OutRun: Exploring Seamful Design in the Development of an Augmented Reality Art Project

Garnet Hertz
Center for Computer Games and Virtual Worlds
Institute for Software Research
University of California, Irvine

Jong Weon Lee
Mixed Reality & Integration Laboratory
Sejong University

Chris Guevara
Center for Computer Games and Virtual Worlds
Institute for Software Research
University of California, Irvine

ABSTRACT

This paper outlines the development process of an augmented reality video game prototype that combines a classic arcade driving game with a real world vehicle. In this project the user, or driver, maneuvers the car-shaped arcade cabinet through actual physical space using a screen as a navigational guide which renders the real world in the style of an 8-bit video game. This case study is presented as a seamful augmented reality (AR) system: a project that exploits inevitable technical limitations of AR. We propose that the concept of seamfulness has important design implications for both AR and electronic media art projects and illustrate this through a description of the OutRun system development process.

KEYWORDS: Seamfulness, seamful design, game design, game innovation, mixed reality, augmented reality, alternate reality, pervasive gaming, electronic art, experimental interfaces, prototyping.

INDEX TERMS: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems – *artificial, augmented and virtual realities*; H.5.m [Simulation and Modeling]: Types of Simulation – *gaming*; J.5 [Arts and Humanities]: Fine Arts, Performing Arts; H.5.m [Information Interfaces and Presentation]: Miscellaneous.

1 INTRODUCTION

The OutRun project is a game and media art project that explores the overlap between the physical world and game environments. OutRun explores the act of driving a vehicle and the interstitial space between everyday life (driving an automobile) and a simulation of it (playing a driving video game) by combining the real world and OutRun, an 8-bit arcade driving game released by Sega in 1986. This project features two main components:

1. Cabinet-Car: A car-shaped sit-down arcade cabinet from Sega's OutRun is converted into a small car that can actually drive. This is done by modifying an existing fiberglass and wood cabinet to include motors, wheels and drivetrain components from an electric golf cart. This customized cabinet-car uses the existing videogame controls (steering wheel, acceleration pedal, brake pedal) to control the vehicle to a maximum speed of 20 kilometers (13 miles) per hour. See Figure 1 for a mockup diagram of the completed cabinet-car.

2. Custom Augmented Reality Software: The screen, which is positioned in front of the driver, renders the real world in the style of the 1986 video game OutRun by Sega. This is done through custom-built software to display a street-level view rendered in the style of the vintage video game. In other words, the driver only sees an 8-bit-style game rendered in their "windshield," which appears as if they are playing the 1986 videogame. Accelerating or turning the car-cabinet in the real world will proportionally change the display. Although the screen will mimic the real world around it, it is expected that the augmented display and the real world will not match perfectly.

Figure 1. A mockup of the proposed OutRun video game concept car functioning outdoors.

This project is motivated by the following concepts:

1. The De-Simulation of Driving - This project de-simulates the driving component of a videogame. Driving game simulations strive to be increasingly realistic, but this realism is usually focused on graphical representations. Instead, this system pursues "real" driving through a videogame as its primary goal.

2. GPS Navigation & Mixed Reality Seamfulness - Driving in a real automobile with a GPS navigation system can be game-like. This project explores the consequences of only using a computer model of the world as a navigation tool for driving. The windshield of this project's vehicle only shows a re-rendered simulation of the immediate environment, and as a result, driving it in the real world will likely be difficult or dangerous. As a result, this project explores and investigates how augmented reality and GPS data differs from the physical world, and what happens when an augmentation of reality envelops and obfuscates reality.

ghertz@uci.edu, jwlee@sejong.ac.kr, cguevara@uci.edu

IEEE International Symposium on Mixed and Augmented Reality 2010
Arts, Media, & Humanities Proceedings
13 -16 October, Seoul, Korea
978-1-4244-9342-5/10/$26.00 ©2010 IEEE

A system that intentionally plays with parallax between the physical environment and mixed reality can be described as *seamfulness*, a concept initially proposed by Mark Weiser in 1994 [27] and expanded since 2002 by Matthew Chalmers and other researchers at the University of Glasgow [21] [8] [7] [2] [3] [6]. Seamful design can be defined as an approach to reveal and exploit inevitable technical limitations in ubiquitous computing systems rather than hiding them. Seamful design starts with the assumption that infrastructures and users in the real world have limited technical resources: patchy network coverage, fluctuating signal strength for wireless networks and inaccuracies in positioning systems are the norm, not the exception. For example, seamful design argues that these seams and gaps should be exploited and used in mobile game design for more fault-tolerant usability and more engaging gameplay.

2 Related Work

Augmented Reality games and art projects “move beyond the traditional computer interfaces and into the physical world to occupy time and place on a human scale.” [14]. As Falk and Davenport describe, these games blur lines of demarcation between games and life: the real world is overlapped with game worlds, and game artifacts are combined with real world objects.

Figure 2. *Seek n' Spell*, where players collect virtual letters by moving through physical space (left) and spell words for points (right).

AR players typically explore a physical game space that is augmented by virtual information, usually as two- or three-dimensional objects overlaid on a map of a physical environment. To augment the virtual information on the maps, players’ positions are tracked by Global Positioning System (GPS) software and information is usually displayed on handheld computers, mobile devices or head-mounted displays. Examples of AR games that use two dimensional location data include Blast Theory's *Can You See Me Now?* [4] and *Bystander* [12], MIT Teacher Education Program's *Environmental Detectives* [13], Radford University's *Alien Contact* [11], ITP's *PacManhattan* [5] and Retronyms' *Seek ‘n Spell* [25]. In addition to two dimensional tracking of location, player orientations can be tracked by computer vision, digital compasses or inertial sensors to create more complicated AR environments. Examples include Bruce Thomas' *ARQuake* [25], National University of Singapore's *Human Pacman* [9], IPerG's *Epidemic Menace* [19], and Acrossai's *Virus Killer 360* [16]. AR systems are widespread, and have also been developed extensively by contemporary artists [10].

2.1 Seamfulness in AR Games

Two game examples – Retronyms' *Seek n' Spell* and Blast Theory's *Can You See Me Now?* – are useful in analyzing the concept of seamfulness in AR game systems.

Seek ‘n Spell is a commercial mobile application developed by Retronyms in 2009 [25]. Virtual letters are located in outdoor space to create a competitive AR word game in the style of Scrabble for a group of people. An augmented map is used to see the location of letters, and the goal is to “pick up” letters before other players by moving quickly through physical space (Figure 2, left). Once collected, the letters can be arranged into words and submitted for a score: each tile can only be used once, and longer words are worth more points (Figure 2, right). The player with the best score when time runs out is the winner.

Seek 'n Spell works best in spaces with a good cellular signal, a good view of the sky, and its developers suggest turning off Wi-Fi to prevent potential interruption and interference with the cellular network. Like many AR games, *Seek 'n Spell* requires a reliable communications infrastructure for engaging gameplay.

On the other hand, *Can You See Me Now?* (CYSMN) is an augmented game that is designed to have the potential for more engaging gameplay when technical infrastructures are degraded. CYSMN is an AR art project built by Blast Theory and the Mixed Reality Laboratory at University of Nottingham that explores the relationships between online players and players in a physical street [15]. The project is an augmented game of catch, where online players navigate through a virtual model of a city to chase people in the real world. These 'runners' – or street players – have to traverse actual city streets in order to escape being caught by online players [4]. Physical street runners carry GPS systems, walkie-talkies, and handheld computers with maps of their immediate environment that include avatars of online players (Figure 3).

Figure 3. A “runner” is pursued by online players while using *Can You See Me Now*, an AR game by Blast Theory and the University of Nottingham's Mixed Reality Laboratory.

CYSMN's gameplay is improved as subtle differences between the physical world and the virtual map of the world are explored: for example, physical runners became familiar with real-world locations that have poor GPS coverage that made it difficult for online users to find them. In other words, technical inaccuracies –

or points at which the technical infrastructure is not seamless – are points of leverage where gameplay is made more interesting. This can be thought of as seamful design. The gameplay is changed instead of suspended, creating an emotionally engaging cultural space that assumes that uncertainty is constant when dealing with mixed reality systems.

In a similar manner the OutRun project has engaged with the concept of seamfulness in its development process and has found that parallaxes between the physical environment and data have provided the most emotionally engaging moments of gameplay. In agreement with Chalmers et al., we propose that seamfulness has especially relevant implications for design, especially to AR and AR art projects. The technical complexity of AR and the budgetary limitations of most media art projects demand a plan for graceful degradation. In addition, seam-minded design provides a path for user metagaming: to play with the game itself, enabling the user to transcend a prescribed rule set of the game, to use external factors to affect the game, or to go beyond the perceived limits or environment set by the game.

3 PROJECT DESIGN AND DEVELOPMENT PROCESS

The OutRun system began as a project proposal by Garnet Hertz in January 2009. Active development started in June 2009 after five undergraduate students at UC Irvine were recruited as interns from the course “Computer Games as Art, Culture and Technology” [20]. The building of the system has two key thrusts: the mechanical construction of a drivable cabinet-car and the design of AR software that renders the real world in the style of the 8-bit driving video game.

3.1 Cabinet-Car Design and Development

The key physical prop in the development of this system is the building of a “video game concept car”, which is the conglomeration of two components: a 350 kg (770 lb) “Deluxe” Sega OutRun system and a four-wheeled 2007 Ez-Go RXV electric golf cart (Figure 4). This car is currently scheduled for release in October 2010.

The humor of building a sidewalk-cruising video game is a strong emotional link into the project. It borrows from a diverse range of sources: automobile showcar culture like Ed “Big Daddy” Roth, extreme computer case modification communities, live action role playing gamers (LARP) [26], and contemporary racing videogame simulators with highly customized computer hardware peripherals, accessories and props to augment game play.

The OutRun project extends the pursuit of immersive simulation by reconnecting the reproduction with its original source: actually driving. This intentional inversion of simulation and reality is a common artistic tactic: notable examples in this field include Tekken Torture (2001) by C-Level, Tazer Tag (2005) by Randy Sarafan, and PainStation (2001–2003) and LegShocker (2002) by Tilman Reiff and Volker Morawe [18]. These systems don't intend to solve a particular problem, but instead are focused on building game experiences that are emotionally or conceptually engaging.

Art-based projects that place an emphasis on emotional engagement often use narratives, costumes, props, and cultural symbols in their work. For example, Pac-Manhattan (2004) is a large-scale urban game that utilized the New York City grid to recreate the 1980's video game Pac-Man. A player dressed as Pac-Man ran around an area of Manhattan while attempting to collect all of the virtual "dots" that ran the length of the streets; four players dressed as the ghosts Inky, Blinky, Pinky and Clyde attempted to catch Pac-Man before all of the dots were collected [5].

Like Seinfeld's George Costanza pushing a *Frogger* arcade machine across a busy street in the real world [17] or people running through New York dressed as characters from Pacman, our project can be partially understood without any particular on-screen content. As a result of its sculptural and performative nature, the video game concept car provides us with more flexibility in building software that is displayed on the system's screen.

3.2 Software Design and Development

Software development of the project has focused on drawing the world like an 8-bit driving videogame (Figure 5): if a road actually appears in front of you that curves to the left, the system would draw an 8-bit style road on the screen that curved to the left.

Development of this software has taken two different paths: 1. Using location-aware technologies like global positioning systems (GPS) to place and move the user in a virtual world, and 2. Using computer vision to detect landscape and objects in the real world. During development, we used the concept of seamfulness to determine which approach was best for our project: one method produced significantly more interesting results when being tested in less-than-ideal technical situations or contexts we had not originally planned for.

Figure 4. Physical components of the OutRun concept car: an electric golf cart and a Sega arcade game system from 1986.

Figure 5. A screenshot from the first level of the original 1986 video game OutRun by Sega, which we used as the basis for our AR game world.

3.2.1 OutRunGPS: Locative Development Path

One development path involves location-aware technologies like global positioning systems (GPS) to locate the user in an 8-bit style virtual world. This approach took data from the physical world – like street data, land shapes and building placement – and built a pre-rendered three dimensional virtual world that could be navigated using real time GPS sensors.

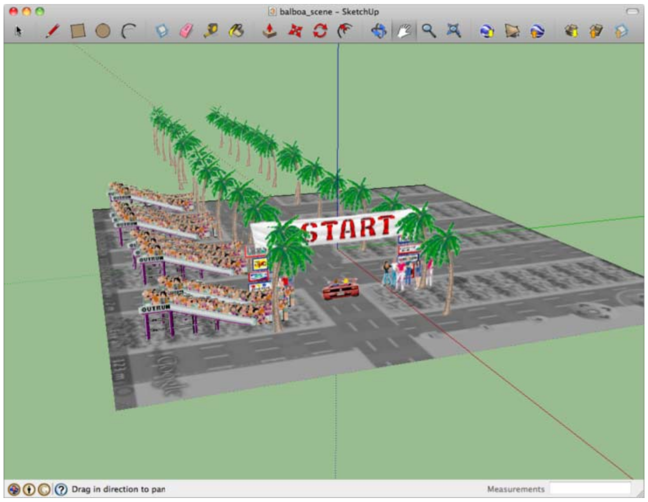

Figure 6. Two dimensional media assets being assembled using Google SketchUp, recreating the start sequence of the original OutRun game on Park Avenue, Balboa Island.

Figure 7. Real-world testing of OutRunGPS software on Balboa Island: due to limited GPS location resolution, users appeared as if they were driving through rows of trees.

The three dimensional virtual world was built as an overlay in Google Earth, and was constructed in Keyhole Markup Language (KML), an XML-based language schema for expressing geographic annotation and visualizations. To simplify development and testing of this approach, we focused on small geographic areas: Balboa Island and the UC Irvine Campus in Orange County, California. Ground maps for this 8-bit-styled environment were built non-programmatically from street and earth data, and objects like trees, buildings, automobiles and people were sampled from the original OutRun game. The world was assembled using Google SketchUp, a 3D modeling program that facilitates the placement of models in Google Earth (Figure 6). After assembling scenes on Balboa Island and the UC Irvine Campus the locations were physically navigated with a Q1UP-XP Samsung UMPC Tablet and a Nokia LD-1W Wireless GPS Module.

Real-world testing of this OutRunGPS system showed that the GPS module had a resolution of approximately ten to fifteen meters and a location refresh rate of about three seconds – which is similar in performance to most GPS-enabled mobile phones. These factors made real-time interaction with the virtual world not as immersive as we would have liked: in Balboa Island tests, the lack of GPS resolution positioned us as driving through a row of trees (Figure 7) and the position refresh rate produced a lurching/hopping sensation as you moved down the street, at least when viewed from the perspective of the original video game.

Although a higher resolution GPS receiver (like a Navcom SF-2050G with an accuracy of 10 cm) could have been implemented to remedy some of these problems, we felt that a seamless correlation between the physical environment and the game world was detrimental for the game experience. Driving a real-world car at 40 kilometers (25 miles) per hour through the 8-bit virtual world, for example, seemed painfully slow in comparison to the 200+ km/h (120+ mph) average speed of the original game. Seamless proportionality highlighted how slow and mundane everyday real-world driving is in comparison to a driving game. In our OutRunGPS implementation the strictly locative system had poor playability, uninteresting seams, and little leverage for metagaming.

3.2.2 OutRunCV: Computer Vision Development Path

An alternate development path for the OutRun AR software began with the development of a realtime computer vision (CV) system using Max/MSP/Jitter 5, an interactive graphical programming environment for music, audio, and media [28]. We used cv.jit.lines developed by Jean-Marc Pelletier to extend the vision and image processing capabilities of Max/MSP to include the ability to detect lines within a live video stream from a camera positioned at the front of the vehicle. Given a grayscale image or video stream, cv.jit.lines detects straight lines by detecting edge pixels using the Canny algorithm and then using a Hough transform to identify straight lines [23].

Since the main activity of our game was driving, we felt that the most important aspect of our computer vision system was to find the shapes of real-world roads. Our tests involved taking forward-looking single camera video footage through a number of different environments that the cabinet-cart might be able to drive, including roads, sidewalks, and paths. This footage was then processed through cv.jit.lines then through a number of custom vision filtering algorithms to try to programmatically determine road shape.

In our opinion, the most important road characteristic was its direction: whether the road was turning to the left, going straight or turning to the right. This decision was influenced by the structure of the original game, which does not feature intersections, stop signs, or parking lots. The world of OutRun is a high speed racetrack that cuts through the landscape, and we therefore felt that our priority was to mimic this.

As of May 2010, we determine road direction by taking the array of endpoints generated by cv.jit.lines and evaluating them to calculate the road's vanishing point with a custom-built software module called VPDetect. This algorithm was inspired by an intuitive method of looking at angled straight edges to find the vanishing point. These straight edges come from lines that are parallel in the real world but converge when converted to a 2 dimensional image. These lines are extended until they point at a single location: the vanishing point. The road is then assumed to curve from the driver's location towards the vanishing point.

Figure 8. VPDetect processing output from cv.jit.lines to determine the road vanishing point, indicated by the white dot in the center of the image.

The object first removes all lines that are perfectly horizontal/vertical, then creates a list of all points of intersection between all pairs of lines. The VPDetect object calculates the general intersection of these lines through the median or mean of the x and y coordinates and is assumed to be the location of the vanishing point (Figure 8).

Our VPDetect algorithm works well in most conditions, but still falters when lines that do not indicate the correct vanishing point are present in a scene. A large amount of irrelevant lines influences the average/median of the coordinates and can result in inaccurate road vanishing point predictions. Shadows, parked cars, crosswalks and other objects in the environment can increase VPDetect's inaccuracy.

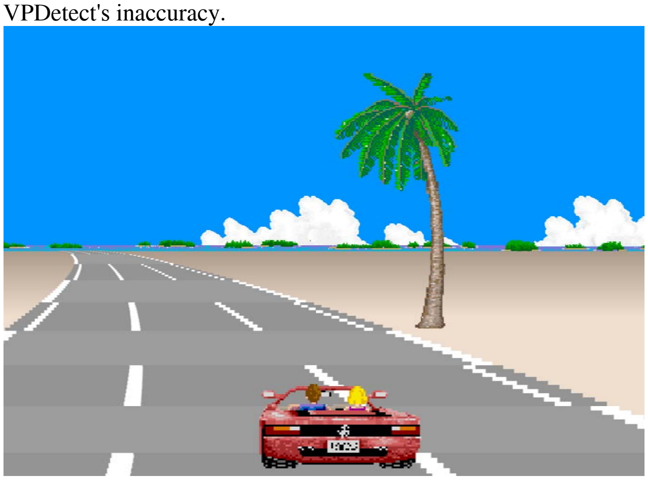

Figure 9. Custom Flash-based game developed in the style of OutRun, with a road endpoint controllable by computer vision data calculated by VPDetect.

Despite some inaccuracies, the calculated endpoint produces interesting results when rendered in the style of an 8-bit driving video game. We constructed a simple Adobe Flash-based driving game that emulated the style of the original Sega OutRun game (Figure 5) that was controllable by our Max/MSP/Jitter vision software (Figure 9). As the VPDetect-calculated vanishing point moved from right to left, our OutRun Flash road endpoint moved from right to left.

Figure 10. The OutRun computer vision system being tested on a Samsung UMPC Tablet in an indoor hallway at UC Irvine.

An unexpected benefit of using computer vision is that the system is enjoyable to play with in several different contexts: driving a bicycle down a path, walking down a sidewalk or hallway, gazing up the trunk of a tree, or using it through a stationary webcam and attempting to look “road-like”. Although it could be considered to be less accurate than a GPS system, it robustly functions without wireless access, satellite tracking or cellular coverage. The inaccuracies of determining vanishing points are made up for by the system's flexibility: when OutRunCV is put on a tablet PC with a camera, it becomes a pervasive game where the user explores their environment for “roadness”.

4 CONCLUSION: SEAMFULNESS AND ARTWORKS

The OutRun project is designed to fail as a device that will seamlessly permeate the boundary of real life and a game. In other words, it is intentionally built as a seamful system with faulty logic at its core: attempting to map a 1986 video game into the real world will never likely result in a seamless transition between game and life. However, this is a key artistic asset of the OutRun project. Like several art-based game projects, OutRun does not propose a solution to a problem: it intentionally magnifies and embraces an impossible gap between simulation and reality. Similar to projects like Blast Theory's *Can You See Me Now?* [4] it does not see a *line* between game and real life, but rather a pleasurably impossible *area*: an uncanny valley between the familiar and a facsimile of it [22].

In our development process, we have favored a computer vision approach to augmenting reality: it offers more flexibility in exploring, extending and playing in the gaps between virtual and physical spaces, and its technical limitations and inaccuracies generally produce emotionally engaging results.

Although our OutRunCV system and Flash renderer are in some ways more of a simulated reality than an augmented reality, we agree with Chalmers et al. that seamfulness has especially relevant implications for mobile locative game design. Because of the relative complexity of most AR software and hardware systems, we see the designing of gracefully degrading and fault tolerant systems as an important task. In our case, seamfulness has not only given us a model to think about the robustness of a system during technical malfunction, but has improved player experience by enabling the user to go beyond the perceived limits set by the game and into the beautiful seams of everyday life.

ACKNOWLEDGMENTS

The authors would like to thank Walt Scacchi, Jeremy Bailey, Gillian Hayes, Christina Lopes, Matt Wong, Erik Olson, David Dinh, Matt Shigekawa, Jessee Joseph, Mike Tang and Alex Szeto for their development efforts and assistance.

The OutRun project is directed by Garnet Hertz and has been supported by the following organizations at the University of California Irvine: the Laboratory for Ubiquitous Computing and Interaction, the Arts Computation Engineering Program, the Center for Computer Games and Virtual Worlds, the Institute for Software Research, and the California Institute for Telecommunications and Information Technology. Support for this research is also provided through grants from the National Science Foundation #0808783; Digital Industry Promotion Agency, Global Research and Development Collaboration Center, Daegu, South Korea. No endorsement implied.

REFERENCES

[1] http://www.acrossair.com/apps_viruskiller360.htm

[2] Barkhuus, L., Chalmers, M., Hall, M., Tennent, P., Bell, M., Sherwood, S., Brown, B. 2005. Picking Pockets on the Lawn: The Development of Tactics and Strategies in a Mobile Game. Proc. Ubiquitous Computing, Tokyo, LNCS 3660, pp. 358-374.

[3] Bell, M., Chalmers, M., Barkhuus, L., Hall, M., Sherwood, S., Tennent, P., Brown, B., Rowland, D., Benford, S., Capra, M., Hampshire, A. 2006. Interweaving mobile games with everyday life. Proceedings of the SIGCHI conference on Human Factors in computing systems.

[4] Benford, S., Crabtree, A., Flintham, M., Drozd, A., Rob Anastasi, R., Paxton, M., Tandavanitj, N., Adams, M., Row-Farr, J. 2006. Can you see me now?, ACM Transactions on Computer-Human Interaction (TOCHI), v.13 n.1, p.100-133.

[5] Bloomberg, A., Boelhauf, K., Crowley, D., Hall, C., Lee, W., Molefe, M., Olson, M., Phalines, M., Romeo, M., Stephensen, O., Thienthong, P., and Vigeant, P. 2004. Pac-Manhattan. http://pacmanhattan.com.

[6] Broll, G. and Benford, S. 2005. Seamful Design for location-based mobile games. Lecture Notes in Computer Science 3711, 155-166.

[7] Chalmers, M., and Galani, A. 2004. Seamful Interweaving: Heterogeneity in the Theory and Design of Interactive Systems. Proc. ACM DIS 2004, pp. 243-252.

[8] Chalmers, M., MacColl, I., and Bell, M. 2003. Seamful Design: Showing the Seams in Wearable Computing. Proc. IEE Eurowearable 2003, Birmingham, pp. 11-17.

[9] Cheok, A.D., Goh, K.H., Liu, W., Farbiz, F., Fong, S.W., Teo, S.L., Li, Y., and Yang, X. 2004. Human Pacman: a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing. *Personal and Ubiquitous Computing*, 8:71-81.

[10] Debatty, R. http://we-make-money-not-art.com/archives/augmented_reality/.

[11] Dunleavy, M., Dede C., and Mitchell, R. 2009. Affordances and Limitations of Immersive Participatory Augmented Reality Simulations for Teaching and Learning. In *J Sci Educ Technol*, 18:7-22.

[12] http://education.mit.edu/drupl/ar

[13] http://education.mit.edu/ar/ed.html

[14] Falk, J. and Davenport, G. 2004. "Live Role-Playing Games: Implications for Pervasive Gaming". Entertainment Computing – ICEC 2004. Lecture Notes in Computer Science 3166. Springer Berlin / Heidelberg. 127.

[15] Flintham, M., Anastasi, R., Benford, S., Hemmings, T., Crabtree, A., Greenhalgh, C., Rodden, T., Tandavanitj, N., Adams, M., and Row-Farr. J. 2003. Where On-Line Meets On-The-Streets: Experiences With Mobile Mixed Reality Games. *CHI 2003*, 5:569-576.

[16] Herbst, I., Ghellal, S., and Braun, A.K. 2007. TimeWarp: An Explorative Outdoor Mixed Reality Game. *SIGGRAPH 2007*, SIGGRAPH Research Posters, August 2007.

[17] http://www.imdb.com/title/tt0697701/.

[18] Laso, P. W. 2007. Games of Pain: Pain as Haptic Stimulation in Computer-Game—Based Media Art. Leonardo 40:3, 238-242.

[19] Lindt, I., Ohlenburg, J., Pankoke-Babatz, U., Prinz, W. and Ghellal, S. 2006. Computing Multiple Gaming Interfaces in Epidemic Menace. *CHI 2006*, pages 213-218.

[20] Losh, E., 2009. Hybridizing Learning, Performing Interdisciplinarity: Teaching Digitally in a Posthuman Age. In Proceedings of Digital Arts and Culture, UC Irvine. http://www.escholarship.org/uc/item/8vq3m5qc.

[21] MacColl, I., Chalmers, M., Rogers, Y., and Smith, H. 2002. Seamful ubiquity: Beyond seamless integration. Proc. Ubicomp 2002 Workshop on Models and Concepts for Ubiquitous Computing, Gothenburg, September 2002.

[22] Mori, M. 1970. Bukimi no tani / The Uncanny Valley (K. F. MacDorman & T. Minato, Trans.). Energy 7:4, 33–35.

[23] Pelletier, J. (n.d.). Cv.jit. Retrieved from http://www.iamas.ac.jp/~jovan02/cv/cv.jit_win32_v1.6b2.zip

[24] http://www.seeknspell.com

[25] Thomas, B., Close, B., Donoghue, J., Squires, J., De Bondi, P., and Piekarski, W. 2002. First Person Indoor/Outdoor Augmented Reality Application: ARQuake. *Personal and Ubiquitous Computing*, 6:75-86.

[26] Tychsen, A., Hithchens, M., Brolund, T. and Kavakli, M. 2006. Live Action Role-Playing Games: Control, Communication, Storytelling, and MMORPG Similarities. Games and Culture 2006 1: 252-275.

[27] Weiser, M. 1994. Building Invisible Interfaces. Keynote talk, ACM UIST94.]

[28] Zicarelli, D. 2010. Cycling 74, http://cycling74.com/.