressed by that, but fundamentally because the act of reading develops critical awareness of the student and becomes a existential experience of its construction as an individual.
Montigneaux (2003) redid a division chronological child development based on the theories of Piaget and Winnicott reviewing concepts for a better understanding of how the child progresses relating to aspects of social and psychic stability (**see Table 1**).

APPROXIMATE AGE RANGE	COGNITIVE DEVELOPMENT	PSYCHOLOGICAL DEVELOPMENT	SOCIALIZATION
0-2	- Thought and action coming; - Limited capacity of reasoning; - Intelligence of the action; - Is the information in a more sensorial and perceptive; - Home of access to symbolic representation of the objects.	- Symbolic relationship with the mother; - Process of separation / individuation in relation mother (awareness of your own existence); - The child differentiates the inner (me) from the outside world (not me).	- It hides itself; - Difficulty of communication; - First steps towards autonomy for control of the body; - Parents are the primary agent of socialization.
2-6	- Ações são exteriorizadas; - É capaz de projetar no tempo e no espaço; - O desenvolvimento da função simbólica lhe permite representar mentalmente os objetos concretos; - Desenvolvimento da linguagem; - O raciocínio se apoia somente sobre a percepção das coisas; - Representação do mundo fortemente personalizada.	-Construction of a mental image of the mother; - Development of ghostly activity (imaginary dream and reality mingle); - Develop your character and acquire sexual identity.	- Development of social activity through imitation of adults; - Development of increasingly cooperative play activities; - The opening remains dependent child's social parents.
7-12	- Is able to perform mentally the ties that bind actions and concrete objects; - Acquire the concepts of logic, arithmetic, timeline; - Proficiency in writing.	- Construction and affirmation of the personality; - Progressive decrease of egocentrism; - Acquisition of moral values;	- Opens in relation to others and the world; - Influence of peer groups and the discovery of names governing group.

Table 1: Stages of child development. Source: Adapted from Montegneaux (2003, p. 32-54).

Table 1: Stages of child development.

In order to group these tools (games) in a more specific way for each type of child, defined primarily by age, rely on classification of Piaget, which categorizes by "changing structures" (Piaget cited Haydt and Rizzi, 1997) by formulating three major categories that correspond to the stages of child development.
Relating to the classification of Piaget, Wallon, we have three age groups correlate and has defined behaviors (**see Table 2**).

Who never had difficulty in solving a math problem? Dealing with such abstract concepts it becomes a challenge for the human mind. However, if well explored some important issues in childhood, such difficulties can be minimized.

Following the concept of playful learning emphasized by researchers of child psychological and pedagogical knowledge, several tools have been developed and even old practices are being reused for teaching basic mathematics and computational logic for children, with the main objective of creating an abstract thought but without losing the "magic"

Logic Blocks A simple game that helps develop mathematical thinking. objective: Recognize four geometric shapes; Identify colors, shapes and thickness; Working sequence; Logical intelligence-athematic Ages: 4-5 years.	

Table 2: Comparing authors and suggesting possible types of games through the minds worked. (**see appendix last page**).

of stories and jokes. In this part of the chapter will present some of these tools, the context of the problem and its objectives.
Since 1950, an important tool called "logic blocks" was developed and became very efficient for students to exercise logic and evolve in abstract reasoning. The logic blocks were created in the 1950s by Hungarian mathematician Paul Zoltan Dienes. Logical blocks were used consistently with children by Russian psychologist Vygotsky, when he studied the formation of children's concepts. It is an extraordinary material to stimulate in the child analysis, reasoning and judgment, leaving the action to then develop language.

Logical block, a set of geometric pieces divided into small squares, rectangles, triangles and circles by means of colors, sizes and thicknesses. They are intended to assist in the learning of children from kindergarten. This material allows the child to develop the rudiments of logic operations and their relationships as matching and classification, which are essential in the formation of math concepts. According to Piaget, "learning math involves physical knowledge and logical-mathematical. For blocks, the physical knowledge occurs when the child picks, observes and identifies the attributes of each piece. The logical-mathematical is when she uses these attributes without having the material at hand (abstract reasoning).

NOVELS AND RPG, THE LEARNING LOGIC THROUGH LITERATURE

Use the RPG (Role-Playing-Game) allied to the practice of reading can bring knowledge of computational logic in high level intruzindo the reader to the world of the seemingly more complex algorithms, Boolean algebra, recursion concepts among others, without the need actually type a line of code.
So this approach child must interpret the story being guided in an organized manner by the author to think like a programmer. Solving small problems, thinking logically,

always communicating with characters within the book, which in dialogues indicate options to be followed. These books are suitable for both children who already know how to read, and can do so independently and then being guided by tutors or even being told by their parents.
Carlos Bueno, Facebook engineer wrote Lauren Ipsum. Lauren a girl lost in the "Land of the User" in a literal translation. She begins her back home with only his wits and the help of friends like the chameleon Xor. Together they have to go through logic gates perfect swamps "Jargon-computational" face a little old lady with a book of puzzles. Where it is necessary to enable logical thinking to get home.

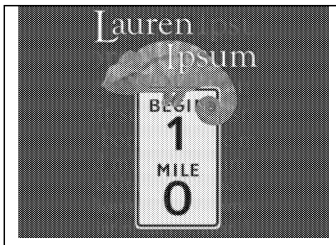
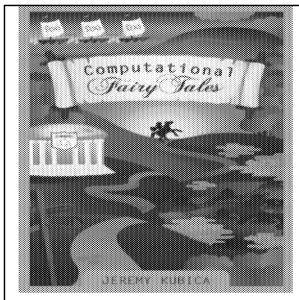
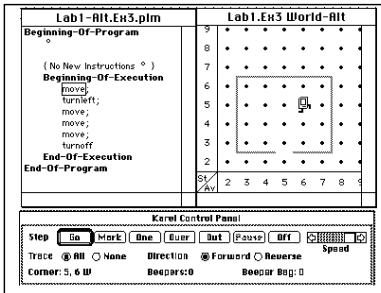
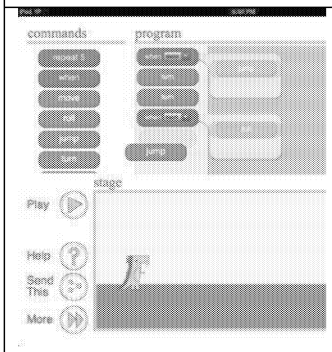
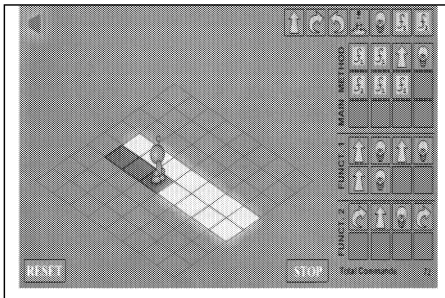
	**Lauren Ipsum** (author Carlos Bueno) Book that introduces algorithms and logic playful way, using methods RPG to guide the reader, games. **Objective:** Familiarization with Algorithms, Boolean logic; Organization of thought; Abstraction, solving puzzles; Logical-mathematical intelligence Linguistic intelligence; Age group (indicated): 6-12 years.
	**Computational Fairy Tales** (author Jeremy Kubica) "Have you ever thought that Computer Science should include more Dragons and Wizards"? **Objective:** Familiarization with Algorithms, Boolean logic; Organization of thought; abstraction; Logical-mathematical intelligence Linguistic intelligence: Age group (indicated): 6-12 years.

Table 3: Some examples.

USE OF MINI-LANGUAGES FOR BEGINNERS

Mini languages are special languages, with the aim of minors provide beginning simpler for beginners. This approach was first introduced by the charts "turtle" LOGO (Papert, 1980), with the success of other mini languages were created and encouraged.
Some features are important in a mini-language, simplicity of syntax and semantics. Most functions should be easily presented to the user, these functions will be performed by the actor in this microworld represented on the screen. The environment must be attractive and meaningful to the intended audience.
Another important advantage is that languages are mini constitruídas in metaphors, based on stories that are engaging and visually appealing. How is a language designed with only one purpose educadional and public with a well-defined target, it can renstringir only the small number of instructions providing only data types and structures essential for these students. (Brusilovsky, P., 1997).
For (Brusilovsky, P., 1997), the mini-language should be a modular language. It should contain a mechanism for the creation of abstract instructions (procedures). This procedure can be considered as a new command of the actor, which can be used for both navigation and programming mode.

	**Karel The Robot** (created by Richard E. Pattis) A programming language for beginners, published in the book Karel The Robot: A Gentle Introduction to the Art of Programming. **Objective:** • Familirization with Algorithms, logic; • Abstraction; • Introduction to the codes; • logical-mathematical intelligence Age group (indicated): 7-12 years
	iPad App Daisy The Dinosaur (created Hopscotch) Children learn the principle of programming, scheduling their own games, interacting with characters, music, sounds. **Objective:** • familiarization with Algorithms, logic; • Abstraction; • Creativity, imagine • logical-mathematical intelligence; • linguistic intelligence; • Musical Intelligence; Age group (indicated): 7-12 years.
	**Light Bot** (created Coolio Niato) Puzzle created to awaken the reasoning of logic programming and burn many neurons to solve their challenges. objective: • Logic programming; • Abstraction; • logical-mathematical intelligence; Age group (indicated): 9-12 years.

FUTURE STUDIES

The main objective of this work was to show the techniques used in studies by psychologists, showing the workings of the mind of a child to be able to from here not only work on games that bring intretenimento but also games that can teach them how they are made. However it is also possible

the academic community or large companies create ways to leverage the teaching of programming without the use of computers, with books, games that teach the principle of how things work, but in a totally fun and attractive.
Becoming or not a programmer in the future, learning the basics of programming is an efficient way for their brains to be trained in certain critical way of thinking.
Many research studies, magazine articles, educators say it is advised that children use the computer after 6 years with parental supervision. In this case further reinforces our goal to create and encourage new ideas for the big kid these tools.

That the future is immeasurable advances in technology and people to think more simply how the software for example.
Merging reality and imagination of children.

ACKNOWLEDGEMENTS

The authors acknowledge the Education Program Tutorial of the Ministry of Educatin (MEC/SESU/BRAZIL) and the *Instituto Federal de Educação do Sudeste de Minas Gerais* for the financial support guaranteed for the accomplishment of this work.

REFERENCES

Bwcker, Fernando 2000 . "O que é construtivismo" ? *Revista de Educação AEC, Brasília, v. 21, n. 83, p. 7-15, abr./jun. 1992.*

Freire, P. 2002. "A importância do ato de ler" 22.ed. São Paulo.

Facci, M. G. D. 2004. "Valorização ou esvaziamento do trabalho do professor? Um estudo crítico-comparativo da teoria do professor reflexivo, do construtivismo e da psicologia vigotskiana". Campinas-SP.

Gardner, H. 1985. "Frames of Mind: The Theory of Multiple Intelligences". New York: Basic Books.

Hargreaves, T. 2002. "Sujeito e subjetividade". São Paulo. *"O ensino como profissão paradoxal". - Pátio, Porto Alegre, ano IV, nº 16, p. 13-18, fev.-abr.*

Machado, M. L. A. 2004. "Educação infantil e sócio-interacionismo". *In: Oliveira, Z.M.R. (Org.). Educação infantil: muitos olhares*. São Paulo.

Morin, EDGAR. 2000. "Os sete saberes necessários à educação do futuro".
Piaget, J. 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).*

Papert, S. 1985. "Mindstorms, children, computers and powerful ideas" . Basic Books, New York.

Rothman, P. 2013. "Ensino Sob Medida".
FIALHO, FRANCISCO, REIS, MARCIA. (2001) Cognição: Piaget e Vygotsky.

Wallon, H. 1975. "Psicologia e Educação da Infância. Lisboa: Estampa". *Do acto ao pensamento. Lisboa: Moraes. 1979.*

Vigotski, L. S. 2000 . "Pensamento e linguagem. Martins "
Rúbio, J. *"Alfabetização Matemática: O universo lúdico infantil potencializando a aprendizagem – Revista Eletrônica Saberes da Educação" – Volume 3 – nº 1. 2000*

Vygotsky, L. S. 1999. "O papel do brinquedo no desenvolvimento". *In: A formação social da mente. São Paulo: Ed. Martins Fontes, 1989. 168p. p.106-118.*

WEB REFERENCES

Salla, F. , Iwaszko, T. - "Piaget. Apêndice Revista Escola, editora Abril, 2013 - Neurociência: como ela ajuda a entender a aprendizagem". Avaliable at [http://revistaescola.abril.com.br/gestao-escolar/neurociencia-como-ela-ajuda-entender-aprendizagem-691867.shtml?page=3.

Computer Weely, "Google Funds, 15,000 Raspberry Pit o Schools" - 29/01/2013. Avaliable at: http://www.computerweekly.com/news/2240177113/Google-funds-15000-Raspberry-Pi-microcomputers]

Forbes Magazine, 09/06/2012. "Why Estonia Has Started Teaching Its First-Graders To Code?" -. [Avaliable at: http://www.forbes.com/sites/parmyolson/2012/09/06/why-estonia-has-started-teaching-its-first-graders-to-code/].

Dignal, L. "Should kids be taught to program**?"** [Avaliable at: http://www.zdnet.com/debate/should-kids-be-taught-to-program/10117927/]

Bueno, C. 2013. – "Book Lauren Ipsum" – Avaliable at: http://books.google.com.br/books/about/Lauren_Ipsum.html?id=CMEsto2zN3UC&redir_esc=y

APPENDIX

COMPARATIVE TABLE OF DEVELOPMENTAL STAGES OF A CHILD

AGE RANGE	PIAGET	AGE RANGE	WALLON	AREAS OF INTELLIGENCE	AREAS TO BE EXPLOITED AND ADOPTED IN GAMES AND / OR ACTIVITIES
0-2	**Sensorimotor stage** (from birth to approximately 2 years): the child plays alone, without using the notion of rules.	0-1	Stage impulsive emotional (1st year of life) this phase predominates with the emotional environment. This is a construction of the subject.	Bodily kinesthetic intelligence: the ability to grasp large and small movements.	The cognitive activity during this stage is based mainly on experience through the senses shown that there is interaction with the environment, this is a practical activity. In the absence of language to describe the experiences and well recall the events and ideas, children are restricted to immediate experience, and so they see and feel what is happening, but have no way to categorize your experience, so the immediate experience during this stage, means that almost nothing exists between the child and the environment, because the mental organization of the child is in a raw state, such that the quality of experience is rarely significant, so that the child learns and how does remain as an immediate experience as vivid as any first experience. The visual search behavior is a sensory-motor and is essential for mental development, because this has to be learned before a very important concept called object permanence.
		1-3	Sensorimotor stage: this period is an intense exploration of the physical world dominated by cognitive relationships with the environment. The child develops practical intelligence and the ability to symbolize. At the end of the second year, speech and conduct representative (symbolic function) confirms a new relationship with reality, which emancipate the intelligence of most immediate perceptual framework. That is, when we speak the word "ball", the child will immediately recognize what it is.		
2-6	**Preoperatively** (from 2 to about 5 or 6 years): children acquire the notion of the existence of rules and start playing with each other games make-account .			Spatial intelligence: interest in breaking heads (shapes of plane figures and solids Logical-mathematical intelligence: interest in problems involving sequences and sorting; Musical intelligence: domain sounds, heights and shades;	For Piaget the most important game is the symbolic play (only happens at this time), this game dominates assimilation. The child, age three already assigns meaning to the design, making risks in horizontal, vertical, spirals, circles, however, gives no name to the drawing. Have a mental image after creating the desenho.Mas age four the child is already more creative and begin to realize their projects in design and drawing what you feel. At this stage the design phase is the most creative and diversified cirança The language in this period begins to be very egocentric, just socialized, ie, the language is focused on children. At this stage the child learns primarily intuitively, ie, performs free associations, fantasies, and assigns unique meanings and logical.
		3-6	Personalism (three to six years, approximately): this stage is the construction of self-consciousness, through social interactions, driving the child's interest for people, so the predominating relationships. There is a mixture affective and personal reworking, in thought, the initial lack of differentiation between intelligence and affectivity.		
7-11	**Stage of concrete operations** (7 to 11 years approximately): children learn the rules and play in groups. This is the stage of the game rules.	6-11	Categorical stage (six years and older): the child directs his interest to the knowledge and the conquest of the outer world, due to the intellectual progress that managed to win so far. Thus, it prints its relations with the surrounding greater visibility of the cognitive aspect.	Linguistic intelligence: ease of use of oral and written language;	is at this stage that reorganizes truly thinking. As I mentioned in the preceding stage children are dreamy, very imaginative and creative. It is from this stage (concrete operations) that begin to see the world more realistically, stop confusing the real with fantasy. It is at this stage that the child acquires the ability to conduct operations. We can define the operation as internalized action - held in thought, composable - composed of several actions; reversible - can return to the starting point. The child can now perform operations, however, need reality to perform the same, or have to get a sense of reality to make it possible for the child perform the operations. Rating - first the child has to group the objects by their class and size, then the sort and consequently acquire concepts. Numerical operations - first the child learns the concept of number and ranking, around seven years after the classification of reality, but this classification will vary according to learning that she will doing over time.

Table 3 : Crossover entre teorias de Piaget versus Wallon, sendo relacionado com conceitos cognitivos.

REAL WORLD APPLICATIONS

A monitoring tool to support remote caretaking of senior citizens

Aryan Firouzian
Henna Nissinen
University of Oulu
Pentti Kaiteran katu 1
90570, Oulu, Finland

KEYWORDS

Serious gaming, Physics and simulation, Design

ABSTRACT

Recent development of technology in the last decades has opened a window through new researches. How could new technologies aid senior citizens suffering from memory illnesses in their daily lives activities? This paper introduces a web application prototype as a tool to fill part of the gap between health care and technology. The prototype is designed like a computer game with an isometric view, to monitor and trace senior citizens around the home and city environment. Caretakers could use the application to monitor the present location of senior citizens and access their health profile.

INTRODUCTION

It is essential to understand how serious the problem concerning the amount of elderly people growing rapidly actually is, by showing actual numbers about the situation. The number of working age people in Finland is decreasing from 65 percent of the total population to 58 percent by the year 2030. This causes the demographic dependency ratio to weaken as the amount of people aged 65 years or more is increasing from 18 to 26 percent (Official Statistics of Finland 2012). According to United Nations (2002), the elderly population worldwide is increasing at a dramatically faster rate than that of the world's total population. It is projected that the relative percentage of elderly people will more than double by the year 2050 (United Nations 2002).

In this paper we discuss how smart technologies and services could support senior citizens in their daily lives and health care. A web application prototype, Old Birds, has been developed to support the main idea. Caretaking and continual observation of senior citizens requires a huge amount of effort, and monitoring just one senior citizen is a very time-consuming task. The prototype presented in this paper provides caretakers with a smart tool for tracing and caretaking numbers of senior citizens remotely, which allows monitoring their current location and health condition in detail. Besides of senior citizens, other people suffering from memory illnesses could also be considered to be monitored in daily life actions. In addition, relatives of each individual patient or senior citizen may want to participate in caretaking. The tool presented in this paper would be a solution to this problem.

A study about aiding tools for people suffering from Dementia in Scandinavia has been conducted in 2006-2008. According to the results, aiding tools can support and help maintaining the functionality of people suffering from memory illnesses. These tools support independency, the feeling of safety and reliability. They prevent harm and accidents and help the senior citizens to navigate in the environment. In addition, they can help to carry out better treatment and care, as well as ease creating and maintaining social contacts. The tools also ease the work done by the relatives of the elderly people and their caretakers and may have a positive effect on the mood of the user (Månsson et al. 2008).

Pulli et al. (2012) have presented a concept of smartphone-based navigation system for senior citizens and their relatives. The purpose of the system is to ease and to support the lives of the elderly people and to improve their quality of lives. The system can be used in the outside environment. Relatives or caretakers of a senior citizen can determine specific safety borders in advance and they can observe the movement of the person with an appropriate interface. The interface allows the remote user to see and review pictures and videos recorded by the mobile terminal of the senior citizen, observe the map and to communicate with the senior citizen. The system increases the awareness of the senior citizens and their relatives or caretakers of the situations that the senior citizens might face, ensuring a feeling of safety.

Siitonen (2011) has studied the safety navigation of elderly people in his Master's thesis. He has designed an imaginary system supporting the elderly people suffering from memory illnesses to navigate in the city environment with the help of a remote support person and a safety service. The functionality of the system was tested based on scenarios of a shopping process of an elderly person in a city environment.

Possakka (2011) has created a set of persona descriptions of elderly people as a part of her Master's thesis to support the scenario based design method of VESC (Value Creation in Smart Living Environment for Senior Citizen) research project. Scenarios are fictional stories or descriptions of some certain activities of the user. By using scenarios, the user point of view of using

technology can be taken into account properly. With persona descriptions, possible perceptions or assumptions of the elderly people that the designers might have can be avoided. These descriptions include essential information of fictional persons, such as the ability to operate, information about person's life and information about the living environment.

The proposed tool in this paper is partially based on the work of Pulli et al. (2012), Siitonen (2011) and Possakka (2011). The tool includes a 2.5D environment in a web application which is run by a game engine. A GPS device is the primary tool to record data of senior citizen's position. It will be possible to use other devices to get information and data about senior citizen's health situation. This is discussed in the future work section at the end of the paper. The data is stored and loaded by the Workflow engine and is displayed in a 2.5D environment graphical interface for the caretaker. Caretaker could monitor all the senior citizens simultaneously and get necessary information about their health condition by hovering the mouse over the avatars. This allows one caretaker to be responsible for monitoring several senior citizens. Each avatar is a representative of a senior citizen. Data related to the senior citizens will be loaded when a caretaker logs into the application.

Senior Citizen and Smart Services

When technology must react and response to the essential needs of senior citizens rapidly, most of the work in this area emphasize on the existence of real time systems. Zheng and Pulli (2007) have gathered the technical requirements of high quality mobile services for senior citizens in smart living environment. The requirements were sorted based on the importance. The most important requirement was the need of emergency function for fast emergency handling.

Using new technology in health care area is about supporting the daily activities of senior citizens, but it is also, more importantly, about involving them in new technology.

It is very important to consider that elderly people usually avoid using new technologies in their daily lives, so the system should be imperceptible and as easy to use as possible (Pulli et al. 2010). There are also some issues raised by smart home and monitoring technologies of elderly people, such as privacy concerns of monitoring capabilities and issues with personal information (McLean 2011).

A study about the factors that make the technology interesting among the elderly people has been conducted by Sorri and Leinonen (2008). The information about the factors was attained by several qualitative methods. The most interesting factors revealed to be the support of safety, independency, memory support, health, and physical exercise (Sorri and Leinonen 2008).

One example is smart kitchen system, presented by Ikeda et al. (2011). The system is designed to aid elderly people suffering from memory illnesses. The main purpose of the system is to support the elderly people with cooking related tasks with the help of a remote person, providing both verbal and visual prompts for the user. A typical smart kitchen is equipped with graphical displays and speakers for prompting the user being supported, and with cameras and microphones for providing a view for the remote user.

Workflow Engine

Workflow in interactive media and software development can be used to improve the development process, as well as adding new features to the system as opposed to using classical software development approaches (Roudaki and Doroodchi 2009).

One of the reasons for the need of workflow engine is to test all the features of the software before implementing it in real world. A database for the workflow engine is designed to predict different probable life situations of senior citizens to have simulation similar to reality. There are different scenarios predefined for some hypothetical senior citizens for the present prototype. These scenarios include the daily routes of the senior citizens, when they are in the outdoor environment and walking around the city. A map is used to pinpoint the routes and GPS coordinates for hypothetical senior citizens. Then the predefined data, such as GPS coordinates, time, description and the whole scenario are stored in a database. The software loads the predefined data via the workflow engine to simulate the senior citizen's current situation and position in a 2.5D environment.

Using a workflow engine seems indispensable because it could play the role of a dynamic database for application. All the information from application and their situations and locations will be stored there, and all the rules and scenarios will come from the workflow engine.

IMPLEMENTATION

In this section, we describe the proposed tool and the design process of the application in more detail.

Web Application

A caretaker user interface should be developed to be user friendly and easy to use. It should be game like application to persuade caretakers to use it for a long time. The monitoring tool should run as a web application in order to be accessible via Internet, and it could be run on all the platforms. Developing the caretaker interface requires a game engine supporting 2.5D development. The designer should design each senior citizen's home for the inside environment, so the game engine must be able to import home models from different 3D model generators, such as 3D Studio MAX

or Blender (3DS Max Autodesk, Blender Foundation). The main purpose for developing a web application is that the tracking device such as GPS system can send information to a server application, and caretakers or families could track the senior citizens and help them in problematic situation.

After comparing different game engines, Unity game engine was chosen as a tool for developing the application. The decision was based on the fact that it had all the necessary capabilities in quality and performance such as supporting workflow, multiplatform, strong communities and so on. Unity allows developing behavior scripts in Unity JavaScript and C#. The editor allows quick playing, testing and editing, making the development process as fast as possible. Another useful feature is the ability to import external files or content objects, such as audio or textures. It is also possible to develop the game for multiple platforms. Unity supports animation and movement from one place to another and it allows modeling of larger environments such as cities (Unity Technologies 2013).

Application environment

2.5D isometric simulation environment makes the application more interesting for the user rather than just a simple 2D environment. Another reason for using isometric view instead of just 2D is considered as a part of future work. By developing and adding more features to the tool in the future, it would be possible to demonstrate what directions the senior citizen is heading and looking at.

Home environment should be designed separately and accurately for each avatar based on the real home decoration. The walls of the simulated home environment must have low opacity, enabling the user of the web application to monitor senior citizen through the transparent walls. In the other words, nothing should block the visibility when the caretakers are monitoring the senior citizens.

The city environment is simulated as simple as possible to be able to locate the avatars easily from the map. The main idea is to use a simple 2D map of the city in a 2.5D view. By simulating just the most important buildings in the city, the tool would be more user friendly for the caretakers (Figure 1).

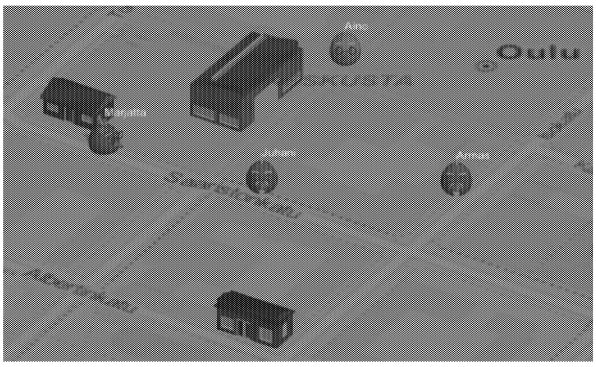


Figure 1: Simulating Senior Citizen's Home and Main Buildings of the City

It is possible to simulate different environments and places for each senior citizen, according to their daily lives. For example, the main shopping mall of the city of Oulu, Finland, is designed in the present prototype just because many local senior citizens spend some of their daily time in there (Figure 2).



Figure 2: Open View of Stockmann Shopping Mall in Oulu City, Finland

All views are combined and the avatars are able to change position from the inside view to the outside view when moving out from the home or from the shop, and vice versa (Figure 3). The original game coordinates are replaced with real map coordinates, so the tool shows the exact position of the senior citizen based on the real position in the city.

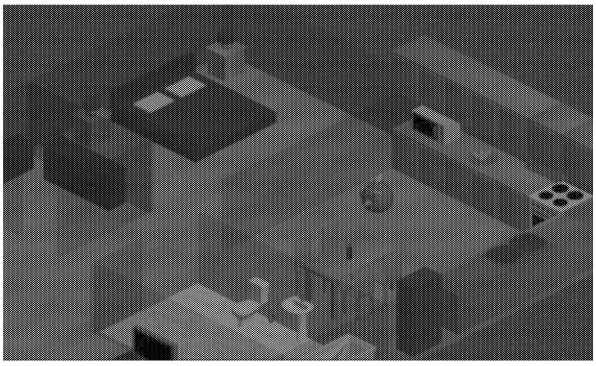


Figure 3: Senior Citizen Avatar Walking Inside the Home

Avatars

The main idea is to simulate senior citizens' daily lives in both, indoor and outdoor environment and thus, there is an essential need to provide models for senior citizens simulation. Human-like model is an interesting option for modeling senior citizens. In the first version of the tool all the avatars were human-like models and the models were representing the hypothetical characters. In this way the caretakers could imagine how the senior citizens look like and what is their gender. However, even though using human-like model avatar seems to be an appropriate solution, it may cause some serious privacy issues, as mentioned by McLean (2011). Simulating people may end up being offensive in case the designer makes mistakes in designing the human-like models.

To avoid privacy issues and to make the application more persuasive and interesting for the caretakers, the avatars are designed in cartoonish style. Old Birds models have been created to demonstrate senior citizens in the inside and outside views (Figure 4). There are different kinds of models for both genders and different color options so that the caretakers can easily recognize different avatars. The avatars are able to move from one place to another in the isometric view, for example from home to a shopping mall. In further development, the avatars should be able to follow a route defined in the map.

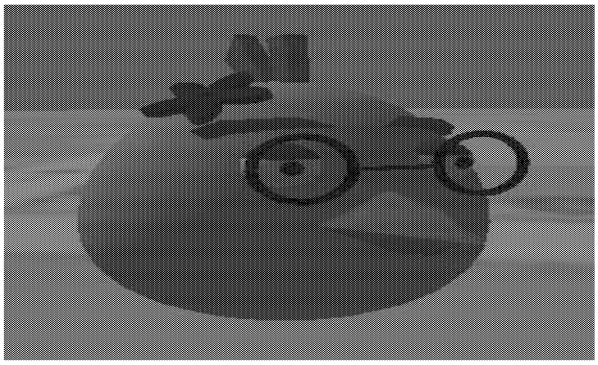


Figure 4: Female Cartoon like Avatar

The monitoring tool has an information box for each avatar. It is opened by hovering the mouse over each avatar. The information box contains personal information and health profile of the senior citizens designed by Possakka (2011). In the current version of the application this information includes age, medical information, living status, memory status and technology skills of senior citizens (Figure 5).

As a result, a video demonstrating the functionality of the system by using a typical elderly people life situation scenario was produced.

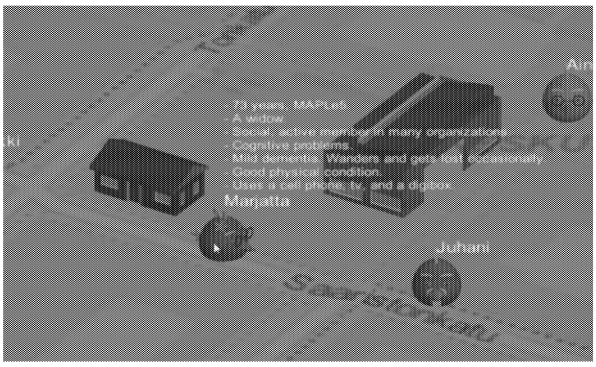


Figure 5: Information Box Appears Above the Avatar and Includes Essential Data about Senior Citizen

CONCLUSIONS AND FUTURE WORK

The present prototype is applicable only when senior citizens are using portable GPS devices. In the future, the application could determine the exact location of senior citizen's with the GPS information from the smart device carried by the senior citizen, as suggested by Siitonen (2011) and Pulli et al. (2012). However, there is an essential need to gain accurate information when senior citizens are moving inside the home as well. In the future, the tool could be developed further to simulate the kitchen environment, especially the smart kitchen by Ikeda et al (2011).

Another future feature which could increase the usability of the present application, is designing an additional caretaker interface for communication, such as allowing a phone call or Skype call. The communication tool would allow the remote assistance to have a better understanding of senior citizen's situation. The information box above each avatar could contain specific buttons, allowing the caretakers to make a phone call to senior citizens and their relatives.

Improvement in the quality and size of cameras and Internet speed in the last decade will lead to use wearable cameras in the near future. Wearable cameras could facilitate present application to capture snapshots or record videos from senior citizens' views.

ACKNOWLEDGEMENTS

This contribution is based on results from Spring 2013 semester student project at University of Oulu, Department of Information processing Science. We thank our supervisors Petri Pulli, Eeva Leinonen, Zeeshan Asghar and Davide Fucci for ideas and guidance. We also thank the other team members of the Old Birds project group: Huang Jiao, Hu Chuan and Keino Eliud.

REFERENCES

Ikeda, S., Asghar, Z., Hyry, J., Pulli, P., Pitkanen, A., Kato, H. 2011. "Remote Assistance Using Visual Prompts for Demented Elderly in Cooking". In *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, *ISABEL '11* (Barcelona, Spain, Oct 26-29). pp. 1-5.

McLean, A. 2011. “Ethical frontiers of ICT and older users: cultural, pragmatic and ethical issues”. In *Ethics and Information Technology*. 13 (4), pp. 313-326.

Possakka, S. 2011. *Ikäihmisen persoonakuvaus skenaariosuunnittelun avuksi* (Master's thesis). University of Oulu, Finland.

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

Pulli, P., Hyry, J., Pouke, M. and Yamamoto, G. 2010. "User interaction in smart ambient environment targeted for senior citizen". In *ISABEL 2010 The 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies* (Rome, Italy, Nov 7-10). pp. 1-5.

Roudaki, A. and Doroodchi, M. 2009. "Interactive Software Development Using Workflow". In *International Conference on Future Computer and Communication, ICFCC* (Kuala Lumpur, Malaysia, Apr 3-5). pp. 3-7.

Siitonen, M. 2011. *Muistihäiriöisten vanhusten turvanavigointi.* (Master's thesis). University of Oulu, Finland.

Sorri, L. and Leinonen, E. 2008. “Technology that Persuades the Elderly”. In *Proceedings of Persuasive 2008 - The third International conference on Persuasive Technology* (Oulu, Finland, June 4-6). pp. 270-273.

Månsson, I., Hurnasti, T. and Topo, P. 2008. *Apuvälineet ja Dementia Pohjoismaissa: muistia ja muita kognitiivisia toimintoja tukevat apuvälineet dementoituvan ihmisen arjessa: haastattelututkimus Suomesta, Islannista, Norjasta, Ruotsista ja Tanskasta.* Helsinki: Stakes. pp. 6. Retrieved from: http://urn.fi/URN:NBN:fi-fe201204193594

United Nations. 2002. *World population ageing, 1950-2050*. New York: United Nations. pp. 33. Retrieved from: http://www.un.org/esa/population/publications/worldageing19502050/

Zheng, X. and Pulli, P. 2007. “Towards High Quality Mobile Services for Senior Citizens in Smart Living Environments”. *In MUE '07 Proceedings of the 2007 International Conference on Multimedia and Ubiquitous Engineering* (IEEE Computer Society Washington, DC, USA, Apr 26-28). pp. 164-170.

WEB REFERENCES

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.

Unity Technologies. 2013. *Create the games you love with Unity*. Retrieved September 15, 2013, from http://unity3d.com/unity

3DS Max Autodesk. 2013. *3D modeling and animation software.* Retrieved October 30, 2013, from http://www.autodesk.com/products/autodesk-3ds-max/overview

Blender Foundation. 2013. *Blender is 3D creation for everyone.* Retrieved October 30, 2013, from http://www.blender.org/

SERIOUS GAMING HELPS CHILDREN VIEW HIGHWAY CODE FROM ANOTHER PERSPECTIVE

Anthoula Alipasali
Anton Eliens
Department of Information Sciences
Vrije Universiteit Amsterdam
De Boelelaan 1105,
1081 HV Amsterdam

KEYWORDS

Serious gaming, Highway Code, children, quiz.

ABSTRACT

Technology has made great strides nowadays and continues to do so in a very rapid manner. Therefore, at the moment, it is not only used just for scientific but also for educational purposes. It is exactly for that reason that people have started to use e-learning platforms. These e-learning applications help students study with a very different perspective.

However, even though e-learning has improved significantly the way of studying, there are some general issues that do not meet the anticipated expectations, especially as students at earlier ages are concerned. Those problems could be alleviated via the implementation of aspects of serious gaming. In this essay a description of the theoretical background of serious gaming will be attempted. After that, significant benefits of Highway Code will be given, as well as and the proposed idea for future work and dissemination of the game.

INTRODUCTION

Serious games refer to digital games used for purposes other than mere entertainment, but are used for advertising, training, education, or simulation that are designed to run on personal computers or video game consoles. They simulate environments and systems, and allow learners to experience cases that are not possible in the real life for reasons of safety, cost, time, etc. Analyses that have been conducted over the years consistently show that games promote learning.

Serious gaming is started being used for educational purposes a few years ago. Before that, people did not realize how many benefits serious gaming might had, the possibilities it presented and its full potential. Educational games transferred the learners to another world via experimental exercises. In those scenarios, the learners apply their skills, strategies and knowledge in order to manage the execution of the roles assigned [2]. In other words, serious gaming is a new type of e-learning that changes the way that students perceive the learning process and make them wish to continue learning. It can be said that serious games originated from e-learning, after all serious gaming has been characterized as —the newest trend in e-learning (Marc Prensky, Digital Game-Based Learning, 2000). In addition to these, serious gaming engages the students to encourage behavioral changes. It also helps students improve their learning results through training in a restricted and controlled environment and develop their practical skills.

Moreover, the Federation of American Scientists suggest that kids from 8 to 18 years old usually spend approximately 50 minutes per day playing video games [3]. This means that they are already adequately familiar with that kind of technology.

Furthermore, serious games cannot be considered as a theory, at least not yet. Commonly, many websites describe serious games as wanting to achieve something more than entertainment, and considers it more of a movement than a defined area of its own. For example, it has been described as a movement that is —cross-appropriating video game techniques, technologies, tools and structures from the video game industry to other fields of human endeavor (outside of entertainment) like policy and management issues. However, the domain of serious gaming has potentials to become an important one, when developments will make its limits more specific, its purposes more clear and the games that will be created are more based on science (psychology, sociology etc.) to target specific areas of interest to the players.

SERIOUS GAMES AND EDUCATION

Learning is acquiring new, or modifying existing, knowledge, behaviors, values, skills, or preferences and may involve synthesizing different types of information. The ability to learn is processed by humans, animals and some machines. [6]

A knowledgeable pedagogue who constructed and transmitted knowledge on a specific topic to people (learners) via

some accepted instructional technologies such as books, prelections, and articles is a typical procedure for learning. It was believed that this process is effective. The more isomorphism between the message sent and the message which is received, the better the communication. Also a better education can be achieved with more complete learning. However, there are seven main limitations of that learning paradigm. Bellow we mention those limitations: [7]

- This framework implies that learning and teaching are integral linked activities.
- The ultimate testing and skill acquisition and knowledge it is not usually for knowing but it is usually in the ability to use the knowledge appropriately.
- The paradigm that used in schools tends to emphasize the transmission of knowledge from an acknowledged special to individuals in isolation.
- Learning only what a person intends is not necessarily the most desirable.
- The physical layout of classrooms the structure of classes, and the traditional model transfers a number of metamessages about knowledge creation, acquisition, and use.
- The classic model of education has small capacity to accommodate issues of emotion and the practical and theoretical link ages between affective, behavioral, and cognitive domains.
- Those traditional learning paradigms are often too predictable, unchallenging, and static. In other words we could say that are "boring".

However, the dissemination of gaming, the need to create more engaging educational practices and the worldwide use of the Internet have led us to think serious games as a new type for training and education. Even though many people begun to see a potential for serious gaming because this could augment and supplement traditional typical education, it is believed that the potential of serious games is to offer a paradigm shift. This paradigm shift shows us how the education and training should be delivered in the 21st century. This paradigm shift will comprise the espousal of: "distributed tutoring models using avatar-driven scaffolded approaches, models of assessment and accreditation towards peer- and personalized modeling of the learner and provide an emphasis upon social interactive learning based upon dialogue and social interactions rather than tutor-based and individual study."(S.Freitas & F. Liarokapis). This would increase the opportunities for feedback loops and generally for synchronous feedback. It is also means the integration of multimodal interfaces (haptics and brain- computer interfaces). [8] As mentioned in previous section, serious gaming is used for educational purposes. The markets that serious games addresses are: educational games,military games,corporate games, government games, ,healthcare games, and art,political, and religious games (Michael & Chen, 2006). Serious gaming could enhance education but the scenario of the game must be targeted according to each case. Games can be used as simulators for some educational purposes. In case that the actual training would be costly or dangerous, simulations in computer and serious gaming would be extremely helpful. But it would not be optimal for every type of education. Why for example to simulate a programming environment when you can practice on the actual computer environment. We should be careful how we use serious games because it can be helpful for all educational purposes but it should be well targeted. Do not create a simulation of a computer environment game but a game with critical questions and fast thinking about programming, as a learning software.

KNOWLEDGE TYPES THAT DEVELOPED BY USING SERIOUS GAMING

Serious games players receive and process information, facts, and descriptions. The gamers also gain more skills through experience and education. Knowledge can be gained from serious gaming. Still there are things that cannot be transferred through serious gaming. Think of smell, sense, weather, injuries, hunger and tiredness. Take for example people that play car and racing games. They do not mind driving fast and reckless. That is because they know they will not get harmed. Also computers are trustworthy and consistent compared to human beings. Thus, there is a poor development in human social aspect. Knowledge types that can be developed using serious gaming:

- Communication knowledge is knowledge being transferred through symbols, imitation and observation, verbal exchange, and video and audio recordings. Throughout the era of videogames or serious games knowledge is being transferred through audio and video.
- Situated knowledge is knowledge specific to a particular situation. In the military they use serious gaming so the trainees and soldiers get familiar with certain situations.
- Tacit knowledge is sometimes referred to as know-how (Brown & Duguid 1998) and refers to intuitive, hard to designate knowledge that is largely experience based; it can be practical skill or expertise.
- Imperative knowledge: the knowledge exercised in the performance of some task. If through repetition one performs the same task, his performance will reach a higher level.
- Explicit knowledge: the knowledge that has been or can be articulated, codified, and stored in certain media.

- Embedded knowledge: the knowledge that is locked in culture, products, processes, routines, structures, or artifacts.
-

KNOWLEDGE PROCESSES THAT ARE SUPPORTED

Serious gaming can support knowledge processes like hand and eye co-ordination, spatial awareness, memory en lateral thinking, cognitive thinking, self-assessment and analysis, spatial abilities, more accurate, the capacity for mental rotation. Serious gaming can also support people, job-specific skills, organization, communication skills, and strategy skills (Michael and Chen, 2006). Eventually the degree of skill and knowledge must be transferred from the virtual to the operational environment.

SOCIAL ASPECTS THAT INVOLVED IN SERIOUS GAMING

Serious games are (digital) games used for purposes other than mere entertainment. Serious games can be applied to a broad spectrum of application areas, e.g. corporate,government,military,educational, and healthcare. It is noticeable that yet the concept is defined in many ways; definitions agree on some matters, but also vary depending on different perspectives and interests. Moreover, serious games encompass the same goals as edutainment, but extend far beyond teaching facts and rote memorization, and instead include all aspects of education – teaching, training, and informing – and at all ages (Michael & Chen, 2006). There are some aspects that involved in serious gaming. First of all, serious gamings allow the learners to experience situations that are not possible to experience in the real world for reasons of safety, cost, time, etc. That can also have positive impacts on the players, development of knowledge and of a number of different skills. Furthermore, serious gaming can support the development of a number of different skills, as discussed by Mitchell and Savill-Smith (2004); analytical and spatial skills, insight and strategic skills, learning and recollection capabilities, psychomotor skills, visual selective attention, etc. It is also used for improvement of problem recognition, self-monitoring, and problem solving, better long-term and short term memory, decision making, and increased social skills such as negotiation, collaboration, and shared decision making. Last but not least, gamers develop their thinking strategies towards more analogical thinking rather than trial-and-error thinking (Hong & Liu, 2003) and that game elements such as competitive scoring, role playing, and increasingly difficult levels have proven useful in corporate training (solution to business problems) Except from that, the categorization of serious games into a number of markets, provided by Michael and Chen (2006), is adopted. Video game playing has a number of advantages for the military, such as improved hand-eye coordination, the ability to work in a team using minimal communication, improved multitasking skills, and the willingness to take aggressive action (Michael & Chen, 2006) Games that developed for the government may concern a number of different kinds of tasks and situations, like different types of crisis management, for instance, dealing with terrorist attacks, disease outbreaks, biohazards, health care policy issues, city planning, traffic control, firefighting, budget balancing, ethics training, and defensive driving (Michael & Chen, 2006; Squire & Jenkins, 2003). Games can help in the development of a number of various skills: planning, strategic thinking, group decision making, communication, and collaboration. Finally from a Healthcare perspective serious gaming has a number of advantages, such as *physical fitness*, *education in health/self-directed car*, *distraction therapy*, *recovery and rehabilitation*, *training and simulation*, *diagnosis and treatment of mental condition or mental illness*, *cognitive functioning*, and *control*.

SIGNIFCANT BENEFITS OF HIGHWAY CODE GAME

The Highway Code serious game is developed in order to teach the children the Highway Code. Through that application children will be able to exercise and learn the signs. Except the basic goal of that application there are some other benefits that need to be referenced. It is noticeable that that these benefits have already discussed in previous sections. Children will have some benefits if they use that application. This application will help them to increase their skills and to obtain some different types of knowledge. Bellow we will mention some of the most significant benefits that obtained from the use of Highway Code:

- Situated knowledge
- Tacit knowledge
- Embedded knowledge that is locked in processes
- Hand and eye co-ordination
- Spatial awareness
- Memory en lateral thinking
- Cognitive thinking
- Self-assessment and analysis and
- Spatial abilities

STEPS THAT FOLLOWED DURING THE DEVELOPMENT PROCESS

This mobile application is built for iPhone devices. It is built only for portrait mode. We used a platform in order to write and execute the code. The platform that we used is called Appcelerator Titanium. It is a platform for developing mobile, tablet and desktop applications using web technologies. In other words it is a phone web based application framework that allows web developers to apply existing skills in order to create native applications for iPhone and

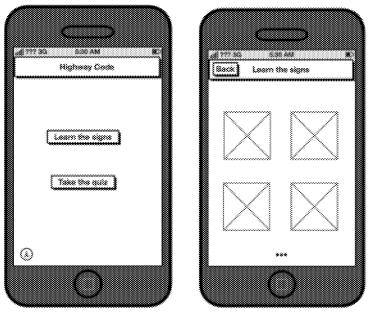
Android. [9] Users need to use JavaScript syntax. Also users need to learn the Titanium API. Titanium API is a framework which contains modules and components that developers can use for creating applications. Moreover, in order to complete our application we followed some steps; Brainstorming, creation of wireframes, writing the code, build the application. Below we will explain each step with more details.

Brainstorming Analysis

The first think that we thought is that we need to create an application which will be helpful and innovative (as innovative as it could be nowadays) at the same time. Children have much less limited attention span than the adults. In order to concentrate on something they really need to have fun with it. Thus we thought about creating an application for children that will give them the opportunity to learn about the Highway Code.

Creation Of Wireframes

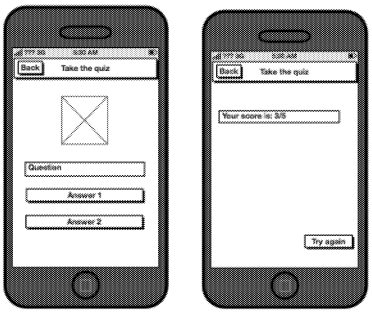
Wireframes help us to analyze the characteristics that we wish to be concluded in an application. The application should be user friendly in order to motivate people to play with it. In screenshots bellow you can see how the application will look like.



1. Main Window (left) and "Learn the signs" window (right)

The screen bellow is the main window of our application (Figure 1). In the top of the screen there is a label which contains the title "Highway Code" of the application. In the middle of the screen there are two buttons; "learn the signs", and "take the quiz". Users can tap on both buttons. The third button is the information button which is placed in the left down corner of the window. From main window, if we tap on the "Learn the signs" button we will be transferred to the window is shown in figure 1. This window does also have a label on the top of the screen with the title of the window. It does have a back button as well. If we tap on back button we will be transferred to main window. The main concept for the "Learn the Signs" window is to contain different signs and teach the users about the signs. Users can scroll through different views and watch different signs. They are able to tap on the signs. When they tap on the signs an alert is shown which contain information about the specific sign. If users click on "OK" button they will see again the whole view which contains the sings. It is noticeable that the users are able to go back to main window from each view. For instance, if we have 4 views with sings and the user is watching the third, they can tap on the "Back" button and move to main window.

The other option from the main window is the "Take the Quiz" button. When you tap on that button you will be transferred to the appropriate window (figure 2). This window contains a label where the title "Take the Quiz" is showed. That window does also have a "Back" button which moves the users to the main window. The main idea of that window is to take a quiz based on the signs that have already showed in the "Learn the Sign" window. In the middle of the screen there is a label which contains the questions. Bellow there are two buttons. In the first button the first answer is showed. In the second button the second answer is shown respectively. When the user taps either on the first button or on the second button they will be transferred to next question. This procedure will continue until the quiz is finished. When the quiz is finished, a view with result will be showed (figure 2).



2. "Take the quiz" window (left) and Quiz Results (right)

Writing The Code

As we mentioned above in order to write and execute the code we used the Titanium Platform. Bellow we will give you a first taste of what should be used in order to create a window, a button, and a label (which are the most common objects that we used).

a. Create a window: Ti.UI.createWindow({})

b. Create a label: Ti.UI.createLabel({})

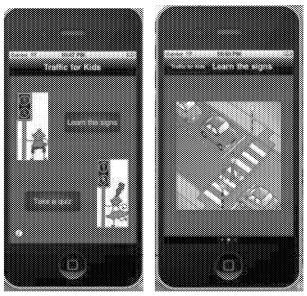
c. Create a view: Ti.UI.createView({})

d. Create a button: Ti.UI.createButton({})

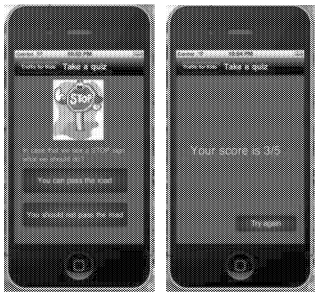
It is noticeable that inside the brackets we add different properties for each object such as position properties (top, left, right).

Build The Application

Once we wrote the code we need to build the application and see the final results. The screenshots below will give you an insight of the application.



3. Main window (left) and "Learn the signs window" (right)



4. "Take a quiz" window (left) and "Quiz results" (right)

DISSEMINATION OF THE GAME

The Highway Code game will be uploaded in the Apple store as it is an application for IPhone devices. This application will be free of charge. That means that this application could attract more people to download it than if it would cost some money. Moreover, in order to achieve a bigger dissemination of the game we should focus on it is wise to focus on schools. If teachers could understand the importance of that application it should be easier to persuade the children to download the application. Highway Code is something important for our everyday life. Thus it would be good if people be taught that from their early ages. Another group of people that we should target is the parents group. We could reach that group via social networks. We could create for instance a group on Facebook or Twitter where we will describe the main aspects of the application. That will be helpful for parents because they are always concern about their children. This application is a tool that could reduce their concern. As we mentioned before children generally have limited attention span so it would be easier to learn the signs through that application than to read a book for example.

FUTURE WORK

For future work we need to think out of the box. We should be focused on how children perceive reality and add some characteristics that improve their social skills. For instance we could design a community that children can talk with other children. The latter seems of high importance, especially considering that nowadays children are used to play at home, more often than not alone, thus not interacting with one another. Another perspective for our game is to design more intelligent and adaptive tools. Those tools will give the opportunity to children with special needs to use that application as well. For instance, children who are blind or have a percentage of blindness will not be able to read the quiz questions. Thus, we should create a system that tackles that problem.

CONCLUSION

In this paper an attempt to decipher how aspects of serious gaming can achieve better results in educating children in a much more efficient way than the traditional learning methods. Children are in a lot of ways pretty different from adults. Therefore, it is only reasonable to make some adaptations in order optimum results to be achieved. This paper has given some insight about serious games and some ideas on how an educational game aiming at children wanting to learn about Highway Code can be implemented. However, a lot more can be done, but that could be the subject of further research.

REFERENCES

[1] Bates A. W. T. 2005. "*Technology, e-learning and distance education*" .

[2] Gredler M. E. 2009. "*Games and Simulations and their relationships to e-learning"*. University of South Carolina,

[3] Derryberry A. 2007. "Serious games: online games for learning".

[4] Kaiser family foundation, available at <http://www.kff.org/entmedia/entmedia012010nr.cf>

[5] Cole et al. "*Color Psychology: Children vs. Adults"*

[6] Definition of Learning, available at <http://en.wikipedia.org/wiki/Learning>

[7] Ruben B.D. 2002. "*Simulations, games, and Experience-Based Learning: The new Quest for a New Paradigm for Teaching and Learning." Rutgers University*

[8] Freitas S. and Liarokapis F. "*Serious Games: A new Paradigm for Education". Coventry University.*

WEB REFERENCES

[9] Appcelerator Titanium, available at < http://en.wikipedia.org/wiki/Appcelerator_Titanium>

APPLYING THE ALFIL CROWD SIMULATOR TO A REAL EVACUATION INCIDENT: THE NEWS DIVINE SCENARIO

César García-García, Victor Larios-Rosillo and Hervé Luga
CUCEA-Postgraduate and Research Department
Universidad de Guadalajara
Periférico Norte 799, Guadalajara
Mexico
cesar.garcia@cucea.udg.mx

KEYWORDS
Multi-agent systems, Crowd simulation, Personality modeling, Emergency response, Serious games

ABSTRACT

This paper presents a reconstruction of the News Divine Discoteque, one of the most well-known and well-documented evacuation catastrophes in Mexico in recent years. A simulated evacuation is compared to the official reports using mean squared error and an error percentage.

INTRODUCTION

The News Divine incident is without a doubt one of the most media-covered and well-documented evacuation tragedies in Mexico (BBC News 2008), (National Commission of Human Rights, México 2008a;b; 2009) and more recently (El Universal Online México 2011).
The fact that several recreations and investigations have taken place regarding this incident makes it a good referral for simulation evacuation. In this particular case, we are not trying to explain the causes of this tragic event, only to reproduce the events using the ALFIL model (García-García et al. 2012a).
The main difference with comercial evacuation software such as PEDSIM (PedSim 2002), Maya Crowd (Maya 2011), Golaem (Golaem 2013) or INCONTROL (INCONTROL 2013) is that ALFIL uses a personality-driven behavior model (García-García et al. 2011; 2012b) to simulate each individual.

PROBLEM DESCRIPTION

According to the official report the evacuation progressed in three phases: in the first three minutes approximately 65 people left the building and were transported in the first bus, with no report of injuries; in the next three minutes approximately 65 more people were transported in a second bus, with no report of injuries; 6 minutes into the raid, and with no more buses available, the police locks the door, effectively containing the remaining people inside; eleven minutes into the raid, the police containment is relaxed and the doors are finally opened, finding 12 dead and 102 injured people; 27 minutes into the raid, a single ambulance will show up. Summarizing:

- Seconds 0 to 180: approx. 65 people evacuated, assuming 0 injured
- Seconds 181 to 360: approx. 65 more people evacuated, assuming 0 injured
- Seconds 361 to 660: assuming 0 people evacuated, 114 found injured
- Seconds 661 and onward: The police containment is relaxed and the doors are finally opened

APPROACH

A virtual environment similar in size and distribution to the real venue of the incident was filled with 350 civilian agents, the approximate number of occupants in official reports (National Commission of Human Rights, México 2008a;b) and news sources (El Universal Online México 2011), with initial reports suggesting up to 500 occupants (BBC News 2008).
The artificial crowd was then evacuated under different scenarios to measure the evacuation time, speed of evacuation, number of injured, and number of dead; as well as the average presence of fear, and the number of people in panic.
Official regulatory sources (Pérez-Guerrero 2006) refer an evacuation time of 2.5 minutes (150 seconds) for 100 people using a 1 m door, assuming a walking speed of 1 m/s, or 0.67 people per second, or *pps*. Cross-cultural studies (Levine and Norenzayan 1999) place the average walking speed of a healthy adult at 1.4 m/s. Training materials refer an acceptable evacuation time of up to 3 minutes (UEPCYBJ 2012).
Using this model we estimate the required evacuation time for the venue described in the official report to be of 525 seconds or 8 minutes 45 seconds, due to the fact that it only has one door, and not the regulation-

mandated 1 door per 100 people for the target time of 2 minutes 30 seconds with 0.67 pps.

Evacuation scenario as per the official report
From the official report data, a very simple linear regression model was used to determine evacuation and injury rates. Figure 1 illustrates the amounts of evacuated and injured people over time and Figure 2 illustrates the evacuation and injury rates.

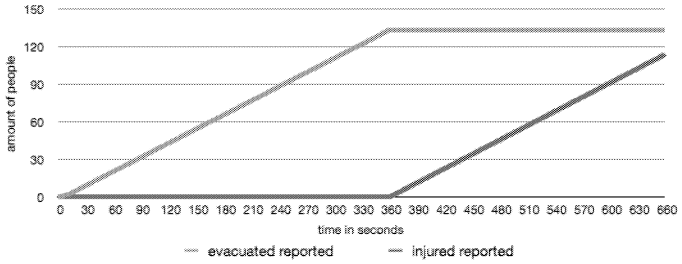


Figure 1: Evacuated and injured people according to the official report.

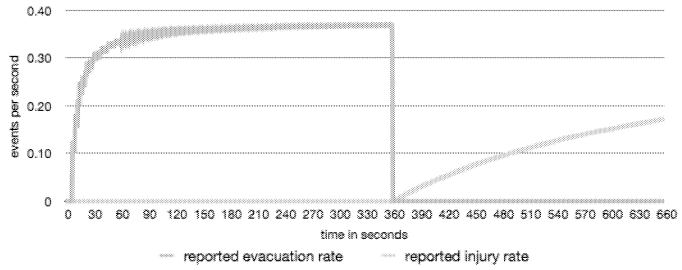


Figure 2: Evacuation and injury rate from a linear regression based in the official report.

ALFIL simulation of the official scenario
This scenario attempts to recreate the events in the report and is divided in two phases:

- Seconds 0 to 360: The doors are fully open and, while the police is directing the evacuation, there is no physical pressure to do so. An agent is considered evacuated when he crossed the door.
- Seconds 361 to 660: The doors are closed and the police officers are unaware of this fact, thus begin pushing the detainees. An agent who falls prone or who sustains too many collisions is considered injured.

We can observe from the simulated evacuation a total of 94 evacuations and 69 injuries. Figure 3 illustrates the amounts of evacuated and injured people over time. Figure 4 illustrates the evacuation and injury rates, with evacuation average of 0.55 and injury average of 0.09.

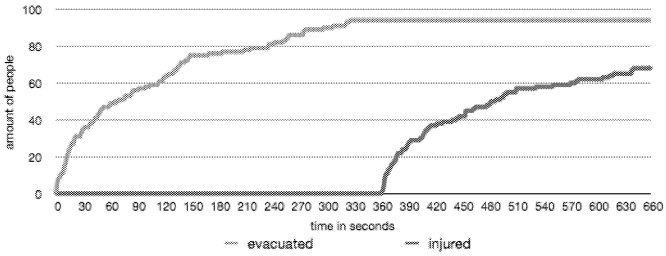


Figure 3: Evacuated and injured people from the ALFIL simulation.

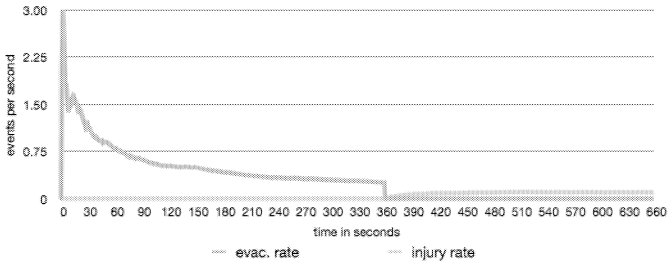


Figure 4: Evacuation and injury rate from the ALFIL simulation.

RESULTS

We establish a comparison point between the linear regression from the report and the simulated amount of evacuated and injured people. A shown in Figure 5, both the evacuation and injury rate behave in a similar way, even if there is a clear difference between linear regression and simulation.

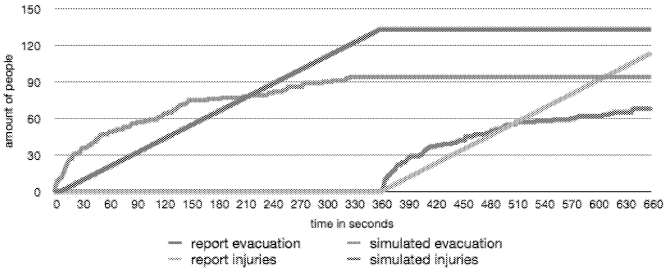


Figure 5: Evacuation and injury comparison from the report and simulation.

From this comparison we establish an accumulated mean squared error or *MSE*, which can serve as a comparison with other simulations of similar events. The accumulated MSE is shown in Figure 6 and clearly shows that the simulation rapidly loses accuracy and is unable to compensate for the accumulated errors.

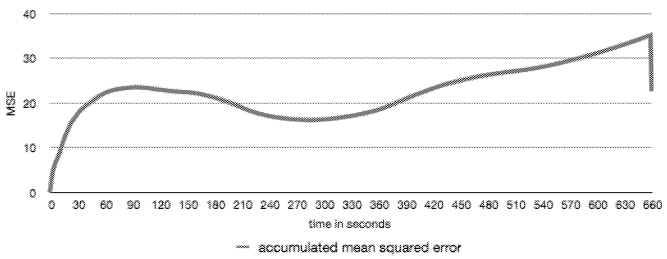


Figure 6: Mean squared error for the ALFIL simulation.

Finally, an error percentage and error percentage average is calculated for both evacuations and injuries. Figure 7 shows this result, with an average evacuation error of 32.74% and an average injury error of 34.50%.

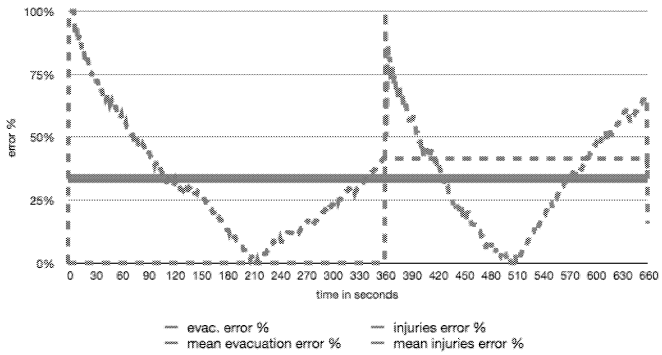


Figure 7: Error average percentage for the ALFIL simulation.

CONCLUSIONS

First, we want to reproduce textually the conclusions reached by the official incident investigation by National Commission of Human Rights, México (2008a;b):

- The New's Divine venue operated illegally
- The police raid was badly planned
- The police raid was badly executed
- The consequences of the police raid where badly managed

We observe that the average evacuation speed of 0.55 *pps* in the simulated scenario suggests that the venue was probably over full capacity, closing in the estimated 8 minutes and 45 seconds and going beyond the regulatory 2 minutes and 30 seconds.
The evacuation speed of 0.37 pps, obtained from the official report's linear regression suggests that the evacuation was proceeding slowly, accounting to restricted movement by the police watching the doors.

While we are in no position to support or challenge the official findings, given that the physical venue no longer exists and that we had no access to official expert forensic reports; we can conclude that the behavior model used to simulate the evacuation produces results that are close to a 30% margin of error, which is the greatest error tolerable for simulation of socio-organizational phenomena.

Further work

Data from this simulation will be used to further refine the ALFIL behavior model to improve its accuracy and reduce both the mean squared error and error percentage.

ACKNOWLEDGEMENTS

This project is funded by CONACYT and COECYT-JAL grants. The DVRMedia2 framework is a cooperative 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

BBC News, 2008. *Deadly stampede at Mexico disco.* URL `http://news.bbc.co.uk/2/hi/americas/7466950.stm`.

El Universal Online México, 2011. *News Divine: Cuentas Pendientes.* URL `http://www.eluniversal.com.mx/graficos/especial/EU_divine_gnd/`.

García-García C.; Fernández-Robles J.; Larios-Rosillo V.; and Luga H., 2012a. *ALFIL: A crowd simulation serious game for massive evacuation training and awareness. International Journal of Game-Based Learning*, 2, no. 3, 71–86.

García-García C.; Larios-Rosillo V.; and Luga H., 2012b. *Intelligent Computer Graphics 2012*, Springer, *Studies in Computational Intelligence*, vol. 441, chap. Agent behaviour modeling using personality profile characterization for emergency evacuation serious games.

García-García C.; Torres-Nabel C.; Larios-Rosillo V.; and Luga H., 2011. *Towards a Behavioral Model for Personality-driven Artificial Humanoids in Serious Games.* In D. Plemenos (Ed.), *14th International Conference 3IA'2011, the International Conference in Computer Graphics and Artificial Intelligence.* Athens, Greece, 131–136.

Golaem, 2013. *Golaem Crowd: Artist driven crowd simulation*. URL `http://www.golaem.com/content/products/golaem-crowd/overview`.

INCONTROL, 2013. *Enterprise Dynamics*. URL `http://www.incontrolsim.com/`.

Levine R.V. and Norenzayan A., 1999. *The pace of life in 31 countries. Journal of Cross-Cultural Psychology*, 30, no. 2, 178–205.

Maya, 2011. *Maya Crowd Simulation*. URL `http://www.mayacrowd.com/`.

National Commission of Human Rights, México, 2008a. *Informe Especial sobre violaciones a los derechos humanos en el caso New's Divine*. Tech. rep., National Commission of Human Rights, México.

National Commission of Human Rights, México, 2008b. *Informe Especial sobre violaciones a los derechos humanos en la procuración de justicia en el caso New's Divine*. Tech. rep., National Commission of Human Rights, México.

National Commission of Human Rights, México, 2009. *CNDH Newsletter 193*.

PedSim, 2002. *PEDSIM: A Microscopic Pedestrian Crowd Simulation System*. URL `http://pedsim.silmaril.org/`.

Pérez-Guerrero A., 2006. *NTP 436: Cálculo estimativo de vías y tiempos de evacuación*. Insituto Nacional de Seguridad e Higiene en el Trabajo, España.

UEPCYBJ, 2012. *Diagnóstico de riesgos, conformación de brigadas, señalética y simulacros (DIACOSESIM)*. Unidad Estatal de Protección Civil Jalisco, Jalisco, Mexico.

AUTHOR LISTING

AUTHOR LISTING

Alipasali A. **94**
Alvarenga E.R.Q. **80**
Andrade F.L.C.D. **69**

Barbas H. **36**
Bhikharie S.V. **8**

Černy M. **52/59**
Clua E.W.G. **20**
Correia R.F. **20**

Dias L.L. **69**

Eliëns A. **8/94**

Firouzian A. **89**

Garcia-Garcia C. **99**
Gemrot J. **52/59**

Holaň D. **52**

Johansson M. **15**

Kehoe J. **5**
Khan U.A. **31**

Lappi E. **76**
Lappi M. **76**
Larios-Rosillo V. **99**
Livingstone D. **45**
Luga H. **99**
Lunardi A.D.C. **20**

Machado A.F.V. **20/69/80**

Nissinen H. **89**

Okada Y. **31**

Padovani R.R. **69**
Pinto T.P.C. **69**

Stiegler A. **45**
Strååt B. **15**

Warpefelt H. **15**