# OutRun: Perversive Games and Designing the De-Simulation of Eight-Bit Driving


Garnet Hertz
Center for Computer Games and Virtual Worlds
Institute for Software Research
University of California, Irvine
209 Information and Computer Science 2
Irvine, CA 92697, USA
ghertz@uci.edu (http://conceptlab.com/)



## ABSTRACT

This paper outlines the development process of a mixed reality video game prototype that combines a classic arcade driving game with a real world vehicle. In this project the user, or player, 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 "perversive game": an attempt to disrupt the everyday by highlighting and inverting conventional behavior through humor and paradox.



## Categories and Subject Descriptors

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.

## General Terms

Design, Experimentation, Human Factors, Theory.

## Keywords

Game design, game innovation, mixed reality, augmented reality, alternate reality, pervasive gaming, live action role playing, electronic art, experimental interfaces, prototyping.


## 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 eight-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 with motors, wheels and drivetrain components from an electric golf cart. The original arcade cabinet is modeled after a 1984 Ferrari Testarossa. This customized cabinet-car will use the existing videogame controls (steering wheel, acceleration pedal, brake) to control the vehicle. It is expected that the maximum speed of the car will be no more than 20 miles (32 kilometers) 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. This is done through custom-built computer vision software and GPS sensors that calculate the location of the experimental vehicle and display a street-level view rendered in the style of the vintage video game. In other words, the driver only sees an eight-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. The proposed OutRun cabinet-car.**

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 Parallax** - Driving in a real automobile with a GPS navigation system can be game-like. This project explores the consequences of only using only 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 is often 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.

## 2. CONTEXTS IN GAME CULTURE

The OutRun project shares similarities to several existing genres of game design, namely pervasive games (including live-action role-playing games, alternate reality games, and big games) and driving simulators.

Pervasive games “move beyond the traditional computer interfaces and into the physical world to occupy time and place on a human scale.”[Falk, Jennica; Davenport, Glorianna (2004). "Live Role-Playing Games: Implications for Pervasive Gaming" (PDF). Entertainment Computing – ICEC 2004. Lecture Notes in Computer Science. 3166. Springer Berlin / Heidelberg. page 127.] As Falk and Davenport describe, pervasive games blur lines of demarcation between games and life: players are mixed with game characters, the real world is overlapped with game worlds, and game artifacts are combined with real world objects. Contemporary pervasive games consist of at least three subgenres: live action role-playing games, alternate reality games and big games.

Live action role-playing (LARP) games are pervasive role-playing games where individuals physically act out their characters' actions, typically in a fictional setting that is represented by the real world. Players embody their character roles through costumes and physical actions, maintaining a clear sense of being in-character or out-of-character [Tychsen et al., Live Action Role Playing Games, page 255]. Alternate Reality games, on the other hand, are pervasive games that attempt to delete the line between being in-character and out-of-character: “they do everything in their power to erase game boundaries – physical, temporal and social – and to obscure the metacommunications that might otherwise announce, 'This is play.'”[Jane McGonigal, A Real Little Game: The Performance of Belief in Pervasive Play]. Instead of fictional character play, alternate reality games have an aesthetic of trying to not look like either a game or a fantasy. Inspiration for alternate reality games is primarily driven by an extension of gaming environments through mobile network technologies and ubiquitous computing. [Jane McGonigal, A Real Little Game: The Performance of Belief in Pervasive Play].

Within the category of pervasive games, the OutRun project perhaps has the most similarity with some styles of “Big Games,” especially the Pac-Manhattan project led by Frank Lantz at NYU's Interactive Telecommunications graduate program in 2004 (Figure 2). Pac-Manhattan 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 runs 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 [Pac-Manhattan, http://pacmanhattan.com/about.php, 2004].


**Figure 2. Pac-Manhattan, a “big” game developed in 2004 where individuals play a Pac-Man inspired physical game through the streets of Manhattan.**

Like the OutRun project, Pac-Manhattan takes an existing eight-bit game and explores what happens when it is mapped into the larger "real world" of buildings, streets, and cities. In comparison to LARP or Alternate Reality games, Big Games like Pac-Manhattan have a simple premise: take an existing video game and explore what happens when it is scaled up into the real world. OutRun does this, but also does the inverse: it attempts to take the real world and translate it into a “small game” on a computer screen.

**Figure 3. The F1Showcar, a full size motion simulator computer game car system.**

The OutRun project also borrows from non-pervasive game systems in its concept. The contemporary racing simulator video game community, for example, has developed a substantial selection of customized computer furniture and hardware peripherals to augment its genre of game play. Basic systems include foot pedal and steering wheel peripherals, while more involved configurations include full scale car cockpits and hydraulic motion tables that simulate the physics of real-world driving (Figure 3) [F1 ShowCar, http://f1showcar.com/ ].

The OutRun project extends this pursuit of immersive driving simulation by reconnecting the reproduction with its original source: actually driving. In the process, OutRun borrows from pervasive gaming by using the real world as its game space.

The end result explores the relatively new phenomenon of driving an automobile and simultaneously looking at a computer screen. Contemporary driving increasingly involves computing: using a GPS navigation system, in-dash entertainment, or text messaging while driving are progressively part of real-world driving culture. Although automobile GPS navigation is not considered a game, malfunctioning wayfinding algorithms and inaccuracies in street-level data can produce a humorous, frustrating or dangerous "pervasive game." These overlaps have serious consequences and deserve to be explored: in 2008, the British newspaper The Mirror estimated that automobile GPS navigation systems have caused 300,000 accidents in the UK [The Mirror, SatNav danger revealed: Navigation device blamed for causing 300,000 crashes, Tanith Carey, 21/07/2008]. OutRun is not proposing a solution to precarious uses of computers in vehicles, but intentionally magnifying and embracing the problem. In the process, the system strives to function in the historical role of a trickster: to disrupt the everyday by highlighting and inverting conventional behavior through humor and paradox. [Hertz, Un-Simulations, Tricksters and Radical Thought, Blackflash Magazine, 2009] OutRun extends simulation and perverts it at the same time: it is like a fantasy taken to its extreme and gone humorously wrong. Being a clown doesn't accomplish much, but trying to intelligently invert social conventions and assumptions can constructively and artistically rewire our understandings of who we are.

# 3. PROJECT DESIGN AND DEVELOPMENT PROCESS

The OutRun system began as a project proposal 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" [Losh, DAC09, 2009]. I oversaw development production, and the team was split up into the following general areas – Hardware: Matt Wong and Erik Olson; Game and Visualization Software: Chris Guevara; Media Assets: David Dinh; and Documentation: Richard Vu. The Laboratory for Ubiquitous Computing and Interaction at UC Irvine provided intern office space and software licenses, while the Arts Computation Engineering program at UCI provided large format studio space for the physical construction of the cabinet-car.

## 3.1 Software Design and Development Process


**Figure 4. A screenshot from the first level of the original 1986 video game OutRun.**

The original OutRun game featured a pixellated 1984 Ferrari Testarossa that the user controlled down winding roads through a number of different landscapes. For the purposes of this project, only the first level will be used as a source of inspiration: it is an idealized version of a California-esque beach community with palm trees, windsurfers and surf shops – plus a six lane freeway barreling through it all (Figure 4).

Software development of the project to date has focused on building a software system that is able to render the real world as it would be drawn in the original OutRun game. In other words, the software's key task was to draw the world like an eight-bit driving videogame: if a road actually appears in front of you that curves to the left, the system would draw an eight-bit style road on the screen that curved to the left.

Development of this software has taken two different paths: 1. Using computer vision to detect objects in the real world, and 2. Using location-aware technologies like global positioning systems (GPS) to position the user in a virtual world.

### *3.1.1 Computer Vision Development Path*

Development of the OutRun software began with the development of a realtime computer vision system using Max/MSP/Jitter 5, an interactive graphical programming environment for music, audio, and media [Cycling '74, Max/MSP/Jitter, http://cycling74.com/ ]. Jitter – an architecture optimized for use with video – was used to take video data and store it in a two dimensional array. Cv.jit, a collection of of Max/MSP/Jitter computer vision tools by Jean-Marc Pelletier, was used for its ability to detect line edges in video data. Along with pre-defined Max/MSP/Jitter objects and a line detection tool in cv.jit, we created our own computer vision objects in Java to tackle what we saw as our main problem: to detect the shape of roads.

**Figure 5. Driving video processed using Jean-Marc Pelletier's cv.jit.lines, a Max/MSP/Jitter tool for detecting line edges in video data.**

Cv.jit detected numerous lines in a scene (Figure 5) and to focus on roads, we developed a number of custom filtering algorithms to remove lines that we thought weren't road-like. Then, by calculating the convergence point of the road-like lines, we tried to determine the general direction of the road (Figure 6).

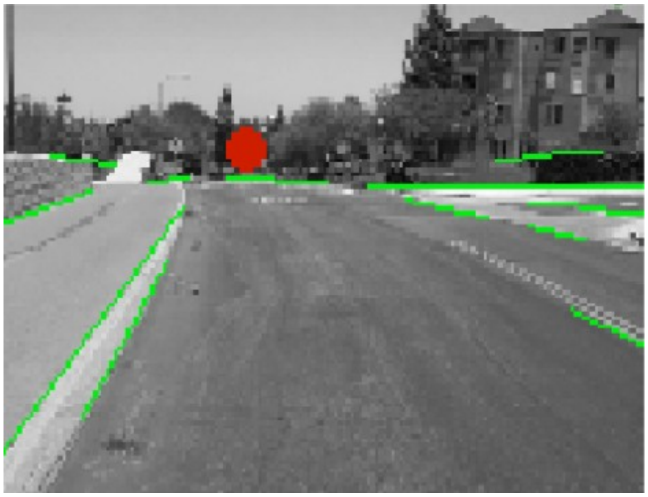

**Figure 6. Driving video processed with cv.jit.lines and custom filtering modules. The dot in the center of the image shows the calculated direction of the road, which is the average convergence point of filtered lines.**

This approach has been useful in programmatically determining the direction of real-world roads, but is far from perfect. The current system is only set to detect a single road endpoint, does not calculate road width, and gets confused by shadows, crosswalks and other variables.

Despite these problems, the calculated endpoint produces interesting results when rendered in an eight-bit driving video game style. To do this, we constructed a simple Adobe Flash-based driving game that was controllable by our Max/MSP/Jitter vision software. We used Antony Dzeryn's Retro Racer as a starting point for this development, a Flash-based open source driving game that was similar in style to the original OutRun game. By modifying the ActionScript and replacing media assets, we modified Retro Racer to look more like OutRun and enabled communication between Max/MSP/Jitter and Flash (Figure 7).

**Figure 7. Custom Flash-based game developed in the style of OutRun, with road endpoint controllable by computer vision data in Max/MSP/Jitter.**

### *3.1.2 Locative / GPS Development Path*

The second development path that is being pursued for this project is through location-aware technologies like global positioning systems (GPS) to position the user in an eight-bit style virtual world. This development path has been pursued with undergraduate students Matt Shigekawa, Jesse Joseph and Mike Tang starting in September 2009.

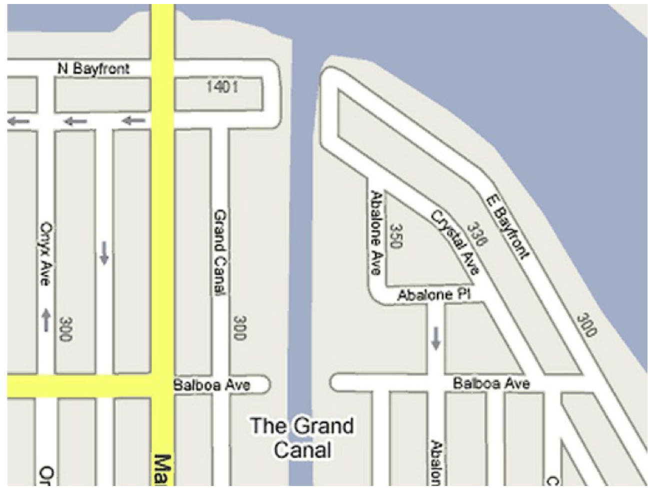


**Figure 8. A standard Google Map view of Balboa Island, California that shows street data and land outlines.**

**Figure 9. Street data and land outlines of Balboa Island drawn in an eight-bit videogame style.**

This approach takes data from the physical world – like street data, land shapes and building placement – and builds a pre-rendered three dimensional virtual world that can be navigated by using a real time GPS sensor that controls a handheld display.

The three dimensional virtual world was built to be displayed in Google Earth, and was constructed in Keyhole Markup Language (KML), an XML-based language schema for expressing geographic annotation and visualization on two-dimensional maps and three-dimensional Earth browsers [Google, KML – Google Code, http://code.google.com/apis/kml/]. 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 eight-bit-styled environment were built non-programmatically from street and earth data (Figures 8 and 9), and objects – like trees, buildings, automobiles and people – were taken 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 (Figures 10 and 11) [Google SketchUp, http://www.sketchup.com ].

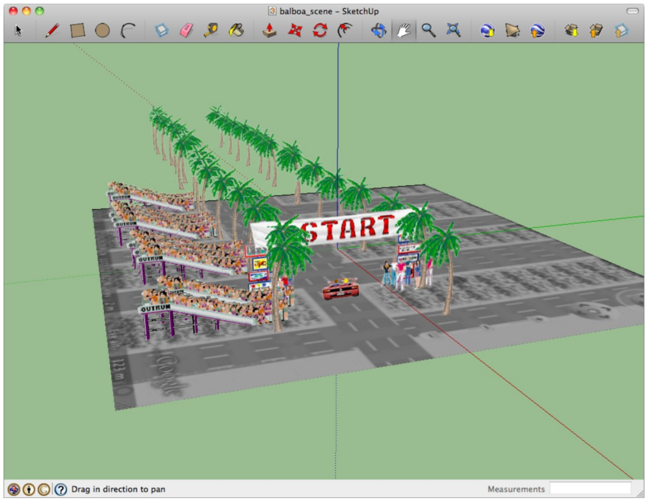


**Figure 10. Two dimensional media assets being assembled using Google SketchUp, recreating the start sequence of the original OutRun game on Park Avenue, Balboa Island.**

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, with position data fed from the GPS module to Google Earth by GooPs Pro, a Google Earth GPS Tracking and Navigation tool.


**Figure 11. View down Park Avenue of Balboa Island in Google Earth with placed OutRun-style media assets.**

Real-world testing of this locative/GPS system showed that the Nokia LD-1W Wireless GPS module had a resolution of approximately ten to fifteen meters and a location refresh rate of about three seconds. 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 and the position refresh rate produced a lurching/hopping sensation as you moved down the street, at least at the perspective of the original video game. A locative development path will likely require a high resolution GPS receiver (like a Navcom SF-2050G with an accuracy of 10cm) and tweening between location waypoints.

The sensation of speed was also considerably different from the original game: driving a real-world car at 40 kilometers per hour through the eight-bit virtual world, for example, seemed painfully slow in comparison to the 200+ km/h average speed of the original game car. In our mixed reality GPS prototype it felt as if you were crawling through the game world, highlighting that in everyday driving one normally doesn't race a car down residential streets. This gap between the expectations of the game world and action in the real world could have some interesting implications for gameplay: it could either encourage players to drive recklessly in the real world to recreate the original eight-bit game experience, or highlight how slow and mundane everyday real-world driving is in comparison to a driving game.

## 3.2 Cabinet-Car Design and Development Process

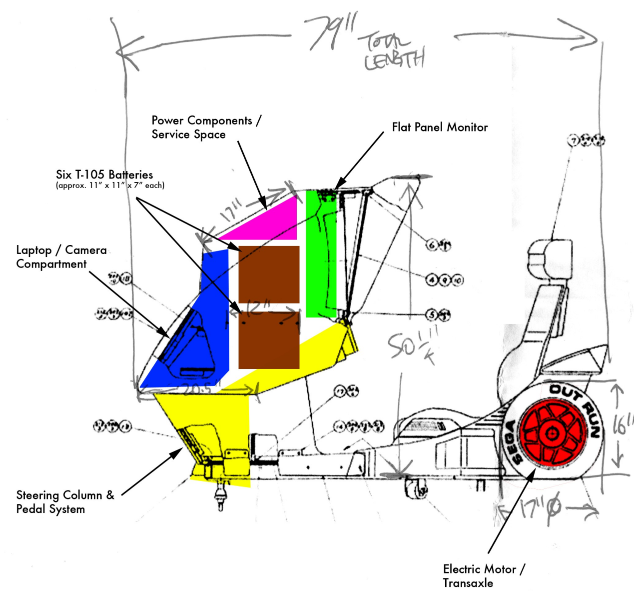


**Figure 12. Proposed Cabinet-Car Component Layout, using side profile of video game cabinet as a reference.**

The cabinet-car is the conglomeration of two components: a sit-down video game arcade cabinet and an electric golf cart (Figure 12). The arcade cabinet chosen was the "deluxe" OutRun configuration offered by Sega, weighing with 350 kg (770 lbs) with hydraulic side-to-side motion, a large fiberglass driving cockpit and a 25 inch CRT monitor. Our cabinet was purchased from a private party in Maine (USA) in July 2009 (Figure 13).


**Figure 13. Original OutRun "Deluxe" arcade game cabinet (background) and 1959 Turfrider Mark IV golf cart (foreground).**

In August 2009 a three-wheeled golf cart was purchased as a drivetrain for the system: a 1959 Turfrider Mark IV (Figure 13). The Mark IV golf cart was selected due to its three wheel configuration, unique "googie" space-age styling and similar form factor to the OutRun cabinet. However, after testing the system it lacked efficiency, power, stability and the tricycle steering mechanism would be difficult to interface with the arcade steering wheel. As a result, the classic golf cart was abandoned in favor of a four-wheeled 2007 Ez-Go RXV, which grants at the Institute for Software Research supported since Fall 2009. The Ez-Go RXV electric golf cart provides a more functional drivetrain, with greater reliability, power, battery duration and serviceability to make the make the system more viable as a robust exhibit/vehicle device.

## 4. CONCLUSION: PERVERSIVE GAMING

In 2003 Jane McGonigal humorously raised the idea that the combination of pervasive and immersive play results in "perversive" gaming: where "players are prone to falling for the games' dissimulative rhetoric [...] They wind up believing in their play too much for their own good." [Jane McGonigal, "A Real Little Game: The Performance of Belief in Pervasive Play." Digital Games Research Association (DiGRA) "Level Up" Conference Proceedings. November 2003.] Through the OutRun project, I'd like to return to McGonigal's concept of perversive gaming and extend it.

McGonigal sees that in pervasive and immersive game situations gamers maximize their play experience by *performing* the role of believing that they are able to permeate the game-reality boundary. In other words, players don't actually believe that real life and games are the same thing, but their enjoyment flows from pretending that the two worlds are able to be intermingled. McGonigal describes this gameplay as a combination of the "Pinnocchio effect" – the desire for a game to be transformed into real life – and conversely, the desire for everyday life to be transformed into a "real little game." McGonigal focuses on two examples in her paper: the 2001 immersive game known as the Beast, and the Go Game, an ongoing urban superhero game [ibid].

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 with faulty logic at its core: attempting to map a 1986 video game into the real world likely will never result in a seamless transition between game and life. When the game is extended beyond its normal constraints as a videogame, it results in a malfunctioning: confusion, collisions or accidents. However, this is part of the fun of the OutRun project.

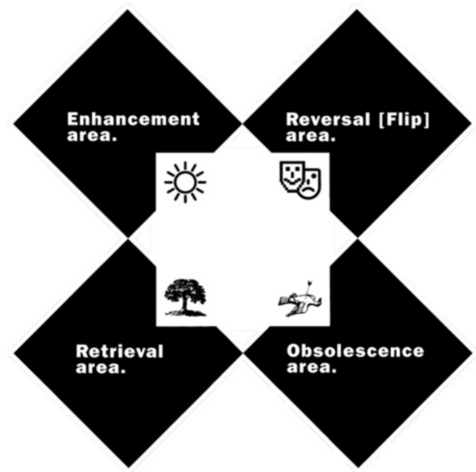


**Figure 14. McLuhan's Tetrad diagram, which proposes how communication and representation technologies have the potential to change in four distinct ways: obsolescence, retrieval, enhancement, and reversal.**

A useful framework for visualizing and rethinking this dynamic includes Marshall and Eric McLuhan's Laws of Media: The New Science (1989), which proposes that media technologies have the potential to change in four distinct ways [Marshall and Eric McLuhan, Laws of Media: The New Science, 1989]. A poetic four-region model is presented to envision the characteristics of obsolescence, retrieval, enhancement, and reversal (Figure 14). The reversal, in McLuhan's eyes, occurs when something is pushed to its limits.

The main point of relevance to this discussion on pervasive and immersive games is that when mediating technologies are pushed beyond their ordinary limits, they can reverse or flip in their intent or use. The McLuhan tetrad diagram gives a visual graph to think about how game projects may simultaneously amplify, invert, revive, and subsume – they swirl around and don't simply proceed in a straight line. When a simulation is taken beyond its role as a safe fantasy and pushed to envelop and take over reality, it becomes perverted. Problematizing a binary virtual/physical view of the world, for example, can be done by taking virtuality to an extreme. By taking a belief or simulation too far, it stops being a comforting diversion and flips into an absurdity, an obsession or a dark dream [Hertz, Un-Simulations, Tricksters and Radical Thought, Blackflash Magazine, 2009]. In the process, beliefs are "perverted" [Rafael Lozano-Hemmer, Perverting Technological Correctness, Leonardo 29:1, 1996].

OutRun does not propose a solution to a problem: it intentionally magnifies and embraces an impossible gap between simulation and reality. It does not see a *line* between game and real life, but rather a pleasurably impossible *area*: an uncanny valley between the familiar and the almost-familiar [Masahiro Mori, 1970]. Exploring this "perverted" and uncanny valley by over-extending simulations is a useful approach for game design. Intentional misuse of technology, incorporating physical pain, questioning cultural concepts of progress, creating intentionally useless devices and bending cultural stereotypes are all useful approaches to embrace in developing the uncanny [Rafael Lozano-Hemmer, Perverting Technological Correctness, Leonardo 29:1, 1996].

In this perversion, the game strives to function to break the "natural" rules to highlight and invert common knowledge. To build perversive games is to explore the historical role of a trickster: to disrupt the everyday by highlighting and inverting conventional behavior through humor and paradox. In the process, the contractual magic circle of engagement is extended spatially, temporally, and socially [Markus Montola, Games and Pervasive Games, Page 7].

## 5. ACKNOWLEDGMENTS

Thanks to the development efforts of Walt Scacchi, Professor Jong Weon Lee, Jeremy Bailey, Matt Wong, Erik Olson, Chris Guevara, David Dinh, Matt Shigekawa, Jessee Joseph, and Mike Tang.

The OutRun project has been supported by the Laboratory for Ubiquitous Computing and Interaction at UC Irvine, Gillian Hayes, Christina Lopes, the Arts Computation Engineering Program at UC Irvine, and is currently supported by the Center for Computer Games and Virtual Worlds and the Institute for Software Research at UC Irvine. Support for this research is 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.

## 6. REFERENCES

[1] Cycling '74, Max/MSP/Jitter, http://cycling74.com/.

[2] Google, KML – Google Code, http://code.google.com/apis/kml/.

[3] Google SketchUp, http://www.sketchup.com.

[4] Garnet Hertz, Un-Simulations, Tricksters and Radical Thought, Blackflash Magazine, 2009.

[5] Elizabeth Losh, Hybridizing Learning, Performing Interdisciplinarity: Teaching Digitally in a Posthuman Age, Digital Arts and Culture 2009.

[6] Jennica Falk and Glorianna Davenport (2004). "Live Role-Playing Games: Implications for Pervasive Gaming" (PDF). Entertainment Computing – ICEC 2004. Lecture Notes in Computer Science. 3166. Springer Berlin / Heidelberg. Page 127.

[7] F1 ShowCar, http://f1showcar.com/.

[8] Rafael Lozano-Hemmer, Perverting Technological Correctness, Leonardo 29:1, 1996.

[9] Markus Montola, Games and Pervasive Games, Page 7.

[10] Masahiro Mori, 1970.

[11] Jane McGonigal, "A Real Little Game: The Performance of Belief in Pervasive Play." Digital Games Research Association (DiGRA) "Level Up" Conference Proceedings. November 2003.

[12] Marshall and Eric McLuhan, Laws of Media: The New Science, 1989

[13] The Mirror (newspaper), SatNav danger revealed: Navigation device blamed for causing 300,000 crashes, Tanith Carey, 21/07/2008.

[14] Pac-Manhattan, http://pacmanhattan.com/about.php, 2004.

[15] Tychsen et al., Live Action Role Playing Games, page 255.