DESIGN AND ENGINEERING OF GAME-LIKE VIRTUAL AND MULTIMODAL ENVIRONMENTS

A workshop of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems

Berlin, Germany
June 20, 2010

Chris Raymaekers, Hasselt University, Belgium
Karin Coninx, Hasselt University, Belgium
Juan Manuel González-Calleros, Université Catholique de Louvain, Belgium

Contents

Session: design

Session: development

Session: evaluation

Session: serious games

Designing and Evaluating Engaging Social and Emotional Experiences in Games and Virtual Worlds

Katherine Isbister
New York University's
Polytechnic Institute
Six Metrotech Center
Brooklyn NY 11201
isbister@poly.edu

ABSTRACT

In this position paper, I outline the research methods and approach our group at the Social Game Lab takes to understanding and innovating successful social and emotional game and virtual world experiences.

Author Keywords

Games, user experience, social, emotional, design, player experience.

INTRODUCTION

The construction of compelling and believable virtual environments for games and simulation experiences presents many technical challenges, and the design of such environments is further complicated by the rapidly changing nature of the technology.

What changes less dramatically, over time, is the psychological qualities of the users/players who will engage with the technology. Our lab takes an approach to the design of games and virtual worlds grounded in player experience, with emphasis on psychological research and findings about how people respond socially and emotionally. We use (and innovate) methods based in the Human Computer Interaction (HCI) tradition, to create games and game-like experiences that provide compelling social and emotional experiences to players/users. It is our hope that this position paper will spark discussion and perhaps also collaboration with other research teams, who might benefit from application of these techniques to their own game/virtual world projects.

RESEARCH APPROACH

Unlike productivity software such as spreadsheets and word processors, game and virtual world experiences derive a great deal of their value from being engaging and enjoyable

during the use of the software (rather than primarily from success in outcomes). Even 'serious' games and simulations must provide enjoyment to users in order to succeed [7].

Putting the moment-to-moment experience foremost presents a new set of challenges to software engineers who may be more familiar and comfortable with an approach to user experience that focuses on task efficiency and effectiveness. While a game or simulation must also provide classic usability and goal-driven satisfaction, it will not be considered a success unless the journey is also pleasurable (or at least engaging).

In the Social Game Lab, we believe this shift in emphasis requires modification to both the design/engineering and evaluation processes.

It is important to gain an understanding of what qualities of a system lead to fun and engagement, what sorts of design/engineering process produce these qualities, and how best to measure whether these qualities have been achieved for players/users.

System Qualities that Enhance Fun and Engagement

There are of course many facets to producing a game or simulation environment that results in a fun and engaging experience (see [8] for a small literature review on this topic). In our work at the Social Game Lab, we have focused on two primary system qualities that heighten fun and engagement—the presence of social and pseudo-social contact as part of the game [3, 4, 6], and the presence of game mechanics and other cues that heighten emotional engagement with the experience [8, 9].

More than half of players surveyed in the United States say they are playing games with others more of the time than alone [1], and NPCs (non-player characters) have been a part of digital games since the earliest days of game design. So it is safe to say that most gaming experiences include some form of social interaction.

Typically, players are acting in a game or virtual world through an avatar (player-character). Even in the case of first-person-view worlds, there is often an implied player-character that provides psychological framing for the player

of who she is in the game world and her capabilities, social connections, and motivations.

Games that allow for multi-player interaction typically allow communication among players through their avatars (sometimes also in parallel through channels like voice or text chat). Co-located game play also provides a backdrop for social interaction among players who are in the same room, as well as spectators. A significant portion of the fun of co-located games may have to do with in-room engagement that is sparked by what is on-screen (see [3, 4]).

Games are also known for their capacity to provoke emotion [8]. Like film, television, and other traditional media, games use narrative elements to draw a player into caring about characters and situations at hand. Games are also able to elicit emotions that have to do with personal agency, such as guilt or pride, because a game can provide the player with opportunities to make decisions and act on her own initiative, and see the consequences [6]. Games that make use of real physical movement have an additional means to provoke emotion, as it has been demonstrated that moving as if you feel a certain way leads you to label your own emotional state differently (e.g. if I move in a happy manner I begin to feel happy and to label myself as feeling happy) [12]. In addition, observed cues of emotions are readily internalized in observers, leading to 'emotional contagion' effects that can be exploited by game and simulation designers in crafting both co-located play and avatar-based play social play situations [2].

Design/Engineering Processes that Produce Fun
Based both on interviews with practicing game designers [7], review of available literature [8], and our own design process in our lab, we have come to believe game and virtual world design for social and emotional engagement requires a design and engineering process that takes a very prototype-driven approach [8]. Natural-seeming social and pseudo-social interaction cannot really be specified ahead-of-time, because it is co-reliant on pushing the limits of technology (believable animation, engaging dialog, reactive movement, intuitive interface for puppeteering an avatar, and the like) and on the emergent qualities of an interaction among players and the system.

We use small design sketches built in a game engine (see Figure 1.) to prototype concepts in our lab for engaging social and emotional game play experiences. We find these prototypes essential both to crafting a final refined game, and also, to informing our research about which aspects of game play are engaging and why, toward generating design theories and recommendations. Prototypes must be of relatively high resolution in terms of reactivity to input, in order to be useful for design iteration and for theorizing (see [8] for an extended description of our recommendations for doing this kind of 'supple' design).

As an example, we have been creating a series of game prototypes around the general area of enhancing social connection through a combination of game mechanics and physical play. We have created three small game experiences that explore this terrain—games called 'pass the torch', 'the wave', and 'bonk'. Pass the torch (see Figure 1.) is a game in which two players try to move a flame from one podium to another, using body movement. The player operating the green avatar leans toward the podium on his right, picks up the flame on his torch, and then the two players must physically lean toward one another so the flame can be passed to the purple player, who must then lean toward the other podium to settle the flame there.

Figure 1. A prototype of a social game mechanic that we built in our lab to try out some concepts about forging social connection through collaborative movement.

The game requires close physical cooperation and collaboration, which in turn can engender a greater feeling of social connectedness. We have kept the concept and game parameters quite simple, but the game is well-realized enough to be playable and engaging. This state (simple but playable) was reached through several phases of iteration within the lab group on a rapid time cycle. We spent perhaps 6 weeks total in developing the game, and were able to leverage and extend core code within the game engine that we have re-used and extended to create the other games in this cycle as well. We will use pass the torch (and the other two social connectedness prototypes) in an experiment in the fall, in which we compare play using the Wiimotes (physical play) to play of the same games using keyboard controls instead. This will allow us to isolate the influence of the physical movement mechanics on player feelings of social connectedness, to test out whether our hypothesis that cooperative physical play increases social connectedness is in fact true.

Innovating Evaluation
Measures in the HCI community for usability, including satisfaction measures, have in the past been focused on very goal-driven types of software, which lead users to have

particular kinds of interest and attention. As mentioned before, games and virtual worlds must be engaging in the moment, and in fact it may be okay if the experience of using them is sometimes confusing or frustrating, if this heightens the experience itself. This means that designers/engineers must rethink what they are trying to ascertain from users/players when evaluating the system both during and at the end of the design cycle.

Our lab combines rich qualitative observation with quantitative measures and approaches to the study of game play. We frequently begin with video observations of play that make use of multiple cameras and a screen feed from the game to really capture what is happening during play (see Figure 2).

In this example, we were interested in whether amount of player movement contributes to engagement. We had groups of players play movement-based games with low, medium, and high amounts of physical movement required by the game mechanics, to see if we found significant differences in player ratings of their experiences with the game.

Figure 2. Multi-camera set-up for recording social game play.

We have had to get creative with evaluation methods to tackle novel contexts of use—for example, in this study, we made use of classroom 'clickers'—handheld devices that can be tucked into the pocket when not in use—to collect player ratings of satisfaction and frustration during movement-based game play (see Figure 3).

We have also done some radical evaluation method innovation in the aid of better eliciting player emotions, including a nonverbal self-report method that involved the use of handheld sculpted objects (see [9, 11] for an overview of this work, which received a best paper nomination at the CHI conference).

It is clear from the state-of-the-art of user experience evaluation even among the game developer community's foremost practitioners [10], that there is much work yet to be done in evolving robust and well-tuned evaluation methods for determining social and emotional engagement with games and virtual worlds.
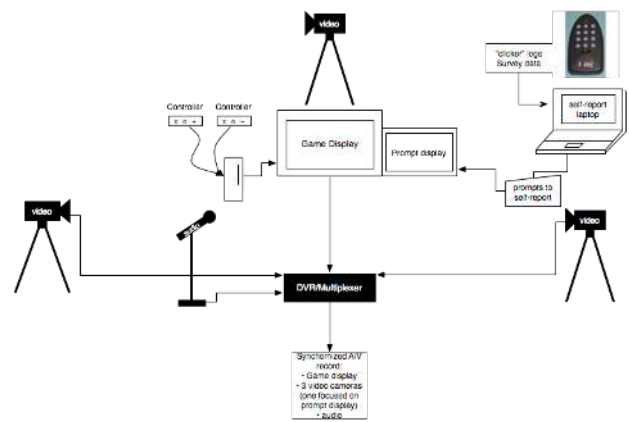

Figure 3. Diagram of recording and data-collection set-up for recording social game play.

CONCLUSIONS

In this paper, I've presented a brief overview of some key challenges the Social Game Lab has taken up, in advancing the state of the art in game and virtual world development practices. Our research places emphasis on social and emotional qualities of these experiences—how to design them, and how to evaluate them—toward generating best practices that can be applied both within the game development community and also in 'serious' application contexts such as games for learning [7].

I look forward to connecting with other practitioners at the workshop, to trade insights and methods, toward the continued evolution of our field.

ACKNOWLEDGMENTS

REFERENCES

1. Entertainment Software Association (2008) Essential facts about the computer and video game industry. Downloaded from http://www.theesa.com/facts/pdfs/ESA_EF_2008.pdf on July 17.

2. Hatfield, E., Cacioppo, J.T., and Rapson, R.L. (1994). Emotional contagion. Cambridge University Press.

3. Isbister, K. Enabling Social Play: A Framework for Evaluation. In Bernhaupt, R. (Ed). Evaluating User Experience in Games, Springer Verlag, pp. 11-22.

4. Isbister, K. (2009). Step two: Understand social play. In Bateman, C. (Ed.) Beyond Game Design: Nine Steps Toward Creating Better Videogames, Charles River Media.

5. Isbister, K. 2008. Social Psychology and user research. In Isbister, K. and Schaffer, N. (Eds.) Game Usability: Advice from the Experts for Advancing the Player Experience, Morgan Kaufmann.

6. Isbister, K. Better Game Characters by Design: A Psychological Approach. Morgan Kaufmann, 2006.

7. Isbister, K., Flanagan, M. and Hash, C. Designing Games for Learning: Insights from Conversations with Designers. Proce. of CHI 2010, ACM (2010).

8. Isbister, K. and Höök, K. On Being Supple: In search of rigor without rigidity in meeting new design and evaluation challenges for HCI practitioners, Proc. of CHI 2009, ACM (2009), 2233-2242.

9. Isbister, K., Höök, K., Laaksolahti, J., & Sharp, M. (2007). Sensual evaluation instrument: Developing a trans-cultural self-report measure of affect. International journal of human-computer studies 65(4): 315-328.

10. Isbister, K. and Schaffer, N. (Eds.) Game Usability: Advice from the Experts for Advancing the Player Experience. Morgan Kaufmann, 2008.

11. Laaksolahti, J., Isbister, K. and Höök, K. (2009). Using the sensual evaluation instrument. Digital Creativity 20(3): 165-175.

12. Strack, F., Martin, L.L., and Stepper, S. (1988). Inhibiting and facilitating conditions of the human smile: A nonobtrusive test of the facial feedback hypothesis. Journal of Personality and Social Psychology 54: 768-777.

Five Grand Challenges in the Engineering of Networked Digital Games

T.C. Nicholas Graham
School of Computing
Queen's University
graham@cs.queensu.ca

ABSTRACT

In this position paper, I introduce five grand challenges in the development of networked digital games. While this list is not intended to be exhaustive, I argue that these challenges are significant in that they are holding back the development of the next generation of innovative games.

INTRODUCTION

Digital games have become an important entertainment medium, with the Entertainment Software Association reporting that 68% of American households play computer or video games [3]. Within these, multiplayer games, those that allow groups of people to play together, have become particularly popular.

Recent years have seen stagnation in the design of digital games. While innovative input technologies such as the Wii Remote have led to new forms of interaction in games, many successful games are derivative remakes of existing games [5]. The year 2010 has already seen the release of Bioshock 2, Mass Effect 2, God of War 3, Command and Conquer 4, among others.

In this position paper, I identify five challenges that are holding back innovation in the development of multiplayer digital games. While these challenges are by no means exhaustive, they are all important, and the gaming industry would benefit from their solution. The challenges are development cost, real-time consistency maintenance, supporting truly massive multiplayer play, allowing player-generated content, and security. None of these have obvious solutions, and therefore can serve as grand challenges for researchers in the engineering of networked digital games.

CHALLENGE 1: DEVELOPMENT COST

The cost of developing games has increased greatly, with modern "AAA" games costing upwards of $20 million to produce. But the game industry is hit-driven, where only a minority of titles recoup their development costs: accord-

Figure 1. Game sketching with Raptor can help evaluate fun before the game is even prototyped, liberating designers to try new ideas.

ing to the New York Times, as of March 2009, only 16 of 486 released titles for the Nintendo Wii console had been profitable [10].

Very large studios can allow a few hits to pay for many titles that fail. Smaller studios do not have this luxury, as a single failure can lead to bankruptcy. Many studios therefore act conservatively, releasing only revisions of games that have already proven successful.

Our first challenge problem is therefore to find ways of evaluating whether a game idea will be fun before the game has been implemented, allowing game studios to innovate with lower risk. One emerging solution is game sketching [1, 13], which employs *Wizard of Oz* techniques to allow games to be played before even prototypes are available (figure 1).

CHALLENGE 2: CONSISTENCY MAINTENANCE

Entertaining gameplay often depends on players' having a consistent view of the game world [12, 4]. For example, in a first-person shooter game, players need accurate representations of their opponents' positions in order to be able to aim at them [8]. In practice, because of latency in the networks connecting the players' computers, inconsistency in players' views is inevitable (figure 2).

One current solution to this problem is to design the game to be less real-time. For example, in World of Warcraft, players perform commands that trigger an action at some point in the
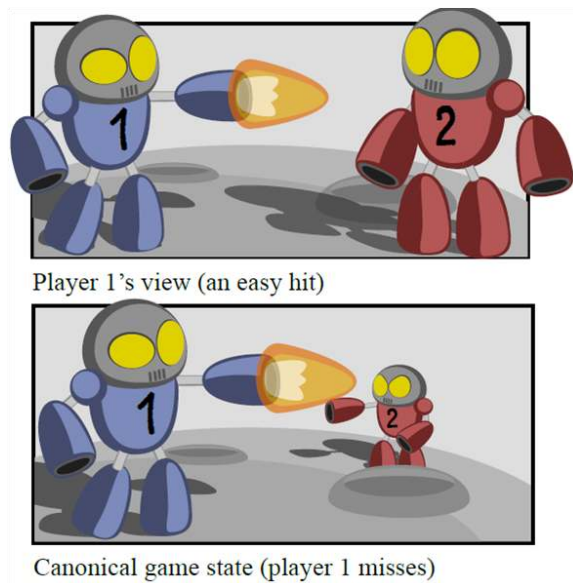
Figure 2. Consistency problem in first-person shooter. Maintaining consistency of players' views in real-time is becoming increasingly important with the introduction of motion-tracking input devices.

Figure 3. The City of Heroes' *Mission Architect* **is a rare example of a tool allowing players to create content within an online game.**

near future (e.g., casting a spell or using a special sword attack.) Since the actions are not instantaneous, latency can be masked. Another increasingly popular solution, *local lag*, makes latency predictable by inserting delays in local processing of input [11].

Modern game controllers, however, such as the Wii Motion-Plus, Microsoft Natal and PlayStation Move encourage increasingly real-time styles of play, where players' actions are reflected in the game world as they occur.

Possible avenues for addressing this problem include finding accurate approaches for client-side prediction, better approaches for determining the consistency requirements of different circumstances, and imaginative consistency maintenance algorithms tuned for specific game situations.

In summary, our second grand challenge is how to accommodate truly real-time play in the presence of network latency.

CHALLENGE 3: MASSIVELY MULTIPLAYER
Games increasingly allow large groups of people to interact in real time. Massively multiplayer online games (MMOGs), for example, allow thousands of people to concurrently log in to the same world. Current games have, however, suprisingly small limits on how many people can interact in the same virtual space. This is due to bandwidth limitations (conveying the positions and actions of large numbers of players from server to client in real time), CPU limits on the server, and limitations of the graphics processing unit of the client in rendering large numbers of detailed models.

To address this problem, MMOGs carefully funnel players so that only a few dozen are involved in the same combat

(or other activity.) When numbers of players become large, games frequently "instance" playing areas, creating multiple copies of an area, each capped in population. Other current solutions include level of detail updates [2] and predictive client-side updates [9].

Our third grand challenge is to allow truly large-scale concurrent play, as necessary to support battles with hundreds of units per side.

CHALLENGE 4: PLAYER-GENERATED CONTENT
Players of online games consume content dramatically faster than developers are capable of creating it. One solution to this problem is to permit players to create content and add it to the game for themselves and others to enjoy. Being able to participate in meta-gaming (the game of creating games) has the potential to extend players' interest in the game world. Examples of this approach already exist. NCsoft's City of Heroes MMOG includes a "mission architect" feature (figure 3) that allows players to create quests for other players' to carry out. The feature includes a voting function, allowing players to rate quests and therefore allow the best player-created content to be easily found. EA-MAXIS' Spore game includes a sophisticated model editor that allows players to create creatures, buildings and vehicles for inclusion in the game (figure 4). While Spore is a single-player game, player-created models are automatically transmitted to a server, and subsequently may appear in other players' game sessions.

There are many barriers to the broad adoption of player-generated content in a networked context. Since player-created models cannot be dynamically loaded onto the client,

2

Figure 4. EA-MAXIS' *Spore* **provides a sophisticated editor allowing players to create 3D models for inclusion in the game. Unfortunately, the editor can be used to create content that many players may find objectionable.**

Figure 5. The challenge of allowing player-created content is illustrated by one of dozens of "penis monsters" created by players of Spore shortly after its release.

Figure 6. The security challenge: virtual gold sales through third-party web sites can imbalance game economies and can provide incentive for theft of in-game items that now have real monetary value.
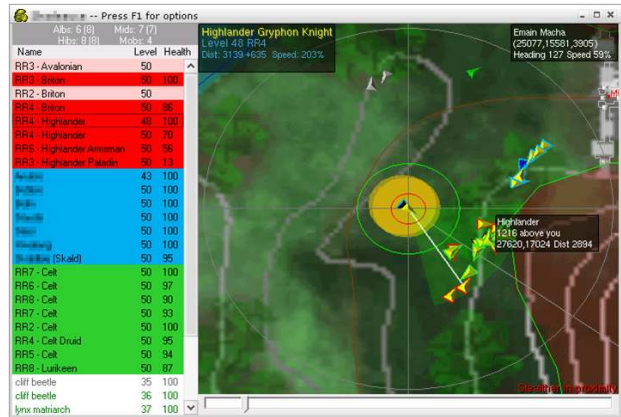

Figure 7. DaocSkilla tool used to cheat in EA-Mythic's *Dark Age of Camelot* **Massively Multiplayer Online Game.**

prohibitive levels of bandwidth may be required from the server to the client. The scalability implications of player-generated content can be seen in Linden Labs' Second Life, which according to Kumar et al. can host only 40 players per server as opposed to thousands in traditional MMOGs [7].

Game companies may inadvertently find themselves redistributing player recreations of copyright IP, leading to legal problems. Players may use the games' mechanisms to engage in illegal activity, e.g., setting up a casino. Finally, players may use these tools to create offensive or lewd materials inconsistent with the game's intended market (figure 5).

Game companies do not typically have the resources necessary to evaluate all player content, and therefore must either rely on mechanisms allowing players to report undesirable content, or provide no official channel for distribution of player-created content. Because of the inherent limitations

to these approaches, most virtual world games do not allow players to create and upload their own content.

Solutions to this problem must lie in the technical sphere of finding ways of predicting what content a player is likely to need in the near future and distributing it to the player's client asynchronously, through to attempting to find technical means of identifying objectionable or illegal content quickly.

Our fourth grand challenge is therefore to find mechanisms enabling players to unleash their creativity by creating content for use in networked games.

CHALLENGE 5: SECURITY
Cheating in online games can lead to reduced enjoyment for legitimate players, and even to financial loss from the theft of online goods [6]. Examples of cheating include the pur-

3

chase of virtual currency for real money (figure 6) and the use of cheating programs to gain advantage over other players (figure 7).

One set of current techniques for combatting cheating is architectural. These include ensuring that no game-critical information is resident in the game client (in case it is hacked), that all game actions are audited on the server (to detect suspicious behaviour), that no unnecessary information is in the client-server protocol (in case it is snooped), and that the protocol includes encryption and challenge-response features (in case of man-in-the-middle attacks.) These features constrain possible implementation architectures, notably ruling out peer-to-peer architectures, in turn constraining the solution space for our other four grand challenges.

Another approach is to design the game so that cheating is less desirable. For example, the *DaocSkilla* program (figure 7) provides a radar view, easing player-versus-player combat in EA-Mythic's Dark Age of Camelot game. Later MMOGs such as World of Warcraft have included radar views in the game client, removing the advantage conferred by this form of cheating.

Our fifth grand challenge is therefore to find techniques for reducing the opportunities of cheating in games, helping to free developers to pursue development of gameplay instead of anti-cheating methods.

CONCLUSION

In this position paper, I have identified five grand challenges in the design of networked digital games. While this list is not intended to be exhaustive, each of the listed problems is holding back innovation in multiplayer games. Each problem is complex and has no obvious solution, and therefore is appropriate for investigation within the academic community.

REFERENCES

1. M. Agustin, G. Chuang, A. Delgado, A. Ortega, J. Seaver, and J. Buchanan. Game sketching. In *Proceedings of the 2nd international conference on Digital interactive media in entertainment and arts*, pages 36–43. ACM, 2007.

2. M. Brockington. Level-of-detail AI for a large role-playing game. In *AI Game Programming Wisdom*. Charles River Media, 2002.

3. Entertainment Software Association. Industry Facts. http://www.theesa.com/facts/index.asp.

4. R. Fletcher, T. C. N. Graham, and C. Wolfe. Plug-replaceable consistency maintenance for multiplayer games. In *Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games*, pages 34–37. ACM, 2006.

5. U. Hagen. Where Do Game Design Ideas Come From? Invention and Recycling in Games Developed in Sweden. In *DiGRA*, pages 1–11, 2009.

6. G. Hoglund and G. McGraw. *Exploiting online games: cheating massively distributed systems*. Addison-Wesley, 2007.

7. S. Kumar, J. Chhugani, C. Kim, D. Kim, A. Nguyen, P. Dubey, C. Bienia, and Y. Kim. Second Life and the new generation of virtual worlds. *Computer*, 41(9):46–53, 2008.

8. M. Mauve. How to keep a dead man from shooting. In *Interactive Distributed Multimedia Systems and Telecommunication Services*, pages 741–779. Springer, 2000.

9. L. Pantel and L. Wolf. On the suitability of dead reckoning schemes for games. In *Proceedings of the 1st workshop on Network and system support for games*, pages 79–84. ACM, 2002.

10. M. Richtel. Video game makers challenged by the next wave of media. *New York Times*, March 30, 2009.

11. D. Stuckel and C. Gutwin. The effects of local lag on tightly-coupled interaction in distributed groupware. In *Proceedings of the ACM 2008 conference on Computer supported cooperative work*, pages 447–456. ACM, 2008.

12. J. Vogel and M. Mauve. Consistency control for distributed interactive media. In *Proceedings of the ninth ACM international conference on Multimedia*, pages 221–230. ACM, 2001.

13. C. Wolfe, J. Smith, W. G. Phillips, and T. C. N. Graham. A model-based approach to engineering collaborative augmented reality. *Engineering of Mixed Reality*, pages 293–312, 2010.

Towards Model-Based development of 3D User Interfaces for Games

Juan Manuel González-Calleros, Jean Vanderdonckt
Université catholique de Louvain
Place des Doyens 1, 1348
Louvain-la-Neuve, BELGIUM
+32 (0)1 047 8349
{juan.m.gonzalez, jean.vanderdonckt}@uclouvain.be

ABSTRACT

Video games industry is the leader of the virtual reality (VR) market. Along as any VR reality application they used 3D User Interfaces (3DUIs) for the interaction with the user. Even that several games nowadays have very impressive realism and innovative representations, this is not the case of 3DUIs. In this position paper we state that 3DUIs for any 3D application, including video games, should be revised in order to reduce the gap between the innovative representation of VR application and their 3DUIs.

Categories and Subject Descriptors

D2.2 [**Software Engineering**]: Design Tools and Techniques – *Modules and interfaces; user interfaces*. D2.m [**Software Engineering**]: Miscellaneous – *Rapid Prototyping; reusable software*. H.1.2 [**Information Systems**]: Models and Principles – *User/Machine Systems*. H5.2 [**Information interfaces and presentation**]: User Interfaces – *Prototyping; user-centered design; user interface management systems (UIMS).*

General Terms

Design, Experimentation, Human Factors, Verification.

Author Keywords

User Interface, 3D User Interface, 3DUI, Model-Based User Interface Development, Information Systems design.

INTRODUCTION

3DUIs are becoming the primary subject of interest of a growing community of researchers and developers adopting different approaches for specifying and creating 3DUIs. Providing development methods and software support for 3DUIs is a complex problem. Most research and development is focusing on technological issues, as reported in a survey of major publications on 3DUIs [11].

overcome hardware and software issues [2]. Little or no attention is devoted to the design knowledge that should drive the development life cycle of 3DUIs.

This is also the case of VR for games or game-based VR applications. It is very frequent that 3DUIs are reduced to just imitate a 2DUI. For instance Figure 1, illustrates the representation of a 3DUI using a toolkit CEGUI (www.cegui.org.uk) for the game engine OGRE (www.ogre3d. org). Notice that the components regarding the User Interface remains in 2D. Consistency is one of the most important aspects to be considered for the usability of a system. It looks like this has been forgotten in the design of 3DUIs where there is always a break in the 3D metaphor.
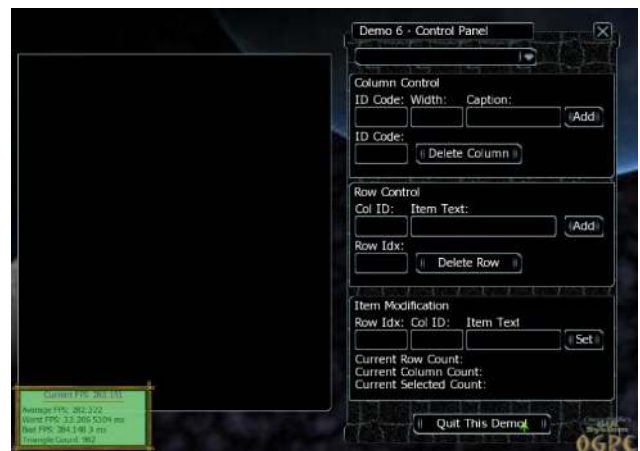
Figure 1 Game user interface created with Crazy Eddies' toolkit.

In this paper we argue that developing 3DUIs for VR applications is an activity that would benefit from the application of a development method composed of: (1) a set of models defined according to an ontology, (2) a language that expresses these models, and (3) a principle-based approach manipulating these models based on principles. Principles including guidelines and design knowledge helping in the task of creating 3DUIs more close to 3D than 2D.

STATE OF THE ART

Game engines, such as: Crystal Space 3D (www.crystalspace3d.org), Open Scene Graph (www.openscenegraph.com), Panda3D (panda3d.org), Microsoft XNA, are mostly promoting the development of 3DUIs by hand. Although there are some toolkits that help designers in this task, for instance, Crazy Eddie's GUI System assists the development of 3D GUI for the Object-Oriented Graphics Rendering Engine (OGRE). The set of elements that are supported are just limited to those existing in a GUI. There is no particular innovation in the representation as it just imitates a 2D GUI (Figure 1). There is a plethora of methods to develop 3DUIs; in [4] we explored some of them. Although, methods share some similarities on their steps, several conceptual dissimilarities differentiate them. Consequently, it is challenging to transfer one abstraction from one method to another in order to have one consistent framework supporting the development lifecycle of 3DUIs. Moreover, these methods rarely provide the design knowledge that should be typically used for achieving each development step. Most approaches cover aspects such as task modelling, dialog modelling and implementation aspects. While some knowledge in explicit of existing method most of the time it is hard to find the abstracted models, the transformational approach along with its rules.

3D USER INTERFACES GUIDELINES FOR SELECTING 3DUIS

In this section we define concepts related to 3DUIs, i.e., the representation of 3D widgets, their characteristics, and how we can derive the 3DUI. For doing that we need: models of the 3DUI, a method and a language. Our models were incorporated to UsiXML [8] to support:

- 3D Rendering of 2D User Interfaces [10]. The UsiXML model that considers 2D GUI was specialized to consider the attributes specific for their 3D rendering. Then, this is compliant to current approaches.

- 3DUIs involving a true 3D presentation of the 3DUI [5]. By true representation is meant going beyond an imitation of their 2D GUI counterpart. In that scenario it was not enough to rely on the existing model as concepts such as appearance, texture, shape, and behaviour were needed.

- Hapgets Haptically enhanced 3D widgets [7]. This to support another modality of interaction.

Regarding the method, the goal is not to come up with yet another Software Development Method but to reuse existing work and structure it accordingly. The result is a method that structures the development life cycle of a 3DUI of an IS in a principle-based way. The method follows an exploratory approach as its goal is to show a variety of possibilities to encourage design. It is said that is structured as it is based on a structured Framework, the Cameleon Framework [1]. Second, a set of instruments (evaluation guidelines [5], task patterns [5], a list of

canonical task types [5], and rules for model to model transformations [4]) guide de use of the method then making it principle-based. The method (Figure 2) includes an evaluation of each development step, which can be performed manually or automatically. We rely on principles expressed as guidelines for modelling the task and abstract user interface (AUI) model. They are applied manually and refine the models. At the concrete user interface (CUI) model direct evaluation over the code is performed by evaluating rules and conventions or recommendations. The refined model is then used for the code generation. The next subsections details the development steps.
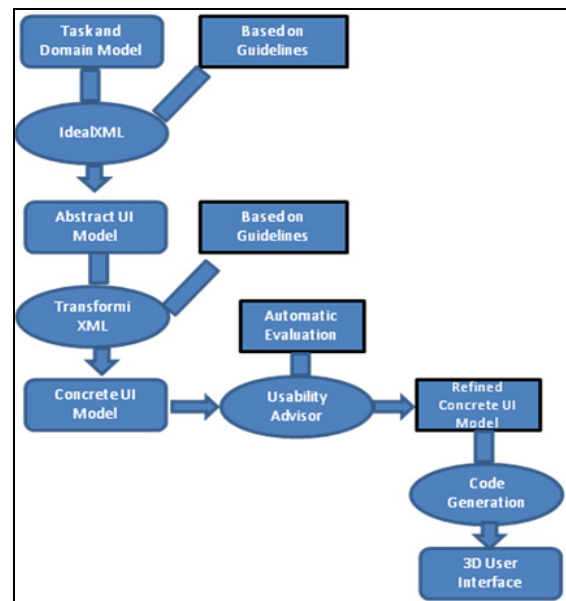

Figure 2 Outline of the method for developing 3DUIs

The method has been already reported in the literature but we would like to stress the use of the mechanism for selecting widgets based on a taxonomy of 3DUIs, this is done in the step from abstract to concrete user interface model.

After identifying the appropriate 3DUI interactive object that is needed for the VR application, for instance, a button. At this point the designer could ask himself what is the appropriate representation of a 3D button? Should the 2D desktop metaphor still be used or are there alternative visualizations or metaphors? Several attempts go towards defining a new toolkit of 3D objects which are natively appropriate to 3D applications. Again, this represents an advantage to have a predefined collection of such 3D-widgets, but then the interaction is reduced by what they offer natively.

To help designers in selecting more systematically the 3DUI elements we propose the following steps. We build a taxonomy of 3DUIs (Figure 3 illustrates the taxonomy of a radio player).
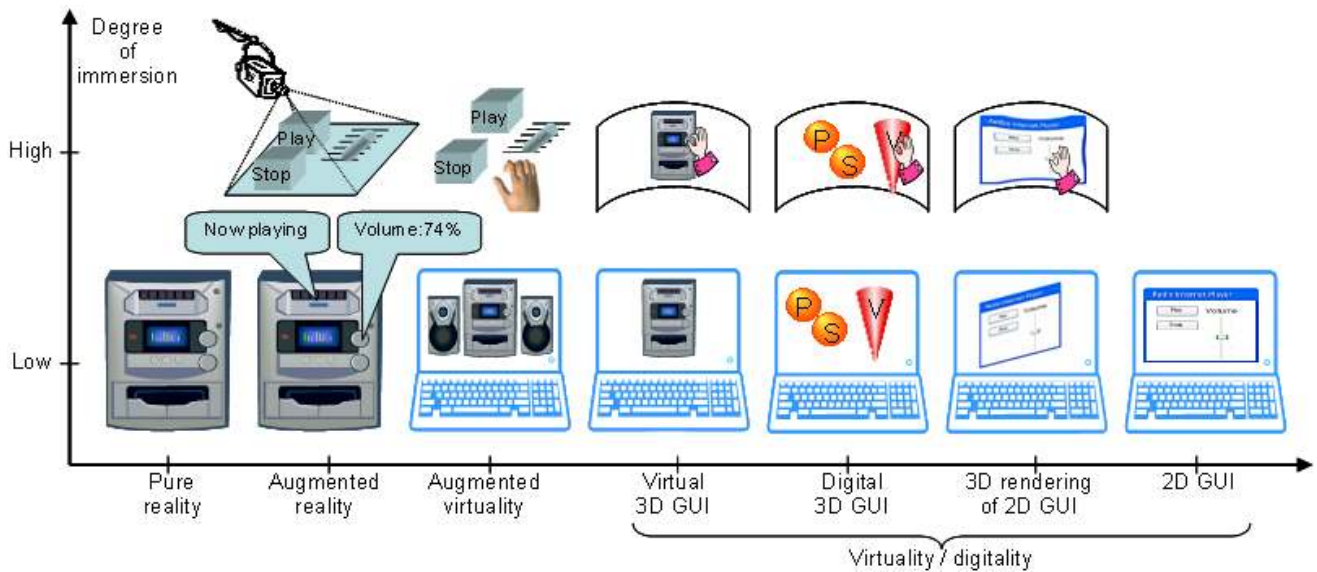
Figure 3 Three-Dimensional Sliders Taxonomy, source [4].

The taxonomy can be of some help in making that decision as a design space of potential representations of the 3DUI elements. The use of this taxonomy could changed by any other form that fits authors needs, for instance, Figure 4 shows a list of metaphors used to represent a toggle button.

Figure 4 Graphical representation for a toggle button

The use of the taxonomy provides hints to actual solutions for designers. More important is the fact that there can be more than the examples shown in each level, further investigation can fill the gaps or add more examples to the existing levels.

In a second step, the possible renderings are examined. The comparison between the different representations of the same component that are presented must be examined using criteria like the 2D and 3D consistency, the development complexity, the usability and how much intuitive the representation was. This evaluation of the representation could be as formal as the project demands. The result of evaluating the different renderings ends with a selection of the most appropriate representation for a 3DUI element, depending on the special needs of the users. For instance, when the target group is the blind users, there is no need to keep a 2D-3D consistency, as users cannot see the 2D components to make the comparison and more easily identify its 3D counterpart. In this case, the major criterion is to represent a 3D component in a way that is easy to comprehend and be used by the blind users, even if this representation has nothing to do with the 2D representation of the corresponding 2D components. The method for selecting the representation of the 3DUI element that we propose is the Question, option, criteria [9]. Accordingly to this method, a set of criteria can be evaluated when a question is raised and a set of possible answers exists (3D renderings in this case). In this case four criteria were identified to consider: 2D to 3D consistency, easy to develop, intuitional and the usability. The weight attributed to each criterion is based on the users who evaluate those representations.

In Figure 5 we illustrate this process for adding weight to the toggle button representation. The meaning of the links is: (++) means strongly supported, (+) means supported, (~) means neutral, (-) means denied and (..) means strongly denied. If we consider that the haptic channel is desirable and that the graphical representation is not of interest due to fact that our end user in visible impair. Then it is clear that the selection must be towards the haptic representation.

Notice that this analysis only provides a general view of the different properties related to the development of the 3DUI

elements and does not arrives at a conclusion about which representation is generally the best.

Question	Answer	Score	Options
What will be the representation of the 3D Toggle Button?	Switch	-	(1) 2D-3D Consistency
		--	(3) Easy to develop
		+	(4) Intuitional
		~	(5) Usability
	Sphere	~	(1) 2D-3D Consistency
		~	(3) Easy to develop
		~	(4) Intuitional
		~	(5) Usability
	2D representation	++	(1) 2D-3D Consistency
		+	(3) Easy to develop
		+	(4) Intuitional
		+	(5) Usability
	Haptic	+	(1) 2D-3D Consistency
		+	(3) Easy to develop
		+	(4) Intuitional
		+	(5) Usability

Figure 5 Questions and answer criteria to select a toggle button.

This step is part of a complete methodology supporting the development life cycle of 3DUIs [4,5,6,7] which is compliant with the Model Driven Architecture paradigm promoted by the OMG (www.omg.org). In order to be fully-MDA compliant, this works need a User Interface Description Language (UIDL). A review of the literature of existing UIDLs was conducted in [3] to justify the selection of UsiXML [8] as UIDL for our work.

CONCLUSIONS

The method presented is aimed at promoting a more systemic approach for the development of 3DUI for VR applications. This element we feel that has been considered of less importance due to the complexity of VR applications. The method shows the different steps to select 3DUI elements in a structured way. This development step reinforces existing knowledge on 3DUI development methods as we provide some means to identify the diversity of concretizations of the 3DUI (i.e. its representation).

ACKNOWLEDGMENTS

REFERENCES

1. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J.: A Unifying Reference Framework for Multi-Target User Interfaces. Interacting with Computers, Vol. 15, No. 3 (June 2003) 289–308.

2. Dünser, A., Grasset, R., Seichter, H., Billinghurst, M., Applying HCI Principles in AR Systems Design, *Proc. of the 2nd Int. Workshop on Mixed Reality User Interfaces: Specification, Authoring, Adaptation* (*MRUI'07*), Charlotte (USA), 11 March 2007.

3. Guerrero García, J., González Calleros, J.M., Vanderdonckt, J., Muñoz Arteaga, J. A Theoretical Survey of User Interface Description Languages: Preliminary Results, Proc. of Joint 4th Latin American Conference on Human-Computer Interaction-7th Latin American Web Congress LA-Web/CLIHC'2009 (Merida, November 9-11, 2009), E. Chavez, E. Furtado, A. Moran (Eds.), IEEE Computer Society Press, Los Alamitos, 2009, pp. 36-43.

4. Gonzalez-Calleros, J.M., Vanderdonckt, J., Muñoz-Arteaga, J.M., A Method for Developing 3D User Interfaces of Information Systems*, Proc. of 6th Int. Conf. on Computer-Aided Design of User Interfaces* (*CADUI'2006*), Bucharest (Romania), 6-8 June 2006, Chapter 7, Springer-Verlag, Berlin, 2006, pp. 85-100.

5. Gonzalez-Calleros, J.M., Vanderdonckt, J., Muñoz-Arteaga, J., A Structured Methodology for Developing 3D Web Applications, in T. Spiliotopoulos, P. Papadopoulou, D. Martakos, and G. Kouroupetroglou (eds.), Integrating Usability Engineering for Designing the Web Experience: Methodologies and Principles, Chapter 2, *IGI Global Inc.*, Hershey, 2010, pp. 15-43.

6. Gonzalez-Calleros, J.M., A Model-Driven Approach for Developing Three-Dimensional User Interfaces of Information Systems in a Principle-based Way, PhD Thesis, Université catholique de Louvain Press, February 2010.

7. Kaklanis, N., González-Calleros, J.M., Vanderdonckt, J., Tzovaras, D., Hapgets, Towards Haptically-enhanced widgets Based on a User Interface Description Language, *Proc. of Workshop on Multimodal Interaction Through Haptic Feedback* (*MITH'2008*), Naples (Italy), 31 May 2008.

8. Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., Lopez, V., UsiXML: a Language Supporting Multi-Path Development of User Interfaces, *Proc. of 9th IFIP Working Conference on Engineering for Human-Computer Interaction jointly with 11th Int. Workshop on Design, Specification, and Verification of Interactive Systems EHCI-DSVIS'2004,* Hamburg (Germany), July 11-13, 2004. Lecture Notes in Computer Science, Vol. 3425, Springer-Verlag, Berlin, 2005, pp. 200-220.

9. MacLean, A., Young, R.M., Bellotti, V., and Moran, T.P. Questions, Options, and Criteria: Elements of Design Space Analysis. Human-Computer Int. 6, 3-4 (1991) 201–250.

10. Molina, J.P., Vanderdonckt, J., Montero, F., Gonzalez, P., *Towards Virtualization of User Interfaces based on UsiXML*, Proc. of 10th ACM Int. Conf. on 3D Web Technology Web3D'2005 (Bangor, 29 March-1 April 2005), ACM Press, New York, 2005, pp. 169-178.

11. Swan, J.E., Gabbard, J.L. Survey of User-Based Experimentation in Augmented Reality? *Proc. of 1st Int. Conf. on Virtual Reality,* Las Vegas (USA), 2005.

Developing video games and virtual environments with the Crystal Space engine

Jorrit Tyberghein, Eric Sunshine, Frank Richter, Mike Gist
Crystal Space team
http://www.crystalspace3d.org

Christian Van Brussel
Communications and Remote Sensing Laboratory
Université catholique de Louvain
Louvain-la-Neuve, Belgium
christian.vanbrussel@uclouvain.be

ABSTRACT

In this paper, we present the Crystal Space engine (CS – [1]), an open-source software development kit for the development of modern video games and virtual environments. We describe the various components of the framework, their interaction with 3D design tools, and how it is possible to use them to create applications with complex virtual worlds. We present more closely the available features and the level of abstraction introduced in the CS SDK and the Crystal Entity Layer (CEL). We also present CELStart, a runtime package environment on top of CEL. The presentation is accompanied by an overview of the game engine middleware and compares CS to the main other related projects.

Keywords

Game engine, software development kit, virtual reality, video games, 3D rendering

INTRODUCTION

Nowadays, in accordance with the increasing presence of video games and virtual reality-based software in everyday computer applications and entertainment, there has been a growing need for a greater number of realistic virtual worlds with ever increasing complexity. The size, the level of detail, the realism of the scenes, as well as the need for immersive interaction have all been increased, in turn forcing developers to invent ever more complex solutions to address the issues that are created by such requirements.

In the domain of video games for instance, the size of the team needed to develop a game has become bigger and bigger in order to be able to address the increasing complexity and quantity of the technical functionalities and artistic creations needed. Because of this, the budgets involved in the production of modern commercial games have rapidly caught up with those of Hollywood's big productions.

It has been difficult for independent and low-/mid-budget games to compete with such a level of complexity, and as a result their productions have been ghettoized in to simpler games, e.g. in 2D instead of 3D. For example, the quantity, quality and diversity of games made with the Flash technology attests to the willingness of the community to create such applications from the moment the available technology made it possible to concentrate mainly on the content and logic of the game, instead of on technical aspects of the simulation.

In these last few years there has been a flourishing range of middleware and complete frameworks attempting to address this problem, solving part of (or as much as possible of) the technical needs of modern games and virtual reality applications. In the rest of this paper, we will present an overview of these software development kits, and then we will present more closely one of them, the Crystal Space engine, and compare it to the main other related projects.

CONTEXT AND RELATED PROJECTS

One of the first types of middleware for virtual reality appeared with video games such as id Software's Doom and Quake in the early and mid 1990's. Part of the huge success of these games came from the fact that they were shipped with tools and mechanisms allowing anyone to modify and extend the content of the game. Soon, many teams of fans formed across the Web and started the creation of new variants of the initial games. The concept of 'mods' (for 'modifications') was born, a phenomenon which had great influence on the future evolution of the game industry. These 'modifiable' games played the role of sandboxes for the imagination of the community, allowing many new concepts to emerge from all these contributions. The noticeable success of mods such as 'Counter Strike' and 'Team Fortress', both of which were later adopted by Valve Software and converted into high-quality commercial products, testify the importance of the phenomenon.

Even before that, companies such as id Software (and later on Epic Games with the Unreal license) had started to license the software technology of their games (their 'game engines') so that other commercial companies could develop new games on top of the technology already provided in the game engines. 'Half-Life' for instance, the first game produced by Valve Software, was using a modified version of the Quake engine, whereas this same

company now licenses its home built engine 'Source', one of the main solutions for game engines today. Besides this new commercial market and along with the emergence of the open-source movement, new open-source projects addressing that same task were created, examples of which are Crystal Space, Ogre 3D ([2]) and the Irrlicht Engine.

More complex solutions appeared afterwards, such as the CryEngine of Crytek, featuring what they called a "What You See Is What You Play" editor (WYSIWYP interfaces). Such editors enable the artist to directly visualize the virtual environment concurrently to its design, providing a new interactive way to build 3D worlds. The editor also contains many advanced features allowing manipulation of the environment from a high semantic level, such as the road builder which can create new roads in a few mouse clicks and automatically adapt the landscape, the flora and any objects along the road. Microsoft XNA is another example of a complete framework for the development of video games.

Finally, other companies and open-source projects focused on more specific problems, such as the skeletal animation (Euphoria), the Artificial Intelligence (SpirOps and PathEngine), network management (RakNet and GNE), and physical simulation (NVIDIA PhysX, Havok Physics, Bullet).

All these software solutions cover a wide range of the technical needs for the developing of modern virtual environments. However, it has to be noted that a huge amount of work will most probably still be needed for the creation of the assets of the environment, a process which is currently taking a great part of the man-work needed for a project. In light of this observation, we can probably predict that the developing of new advanced tools assisting the artists in the creation of the assets will be the next challenge to be addressed.

THE CRYSTAL SPACE ENGINE
Overview
Contrary to almost all other open source 3D engines such as Ogre 3D or the Irrlicht Engine, CS is not only a rendering engine, i.e. a system concentrating on the display of 3D scenes, but a complete game engine encompassing as much as possible of the functionalities needed for modern video games and applications using virtual reality.

Crystal Space was founded in 1999 as a personal project hosted on the SourceForge open source software development web site. The project rapidly received keen interest from the open-source community as it was one of the first open projects on 3D real-time rendering. The statistics attest of the success of the project: on July 2000 the SDK was downloaded 15.000 times, and the web site received a record of 1.300.000 visits on March 2005. More than 200 contributors have so far participated to the development of CS and the framework has been used in many independent games, as well as in some academic research projects and some commercial games, and it has been a testbed for many experimental features that were later used in other projects, such as the Verse network protocol funded by the EU Sixth Framework Programme

([3]). CS is also more or less closely related to many other open-source communities, such as the teams of Blender and PlaneShift.

The CS project is a typical mid-scale open-source project, based on spontaneous and voluntary contributions instead of on a planned design with financial funding. The work is coordinated through open discussion, under the monitoring of the administrators of the project. Due to the nature of the participations to the development process, there has been no real long-term or global planning of development other than guidance from the main administrators and contributors. Special care has therefore been needed in order to keep a clean and coherent global architecture of the system.

General architecture and workflow
CS is licensed under the GNU Lesser General Public License. It is written mainly in C++ using very few non-standard extensions, allowing it to run on a wide range of POSIX platforms such as GNU/Linux, most BSD's, Mac OS/X, as well as on all 32- and 64-bit MS Windows. It has also been recently used successfully on an ARM architecture. Finally, CS supports a wide range of compilers and IDE such as GCC, Visual C++ 8 and 9, MinGW/MSYS and Cygwin.

CS encompasses two main components, the CS SDK, providing the low-level functionalities such as rendering, sounds and physics, and the Crystal Entity Layer, providing more advanced features and a higher level of abstraction for manipulating the objects of the virtual environment.

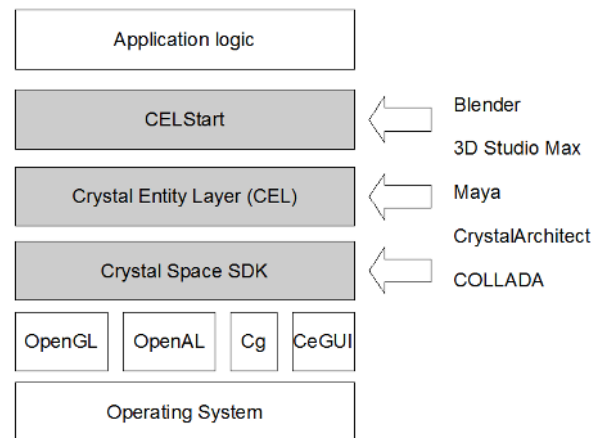*Figure 1 General architecture of the Crystal Space framework.*

The general architecture of the whole framework is presented in figure 1. The CS SDK is above the operating system, and uses some few external libraries such as OpenGL, Cg, OpenAL and CeGUI. CEL is on top of the CS SDK, while the logic of the user applications is defined on top of CEL (or directly above CS according to their design choices).

Above CEL sits CELStart; a runtime package environment that allows the developers to distribute 3D applications in a rather innovative and elegant way. CELStart consists of a generic launcher application for 3D software comprised of a single self-contained package, holding the entire assets and logic of the 3D application. The logic of the application is defined through portable scripts written in Python or in a dedicated XML compliant description language. As the data files and scripts are self-contained and portable, the CELStart launcher has only to be installed once on a system. A new application then needs to only distribute a single data package, allowing for an easy distribution, installation and management of application packages.

The assets, as well as the descriptions of the virtual scenes, and the events and behaviors of the objects in the scenes, can be described in CS's XML compliant description language. They are imported into CS and CEL from 3D design tools such as Blender (thanks to the related open-source project blender2crystal), 3D Studio Max, and Maya. CrystalArchitect, another related open-source project, is a design environment with a particular emphasis on CS which allows the direct visualization of the 3D environment concurrently to its design (WYSIWYP interface). There are also some importers for 3D scenes file formats such as COLLADA.

To develop an application using CS, one has to choose at first if he will use or not CEL, because a later integration of it will be difficult. As we will see, CEL is versatile and provides a solution for the developing of applications needing almost no programming. With CEL, it is possible to design all or a great part of the application from graphical editors such as Blender and CrystalArchitect, with no or a minimum of coding. However, the level of abstraction and the functionalities provided may not be judged interesting regarding the architectural restrictions imposed. In this case the developing can be made atop of the CS SDK, but it will need the coding of more functionalities (but with more freedom).

As CS covers a wide range of the technical needs of the application, most of the coding can still be concentrated on the logic of the application, i.e. the various concepts and objects of the environments and how they interact. A good start for a new project is to look at other demo and test applications such as CrystalCore or YoFrankie!

The Crystal Space SDK
An important feature of the CS SDK is its modularity, achieved through the Shared Class Facility (SCF), a component-based system based on the COM model. It separates the API of the components from their implementations, which are accessible in the form of shared libraries. The system allows the dynamic loading of the plugins, providing the ability to switch or select plugins at runtime or during development. One main benefit is the structuring of the code that is imposed by this system, forcing the decomposition of the system in independent and interchangeable components and allowing an easy integration of new features. This also helps to address

problems such as incompatibilities on different platforms, and makes it possible to extend it with other communication mechanisms for SCF, for example inter-process communication through pipes or CORBA.

Another important characteristic of the CS SDK is the wide range of functionalities which are already integrated inside the SDK, providing the base tools to develop any kind of virtual environments, and not being restricted to a single type of application or game. This is reinforced by the fact that these functionalities are organized through SCF and can be selected according to the needs.

Notable features of the CS SDK are a very descent modern 3D rendering engine with shaders, dynamic lighting and post-processing effects, many types of different meshes, and many mechanisms for the management of Level Of Details (LOD) and big and/or external environments. It has good support for physical simulation thanks to the Bullet and ODE libraries, and a strong skeletal animation system using animation blending trees, with support for ragdoll and inverse kinematics. It has also many other features such as abstraction of the operating system and of the hardware peripherals, text localization, font server, integrated graphical user interfaces with CeGUI and wxWidgets, and 3D sounds with OpenAL.

Finally, CS has also bindings for other programming languages through SWIG. It has support for Python, Java, and Perl, and some experiments have been made for Lua and C#.

The Crystal Entity Layer
The main addition of CEL is an entity system which abstracts all technical aspects of the management of the objects, letting the designer concentrate on the content of the virtual environment. Objects are defined as entities, on which property classes defining a facet or aspect of what the objects are and how they behave are attached. An event system enables generic communication between property classes. There is a wide range of property classes already available, covering many of the needs of modern 3D applications. These can also be combined to create complex entities and behaviors. Main property classes are for avatars and Non-Player Character management, camera management, scene and world management, binding of peripheral events, selection and grabbing of objects, advanced physics (e.g. fluids, explosions, wheeled vehicles, gradual destruction of objects), artificial intelligence (behavior trees, path finding, steering behaviors, neural networks and genetic algorithms).

As an example of the combination of property classes, the entity of a user avatar would typically have an 'input' property class to redirect the peripheral events to the entity, a 'mesh' property class for the 3D model of the avatar and its animations, a 'movement' property class for the synchronization of the user actions with the mesh, its position, its animations, and the collisions with the environment, a 'select' property class to enable the avatar to select and grab objects near him, an 'inventory' property class to manage the objects that are grabbed, and finally a

'camera' property class to have the camera automatically follow the avatar.

The level of abstraction introduced by CEL provides an elegant way to manipulate all the objects of a virtual environment. Some tools benefit from this, such as the persistence layer used for saving and loading world states, and the experimental networking layer achieving the synchronization of data among the nodes of a multi-player environment.

The logic of the application is defined through the list of entities and their property class. However the user still has to define how these entities are updated, and how they behave precisely. This can be achieved through C++, Python, or a dedicated XML compliant description language. The quest manager tool also helps by providing the infrastructure for user interaction and for the triggering of events.

CELStart is actually just a small piece of software on top of CEL, providing the functionalities for managing the application package, accessing the XML description of the application and its entities, loading and setting up all entities and property classes, and updating them thanks to the XML and Python scripts incorporated in the package.

The biggest restriction introduced when using CEL is the fact that most of the concepts and objects of the virtual environment will have to be expressed in terms of entities and property classes in order to be able to interact with CEL. This would allow benefiting of all the generalization features of CEL, but may not be appreciated. That is the reason why one has to choose early in the development process whether he will use CEL or only the CS SDK.

FUTURE WORK

The first improvements which are needed in CS concern its accessibility to the public, with for instance a more sustained release cycle, as well as the release of binary packages and the development of more attractive and explicative demonstration applications.

New features are also planned, in order to match the evolution of the new technical needs. The CS SDK has recently received the addition of a new system for advanced skeletal animation, many features for which have already been added and many others are planned (e.g. facial animation, parametric motion graphs and avatar personalization). Other domains that are worked on in the CS SDK are advanced lighting techniques such as radiosity and deferred shading, more featured post-processing and environmental effects, hardware occlusion culling, advanced LOD management, and better support for import tools for 3D Studio Max and Maya. For CEL, future improvements are needed in artificial intelligence with e.g. more advanced features for the behavior trees and the path finding system, as well as in networking support.

CONCLUSION AND USER PERSPECTIVE

In this paper, we have presented the Crystal Space framework and its main parts, the CS SDK, CEL, CELStart, and their related software tools and projects. This "Crystal Space umbrella" forms a complete open-source middleware solution covering most of the needs for the development of modern video games and virtual reality environments.

When having to choose for a game or a rendering engine, one has to consider its real needs. For a big-budget production, the best choices are commercial solutions such as Unity, the Source engine, the Unreal engine, or the CryEngine, providing both a high-level of quality and guaranteed user assistance. For academic, independent and low-/mid-budget productions, the open-source counterparts are nowadays more and more interesting. If one simply needs a 3D rendering engine, then CS, Ogre 3D and Panda 3D are good choices. If the user wants to work specifically on the rendering techniques, then he may orient himself to other projects such as GTP ([4]) or G3D ([5]). For some specific video game type, there are some engines well suited, such as ORTS ([6]) and the Spring engine for real-time strategy games. For other less specific or more substantial projects, then a general game engine such as CS is a really good choice, other solution being the Blender Game Engine and the GameKit ([7]).

REFERENCES

1. http://www.crystalspace3d.org

2. Gregory Junker, Pro OGRE 3D Programming, Apress, 2006.

3. Uni-Verse project, Deliverable D 1.2, Sixth Framework Programme, 2005.

4. Mateu Sbert, Jordi Palau, GameTools: Advanced Tools for Developing Highly Realistic Computer Games IVth ITRA World Conference, Alicante, July 2005.

5. Morgan McGuire, The G3D Graphics Engine: Advanced language features for simplicity and safety in a graphics API, C/C++ Users Journal, 2004.

6. M. Buro and T. Furtak, On the Development of a Free RTS Game Engine, GameOn'NA Conference, Montreal 2005.

7. http://code.google.com/p/gamekit/

User Experience Evaluation for Multimodal Interaction in Games

Regina Bernhaupt and Nicolas Linard
IRIT
118 Route de Narbonne
31062 Toulouse, Cedex 9, France
+33 561 55 7446
Firstname.Secondname @irit.fr

ABSTRACT

User experience (UX) evaluation in games has become an active research area in human-computer interaction and more general in computer science. Based on this research it became clear that UX evaluation for games has to be adapted to the various game genres and interaction approaches. Still a general understanding of user experience in games that use multi-modal interaction, like for example table top games, virtual reality approaches or simply new forms of interaction in standard environments (e.g. exertion games) is an open research question.

Based on a classification of user experience evaluation methods for games, this contribution shows how to adopt user experience evaluation for the area of multi-modal interaction in games. Based on a general outline of available methods, we show how methods shall be combined to evaluate games including multi-modal interaction.

Keywords

user experience, evaluation, games, multi-modal interfaces

1. INTRODUCTION

User experience evaluations in games and more general in interactive entertainment systems have been performed from early on in games development. Programmers of the first computing systems started to develop a first versions of digital games, and already established a very basic form of user experience evaluation by simply trying to play the game – and trying to understand why it was not fun in the end. The introduction of video games like Tetris showed that small changes in game play or story heavily influence the overall user experience of the game (Novak, 2008).

In industry today a variety of methods is deployed to understand the factors that contribute to the overall gaming experience. The term user experience was only rarely used in the games industry (Federoff, 2002), but became extremely prominent in the field of human-computer interaction (HCI) during the last ten years. Since then the scientific communities of human-computer interaction and game research are starting to learn from each other. On the one hand, user experience evaluation methods from HCI are used during the game development to improve user experience, on the other side HCI was borrowing and investigating aspects of the gaming experience like immersion, fun or flow to better understand the concept of user experience.

To understand how to evaluate user experience of games that are based on multi-modal interaction (independent of the extent of the virtual world) it is necessary to agree on the term user experience (section 2: defining user experience) and what factors are related to user experience (section 3: user experience decomposed). Next we present an overview on existing methods (section 4: UX Evaluation Methods) and briefly describe multi-modal interaction in games (section 5). Finally we propose a method-mix that allows evaluation of user experience for games, and show benefits and limitations.

2. DEFINING USER EXPERIENCE

How user experience should be defined is currently discussed in the HCI community. User Experience (UX) still misses a clear definition. As of today the term user experience can be seen as an umbrella term used to stimulate research in HCI to focus on aspects which are beyond usability and its task-oriented instrumental values (Hassenzahl, 2003). User experience does include a look on all the (qualitative) experience a user is making while interacting with a product (McCarthy and Wright, 2004), or on experiences made during interacting with a special type of product e.g. a mobile phone (Roto, 2006). The current ISO definition on user experience focuses on a person's perception and the responses resulting from the use or anticipated use of a product, system, or service. From a

psychological perspective these responses are actively generated in a psychological evaluation process, and it has to be decided which concepts can best represent the psychological compartments in games to allow to measure the characteristics of user experience (Takatalo et al., in Bernhaupt, 2010).

3. USER EXPERIENCE DECOMPOSED

User experience in games has been evaluated using a variety of concepts including immersion, fun, presence, involvement, engagement, and flow (for an overview see Takatalo et al., in Bernhaupt, 2010), play and playability – and what makes play fun, including social play (Isbister, in Bernhaupt, 2010). Given that user experience is understood as the subjective relationship between user and application (McCarthy and Wright, 2004), the CEGE approach (Calvillo-Gamez et al., in Bernhaupt, 2010) allows to differentiate between different subjective experiences. Other evaluation approaches focusing on experiences and attitudes towards video games are the use of semantic differentials (Lemay et al., in Bernhaupt, 2010). From an industry perspective the tools currently in use to enable the evaluation of user experience are closely connected to the game development phases. During the concept phase, classical user-centered development methods like (paper) prototyping are used to allow to understand if the game is fun to play, or the tech demo allowing to understand not only the robustness of the technology but enabling a first evaluation of the fun potential. In the pre-production phase evaluation methods like heuristics are used, during production user testing is one of the most successful methodologies to evaluate user experiences. Post launch user experience evaluations are based on reviews and online forums (Brown, in Bernhaupt, 2010).

Decomposing various aspects of user experience in games, we have to understand that the user experience depends on *the individual player experiences*. Evaluation of user experience thus has to take into account the abilities of the player and its goals. Casual gamers will have different experiences in games compared to experience gamers playing for example World of Warcraft for years. Some researchers started to focus on these different gaming experiences: Desurvire et al. (in Bernhaupt, 2010) propose the game approachable principles that help evaluate and shape UX with respect to inexperienced gamers.

Game phases, especially the post game experience allows to understand why games provoke different forms of experiences. A first questionnaire on Post-Game Experiences was developed by Poels et al. (in Bernhaupt, 2010).

Evaluating aspects of user experience and especially understanding what kind of *game play* will evoke what type of user experience is a task that is difficult to accomplish. There is a broad variety of work on what game play aspects might evoke what kind of experience including investigations that used experiments to understand for example the influence on non-player characters on the overall user experience (e.g. Lankes et al. in Bernhaupt, 2010)

Evaluation additionally becomes difficult when the goal of the game is not only entertainment but *reaches beyond* and wants to convey a message, to *educate* or to change people's habits.

Finally evaluation of user experience is not only depending on the various constructs and factors that contribute to the general experience, but can be heavily influenced by the *interaction technique* and *format of the game*. Games like exertion games (Mueller and Berthouze, in Bernhaupt, 2010) need a completely different set up for evaluating the user experience during performing the exertion game compared for example to a table top game (Köffel et al., in Bernhaupt, 2010).

Additionally important for new forms of interaction and formats is the evaluation of input and output devices. Evaluation for in- and output pose the problem to what extend traditional usability is evaluated and to what extend user experience. It is still and open question if usability might be one factor of user experience, or if usability is an independent factor of user experience.

4. UX EVALUATION METHODS

Game development can be structured in a set of development phases. Most of these phases are used in standard software development processes. Other phases are special for game development. Following Novak (2008) the following phases are used to structure the overview on methods for evaluating user experience in games:

- Concept: This phase is dedicated to the initial game idea and is devoted to producing a first concept document describing the game. The development team in this phase is typically small (e.g. consisting of designer, programmer, artist and producer).
- Pre-Production Phase: This phase includes the development of art style guides, production plans and first description of the game design and the technical design document.
- Prototype: Goal of this phase is a first working piece of software allowing to demonstrate key characteristics of the game and enabling to understand basic concepts related to the general user experience of the game ("Is the game fun to play?").
- Production: The production phase can range from few weeks development to years of programming. This phase can be structured additionally, following approaches like an increment to completion approach, a cascade approach or an "iterative- until you drop" approach (Irish, 2005).
- Localization: an important phase for games that will be delivered to different markets (countries) is the localization phase. In this phase game-play can be adjusted to suit the tastes of the market, to allow

language translation and modifications due to local regulatory authorities.

- Alpha-Phase: This is the phase when a game is playable from start to finish, allowing different evaluation methods to be applied to better understand aspects like fun, playability and UX.
- Beta-Phase: Main goal during this phase is normally to fix bugs. In terms of user experience in this phase lots of fine-tuning is necessary to improve the overall user experience. The beta-phase includes steps like certification or submission (the hardware-manufacturer of the proprietary platform will test the game).
- Gold: In this phase the game is sent to be manufactured.
- Post-Production: In this phase subsequent versions of the game may be released (including patches and updated) and allows to improve the user experience of the game.

Goal of evaluating user experience during the **concept phase** is to understand if the game will be fun to play and what kind of experience the player will have during game play.

During the concept phase and other early development phases like the **pre-production and prototype phase**, methods and approaches used are:

(1) User-oriented evaluation methods

- Focus Groups
- Interviews
- Informal Play-Testing
- Questionnaires
- Paper Prototypes
- PIFF questionnaire (Takatalo, in Bernhaupt, 2010)

(2) Expert oriented evaluation:

- GAP (Desurvire, in Bernhaupt, 2010)
- Heuristic evaluation using guidelines (Desurvire, 2004, Sweetser, 2005)

During the **implementation and testing phases** the following methods have been successfully used:

(1) User-oriented evaluation methods

- play testing (including bio-metrical measurements)
- (semi-structured) interviews
- observation
- video coding
- quantitative comparisons of gamers behaviours
- Questionnaires focusing on users attitudes, experiences, ….

(2) Expert oriented evaluation:

- Heuristic Evaluation (including heuristics for playability, table top, …) e.g. Mueller & Berthouze, in Bernhaupt 2010; Köffel et al., in Bernhaupt 2010.

UX evaluation methods thus can be classified based on their applicability in the different development stages, based on expert or user involvement and additionally they can be classified based on their ability to evaluate a certain game formats and the interaction techniques used.

5. UX EVALUATION OF MULTI-MODAL INTERACTION

How to evaluate UX in games that use multi-modal interaction, like for table top games, VR approaches or simply new forms of interaction in standard environments (e.g. exertion games) is still an open research question. Based on a classification of user experience evaluation methods for games, we develop key characteristics that have to be taken into account when evaluating UX in games with multi-modal interaction.

Games using multi-modal interaction are currently available on a commercial basis (e.g. the Wii system additionally uses speech input, EyeToy uses gesture, Nintendo DS uses pen/touch input with speech). In the scientific literature a variety of multi-modal interaction has been proposed. For table tops Cheok et al. (2005) develop mixed-reality interactions, or Tse et al (2007) developed rich gesture and speech interaction. For an overview and classification of multi-modal interaction in table-tops see Magerkurth et al. (2003).

Evaluation of multi-modal interaction in games has been (until very recently) conducted on an informal basis. For example Magerkurth et al. (2003) mention in the section on user experience evaluation that the game was simply played and people provided informal feedback.

When evaluating user experience of games using multi-modal interaction we propose the following procedure.

(A) Goal of the UX Evaluation

(1) Scope:
It is necessary to define the scope of the evaluation. As multimodal interaction is heavily influencing the usability (especially the learn-ability of the game) it has to be decided if usability is seen as a factor included in the overall user experience.

(2) Context:
As multi-modal interaction is influences by the usage context, it has to be defined in what contexts the game will be used in, and if evaluation shall address all of these usage contexts.

(3) Factorial approach: The authors are not aware of any method allowing the evaluation of user experience in a general or holistic way, but only methods that investigate a certain set of factors. To refine the goal of the evaluation, it is thus necessary to focus on a set of factors.<

(B) Method Mix:

When evaluating a method mix can help to improve user experience during the whole player-centered development of the game.

(1) Early stages

First paper prototypes can help to understand what combination of modalities is used in the various gaming situations.

Heuristic Evaluation using game play guidelines (Desurvire, 2004) or focusing on playability (Sweetser, 2005).

Based on the application domain (e.g. table tops) some specific heuristics are available (e.g. Köffel et al., in Bernhaupt, 2010).

(2) Later stages

First functional prototypes allow investigation of the multimodal interaction. As the usability of the multimodal interaction is influencing the overall user experience we propose a method mix:

Combination of usability study, play testing and user experience evaluation using questionnaires or retrospective think aloud.

When evaluating user experience with the above method mix it is necessary to cope with the specifities of the multimodal interaction:

- not every type of interaction can be evaluated

- the prototypical implementation might not allow to use all combinations of the modalities that shall be available in the final game.

To overcome these limitations a combination of usability study and modeling can be helpful (see Bernhaupt, Navarre, Palanque, Winkler, 2007). This approach might be expensive for one game, but in general it can be helpful to model a specific set of multi-modal interaction that (afterwards) can be deployed at several games.

6. CONCLUSION

There is a variety of methods that allow user experience evaluation in games. When looking at multimodal interaction the current set of methods must be combined to allow address the specifities of multimodality. First the scope of the evaluation has to be defined (as for multimodal interaction in games usability and user experience are two constructs that are closely related) and second the evaluation has to be based on a set of methods (method mix) to allow cope with the limitations of the methods.

ACKNOWLEDGMENTS

REFERENCES

Anderson, R.E. (1992) Social impacts of computing: Codes of professional ethics. *Social Science Computing Review 10*, 2 (Winter 1992), pp. 453-469.

Bernhaupt, R. (2010) User Experience Evaluation in Games: Concepts and Methods, Springer.

Bernhaupt, R., Navarre, D., Palanque, P. and Winkler, M. (2007) Model-based Evaluation: A new way to support Usability Evaluation of multimodal interactive Applications. In Law, E. et al. (Eds.) Maturing Usability: Quality in Software, Interaction and Value. pp. 95-127, Springer.

Cheok, A. D., Goh, K. H., Farbiz, F., Liu, W., Li, Y., Fong, S. W., Yang, X., and Teo, S. L. (2004) Human pacman: a wide area socio-physical interactive entertainment system in mixed reality. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems* (Vienna, Austria, April 24 - 29, 2004). CHI '04. ACM, New York, NY, pp. 779-780.

Desurvire, H., Caplan, M., and Toth, J. A. (2004) Using heuristics to evaluate the playability of games. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems* (Vienna, Austria, April 24 - 29, 2004). CHI '04. ACM, New York, NY, pp. 1509-1512.

Federoff, M. A. (2002) Heuristics and usability guidelines for the creation and evaluation of fun in videogames. Master's thesis, Department of Telecommunications, Indiana University

Hassenzahl, M. (2003) The Thing and I: Understanding the Relationship between User and Product. In: Blythe, M. A., Monk A. F., Overbeeke, K., Wright, P. C., (Eds.) Funology: From Usability to Enjoyment, Kluwer Academic Publishers, Netherlands.

Irish, D. (2005) The Game Producer's Handbook. Thomson Course Technology PRT.

Magerkurth, C., Stenzel, R., Streitz, N., and Neuhold, E. (2003) A Multimodal Interaction Framework for Pervasive Game Applications. Workshop at Artificial Intelligence in Mobile System 2003(AIMS 2003), Seattle, USA, Oct. 12, 2003.

Novak, J. (2008) Game Development Essentials. Delmar Cengage Learning.

Roto, V. (2006) Web browsing on mobile phones – characteristics of user experience, PhD Thesis, University of Helsinki

Tse, E., Greenberg, S., Shen C., and Forlines C. (2007). Multimodal multiplayer tabletop gaming. Comput. Entertain. 5, 2 (Apr. 2007), 12.

Sweetser, P. and Wyeth, P. (2005). GameFlow: a model for evaluating player enjoyment in games. *Comput. Entertain.* 3, 3 (Jul. 2005)

Triadic Game Evaluation: A Framework for Assessing Games with a Serious Purpose

Casper Harteveld
Delft University of Technology
Jaffalaan 5, 2600 GA
Delft, the Netherlands
+31 1527 81874
c.harteveld@tudelft.nl

ABSTRACT
Evaluatory research for games with a serious purpose is still at its infancy. It especially faces a lack of methodology. To contribute to this gap, this paper puts forward an evaluation framework called "Triadic Game Evaluation (TGE)." This framework stresses that three different perspectives have to be taken into account in the beginning, during, and at the end of the evaluation process.

Keywords
Triadic Game Design, evaluation, methodology, serious games, framework, assessment

INTRODUCTION
The use of games for serious purposes has gained much attention since the beginning of this 21st century [2]. In contrast to the century before, games, and especially digital games, are also applied for a wider variety of domains and purposes. Most institutions nowadays belief games are a *potentially* powerful tool. The question remains, will it reach its potential?

For this to be answered, *evaluatory* research is necessary. By assessing whether a game reaches its serious purpose and in what manner, it can be concluded to what extent games really are such powerful tools. Additionally, we can use the results to improve existing designs and practices. For pursuing this, we hit, however, another question: How do we perform evaluatory game research?

This question cannot be easily answered, because we are facing a *methodological gap* when it comes to evaluating games. This gap can be noted from the huge inconsistency in evaluation studies that have been done so far [3]. Scholars evaluate their games without a structured way of thinking of how to evaluate them.

To contribute to this methodological gap, this paper proposes a framework for assessing games with a serious purpose. This evaluation framework is called "Triadic Game Evaluation (TGE)" and is based on a game design philosophy called "Triadic Game Design (TGD)" [2]. In a nutshell, this philosophy stresses that designing a game

involves three equally important "worlds": the worlds of Reality, Meaning, and Play. These worlds have to be taken into account while evaluating a game as well and this is what TGE is about.

In discussing TGE further, I will first go more in-depth about what it takes to assess games. Subsequently, TGD is discussed. Following this, a generic description is given of TGE. Before the paper is concluded, I will highlight how this framework is employed for a specific game called *Levee Patroller* [2].

ASSESSING GAMES
Despite the quite long tradition of using games for serious purposes, which can even be traced back to the beginning of human civilization [2], one of the most fundamental questions on this area – does it work? – remains largely unanswered. This is due to a number of reasons.

First, while a long tradition can be observed, academically game research is still at its *infancy*. We are still exploring the medium; we are trying to understand what it is and to what extent it differs from everything else.

Second, games are *complex* objects to study. Games are rich and dynamic environments that consist of many aspects, such as sound and graphics, and are affected by many environmental variables, such as the context in which the game is played. To study this complexity, a better understanding is needed of what variables need to be considered and how they relate to one another.

Third, it is difficult to *proof* the value beyond the game. For some games this is directly visible, but in many cases, if not most, this is quite difficult. For testing the effects in the real world, many intervening variables can play a role. If it is possible to demonstrate any effects, the question remains to what extent all the effort – in terms of design, deployment, and time spent on playing the game – was actually worth it.

Fourth and finally, performing game research is quite *intensive*. It requires a lot of effort to first design a good game. To setup a good evaluation process in the second place, is often too much to be asked for due to budget and time constraints. Most scholars, therefore, stick to the so-called "smile-sheets" to assess their games [6].

A framework will not solve all these problems. It will, on the other hand, help scholars in determining what to do and

it will help in structuring evaluatory game research. For TGE and possibly other game evaluation approaches, a number of first stepping stones have been laid. I will discuss these before I get to the idea of TGE.

The First Stepping Stones
The work of Kriz and Hense [5] is one of the few attempts to build a methodology for evaluating games. Their theory-oriented evaluation has been important for highlighting that when evaluating a game, the game itself should not be seen as a "black box." Due to the complexity of games, a lot could happen in between. For example, players could get stuck and, as a consequence, get frustrated and stop playing. Therefore, the process, or the "throughput" as they call it, needs to be considered next to the input and output.

Another reason why the work of Kriz and Hense [5] is important is that they made a first attempt to determine what variables play a role in what part of the game cycle and how they relate to each other. For example, they argue that player's expectations, an input, need to be measured as this influences the throughput and eventually the output. In other words, they mapped some causal relationships between important variables.

Both aspects, the importance of the process and the mapping of variables, are illustrated in the "Design, Play and Experience (DPE)" framework as well [7]. This framework explains that in the game design process three components are relevant that each relate to an aspect of the game cycle: the development of a game by the designer ("Design"), playing the game by the player ("Play"), and the experience the player derives from playing the game ("Experience"). Each of the components is subdivided into layers, like user interface and story, which scholars need to take into account when evaluating a game.

Another stepping stone concerns the four dimensional framework by De Freitas and Oliver [1]. This framework stresses that when evaluating a game, four dimensions are relevant: the context, the pedagogy, the representation, and the learner (or player). For each of these dimensions they developed a number of questions, like "Which pedagogic models and approaches are being used?" and "What level of realism is needed to achieve the learning objectives?"

Although each framework focuses at educational games, they are the first attempts to tackle the methodological gap and are, therefore, important stepping stones for evaluatory game research. With this in mind, I will now turn to TGD, a design philosophy for creating games.

TRIADIC GAME DESIGN
Triadic Game Design is a design philosophy that stresses that three different but equally important "worlds" have to be taken into account when designing a game with a serious purpose [2]. Each world is inhabited by different people, practices, theories, criteria, and so forth.

The first world is "Reality." Games have always – how abstract they may be – a relationship with the real physical world. For games with a serious purpose this connection is even more important, as in the end, the real world needs to be affected by the game. This world is grounded in the

disciplines related to the subject matter is represented by the different stakeholders, with their expertise and opinions.

The second world is "Meaning." Although no game can be considered "meaningless," a more elaborate consideration needs to be made to intentionally achieve a meaningful effect beyond the game experience. The world of Meaning is concerned with this creation of value. In considering this, other disciplines, such as the learning sciences, should be involved and, depending on the purpose, people, like teachers, that know how to achieve a certain value.

The third world is "Play." Games are first and foremost a tool (or medium). When we think of games, we think of highly interactive and engaging tools that immerse people into a fictive situation. For creating such an environment, aspects, disciplines, and people, like game designers and modelers, need to be involved.

The three worlds are not disconnected from each other. In fact, they are tightly coupled and even overlap in certain areas. The difficult task for designers is to find a synthesis between the three worlds. This task is difficult, because as I noticed, often the three worlds clash with each other on design issues. The synthesis, or a balance, is necessary, since in the end a game with a serious purpose needs to have an effect on the real world, achieve its purpose, and be fun and engaging to play.

While this philosophy is specific to game design, the ideas can easily form the basis of an evaluation framework. As the DPE framework points out [7], the design of a game influences how it is played and, eventually, what results are achieved. This means that if the principles of TGD are used to design a game, it can (and should) also be tested whether the criteria of all three worlds are achieved. With the development of TGE such an evaluation becomes possible.

TRIADIC GAME EVALUATION
Based on the first stepping stones and TGD an evaluation framework has been developed that involves a way of thinking and a way of working.

The Way of Thinking
Similar to TGD, TGE considers the worlds of Reality, Meaning, and Play. It asserts that the aspects and criteria of each world need to be part of the evaluation process to get strong, reliable, and interpretable results. The reason for this is clear-cut: if all three worlds play an equally important role for creating a good game, variables from each world affect the eventual experience, and thus the results.
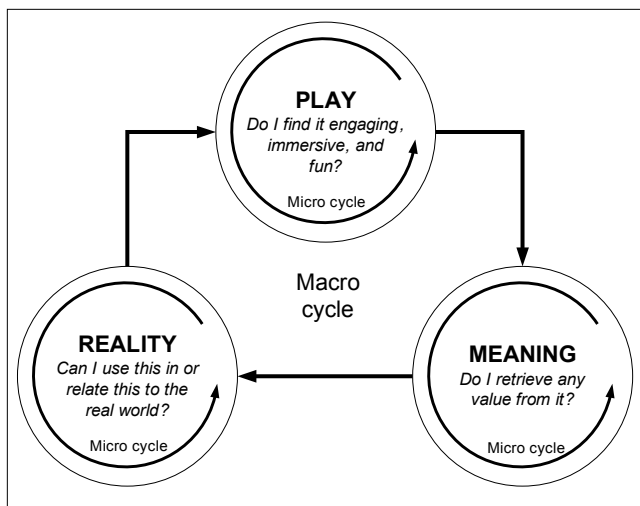
Figure 1. The sensemaking cycle

This means that when evaluating a game, aspects related to each of the worlds have to be looked into. These aspects can be decomposed into as many variables that the researchers find relevant for their evaluation. In doing this, the criteria of each world can be helpful. For example, for Reality the criteria could concern flexibility, validity, and fidelity; for Meaning motivation, relevance, and transfer; and for Play engagement, immersion, and fun.

Similar to De Freitas and Oliver [1], TGE also stresses that the player needs to be part of the evaluation. Unlike their framework, with TGE the player is not a separate dimension. The player can be seen as part of each of the three worlds as he or she can be conceived of having three different roles:

- *Player as person*: the player is a person in the real world. He or she has demographics, a personality, attitudes, and so on, that could affect how the game is experienced.

- *Player as interpreter (or learner)*: people interpret information differently and so do players. This depends, for example, on the existing knowledge, education, learning styles, and expectations.

- *Player as player*: players differ amongst each other. This means that amongst other things previous experience with games as well as game preferences can make a difference in the results.

What researchers want to consider depends very much on their game and the context in which they apply it. They only have to make sure that they structure their evaluation on the basis of Reality, Meaning, and Play.

The Way of Working
Similar to Kriz and Hense [5] and Winn [7], TGE insists that for evaluating a game, it is important to look into what happens "in-between." It has a very specific reason for this. When playing a game, players go through what I call a "sensemaking cycle" (Fig. 1). This cycle is reminiscent of Kolb's learning cycle [4]. It says that players continuously are "making sense" of their play experience by questioning and/or feeling consciously and/or unconsciously whether

they find the game engaging, immersive and fun, whether they learn from it or derive some other specific value from it, and whether this is something they can use or relate to in the real world.

The sensemaking cycle consists of three "micro cycles" and one large "macro cycle". The micro cycles are tied to each world. It could be that the micro cycle in itself is enough for players to continue playing. This happens with most of the entertainment games. The macro cycle connects the three worlds and is critical for the success of a game.

The cycle happens iteratively and continuously. As a researcher it would be valuable to capture this process as the eventual results become better interpretable. To do this, methods should be employed that are able to retrieve insights intermittently. This should be done with cause, because the methods themselves can severely disrupt the sensemaking cycle.

As for the methods in general, TGE does not prescribe any particular ones, but as long as no clear relationship exist, it is important to use various qualitative methods. It further recommends logging the game data. Game data enables to get a clearer picture of the play experience.

THE CASE OF LEVEE PATROLLER
To make the idea of TGE concrete, I will show how it is applied to a game I designed. Before I discuss the setup of the evaluation, I will first explain what the game is about.

The Game
Levee Patroller can be described as a "single-player 3-D first person game." This means the game is solely played by one user from the perspective of the player character. In the game, the player's role is that of a "levee patroller." These are people who inspect "levees" regularly or in cases of emergency. Levees are the natural and artificial barriers that protect the land from flooding. Inspecting these levees is of high importance to the Netherlands due to the high risks involved in a possible levee failure. Therefore, it was desired that patrollers are able to increase their inspection knowledge and skills. This is difficult to achieve in reality, because failures occur rarely. To make it possible for patrollers to get experience, it was decided to develop a game.

In the eventual game, the goal is to find every virtual failure (see Fig. 2 for an example). After finding a failure, players need to fill out a report, possibly measure the failure, and contact the coordinating field office to discuss the severity of the problem. If players do well, they get a high score. If they do not do well, they get a low score. Additionally, a levee breach that floods the whole region may be the result.

The Setup
For the evaluatory study of *Levee Patroller*, I set up a training of three weeks. At the start and the end of the training a short meeting takes place. Aside from filling out the questionnaires, the participants play one exercise at both meetings. The start meeting further includes an introduction to the game, and the end meeting includes a

discussion. In between the meetings, the participants play the game at home. They are required to play two exercises per week.

With this setup in mind, I can explain some of the methods I am using for this study and how they relate to TGE.

Figure 2. A screen shot of a levee failure

Pre- and post-questionnaire
The questionnaires are the prime method in this study. During the pre-questionnaire the background variables that relate to "player as person, interpreter, and player" are first asked. For example, one of the questions asks whether the participant has any experience with First Person Shooters (FPSs), a game genre very similar to *Levee Patroller*.

The post-questionnaire goes into whether the game fulfilled the expectations and deals with the design. The game has, for instance, no sound and one of the questions deals with whether participants missed this. This part deals with the world of Play.

Both questionnaires end with two parts that relate to the worlds of Reality and especially Meaning. One part deals with the perceptions the participants have about levee inspection and the other part consists of real and virtual pictures. For this latter part, participants need to write down what they see and how they would deal with it.

Game log
Every action, from accessing a help menu to measuring a failure, is logged. As such, it becomes possible to analyze how participants play the game. Such analysis could be helpful to improve the design, but is also helpful in understanding how participants go through the sensemaking cycle. The game logs are sent to a central server over the Internet. As a backup, the logs are also saved locally.

In-game evaluation
After each exercise, participants have to fill out a short evaluation form, which is based on the worlds of Reality, Meaning, and Play. This way, it is attempted to capture the sensemaking cycle as well. Similar to the game logs, these forms are saved locally and sent to a central server over the Internet.

CONCLUSION
Evaluatory research for games with a serious purpose is still at its infancy. Some initial work has been done, but we are still exploring how to assess these types of games. In this paper, I put forward a framework called Triadic Game Evaluation (TGE), which can be used for such evaluations. I illustrated its use with the game *Levee Patroller*.

The framework is still under development and the results with *Levee Patroller* will highlight to what extent it is helpful to consistently take the worlds of Reality, Meaning, and Play into account. Additionally, it will become clear what variables of each world matter in this regard. With these insights, the framework can be further elaborated.

REFERENCES
1. De Freitas, S., and Oliver, M. How can exploratory learning with games and simulations within the curriculum be most effectively evaluated? *Computers & Education 46*, 3 (2006), 249-264.

2. Harteveld, C. *Triadic Game Design: Balancing Reality, Meaning and Play*. Springer, London, UK, 2010.

3. Ke, F. A qualitative meta-analysis of computer games as learning tools, in R.E. Ferdig (ed.), *Handbook of Research on Effective Electronic Gaming in Education* (Vol. I, pp. 1-32), Information Science Reference, Hershey PA, 2009.

4. Kolb, D.A. *Experiential learning: Experience as the source of learning and development*. Prentice Hall, Upper Saddle River NJ, 1984.

5. Kriz, W.C., and Hense, J.U. Theory-oriented for the design of and research in gaming and simulation. *Simulation & Gaming 37*, 2 (2006), 268-283.

6. Prensky, M. *Digital game-based learning*. McGraw-Hill, New York NY, 2001.

7. Winn, B.M. The Design, Play, and Experience framework, in R.E. Ferdig (ed.), *Handbook of Research on Effective Electronic Gaming in Education* (Vol. III, pp. 1010-1024), Information Science Reference, Hershey PA, 2009.

A Videogame Management System in Mathematics

Francisco J. Álvarez Rodríguez
Jaime Muñoz Arteaga
Universidad Autónoma de Aguascalientes
Ave. Universidad, 940
Aguascalientes, Ags. 20131 México
+52 449 9108419
fjalvar@correo.uaa.mx
jmunozar@correo.uaa.mx

Arturo Barajas Saavedra
Luis E. Bautista Villalpando
Universidad Autónoma de Aguascalientes
Ave. Universidad, 940
Aguascalientes, Ags.20131 México
+52 449 9108400
abarajas@correo.uaa.mx
lebautis@correo.uaa.mx

ABSTRACT

One of the most relevant subjects for the intellectual formation of elementary and high-school students is Mathematics where its importance goes back to ancient civilizations and which its importance is underestimated nowadays. This situation occurs in Mexico, where 69% of the total population of elementary school students between the third and sixth grade have insufficient level of mathematics knowledge, and this has resulted in the need to use a new mechanism to complement student's classroom learning. This work proposes the use of an Educational Videogame, where the first part of this proposal is a mobile suite of videogames for teaching mathematics and the second is a recommender system, it allows to students to reach contents according to their needs, this is achieved through a core engine that infers from an initial profile that covers the requirements of each user.

Keywords

Mobile learning, E-learning, Mathematics, Videogames

INTRODUCTION

In recent years, Mexico has implemented the use of complementary technologies for the education in public schools. Mainly, the use of computers, which has been implemented with special software that provides diverse resources like bibliographical references, video, audio, fixed maps, images, interactive activities and material of other educative programs. Contents from Microsoft Encarta Encyclopedia have been included also. (INEGI, 2009).

Since 2003, Mexico has implemented the usage of technology for education within public and private schools at all levels, from elementary to higher education. Mainly it has been implemented throughout the use of PCs with special software.

Despite the effort, has found through studies such as ENLACE, which is a test that has as main goal to provide diagnostic information on the extent to which students have acquired themes and contents of the subjects such as, Spanish, Mathematics and Science (Secretaría de Educación Pública, 2009), where the learning levels of those subjects have been very low.

ENLACE test was applied to 15'766,608 students from third grade in elementary school to third grade in junior high school in 2009. The results obtained from this (Table 1) test show the impacts of the different programs that the Ministry of Education has orchestrated and make the challenges evident in which it is important to obtain significant advances.

Table 1. ENLACE test results (Secretaría de Educación Pública, 2009)

Year	Achievement Level (%)				
	Insufficient	Elemental	Good	Excellent	Students
2006	21.0	61.4	16.0	1.6	7,506,255
2007	20.2	57.5	19.0	3.3	7,962,825
2008	22.8	49.5	23.0	4.7	8,108,694
2009	22.1	54.9	21.0	1.9	1,840,417

Extra School Education

Learning is not restricted to the time spent at school. It begins at birth and continues all your life (UNESCO, 2009). Francesco Tonucci argues that the most important experiences for personal development in childhood and youth are lived outside school. A truly meaningful education should also be built on the informal experiences of students that happen before or at the same time of school learning (Vázquez Alonso & Manassero Mas, 2007).

The present pattern, in which we have education at the beginning of our lives, is changing. Lifelong learning is becoming part of modern life. This is because rapid technological change and growth in information require ongoing learning. (UNESCO, 2009).

In modern context, the individuals evolve in an atmosphere strongly influenced and formed by the presence diverse artificial surroundings. The daily possibilities of students having outside of the classroom meaningful and relevant experiences are very intense and important, making the classroom, which is one of the most important sources of knowledge, no longer the primary source of information for people (Vázquez Alonso & Manassero Mas, 2007). New Information and Communication Technologies (ICT) are the modern delivery tools for education, making possible

what was not before. Radio, television, computers, electronic devices, video games and Internet are strongly embedded in the everyday life of individuals.

In contrast with the present paradigm, where formal and non-formal educations are treated as a mutually disjointed, the boundaries between formal and non-formal education must be blurred in order to achieve these meaningful and relevant experiences.

To achieve these objectives both - the improvement of academic standards in math in elementary school children and the introduction of a mechanism which allows to achieve a level of excellence through the elimination of the boundary between formal and non-formal education -, are proposed the use of a system that incorporates a set of resources created to large scale as well as a management system for providing them in manner and in time when they are required.

The first element proposed is a set of videogames developed to run over mobile devices with the next specifications: 1) One problem centered approach, 2) Simple functional graphic user interface; and 3) Challenge and competition promoter.

The second element is a system able to manage this videogames and make them reachable for students according to their needs.

MOBILE VIDEOGAMES FOR TEACHING

A necessity for applying new and more efficient teaching methods that extends the classroom beyond the school into environments that are more strongly ingrained in the students, like videogames, exists.

The classification currently exists game has very few options for access educational resources, this new classification is a strong possibility for the creation of learning content embedded in video games.

It has been observed in many studies, like (Virvou, Katsionis, & Manos, 2005), that the videogame phenomena can be used in advantage to the formal learning process outside school, this due to videogames are much more interesting to kids, and their use as a teaching tools is plausible. Also, others studies (Procuradía Federal del Consumidor, 2009) have showed that videogames are deeply rooted in the daily life of Mexicans kids, showing that 64% of the surveyed people own a videogame console since three years ago, 55% play from one to three days a week and 72% play between one and three hours daily.

On the other hand, the rapid growth of wireless and mobile technologies has resulted in the mobile learning that has been gradually considered as a novel and effective form of learning because this inherits all the advantages of e-learning as well as breaks the limitations of learning time and space occurring in the traditional classroom teaching (Chen & Hsu, 2008).

Thus, the intersection among education, videogames and mobile devices results in a niche that can produce a meaningful improvement in the learning of mathematics in Mexico.

Math Videogames

Ten videogames have been developed in order to cover the four main weak topics in education: fractions, geometry, proportion and variation, and 2D plane. Also these videogames were developed based on textbooks published by the Ministry of Education.
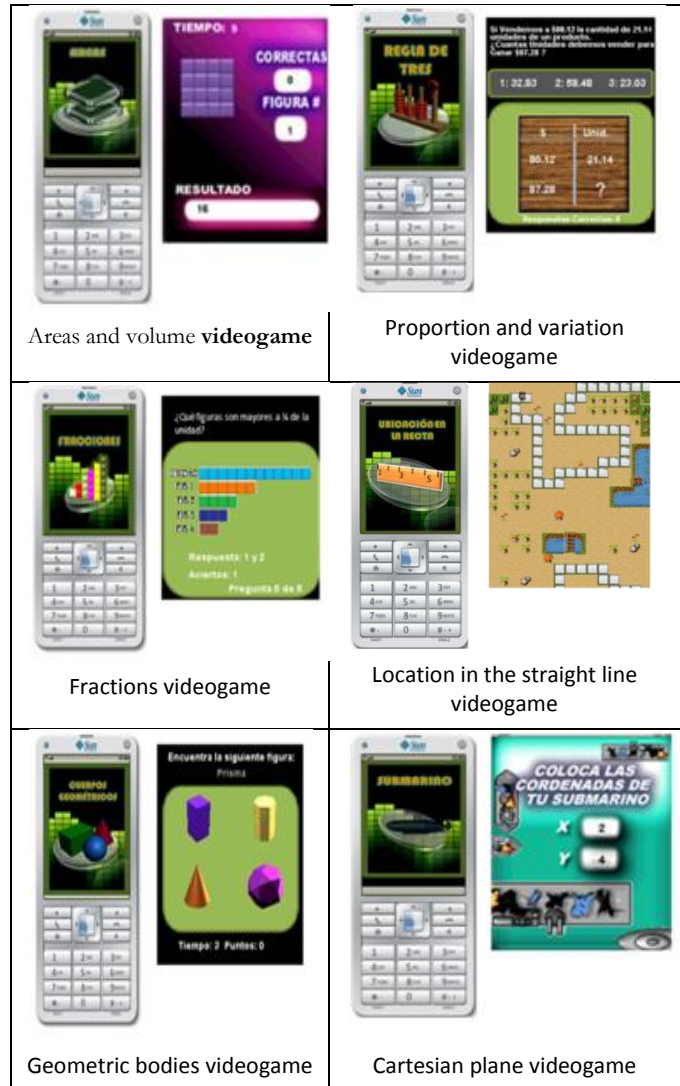

Areas and volume **videogame**	Proportion and variation videogame
Fractions videogame	Location in the straight line videogame
Geometric bodies videogame	Cartesian plane videogame

Figure 1. Partial set of videogames in mathematic

The topics established for the videogames are as follows:

1) Areas and volume, allows children to mentally practice formulas of geometric shapes and bodies. It also enables students to calculate areas and volumes through visual analysis (See figure 1).
2) Proportion and variation, enables the student to distinguish problems where they can apply the technique of the "rule of three", using real-life examples. Set out a random collection of sentences and quantities so that the student does not memorize

the answers to the problems after several executions. (See figure 1).

3) Geometric bodies, allows students to recall names of different geometrical shapes within a limit time. One screen appears four geometrical bodies randomly generated and the student has to select the one that matches the indicated name (See figure 1)..

4) Location in the straight line, its objective is to prompt the student to find points on a line by varying the value of the units and apply basic algebraic operations (See figure 1)..

5) Cartesian plane, allows student to learn and practice the location of points in the Cartesian plane as a strategy game.

6) Fractions, uses strips to enable children to observe and understand the use of fractions with different exercises (See figure 1)..

7) Mathematical concepts, includes a puzzle game to enable students to practice and memorize the basic math concepts.

It is important to settle than these first seven games were developed with Java ME. Finally, two more games were developed under .NET technology, with versions for PDAs and Smart phones. These games emphasize the use of basic operations through four different mini-games: Time Attack, Operators, Scale, and Puzzle.

RECOMMENDER SYSTEM

Once the knowledge acquisition problem in the area of mathematics for basic education indicated above and the use of videogames in mobile devices like a proposed solution for this problem has been defined, the creation of a scheme in which the videogames by means of a repository are managed to be supplied in an efficient way to the possible students according to their needs, that is to say, we suppose that some student has deficiencies in his abilities to solve problems related to the rule of three and nevertheless its knowledge related to geometric bodies is acceptable or excellent, thus we would need some mechanism within the videogame manager that supplies those games on which a particular interest exists by the student, in other words, those games that help to the student to improve their knowledge or abilities in connection with some subject in which is not very efficient. Due to this, is proposed the creation of a Semantic Web scheme which will allow the management of knowledge in an efficient way.

Ontological Representation for Academic Videogames

In the field of Web applications development, there are different mechanisms for the representation of ontological models; in particular, within our investigation was used Protégé 4 software for the edition of Ontologies (Standford Center for Biomedical Informatics Research, 2009), in which we have represented our scheme of videogames through an OWL Ontology.

In our classes definition, initial games were declared as our objects of interest, that is to say and just for representation we declared four games, independent of each other and in

theory created and managed by different videogame programming groups within our development team, which are: BusquedaDelTesoro (Treasure Hunt), Cubos (Buckets), ReglaDeTres (Rule of three) and Submarinos (Submarines).

In this way, we can create equivalent classes for videogame in a way that the reasoner can make an inference according to those classes and also can create an inference tree (Fig.2)
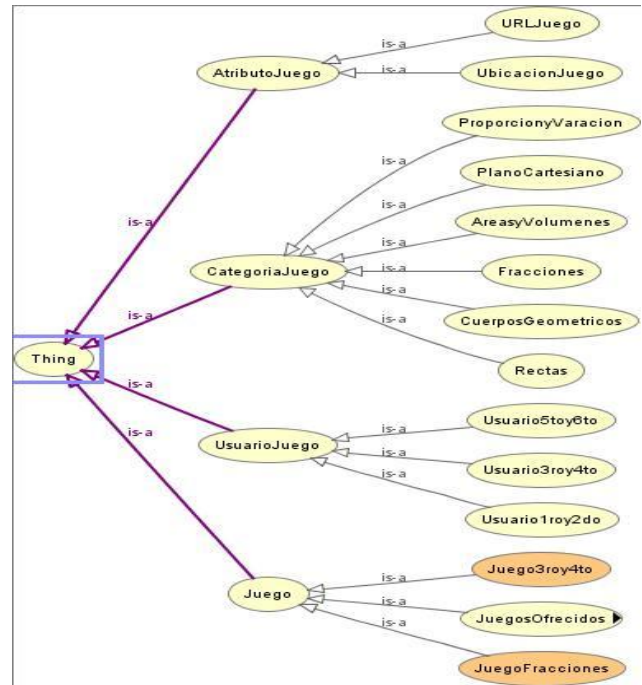

Figure 2. Ontological representation of videogames in mathematic.

Once proven our ontology, we can work with the same within a scheme of XML- RDF files, which shows the ontological description of the videogame later to be to implanted within our structure of videogame manager.

One of the advantages of the use of Ontologies such as OWL is to be able to make use of some reasoner, which allows us to create inferences according to pre-established rules for the solution of some problem. Protégé allows four different reasoners that classify and verify the consistency of the created Ontology, for our study case in particular we used Fact++ (Tsarkov & Horrocks, 2009).

Once defined our games, we can offer them towards determined group of users to be able to customize its use.

Once tested our Ontology, we can work with it within a scheme of XML- RDF files, which shows the ontological description of the videogame to be implanted later within our structure of videogames manager.

Videogames Manager

Once created our Ontology for the organization and as well as the creation of inference of videogames, it is necessary to create a structure to manage the videogames and to be able to provide them in an efficient way to our clients. The goals of our videogame manager are: 1) Record and

organize videogames, 2) Provide the games as well as to create profiles of preferences for users; and 3) Update the information of existing videogames and notify it to the users.

Another part of our videogames manager is *events*. The main idea on which is based the use of events and notification is that an object (videogame) can react to the changes that happen in another object (videogame) through events (update of videogames) that are notified to each other within the network (Coulouris, Dollimore, & Kindberg).

For our research in particular we have created a structure based on this principle where our Videogame Manager is constituted by two main components that are, the *Game Provider* and the *Game Notifier*. Within the Game Provider a catalogue of games previously registered is obtained as result of the above process, this catalogue will be organized subsequently according to the scheme declared in our Ontology to be given to the Game Notifier which will inform to the clients that are interested in one or more games as it can be seen on figure 3.
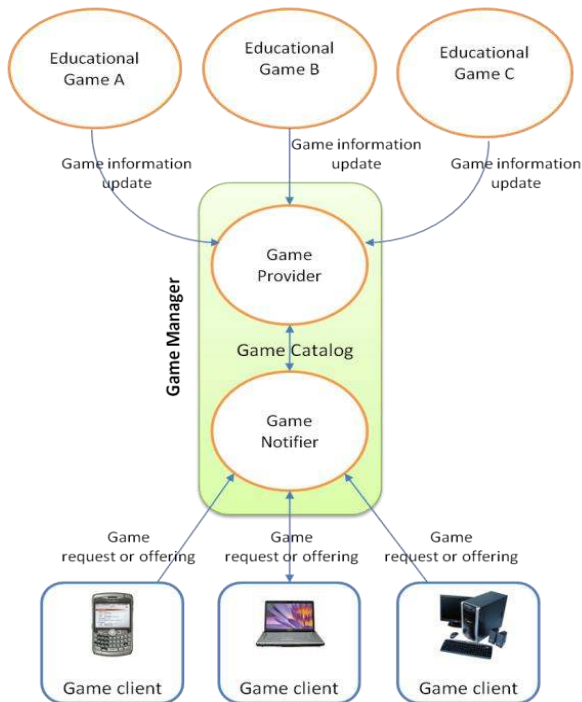

Figure 3. Videogame management system

On the other hand our Videogame Manager through Game Provider, will receive requests to register or update videogames, for example, suppose that the game Treasure Hunt registers continuous changes that modify its category, according to the principle of events and notifications, whenever the game is updated it is generated an updating event which notifies the subscriber of this change, through the included reasoner, where it will infer the possible clients who are interested in the update of the videogame, still more, the same clients will be able to register those games in which they are interested making use of the Game

Notifier, for subsequently updating the Ontology and create new inferences.

CONCLUSION

This work presents the basics for the implementation of a scheme of games recommended and at the same time a form to organize such games where they can work together and creates flow of processes between them to achieve the improvement of student knowledge within mathematics area. The next step in our work is the growing of the game´s repository in a distributed way, where several game creators can share their games through web services and gets a feedback of students to know if their knowledge has been improved after the use of such distributed system.

REFERENCES

1. Chen, C.-M., & Hsu, S.-H. (2008). Personalized Intelligent Mobile Learning System for Supporting Effective English Learning. *Journal of Educational Technology & Society , 11* (3), 153-180.
2. Coulouris, G., Dollimore, J., & Kindberg, T. *Distributed Systems Concepts and Design* (Tercera ed.). Ed. Addison Wesley.
3. Horridge, M. (2009). *A Practical Guide To Building OWL Ontologies Using Protege 4 and CO-ODE Tools Edition 1.2.* The University of Manchester.
4. INEGI. (7 de Diciembre de 2009). *Enciclomedia ¿Qué es?* Recuperado el 7 de Diciembre de 2009, de Enciclomedia Una forma diferente de aprender y enseñar:
5. Martinón, A., & Riera, T. (3 de marzo de 1999). *Importancia de las Matemáticas*. Recuperado el 19 de marzo de 2009.
6. Procuradía Federal del Consumidor. (16 de Febrero de 2009). *PROFECO - Resultados de encuestas y sondeos.* Recuperado el 20 de Marzo de 2009, de Resultados del sondeo sobre consolas y videojuegos:
7. Secretaría de Educación Pública. (2009). *Evaluación Nacional del Logro Académico en Centros Escolares, ENLACE /*
8. Standford Center for Biomedical Informatics Research. (2009). *The Protégé Ontology Editor and Knowledge Acquisition System.* Recuperado el 2009, de Stanford Center for Biomedical Informatics Research: http://protege.stanford.edu/
9. Tsarkov, D., & Horrocks, I. (2009). *FaCT++.* Recuperado el 2009, de School of Computer Science - University of Manchester.
10. UNESCO. (15 de Septiembre de 2009). *Non-Formal Education*. Recuperado el 2 de MArzo de 2010, de ICT in Education,
11. Vázquez Alonso, Á., & Manassero Mas, M. A. (7 de Febrero de 2007). *Las actividades extraescolares relacionadas con la ciencia y la tecnología.*
12. Virvou, M., Katsionis, G., & Manos, K. (2005). Combining Software Games with Education: Evaluation of its Educational Effectiveness. *Educational Technology & Society , 8* (2), 54-65.

Design Aspects for Rehabilitation Games for MS Patients

Sofie Notelaers   Tom De Weyer   Karel Robert

Chris Raymaekers   Karin Coninx

Hasselt University – tUL – IBBT
Expertise Centre for Digital Media
Wetenschapspark 2, 3590 Diepenbeek, Belgium
{sofie,notelaers, tom.deweyer, karel.robert, chris.raymaekers, karin.coninx}@uhasselt.be

ABSTRACT
Computer-supported rehabilitation can benefit many groups of patients. However, when designing such a therapy, it is important to take the characteristics of the patient population and the wishes of the therapists involved into account. This paper therefore focuses on the requirements of rehabilitation games for Multiple Sclerosis patients. As we have created a system for rehabilitating Multiple Sclerosis, based on a virtual environment with force feedback, we will discuss how these requirements can be met using the rehabilitation system as an example.

Categories and Subject Descriptors
J.3 [**Computer Applications**]: Life and medical sciences—*health*; H.5 [**Information Systems and Presentation**]: User interfaces; D.2.8 [**Software Engineering**]: Design

General Terms
Design, Human factors

Keywords
Rehabilitation, virtual environments, haptics

1. INTRODUCTION
Multiple Sclerosis (MS) is a chronic progressive disease of the central nervous system. Depending on the distribution of lesions within the brain, MS may clinically present with impairments of strength, muscle tone, sensation, co-ordination, balance, bladder and bowel function, as well as visual and cognitive deficits, often leading to severe limitations of functioning in daily life. Studies of exercise therapy, focused on balance and walking outcome parameters, have shown a beneficial effect regarding muscle strength, exercise tolerance level, functional mobility and quality of life, while no important deleterious effects were reported [2]. Very few studies have properly investigated the therapeutic potential of arm training in persons with MS. Because training duration and training intensity are considered to be key factors for a successful neurological rehabilitation [5], rehabilitation robotics are introduced to provide additional exercises that can be performed independently of the therapist. Crucially, the patient has to be motivated, and keep being motivated to continue the training regime. We are investigating the value of force-feedback assisted rehabilitation of the upper extremities in persons with MS [4] and how such technologies can be applied in a self-motivating way. More concretely, a virtual environment (VE) has been realized, which provides the patients with training tasks to be carried out and monitors their progress and success rate [3].

In order to realize a good rehabilitation therapy, it is important to take human factors into account. Therapy can only be successful if the patients have to perform rehabilitation tasks, which are challenging, but can still be executed by a patient. Therefore, limitations of the patients' abilities, both physical and cognitive, must be taken into account. Furthermore, rehabilitation tasks must be repeated often in order to be effective. It is therefore important to take motivational aspects into account. This can be realized by providing game-like rehabilitation tasks. Finally, the therapists must be able to define and adjust the patients' therapies. Therefore the rehabilitation system must support the therapists in determining the patients' progress and to make the necessary corrections.

This paper discusses the design aspects that must be taken into account when creating rehabilitation games for MS patients. The requirements from the patients' and therapists' point-of-view are elaborated on. Finally, we will discuss how our solution for the rehabilitation of the upper limbs meets these requirements.

2. REQUIREMENTS
Different aspects must be taken into account when creating rehabilitation games. Both the patients and the therapists must be able to work with the system efficiently in order to achieve the desired results from the therapy.

2.1 Patients' Requirements
Multiple Sclerosis is a disease, which typically worsens in several stages. Patients, which have troubles using their arms, already have the disease in a severe form and are typically elderly people. Often, these people are bound to a wheel chair, and have cognitive problems. A rehabilitation system must therefore be easy to understand and use. In order to give feedback of the patients' movements in 3D, such a system should also provide feedback in 3D. It is however, difficult for these patients to have insight in 3D movements. Furthermore, patients have difficulties using their arms. This includes both problems with muscle strength as with muscle control. The system must therefore be able to also support patients with severe difficulties. Finally, patients must not need to control menus and other typical UI elements as many patients have no to little experience with computers. Even patient who are familiar with this kind of

interface, are not always able to control them in an efficient manner due to their muscle problems.

These leads to the following requirements:

1. the game play must be easy to understand

2. feedback about users' movements must be clear

3. patients who have muscle difficulties must be supported by the game

4. "classical" UI elements, such as menus and slider must be avoided

Next to these requirements, it is important to take the training frequency into account. In order to be successful, a patients needs to train three times per week, during several weeks or even months. As games must be simple in order to satisfy the above-mentioned requirements, they run the risks often getting boring soon. The games must therefore pose enough challenges, without becoming to difficult. Furthermore, the patients must receive feedback about their performance. They need to get the feeling that they are progressing. This can be achieved by awarding the patients when they make progress. In order to not discourage patients who have many difficulties, the amount of progress that needs to be achieved must be adapted to patients' abilities. Alternatively, different types of awards can be defined, ranging from small awards for limited progress to big awards for good progress.

These leads to the following requirements:

5. the difficulty must be adapted to the different patients

6. the games must be challenging

7. progress must be awarded

8. all patients must be able to make progress

Finally, in order to further motivate patients, extra incentives such as social interaction and storytelling [1] can be included in the game. These are not strictly necessary, but are useful to improve the games.

These leads to the following requirement:

9. extra motivational aspects can be taken into account

2.2 Therapists' requirements

In order to meet the patients' requirements, it is important to enable the therapists to define the therapy sessions. As games can include many parameters, including force feedback settings, standard difficulty levels must be provided. Of course, the therapist must be able to assess the patients' progress in order to be able to determine the most suitable settings. Feedback about the patients' progress should therefore also be provided for the therapist.

These leads to the following requirements:

10. therapists must be able to define therapy sessions

11. therapists must be able to assess a patient's progress

12. therapists must be able to change difficulty settings for an individual patient

3. REHABILITATION GAMES
We have created a system for provding MS patients with rehabilitation games. These games take place in a virtual environment, which acts upon a patient's movement. For this purpose, we use a HapticMaster [6] as it can provide the patient with force feedback.

The therapy system provides two types of games: simple games, which clinically trains a particular movement, and advanced games, which combine several movements. Figure 1 shows an example of a simple game, called lifting, where the patient needs to move an object up and down between two targets.
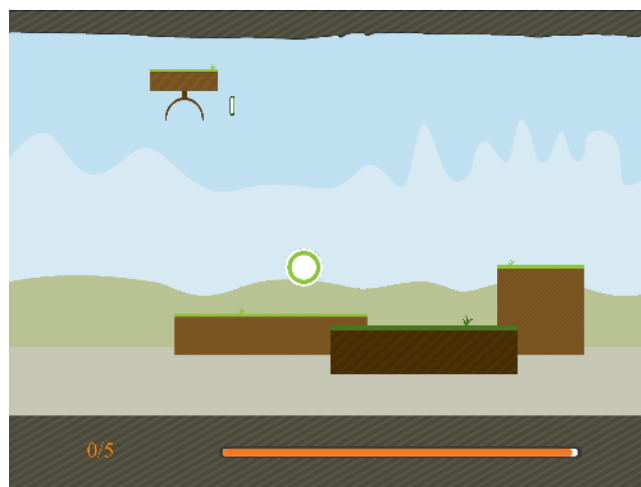

Figure 1: Lifting game

As this game only trains one pariticular movement, it is not difficult to learn. Furthermore, the target areas have a shape that indicate that the circle, controlled by the patients, needs to be docked there. This games thus ensures requirement 1 (*the game play must be easy to understand*). As there is a one-to-one mapping of a patient's movement to the movements of the circle, requirement 2 (*feedback about users' movements must be clear*) is at first sight also satisfied. This is however not sufficient: movements in the 2D plane, in which the targets are located, are clear to the patient. However, the patient can also move forward and backward, which can be difficult to understand. Therefore, the circle changes color according to the forward and backward movement. If the patient holds the circle in or near the plane, the circle is colored green. When the distance with the plane becomes larger, the circle is colored orange, and finally red when it is too far away. This color system ensures that requirement 1 is met.

In order to satisfy requirement 3 (*patients who have muscle difficulties must be supported by the game*), gravity compensation is implemented: the HapticMaster constantly applies

a force, which counteracts the weight of the patient's arm. Furthermore, patients, who experience much difficulty can be guided towards the path that they should follow or even to the target itself. This principle allows to define several difficulty levels, thus ensuring that requirement 5 (*the difficulty must be adapted to the different patients*) is met.

As a game's difficulty can be adjusted for each patient, the patients must perform at their best in order to achieve the goal. This can further be realized, by changing the number of times the target must be successfully reached and the time that is available to achieve this. Hence, requirements 6 (*the games must be challenging*) and 8 (*all patients must be able to make progress*) are met.

We realize however that simple games may become boring after a while. The next section will therefore elaborate on ways to motivate patients to play these games.

4. MOTIVATIONAL ASPECTS

Although the simple games can become boring, it's important for the patients' progress that they play them. Patients must therefore be awarded to play these game. In order to support this, we envision an overall game concept (see figure 2. In this concept, patients can unlock more challenging and engaging games by successfully finishing a simple game. This way, requirement 7 (*progress must be awarded*) is satisfied. Furthermore, by incorporating themes (common elements in different games), a story behind the games can be hinted upon. This meets requirement 9 (*extra motivational aspects can be taken into account*). As a patient can move from one game to another, by simple moving his arm in the direction of the next game, menu's are avoided. This satisfies requirement 4 (*"classical" UI elements, such as menus and slider must be avoided*).
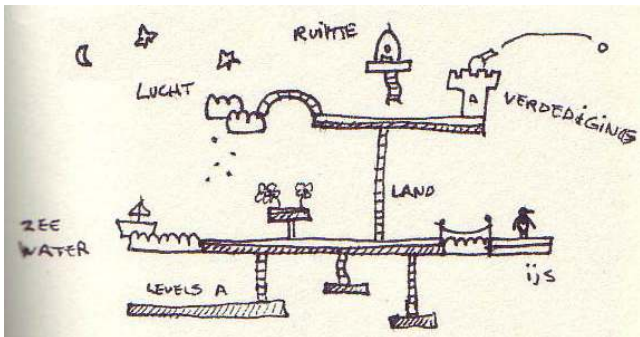

Figure 2: Overal game concept

In order further satisfy requirement 9, we are investigating the usefulness of collaborative games. For this purpose, we have created a game, where two patients (or a patient together with a family member) can play together. Both players can push one side of a beam upwards, as depicted in figure 3. By controlling the height of both sides, a ball can roll over the beam and hit some stars, which represent points. Both the HapticMaster as a WiiMote can be used as input device. More information about this collaborative game can be found in [7].
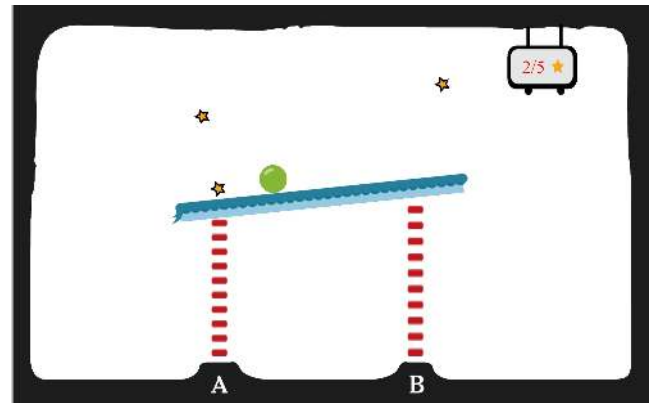
5. THERAPIST INTERFACE

Figure 3: Collaborative game

As MS patients have a different progression of the disease, each patient needs an individual training program in order to have a successful rehabilitation. The therapist therefore needs to be able to create these personalized training programs. We have created an interface that allows therapists to do this (see figure 4). The left column in this figure contains all games that are currently available. The therapist can compose therapy sessions by choosing them from the left column. The gray area in the top of the figure shows the selected games. This process ensures that requirement 10 (*therapists must be able to define therapy sessions*) is met.

In the middle of the figure, the therapist can change the difficulty level according to three axes (general parameters, such as the workspace size, haptic parameters, such as the size of the forces, and the training volume parameters, such as the number of repetitions). This way, the difficulty can be personalized. For endurance training, the haptic difficulty will be kept low whilst the training volume category will be made high. Power training on the other hand requires a high haptic difficulty with fewer repetitions. This way of working satisfies requirement 12 (*therapists must be able to change difficulty settings for an individual patient*).

At this moment, however, our system is not yet able to provide the therapist with good feedback about the patients' progress. This feedback will be defined together with therapists in the coming months. Therefore, we are not yet able to meet requirement 11 (*therapists must be able to assess a patient's progress*).

6. CONCLUSIONS

This paper presented different aspects that must be taken into account when creating a game for the rehabilitation of MS patients. Although, these requirements are very diverse, it is possible to take this set into account. We are currently creating a system for rehabilitating MS patients. Some first game have been realized, which are able to meet most of the requirements. In a next phase, more challenging games and an overall game concept will be realized. Together with feedback about the patients;' progress for evaluation purposes, these will ensure that the remaining requirements are also satisfied.
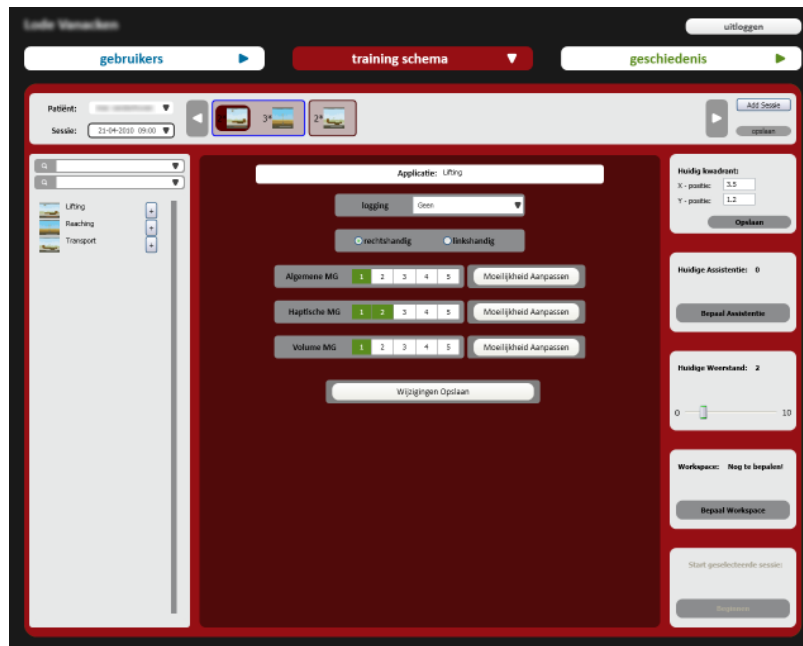
7. ACKNOWLEDGMENTS

Figure 4: The therapist interface. The interface is in Dutch as the target audience consists of native Dutch speakers.

8. REFERENCES

[1] G. Alankus, A. Lazar, M. May, and C. Kelleher. Towards customizable games for stroke rehabilitation. In *Proceedings of the 28th ACM Conference on Human Factors in Computing Systems (CHI 2010)*, pages 2113–2122, Atlanta, USA, April 10–15 2010.

[2] U. Dalgas, E. Stenager, and T. Ingemann-Hansen. Multiple sclerosis and physical exercise: recommendations for the application of resistance-, endurance- and combined training. *Multiple Sclerosis*, 14(16):35–53, 2008.

[3] J. De Boeck, G. Alders, D. Gijbels, T. De Weyer, C. Raymaekers, K. Coninx, and P. Feys. The learning effect of force feedback enabled robotic rehabilitation of the upper limbs in persons with MS - a pilot study. In *Proc. of ENACTIVE08*, pages 117 – 122, November 2008.

[4] P. Feys, G. Alders, D. Gijbels, J. De Boeck, T. De Weyer, K. Coninx, C. Raymaekers, V. Truyens, P. Groenen, K. Meijer, H. Savelberg, and B. Op 't Eijnde. Arm training in multiple sclerosis using phantom: clinical relevance of robotic outcome measures. In *Proceedings of IEEE 11th International Conference on Rehabilitation Robotics (ICORR 2009)*, pages 576–581, June 23–26 2009.

[5] G. Kwakkel, R. van Peppen, R. Wagenaar, S. Dauphinee, C. Richards, A. Ashburn, K. Miller, N. Lincoln, C. Partridge, I. Wellwood, and P. P. Langhorne. Effects of augmented exercise therapy time after stroke: a meta-analysis. *Stroke*, 35(11):2529–2539, 2004.

[6] P. Lammertse, E. Frederiksen, and B. Ruiter. The hapticmaster, a new high-performance haptic interface. In *Proceedings of Eurohaptics 2002*, pages 1–5, Edinburgh, UK, July 8–10 2002.

[7] L. Vanacken, S. Notelaers, C. Raymaekers, K. Coninx, W. van den Hoogen, W. IJsselsteijn, and P. Feys. Game-based collaborative training for arm rehabilitation of ms patients: A proof-of-concept game. In *Proceedings of GameDays 2009*, pages 65–75, March 25–26 2009.