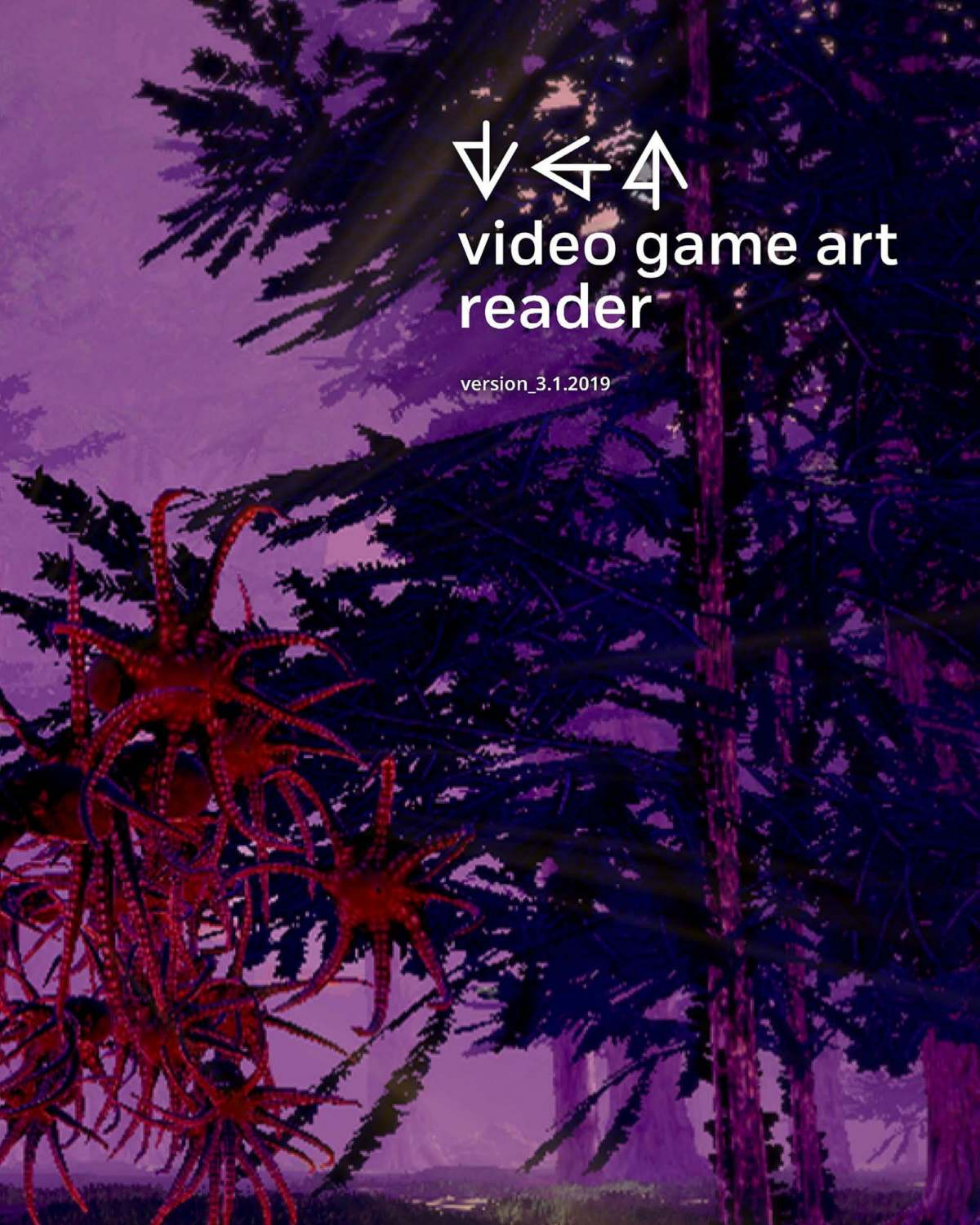


video game art reader

version_3.1.2019

video game art reader©

version_3.1.2019

Video Game Art Reader
version_3.1.2019
Volume 3

The VGA Reader (ISSN 2574-4712) is published by VGA Gallery, an Illinois 501(c)(3): 2418 W. Bloomingdale Ave, Chicago, IL 60647.

Advertising, subscriptions, and back issues
Send inquiries to Michael Reed, c/o VGA Gallery, mreed@vgagallery.org.

VGA Reader Mission The VGA Reader is a peer-reviewed journal for video game audiences and video game practitioners interested in the history, theory, and criticism of video games, explored through the lens of art history and visual culture. Its primary aim is to facilitate conversation and exploration of video game art, documenting and disseminating discourse about the far-reaching influence of video games on history, society, and culture.

Cover
DreamWalker, 2018-2019. Zack Ragozzino, Gabriella Santiago, Enrica Lovaglio Costello, Dr. Zöe J. Wood. Image courtesy of the artists.

Contents

The much-emulated and meme-ified *GamePro* rating system, explained. "*GamePro* Rating Guide," *GamePro Magazine*, c. 1989.

Letter From The Guest Editors

Christopher W. Totten Assistant Professor, Kent State University; founder of Pie for Breakfast Studios
Enrica Lovaglio Costello Associate Professor, California Polytechnic State University, San Luis Obispo, CA

Remember *GamePro* magazine? This American video game magazine was published monthly from 1989 to 2011, sharing news about game releases, previews, interviews, and most notably, reviews. *GamePro* reviews evaluated games based on general categories: Graphics, Sound, Control, Challenge, and something they called "Fun Factor." While those categories seem familiar enough, *GamePro*'s numeric rating scale featured a twist: it was accompanied by a cartoon character that progressed from sleeping (a score of 1) to the throes of "video game mania," with wide eyes and spiked hair (a score of 5). This character rating system sticks in the memory for two reasons: not only did the character express the intensity of emotions we were told video games were capable of producing, but also its depictions and generalizations of video game effects serve as the predominant model for game review media even today. However, in the intervening years between *GamePro*'s heyday and today's vast landscape of print and online review media, we started studying and evaluating games in another way: as works of art, cultural artifacts, and catalysts for change. Though it's tempting to envision the spikey-haired *GamePro* review bro acting out rating categories such as "agent of change" or "indicator of its time," these old evaluative metrics fail to reflect how the built environment is inextricably linked to video game design and the larger mediatized landscape of popular culture.

This is why, dear readers, we bring you this special edition on games and architecture. At games conferences, both academic and professional, there have been whispers about the connections between games and architectural design. There have been some efforts in this area, including by Bobby Schweizer and Robert Yang; there is also Michael Nitsche's book *Video Game Spaces* (2008), and von Borries, Walz, and Böttger's edited collection *Space, Time, Play* (2007). But these are scattered efforts; thus far this area has failed to take root as a fully realized branch of game studies. This is in contrast to the commercial games industry, where architectural principles and expertise are taking hold as a vital part of disciplines like level design, and the creation of games' interactive environments. Architects have even been joining game studios as level designers or to serve as consultants on projects that need architectural or historical context.

With these currents moving within game development and game studies, we felt that it was time to make a concerted effort to collect academic research that not only posited that games and architecture might be aligned, but that found ways to theorize these relationships. Inspired by our strong background in architecture and spatial design—one of us as a practitioner, the other as a theorist—we welcomed with great enthusiasm the idea of joining forces on this issue of the *VGA Reader*. Knowing the rarity of designers and researchers that combine these fields in innovative and thought-provoking ways, we made the call as broad as possible, still loyal to our mission to find critical investigations that reveal how game spaces evoke meaning, enhance game narratives, and explore unconventional themes.

The papers and projects that we present to you in this issue are interesting and provocative in their exploration of human behavior and decision making in a range of both indie and mainstream games. On one hand, there are projects that integrate architecture and architectural principles into the design of games. One such project is game designer and Northeastern University instructor Chris Barney's system for identifying game design "patterns" based on the work of architect Christopher Alexander.

Two of these essays—"*ANX Dread:* A Virtual Reality Experience to Explore Anxiety During Task Completion," and "*Dreamwalker:* A Surreal Virtual Reality Experience that Explores the World of Dreams"—detail game development research projects that explore how video games facilitate self knowledge. *Dreamwalker* aims at making participants feel as if they are inside a dream while in a lucid state, using space and architecture as navigational clues and surreal elements. In *ANX Dread*, in-game architecture induces player anxiety, measured through an external heart rate monitor; players are encouraged to manipulate elements in the game to train them how to tame their unease.

Other essays look to architecture as a framework for aesthetic analysis and building taxonomies of games. For example, the Smithsonian American Art Museum's *Game Spaces* event utilized architectural design principles and different definitions of "space" to establish exhibition guidelines and select works for display. Bobby Schweizer's historic analysis of video game cities before and after *Grand Theft Auto III* charts how "open-world" games are inextricably tied to extraordinary efforts in creating urban spaces for action. Robert Yang's essay offers analysis of the "zero player" aesthetic trend prevalent in the fan communities surrounding games with built-in level editors such as *TrackMania* and *Super Mario Maker*; trends like this, he argues, offer a history of design similar to that found within architecture itself.

Is this the last word on games and architecture? Gosh, we hope not! In bringing you this special issue, we hope to lay a foundation for a disciplinary shift where games are recognized as more than the sum of their graphics, sound, controls, challenge, or "fun factor." Video games and the built environment are complexly related in not only shaping a players' game experience, but also their lived reality.

Figure 1: Games in the museum.

Game Spaces:

Game Architecture as Aesthetic at the Smithsonian American Art Museum Arcade

Christopher W. Totten Assistant Professor, Kent State University; founder of Pie for Breakfast Studios

ABSTRACT

This work considers the curation of games according to the design of elements within the game—artwork, sound, music, and so forth—through 2018's Smithsonian American Art Museum (SAAM) Arcade: Game Spaces. This event, held as part of the yearly SAAM Arcade—a weekend festival at the SAAM featuring classic arcade and console games as well as a selection of independent games selected according to a curatorial theme—selected games by the way they utilized spaces within or around the games to create meaningful play experiences. This case study describes the development of the "Game Spaces" curatorial theme according to how it integrated architectural criticism into the evaluation of video games. The resulting event will also be described, including ways in which it succeeded and failed to inform visitors about the "Game Spaces" theme.

BACKGROUND

Thanks to the popularity of games and events integrating them, game exhibitions are no longer a new phenomenon in museums and cultural institutions. The pervasiveness of these events challenges organizers and curators to find novel ways to select works. Early exhibitions, such as the Smithsonian American Art Museum's (SAAM) 2012 exhibition, The Art of Video Games, showcased games as singular aesthetic and historical objects. Others, such as the Museum of Design Atlanta's 2013 exhibition XYZ: Alternate Voices in Game Design or the Victoria and Albert Museum's 2018 exhibition Videogames: Design/Play/Disrupt, focus on the creators behind games and stories of a game's creation. The aesthetics on display in each reflect pervasive views on games as *objets d'art*: in which games, through a player's interaction with them, express ideas or experiences external to the game itself. Sharp does this when he divides games with artistic intent into the categories *game art*, *artgames*, and *artists' games*.[1] The definitions of these categories are those that use game interactivity as a means of subverting the goals of popular games, those that engage subject matters that games often do not, and those that combine the previous categories, respectively. As large productions in preparation over many years, museum exhibitions must typically lean on these tried-and-true definitions even as game designers and critics are exploring new categorizations.

In 2014, SAAM staff reached out to the organizers of the Washington, DC chapter of the International Game Developers Association (IGDA) asking whether they had ideas on new game-centered events to promote a new game-inclusive exhibit, Watch This! Revelations in Media Art. For SAAM, holding this event was a response to both the upcoming exhibit and to the success of The Art of Video Games, which attracted 680,000 visitors.[2] For IGDA DC, it was an opportunity to create a nearby event that would encourage local developers to polish their ongoing game projects and gain attention for the region's game development scene. The result was the SAAM Arcade, then known as the Smithsonian Indie Arcade: a one-day event in December 2014 where visitors could play video games in the museum's courtyard. The first event featured a lineup of sixteen games from the Washington, DC, Baltimore, and Philadelphia areas, as well as classic games provided by the Music and Games Festival (MAGFest), and attracted over four-thousand attendees. Given the event's success, SAAM and IGDA DC made SAAM Arcade a yearly event that attracted increasingly large audiences.

The goal of the event would be to attract audiences with popular and fondly-remembered games, such as Donkey Kong or Sonic the Hedgehog, while introducing audiences to indie developers' self-published games. In this way, game players familiar with popular commercial games could discover new content that they might have considered unapproachable or that they may have never even seen given many indie

Figure 1: An image from the 2016 SAAM Arcade event, which attracted 11,750 visitors in a single day. Photo credit: Bruce Guthrie, Smithsonian American Art Museum.

Figure 2: An attendee plays What Hath God Wrought? by Mike Lazer Walker. This package of games requires players to utilize a telegraph device to make choices and control on-screen activities. Photo credit: Bruce Guthrie, Smithsonian American Art Museum.

developers' small marketing budgets. Likewise, the presence of the developers themselves was seen as a way to demystify the process of game making itself and introduce attendees to the art form. To support this, local developers, SAAM staff, and representatives of Boolean Girl, a company that makes build-your-own-computer kits marketed for girls, ran game-making workshops throughout the day. Suddenly confronted with the reality of video games in a cultural space, many attendees who had not previously thought much of games took the time to explore them among the landscape of their favorite artworks. Some games chosen for the events aided this process through elements that directly connected them with the media art in the SAAM, such as custom controllers or interfaces.

Subsequent years saw the event attract 11,750 attendees for the 2015/2016 event occurring in January of 2016 and 19,850 for the two-day SAAM Arcade event held in August of 2017. While the first event limited its scope to developers in cities within a few hours' driving distance of Washington, DC, subsequent years included developers from across the United States and eventually worldwide.

GAME SPACES

After establishing the procedures for running the event over the course of three years, the organizers moved forward on a new initiative: curatorial themes. For the fourth annual event, it was decided that the theme would be "Game Spaces," which would align the selected games to the broader landscape of art and art history embodied by the SAAM building itself.

HISTORY OF THE SAAM BUILDING

The building that houses both the Smithsonian American Art Museum and the National Portrait Gallery is a historic one beyond its connection with those institutions. The building itself started life as the US Patent Office and was built between 1836 and 1868. It was designed by architect Robert Mills to "celebrate American invention"[3] and has been cited as a prominent example of Greek Revival architecture. After opening in 1840, the office not only managed the processing of patents, but also publicly displayed patent models, foreshadowing its later use as a museum. During the American Civil War, it served as barracks and a military hospital, and was the site of President Abraham

Lincoln's inaugural ball. After the Patent Office moved out of the building in 1932, it was the office building for the Civil Service Commission until the 1950's, when it was saved from demolition by historical preservation activists. It was given to its current occupants, the Smithsonian and the National Portrait Gallery, in 1958 and has been updated since.

ARCHITECTURE AND GAMES AS FOUR-DIMENSIONAL MEDIA

Beyond structural aesthetics and utility as a place to house artworks, buildings like the SAAM hold important experiential qualities that bring them in creative proximity to the much younger field of digital games. Much has been made in game industry commentary of the similarities between the digital worlds of games and the design of real-world architecture.[5] Among these comparisons are analyses of works from one medium viewed through the lens of the other, as with Totten's analysis of the National Holocaust Museum in Washington, DC according to Jenkin's concepts of games as "Narrative Architecture"[6]. Mueller takes a further step in connecting the two in her study of four-dimensional art and design, which she argues includes film, animation, theater, installations, and games.[7] While the defining component of four-dimensional art is time, Mueller argues that another key component is architecture itself; quoting game designer Jesse Schell, she argues that a key purpose of architecture is to "control a person's experience"[8]. Elements of a space, such as its scale, ambient sounds, topography, and the way that occupants move through it are considered ingredients of creating experiences via the movement of a person through an architectural space. Mueller uses the example of video artist Isaac Julien's *Ten Thousand Waves* (2010)[9]: rather than hang the video screens displaying the work along the four walls of a gallery, they are hung around the center of the space, affecting the path of the viewer through the installation. The same can be said of Eric Zimmerman and Nathalie Pozzi's architectural installations such as *Cross My Heart,* a labyrinth of hanging fabric walls twenty feet

Figure 3: *Desolus* by Mark Mayers. Players can shift between orange and purple-hued versions of the same world, viewing clues to puzzle solutions through windows. Image credit: Mark Mayers.

high in which players play a tag-like game where the "it" player dons a minotaur mask.[10] In both examples, as with the levels—environments that form the gameplay settings of digital games—or boards of many games, the arrangement of spaces affects the players' interactions with the game, one another, and the amount of time that interactions can take. The emotional states of players can also be affected by the proximity at which experiences are placed in such environments, with sudden contrasts and juxtapositions creating very strong reactions.[11]

GAME ARCHITECTURE AS CURATORIAL THEME

As four-dimensional art, games rely on the spaces that their players inhabit, both within the game itself and outside of the game, as a primary tool for controlling the player's experience of the game. As such, the spaces depicted in and existing around a game, if integrated into the gameplay itself, can be judged against the author's intent as an aesthetic quality of the game as one would other interactive elements.

In creating Game Spaces, SAAM Arcade organizers had several goals. First, they wanted to appreciate games not only as self-contained *objets d'art*, but as collections of artworks that include environments designed much in the way an architect would design a building. Secondly, they wanted to highlight the different ways that game players spatially interact with games: physically, playfully, and socially. Likewise, they wanted to investigate games that occur in public and private spaces or whose play affects the world around the game in some way. The theme, therefore, called for games that fit into one or more of these categories:

- Games that make interesting use of space, board, or level design or which use these types of design as a standout feature of the game itself
- Games that transform the real-world spaces in which they are played
- Games that are inspired by or that inspire a strong community or social space

Games were collected via Google Form during a developer submission period that lasted approximately three months from January to April 2018. Developers entering their games into the competition were required to submit a text summary of the game, screenshots of gameplay, a video demonstration of the game, and a working link to a playable version of the game that judges could download. The event attracted one hundred submissions. Judging occurred in two double-blind rounds. The first round was a "review" round where the game would be played by two volunteer reviewers—one a member of the game industry and the other a SAAM employee—and scored based on not only whether it fit into the theme, but also its professional polish, the quality of gameplay, and its novelty. The second round was a "judging" round where event staff and subject matter experts—level designers, architectural critics, art curators, and so forth—selected games based on their review scores and their adherence to the theme. From the one hundred submissions received, fifteen games earned spots in the Arcade.

GAME SELECTIONS AND EVENT OVERVIEW

Once the fifteen games were chosen and their attendance at the event confirmed, preparations for how to exhibit them during the event could be made. This section describes selected entries that exemplify the different theme categories in order to demonstrate how games were chosen for these categories. It also describes how these games were displayed at the event to call attention to the Game Spaces theme.

GAMES THAT MAKE INTERESTING USE OF SPACE, LEVEL, OR BOARD DESIGN: DESOLUS

Desolus is a traditional three-dimensional video game by Mark Mayers. The game features a surreal aesthetic and challenges players to solve puzzles by jumping between two alternate universes via black hole.

The game features a surreal aesthetic with a carefully-chosen color palette and nods to gothic architectural design. The main interactivity mechanics of the game revolve around exploration, and the game often has a haunting and peaceful atmosphere thanks to the aesthetics and soothing music.

Given that this category most literally deals with the architectural qualities of spaces within the game, common principles such as shape, form, color, and composition could be utilized in evalu-

Figure 4: The thorny vines along this staircase provide both a color contrast to the dark architecture and a linear element that leads the player's movements and eye towards the next doorway. Image credit: Mark Mayers.

ating it for entry in the event. Doorways and black holes feature prominently in the game's spatial mechanic, and are an integral part of its world design. Beyond merely placing them around the level, Mayers' designs drip with intentionality: the colors of the different universes contrast one another so that a doorway to one world becomes an easily-spotted beacon in the other. Likewise, level elements such as columns, arches, and foliage draw the player towards these doors where they might continue to search for their end goal.

The way in which players move between worlds is also novel: while atmospheric spatial puzzle games are common in contemporary indie game markets, Desolus's use of alternate versions of single game spaces attracted it to the event's judges. This, and Mayers' thoughtful composition of spaces, made it a strong entry in this category.

GAMES THAT TRANSFORM THE REAL-WORLD SPACE IN WHICH THEY ARE PLAYED: THE OCTOPAD

The Octopad, created by Patrick LeMieux, is not a new game at all, but rather a modified Nintendo Entertainment System (NES) that both subverts and enhances the experience of playing with that console. It accomplishes this by connecting eight controllers to the NES console instead of the standard one or two, but each of these controllers only has one of the standard NES buttons on them. The result is a console that turns even single-player games into comedic social experiences for those playing and those watching the game take place.

The Octopad is typically demonstrated with a popular NES game such as *Super Mario Bros.* or *Tetris* loaded into the console. This creates a game that is at once approachable and nostalgic, but seemingly impossible as many NES games are known to be challenging even on standard hardware.

The Octopad creates a simultaneously familiar and new social atmosphere surrounding the popular NES. Even the most seasoned game players are transformed into novices in the same standing as their less-skilled friends and relatives due to the modified controllers. Suddenly, players must coach one another rather than compete and on occasion, roles among players are reversed as the expert becomes the student. While utilizing literal architectural space, the Octopad creates social frameworks like those argued for by Jane Jacobs in her advocacy for neighborhoods with supportive social roles and spaces that put people in interaction with one another[12]. In Jacob's *The Death and Life of Great American Cities* (1961), she argues for neighborhoods with blended residential, commercial, and public spaces so that people within neighborhoods could develop social ties and adopt roles that help keep the neighborhood maintained and secure. She also argued that people within these communities take on leadership roles helpful for facilitating these social connections. In the case of The Octopad, the game invites players through familiar games like *Tetris* or *Super Mario Bros*., but puts players in different roles based on which button they have. This makes communication essential and creates the social context of a shared effort. Inevitably, a player takes on a leadership role helping facilitate others' play, but the upended gameplay allows this to be anyone rather than the player who is best at the game under normal conditions.

GAMES INSPIRED BY A SOCIAL SPACE: LOST & FOUND: ORDER IN THE COURT

Lost & Found is a series of tabletop card games created by Owen Gottlieb and Ian Schreiber that teaches players real medieval religious law systems. The party version of the game, *Order in the Court*, asks players to become speculative storytellers on how these laws may have been ruled on in a courtroom. One player is a judge and the others must tell a story on how certain laws came to be, such as, "three people may not simultaneously

Figure 5: The Octopad by Patrick LeMieux. This modified NES has eight controllers that function as one, as each has only one of the standard eight NES buttons: A, B, Up, Down, Left, Right, Select, and Start. Photo credit: Patrick LeMieux.

read from the same scroll," and "if a bird drops meat on your field, you may keep it."[13]

In a way, this game creates a social space similar to the one found around the Octopad, as most tabletop board and card games do, through creating a meaningful social context. Game designer Jason Morningstar calls this context the "fruitful void," a space in which game players can play with the game's reality and fill with their own creativity[14]. Alternatively, *Lost & Found* utilizes this space to educate players on particular historical and religious systems. In this way, the designers of *Lost & Found* are bending the ability of game spaces to control the player's experience into one that can also be utilized to teach. This is not a new idea, as the serious and persuasive games sectors of the industry have been actively using games for such purposes for over a decade[15].

SHOWCASING THE CATEGORIES

These examples are merely samples of games in each category. Beyond selecting them, the organizers had the task of making the connections between the games and themes apparent to at-

Figure 6: A game of Lost & Found: Order in the Court being played at SAAM Arcade. The game has players tell stories explaining how medieval laws may have been ruled on.

Figure 7: A team of attendees playing The Octopad at SAAM Arcade. Photo credit: Patrick LeMieux.

tendees. As this was the first year with curatorial themes, it was seen as a test-run for what would be more polished presentations in future years.

In previous years of SAAM Arcade events, volunteers assisted with properly contextualizing the games in the event with the artwork in the museum. For some, arriving at the museum and seeing video games was a jarring event: had they not come to the museum to get away from these types of games? The event volunteers were SAAM volunteers and staff who participated in exhibitions and other art-centered events in the museum. In these contexts, the volunteers were educated and equipped to answer attendee questions on how games fit among the artworks housed in the SAAM. Volunteers would be a major part of the event again and would be an integral part of calling attendee attention to the event's theme.

Another tool employed by the SAAM Arcade organizers was event signage. This came in two forms: major signage at entrances to the event and signs on the tables next to games.

These signs were placed prominently and utilized bright colors matching other event graphics. The posters on the table not only gave information about the games themselves, but also which category the game fulfilled in the context of the Game Spaces theme.

CONCLUSIONS AND IMPROVEMENTS

Even with the successes of SAAM Arcade, there are still many ways to improve the event, especially with the addition of curatorial themes. In general, having a theme was positively received by developers and the theme of Game Spaces was seen as one that fit with popular game design aesthetics. From an organizational end, a theme allowed for a better "narrative" to be told through the entries. In previous years, informal themes were applied once organizers saw the field of entries, but these failed to make it into any formal event literature, unlike Game Spaces.

Submissions from previous years typically averaged about 150 with no theme (completely open submissions). Submissions for Game Spaces were down from that number, but contact with developers during the submission period show that this is mainly due to some developers feeling

Figure 8 (Top Right): Image of a game on display with on-table signage visible. These smaller signs drew attendee attention to the category into which the game fit.

Figure 9 (Above): Event signage outlining the theme and the different categories.

that their game did not fit the theme and could not be entered.

One area that could be improved is the way in which themes are presented to attendees. While signage was present, it was easily missed by attendees rushing in to their favorite classic game cabinet. Signage was present on tables, but was easily overshadowed by the graphic embellishments that game developers use to decorate their tables. Shorter text on posters to better invite engagement, or a handheld guide to the event with the theme info, could improve this awareness while also providing other important information.

Despite these challenges to be addressed in future events, SAAM Arcade 2018: Game Spaces was a success. It was simultaneously a way to honor SAAM's historic venue and a showcase of the ways in which architectural principles could be integrated into the categorization and selection of games for exhibition. Game Spaces showed that games could be understood in terms of elements beyond their interactivity and that future work can be done examining the work of game artists and level designers. Architecture may hold a unique key for understanding games' place in the greater cultural landscape, owing to their shared ability to affect and be affected by their interactive and social contexts in ways that few other media can.

NOTES

1. John Sharp, *Works of Game* (Cambridge: MIT Press, 2015).
2. Jessica Conditt, “Here’s How Many People Saw The Smithsonian’s Art of Video Games,” retrieved from http://www.joystiq.com/2012/10/02/heres-how-many-people-saw-the-smithsonians-art-of-games/.
3. SAAM Architectural History, https://americanart.si.edu/about/history/saam-architecture.
4. Christopher W. Totten, *An Architectural Approach to Level Design* (Boca Raton, Florida: CRC Press, 2014); Bobby Schweizer, “Understanding Videogame Cities,” In DiGRA ‘13 - *Proceedings of the 2013 DiGRA International Conference: DeFragging Game Studies*; Robert Yang, “Level Design in a Day: Level Design Histories and Futures,” Game Developers Conference, 2015.
5. Christopher W. Totten, “Game Levels as Works of Art, Architecture, and Design,”VGA Reader 1, no. 1 (2017): 11-20.
6. Henry Jenkins, “Game Design as Narrative Architecture,” 2004, accessed on October 29, 2019, https://pdfs.semanticscholar.org/f82f/061e7a44530d1dee281b96d9b1640485aa74.pdf?_ga=2.110833029.1001380941.1572311635-1223287680.1572311635.
7. Ellen Mueller, Elements and Principles of 4D Art and Design (Oxford: Oxford University Press, 2016), xix.
8. Ibid., 169.
9. Isaac Julien, *Ten Thousand Waves,* 2010, multi-channel film installation, Museum of Modern Art, New York.
10. “Nathalie Pozzi and Eric Zimmerman - Iterate: Perspectives on Design + Failure,” Design and Failure blog video, 2015, http://www.designandfailure.com/nathalie-pozzi-and-eric-zimmerman/.
11. Totten, *An Architectural Approach to Level Design.*
12. Jane Jacobs, *The Death and Life of Great American Cities* (Cambridge: MIT Press, 1961).
13. *Lost & Found: Order in the Court* game information page, accessed on October 29, 2019, https://www.thegamecrafter.com/games/lost-found:-order-in-the-court-the-party-game.
14. Jason Morningstar, “The Fruitful Void: Principles of Tabletop Design,” East Coast Game Conference, 2013.
15. Ian Bogost, *Persuasive Games* (Cambridge: MIT Press, 2007).

Figure 1: The doors are the portals that allow for players to move from one stage of the experience to the next. Which world they are transported to is not predetermined. Different users will be transported to different places based on how they interact with the environments. *DreamWalker,* 2018-2019. Image courtesy of the artists.

DreamWalker:
A Surreal Virtual Reality Experience That Explores The World of Dreams

Zack Ragozzino Student
Gabriella Santiago Student
Enrica Lovaglio Costello Associate Professor
Dr. Zöe J. Wood Professor
California Polytechnic State University
San Luis Obispo, CA

ABSTRACT
DreamWalker is an interactive virtual reality (VR) experience that is staged inside procedural and surreal worlds personalized to each player. The project's main goal is to make participants feel as if they are inside a dream while in a lucid state. A series of diverse environments, or dream states, are accessible through portals. Once the participants enter a portal, the system generates a new environment in which their interactions affect the emotional flow of the narrative. As players delve deeper into a dream state, they encounter common emotions, such as fear, surprise, awe, and joy while also confronting feelings of claustrophobia, arachnophobia, and acrophobia.

DreamWalker was created by a team of four students inside class settings over the course of two quarters (twenty weeks), supervised by the two faculty members who taught the classes.

Four key components are essential to the final product and drive the VR experience:

The user explores procedurally generated landscapes, which simulate the visuals that people encounter during intense dream states. The procedural generation is influenced by different parameters that modify the emotional tone of the experience.

The user explores and interacts with the virtual world. The interaction with environmental elements triggers changes in the procedural content and emotional tone of the experience.

Users have an enhanced sense of immersion, reached through the use of VR. Virtual spaces and environments are "personalized", which is achieved through populating the scene with photos from the user's Facebook account. This simulates the idea of unlocking memories from one's past as it would occur within a dream.

DreamWalker was primarily intended to be an open-ended experience in which space exploration, narrative creation and emotions blend together. For some, it is simply a zen-like walkthrough surreal environments. For others, it can be a way to confront their emotions or fears. Exposure to heights, for example, is a topic that we explore. Some individuals found the experience to be terrifying, while others found it exhilarating. Most of our design choices were influenced by observations derived from our own personal experiences with dreams as well as prior research about what causes dreams. The choice to use procedural generation and randomness was influenced by the activation-synthesis hypothesis, a theory stating

Figure 2: Animals can be found floating around the environment and swarming together like schools of fish. *DreamWalker, 2018-2019*. Image courtesy of the artists.

Figure 3: The flipped city occurs during peak surrealism. Its overwhelming scale made it a favorite moment for many people who play-tested the project. *DreamWalker, 2018-2019*. Image courtesy of the artists.

that while a person sleeps their brain fires off electrical signals that randomly pull up old memories. These become the basis for narratives that occur during one's dreams (van der Linden 2011). This inspired us to incorporate randomness into the application's generation of the environments so that participants are never fully in control of how their narrative will unfold. We hope that *DreamWalker* might become a platform for discussion regarding the importance of dreams and what we can learn from them.

THE PROJECT

We chose to explore dreams because they are one of humankind's unanswered mysteries. For some, dreams can be overwhelming, as they bring a flood of emotions, memories, and surreal experiences. For others, they are an escape into a place where we can tell ourselves stories and live the impossible. We, as a species, are fascinated by the world of dreams and how they affect our emotions and mental states. We theorize how dreams connect to reality. Some believe dreams have the power to predict our course in life, others say that dreams reveal our true selves. Some individuals even learn how to lucid dream so that they can control a world with no limitations. Our project aims to explore the many aspects of dreams and how they affect us. With the power of virtual reality, we get as close as we can to lucid dreaming and exploring the surreal feeling of it.

ARCHITECTURE OF ***DREAMWALKER***

The design of *DreamWalker* is inspired by how people might experience dreams. There are no tangible motivators presented to the player, the motivation for the exploration is curiosity. We intentionally used unconventional spaces and architectures, such as, for example, an upside-down city or a forest falling onto the user. In order to draw the user towards points of interest, we experimented with eye-catching lighting and colors. We avoided using game-like elements, such as collecting items or defeating enemies, because we felt it would detract from the core goal of aimlessly walking through a stimulating fictional space.

The experience starts in a simple bedroom, which represents a safe space and is meant to ease the user into the surreal visuals that populate the virtual world later on. The player can interact with many different objects in the room, such as books or furniture. This space is where we teach the user how to move in virtual space and use the VR controls.

We intentionally designed the room uncluttered and plain in juxtaposition to the colorful and large-scale visuals that compose the rest of the experience. This has the added function of giving participants, who are new to VR, a chance to adjust to the technology without being overwhelmed or distracted by the dream environments.

After the tutorial is finished, the walls of the bedroom fall towards the ground, and players are given the freedom to explore the world as they choose. For each level, we utilize portals, in the form of "floating" doorways, and architecture to encourage exploration and the making of narratives. Within every level, floating, glowing doors swing open as the user gets near it. These doorways are the user's way of accessing the next dream-space. Every time a doorway is entered, *DreamWalker* procedurally generates a new environment based on parameters that the player is not aware of. The flow of the narrative changes dynamically; one world might depict a serene forest, while the next a vast empty desert, an upside-down city, or a landscape with furniture falling from the sky.

We noticed that many players were naturally drawn toward the doorways. That may be due to the fact that, in the context of a game, some players would seek out the doors to progress. The doors are not attached to any buildings or hallways. These standalone doors encourage player interaction, they also contribute to the dream experience due to their surreal appearance. A house, located in the middle of a wide clearing, is the focus of exploration for one of the levels. This house has a simple layout and many objects that the player can choose to interact with. Eventually, the house is overrun with grass and other foliage, another surreal or strange element that contributes to the dream experience.

The use of open space is as important as the built environments. The user can discover the woods, the empty streets of an upside-down city, or a wasteland filled with furniture. These open spaces allow us to emphasize other elements such as the doorways or more conventional buildings. In other levels, we experimented with claustrophobic environments by changing properties of certain elements. For example, giant redwood trees slowly fall towards the player, or hundreds of animals bounce around erratically like rubber balls. We use these vast procedurally generated open spaces to encourage players to move or behave in different ways, or to interact with objects that can trigger a change in the dream state. Ultimately, we use space to trigger emotions.

While the doors are conducive to transport the player from one state to the next, the player can do this in other ways. For example, if a bear is encountered, which follows and attacks the player, another dream state is triggered, and a new environment unfolds in front of the user's eyes. When trees start to fall onto the player, a different environment unfolds again. This act of transporting the player to a variety of worlds is meant to further emulate the fleeting and impermanent nature of dreams.

At the end of the experience, which lasts three minutes, users are automatically transferred to the final stage of the dream: the brightly lit bedroom already encountered at the beginning. The user is dropped into the room, falling for about 5 seconds, then text appears indicating the end of the dream; from this space, the user cannot continue into other dream states. Dropping the user into a bedroom as the final stage points to the cyclical process of sleeping, dreaming, and waking again.

FEATURES OF ***DREAMWALKER***

Elements to evoke the feeling of dreaming include:

- The "Follow" script, which causes a creature to follow the user. This is used to trigger the feeling of anxiety. This script is attached to a bear that follows and attacks the user.
- Flying animals, such as flying octopuses, sharks, or butterflies, are included to evoke a feeling of awe or curiosity. It is also meant to be surreal, which contributes to the user's sense of immersion within the experience.
- Swarms of insects, such as butterflies or spiders, surround the player. This particular feature explores fears encountered in dreams, such as arachnophobia.

Procedural generation and the creation of content with random noise are used to ensure that no two dreams happen exactly in the same way throughout the experience. We utilized procedural generation to randomize and diversify the content of each dream level. We created the assets and building blocks of the experience and used a procedural generation algorithm to control specific parameters about the environment. These include properties such as the environment type (e.g., city, forest, flat plains, etc.), the density of placement (how many trees to place in our forest), color palette, and the scale of various elements. Terrain's generation and vegetation's placement were both based on prior research regarding the procedural generation of virtual worlds (Freiknecht and Effelsberg 2017). We use Perlin noise to randomly generate grayscale bitmaps (Perlin 2002). The value of each grayscale pixel directly ties to the elevation of a portion of the terrain. This same technique is used when attempting to randomly place vegetation and props in our scenes. A separate grayscale map is generated using Perlin noise and the data from the map is used for determining where to place objects and what types of objects should go where. Instead of using the value of the grayscale texture to control a height parameter, we use the value as an index into a library of assets to pull from.

We ran into issues when we realized that some players had exciting and unique experiences, while others walked through the environments without ever encountering any captivating moment. This is

Figure 4: Vacation memories, family pets, and old high school friends were common photos found scattered throughout the environment. *DreamWalker*, *2018-2019*. Image courtesy of the artists.

the unfortunate nature of relying on randomness to drive a story. We solved this by designing a few unique and rare environments that a participant might encounter once or twice during a single experience. For example, the buildings in a familiar city environment might appear distorted and rotated in odd directions. The addition of these occasional "directed" moments was important because it gave us back some creative control over the flow of the narrative. By manually adding a couple high points we were able to ensure that every player would finish the experience with at least a few noteworthy events.

Another issue that we encountered, due to the procedural generation, was that randomly generated environments of the same kind felt identical. In other words, the algorithm generates a forest that technically is unique, but the lack of artistic merit and reuse of 3D assets made it difficult to distinguish that forest from a different one generated during the experience of another user. This taught us that procedural generation does not always yield interesting results in terms of making compelling and original spaces. We solved this problem by developing "sweet spots," handcrafted areas scattered in the environment for the player to discover. While exploring the forest, one might encounter a circular clearing of trees with a glowing portal in the center, or a large boulder to climb. The incorporation of sweet spots allowed us to give the appearance that an artist had crafted each world while still using the power of procedural generation.

PERSONALIZATION OF THE DREAM EXPERIENCE

An important feature in the experience is the fact that environments are personalized and customized for each user. We used two different strategies to achieve this effect. One involves keeping track of the objects the user chooses to interact with. Depending on those objects, the color of the level overall is altered. We assign a value to each object, which the user can interact with. Such value would either give the user a "good" or "bad" dream. For example, if a player encounters snakes, the tone of the dream turns "bad", and the world around the player will be tinted red. If a player interacts with a serene clearing in the woods, the dream is "good," and the level has a greenish tint. As users traverse each level, this counter of good versus bad changes depending on what the user chooses to explore or interact with. As users make decisions throughout the experience, they unknowingly craft their own narrative. The flow of the story is completely dependent on the user's choice in terms of interaction with certain assets in the world. This is an important element of our project that we plan to focus on during future iterations.

The other strategy used to give the users the feeling of personalized spaces is through Facebook integration. This feature took the user's personalization to the next level. While we never force individuals to log into their Facebook accounts, we offer that option.

The users that are comfortable with providing this information are given the ability to experience truly personalized environments, where assets show photos of the user's pets, past events, family or friends.

This is the closest we were able to get to crafting VR worlds that were truly built around each individual's life. Through this method of personalization, we reflect the user's interests and concerns as closely as possible (Lewis 2014).

ART OF ***DREAMWALKER***

For this project, it was important to balance surrealism and realism. There are elements within dreams that have hints of reality, but they often are combined in such ways that they could not exist in real life. To maintain realism, we found and created realistic assets but used them in ways that would never occur in reality.

We avoided 3D models with low polygon count and items that might have a cartoony art style. Instead, we took the normal, realistic objects, we positioned them in certain ways and assigned to them behaviors that can only be found in a dream. For example, the user may encounter realistic fish flying through the air throughout the exploration of the VR worlds. In another example, the buildings of a cityscape are scaled down in order to make the user feel much larger in comparison. This balance between surrealism and realism is important in conveying the feeling of being in a dream. Surrealism as an art form aims at stimulating the unconscious to unlock the power of the imagination, making it the perfect inspiration for us as we developed the dream worlds.

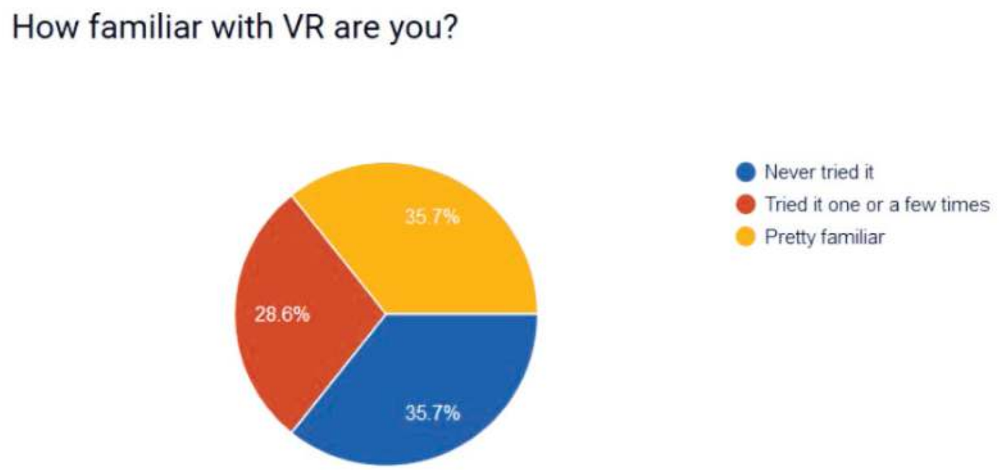
PLAYER FEEDBACK

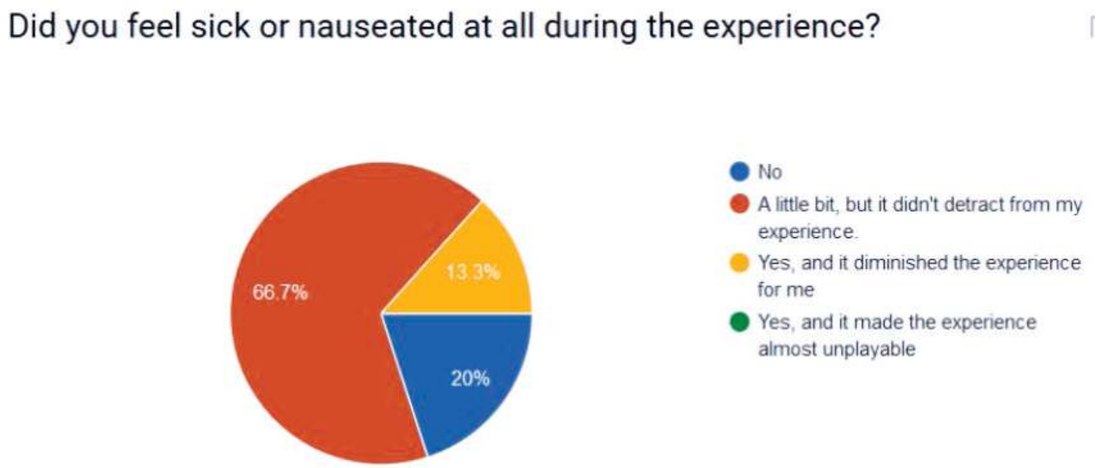
While refining *DreamWalker*, we were most concerned about two topics, one related to the participants' likelihood to be nauseous during play, and the other to the duration of the experience. We quickly realized that our team would not be able to assess these issues without testing the application on willing participants and recording their feedback.

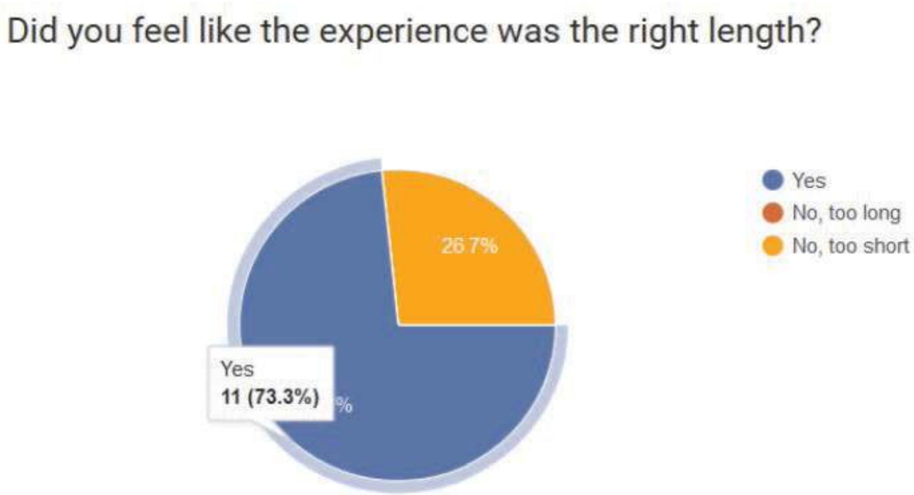
We were surprised to find that while there was some slight nausea experienced among play-testers, most of them did not find it to be too debilitating. This was a relief because we were aware that our VR application contained a lot of overwhelming imagery, and we planned to scale back on the visuals if users felt it was too intense. Surveys seemed to indicate that we were just on the edge of what could be considered a nauseating experience. In the future, we would like to test different types of movements to see if any would reduce motion sickness.

Before we hosted play-testing sessions, we experimented with how long we should make the experience and settled on three minutes. Fortunately, most of the survey results show agreement with this number. In the future, we would like to see if results would have been better with a longer duration such as five or ten minutes.

Play-testing proved to be crucial, not only because it resolved our initial concerns, but it also revealed several issues that we did not account for. The tutorial occurring at the beginning of the ex-







perience, for example, was initially designed very poorly and had to be revised multiple times. We also found that it was valuable to get impressions from people of different age groups. Younger children, for example, were immediately captivated by the experience and we had no concerns regarding whether they would enjoy it. On the other hand, adults over the age of thirty struggled to maintain interest and this inspired us to revise our procedural generation to ensure that there would always be interesting events happening to avoid boredom due to uneventful navigation through the environments.

CONCLUSIONS

DreamWalker explores how VR, procedural generation, and Facebook integration can be combined to create a captivating and personalized experience that people of all backgrounds and ages can enjoy. While we are happy with the current version of the project, we are also aware that there are few weak points that we will improve in future iterations.

Going forward, we plan to add more interactive components in the environments. During testing, we noticed that players tried to grab items in the world expecting interesting things to happen. For example, some would try to pet the animals they encountered and were disappointed to find that the animals did not respond to them. Incorporating interactive elements would encourage participants to explore the environments further and would also add a more playful tone to the experience. These interactions could also be used for more personalized procedural generation. By keeping track of which items the players pick up, we could build the narrative around what they find interesting. If players pick up a basketball in the first world, we might put them in a basketball court filled with bouncing balls in the next world. If they instead try to pick flowers, then we could make the next environment a vibrant garden. This would reinforce the idea that *DreamWalker* is a dynamic experience with a "director" working behind the scenes. We would also like to build upon the usage of Facebook data. Due to technical complications, this feature was only partially integrated into our final build. We were able to project a user's Facebook photos onto cubes that floated in the sky; in future iterations, we would like to incorporate these pictures more organically in the environments. For example, when participants walk into a house, they could find picture frames on a table with familiar photos from their past.

When we originally planned DreamWalker, we knew that we wanted to create a VR experience with procedurally generated worlds that responded to player interaction. It is a project that stands out among other VR applications and we hope that our mistakes and successes can help future developers continue to build captivating interactive products.

BIOS

Enrica Costello is a faculty at the California Polytechnic State University, San Luis Obispo, teaching digital media in Art and Design. She holds master's degrees in architecture (University of Genova, Italy), and in media arts and technology (University of California in Santa Barbara). She is currently pursuing a PhD.

Gabriella Santiago is a student at the California Polytechnic State University, majoring in Liberal Arts and Engineering Studies with a focus in Computer Graphics and Art, and minoring in Computing for Interactive Arts.

Zack Ragozzino is a recent alumni from Cal Poly. His background includes virtual reality, computer graphics, and visual effects. He enjoys using virtual reality as a platform to explore human emotions and traits such as empathy, anxiety, and self-reflection.

As a Professor in the Department of Computer Science at Cal Poly, Zoë Wood's projects unite visual arts, mathematics and computer science. She engages students, via her NSF funded research projects, advising senior projects and master's theses. She is committed to increasing the number of underrepresented students in her field.

NOTES

1. Jonas Freiknecht and Wolfgang Effelsberg, “A Survey on the Procedural Generation of Virtual Worlds,” *Multimodal Technologies Interact* 1, no. 4 (2017): 27.
2. Penelope A. Lewis, “What Is Dreaming and What Does It Tell Us about Memory?” *Scientific American*, July 18, 2014, https://www.scientificamerican.com/article/what-is-dreaming-and-whatdoes-it-tell-us-about-memory-excerpt/.
3. Ken Perlin, “Improving Noise,” In *Proceedings of the 29th annual conference on computer graphics and interactive techniques* (New York: ACM, 2002), 681–682.
4. Sander van der Linden, “The Science Behind Dreaming,” *Scientific American*, July 26, 2011, https://www.scientificamerican.com/article/the-science-behind-dreaming/.

Figure 1: Soldering Puzzle: One of the tasks that the player must complete to escape the malfunctioning ship is to fix the broken door using a soldering iron by tracing a specific path.

ANX Dread:
A Virtual Reality Experience to Explore Anxiety During Task Completion

Chanelle Mosquera Student
Ross De Vito Student
Dr. Zöe J. Wood Professor
Enrica Lovaglio Costello Associate Professor
California Polytechnic State University
San Luis Obispo, CA

ABSTRACT

The use of virtual reality as an assistive technology is of wide interest.[1] In particular, recent work in the use of virtual reality to help users manage psychological challenges has been fruitful.[2] This student capstone project focused on studying users' experience focused on anxiety driven by task completion.

This VR experience, named *ANX Dread*, immerses users into a mildly stressful environment of a malfunctioning spaceship with simple puzzle-like tasks to complete. The experience, built with Unity, includes an integrated heart-rate monitor. The project was built as a part of the capstone experience for the 'Computing for the Interactive Arts' minor for the 2018-19 academic year by a team of five students from various academic backgrounds. This poster presents the system and a study reflecting the user's sense of anxiety within the experience.

INTRODUCTION

The negative overwhelming feeling of anxiety is experienced by nearly 40 million people in the United States. Even more concerning, approximately eight percent of young people, children, and teenagers experience anxiety, developing symptoms at a young age. Virtual Reality (VR) has been shown to be helpful as an assistive technology for users with various physical and mental conditions.[1] Furthermore, recent work on VR in psychological contexts shows great promise.[2] Rooted in a growing concern surrounding anxiety in young people, this project aims to explore the application of virtual reality in support of furthering studies regarding managing stress and anxiety. Inspired by projects such as "Injustice," this project follows a similar approach by putting the user in an immersive environment in order to explore their reactions to anxiety-provoking situations.[3] Specifically, our system was designed to explore users' reactions to simple tasks in simulated stressful and fictional settings, namely a malfunctioning spaceship. Additionally, part of the study examines the user's level of anxiety and stress with respect to specific game elements.

THE EXPERIENCE

The goal of the *ANX Dread* virtual reality experience is to explore users' reactions to stress and anxiety associated with task completion. The project includes a virtual reality experience in which the user must escape a broken spaceship by completing simple tasks in the digital world. To contextualize the project we describe the basic user experience here.

The user starts in a dark control room where only a large control panel appears with a large yellow button that can be pressed. Once pressed, the control panel prompts the user to choose a character; when one of the available characters is selected, however, the ship malfunctions. This is signaled by a stressful series of events: the control room lights turn on, and the only exit door inside the space is broken, as shown by the fact that it loudly moves up and down. A HUD appears displaying the user's heart rate, and a clock starts counting down from five minutes. The user's heartbeat can be heard throughout the entire experience. If the clock runs out before the user reaches the end of the experience, it continues to count in negative time, which gives a sense that the experience will never end.

The user's first task is to figure out how to fix the broken door [fig. 1]. To the right of the door there is a panel of broken pins with instructions to "Solder the broken pin headers" printed directly beneath it. A spotlight illuminates a soldering iron on the nearby shelf to draw the user's attention to the tool necessary to solve the puzzle. The user must use the pin's coloring to trace the correct path of pin headers and fix the door. Once the door has been fixed, the user enters a labyrinth of long, dark hallways, riddled with explosions of sparks and smoke [fig. 2]. The explosions are periodic enough to keep the user in a constant state of suspense. A gradient of colored lights at the end of each corridor guides the user to the end of the labyrinth. Every white light leads to a dead end, while the gradient of colors—starting with blue and ending with red—leads the user to the end of the labyrinth. The user has to figure this out, as no instructions are given. Once the user reaches the end of the maze, the outside of the ship is visible through windows and the user discovers that the ship is floating in space. At that point, the user arrives at a fork in the path. Both the left and right options look the same: each a dark hallway leading to a possible exit door lit dimly by a flickering light. In reality, both options lead to the end of the game, but they are presented in this way to give the user the illusion that one unchangeable decision determines the final outcome of the experience. After the decision is made, the hallway is lit with spinning siren lights, and the user can see a door with a hand scanner at the end [fig. 3]. Before the user is able to reach the end, the hallway elongates to emulate the "vertigo effect" seen in many movies. When reached, the hand scanner fails two times before finally letting the user through the door and into the open space outside. Each failure is accompanied by a reminder that the user must "calm down" in order to be able to exit the ship.

Finally, at the end of the experience, the door opens, revealing that the user must jump out of the

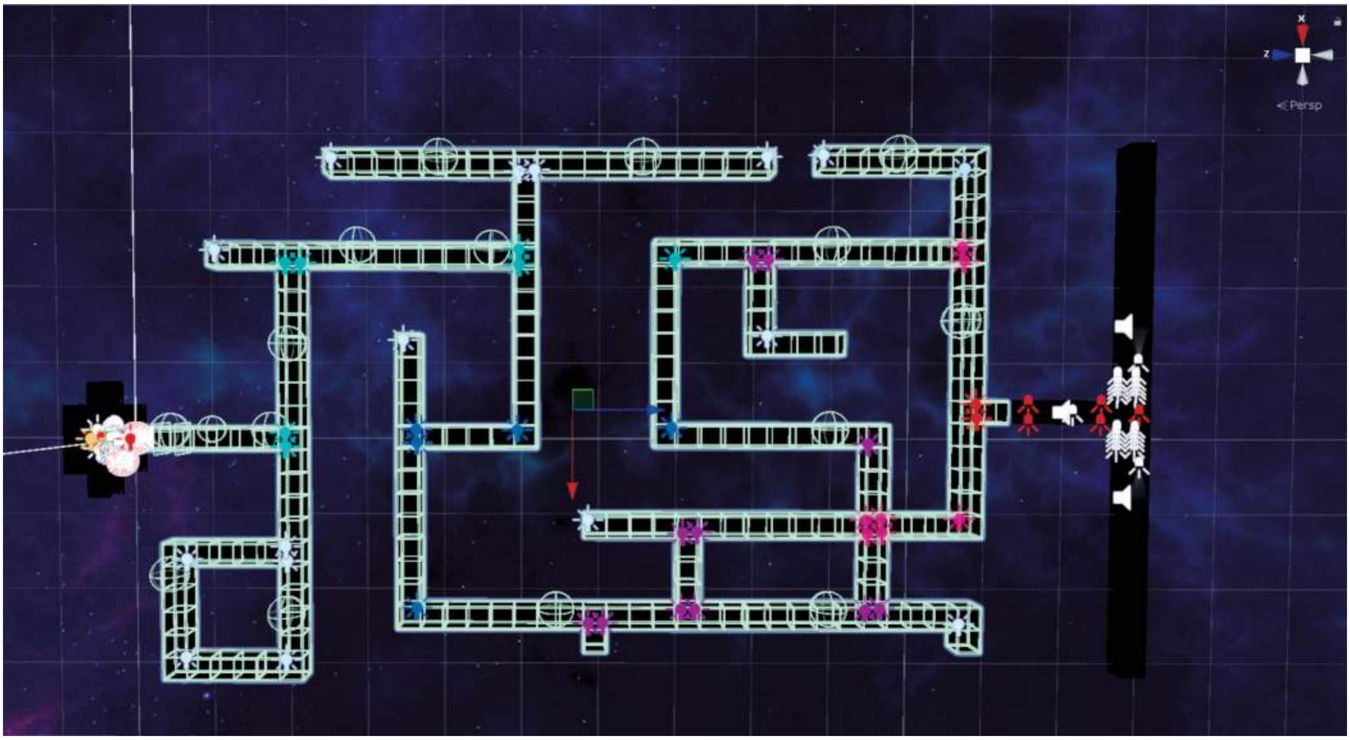


Figure 2: Maze: One of the tasks the player must navigate is to escape the ship in a maze.

ship. Once the user leaps into space, a giant cake shows up, floating below the user. The cake is used as an award and draws the attention of the user. Once the user lands on the cake, the experience ends and credits scroll down the screen, indicating the user's success of escaping the ship.

SYSTEM DEVELOPMENT

ANX Dread was developed in the Unity 3D environment using the HTC Vive VR system. Resources used in making the project include both free and purchased models found in the Unity store and elsewhere online, and original content developed by the team. The majority of the assets in the VR environment, including the walls, floors, ceilings, lights, come from a sci-fi modular pack. Students taught themselves how to use the technology necessary to integrate these assets into the project via tutorials and references.

At the beginning of the VR experience, the user can interact with the environment via the in-game "hand," a 3D model that is mapped to the Vive Controllers (see Figures 1 and 3 for example images). The hands are able to collide with and grab certain objects in the environment as the user progresses through the experience. The user navigates the VR environment by holding onto a specific button on both controllers while simultaneously swinging their arms in a walking motion. Collision detection between game elements is used in the soldering puzzle.

HEART RATE MONITOR

One of the primary goals of the project was to explore anxiety associated with task completion in VR; thus, a heart rate monitor was incorporated into the experience. The PulseSensor heart rate sensor and library, developed by Yury Gitman and Joel Murphy, provides biofeedback data to an Arduino Uno.[4] Example Arduino code was modified for this project to print the current beats per minute (BPM) to the serial port, providing feedback to the Unity VR software (see Figure 8 for example data). An open-source library called WRMHL enables the system to read any data coming in from a specified serial port using Microsoft's .NET Framework 2.0.[5] In order to make the library work for the existing project, any "empty queue" conditions are filtered out from the Arduino. This is implemented via a peek at the queue to filter the incoming data to only contain valid BPMs. In addition, because the VR experience includes a scene change midway through, the heart rate script needs to close communication to the Arduino at the end of the first scene so that it can be reinitiated at the start of the second scene. Without this alteration, Unity would try to have two instances of communication to the Arduino, resulting in no information.

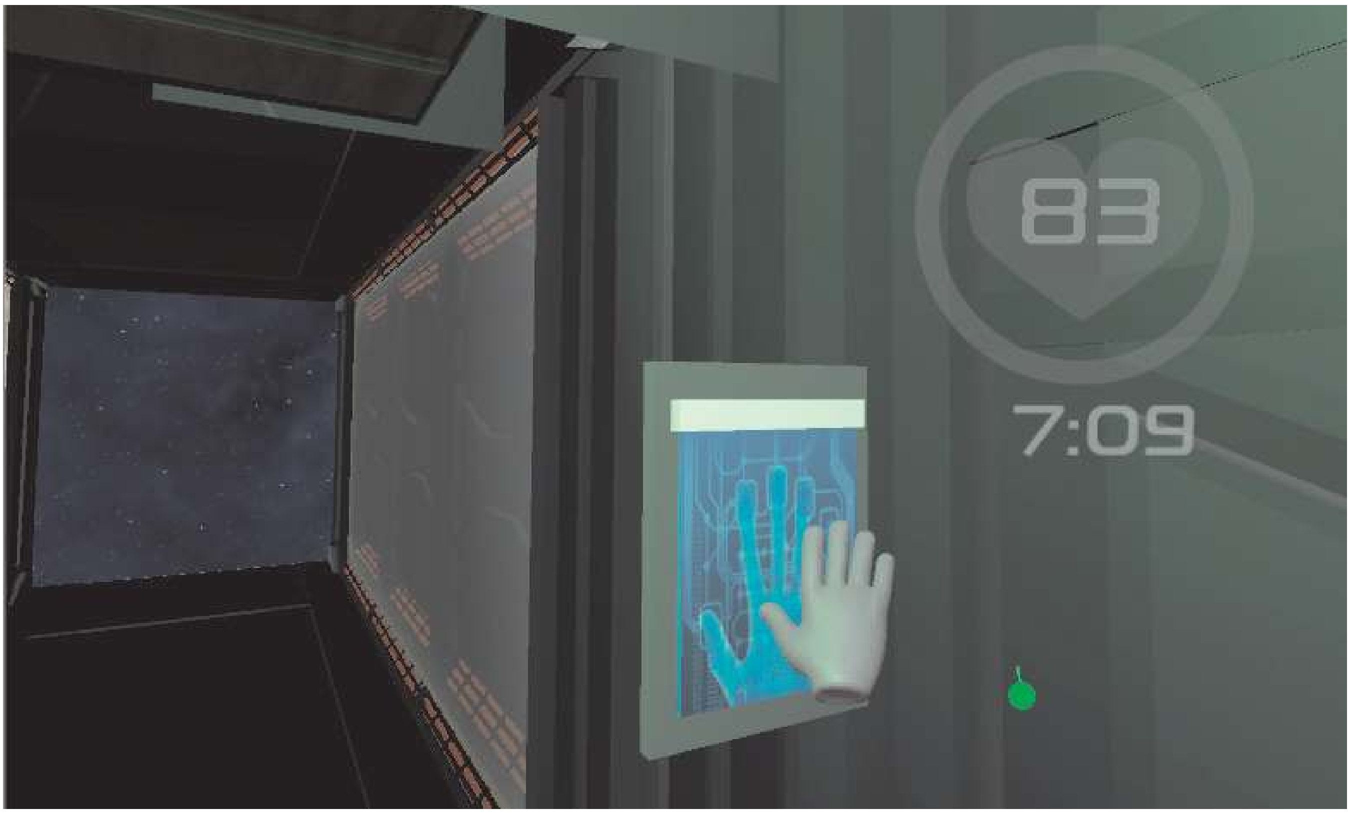
The Arduino and PulseSensor assembly are harnessed in a fanny pack that is wrapped around the user's back during the experience. The PulseSensor comes out of the pack and clips onto the ear lobe, right underneath the headphones. This set up minimizes extraneous movements that could trigger false readings for PulseSensor (Figure 4 demonstrates the minimal impact our system has on a user's VR experience).

USER STUDY

We conducted a user study to explore the user's stress and anxiety levels during task completion. The study measured the user's speed at completing tasks, their heart-rate, and their self-reported experience via a post-task survey. The main questions in the survey include:

1. *Did you experience anxiety while completing the tasks in the experience?*
2. *Please rate each task/experience from most anxiety-inducing (5) to least anxiety-inducing (1):*
   a. *Spaceship Breakdown*
   b. *Puzzle*
   c. *Maze*
   d. *Extending Hallway*
   e. *Heart Rate Door*
   f. *Other*

Figure 3: Heart Rate Door: Initially unable to open the door, the player is urged to calm down as the heart rate must be lowered. Here, the user finally opens the door with the hand scanner successfully.



3. *Could you imagine this kind of experience helping you practice managing stress?*
4. *Did you experience any physical sensations during the experience?*
   a. *Heart Rate Increase*
   b. *Sweating*
   c. *Body Tensing*
   d. *Fatigue*
   e. *Other*
6. *Did the heart rate monitor and heartbeat sound cause you any anxiety?*
7. *Did the countdown cause you any anxiety?*
8. *Did the sound effects throughout the experience cause you any anxiety?*
9. *Did the particle effects, the steam clouds, and sparks, cause you any anxiety?*

SUBJECT SELECTION AND DATA COLLECTION

The pool of subjects included Cal Poly undergraduate and graduate students recruited via in-class announcements and Facebook postings asking for volunteers to participate in a questionnaire/study. No incentives were offered. Thirty participants were observed by investigators. Videos of user experience were collected if participants allowed. In addition, a clip-on heart rate monitor was used with participants' permission to record heart rate information throughout the experience.

RESULTS

Of the thirty participants in the user study, a little more than half (56.7%) reported that they had used VR prior to the *ANX Dread* experience.

In terms of measuring users' anxiety responses, we found the "Maze" component to be significantly more anxiety-inducing than most other components. Specifically, we conducted one-way ANOVA between subjects to compare the effects of the VR experience on anxiety for participants (N = 28) self-rated anxiety levels. With these metrics, the maze condition stands out in comparison to the various tasks in the experience ($F(4) = 3.69$, $p < 0.01$). There was a significant difference in mean anxiety between the "Maze" condition and the other conditions, specifically: "Spaceship Breakdown" ($p = 0.0184$), "Soldering Puzzle" ($p = 0.0348$), and the final "Heart Rate Door" ($p = 0.0184$) [fig. 4]. In addition, post-hoc comparisons indicate that the mean self-reported anxiety level for the "Maze" condition (M = 3.79, SD = 1.07) was significantly higher than the mean level for "Spaceship Breakdown" (M = 2.75, SD = 1.32), "Puzzle" (M = 2.82, SD = 1.28), and "Heart Rate Door" (M = 2.75, SD = 1.40) (see fig. 5 for complete data).

The "Other Experience" components had a mean self-reported anxiety level of 3.46 out of 5. While we cannot statistically say it was significantly higher than other components, we can conclude that they were the second highest mean anxiety level, right behind the Maze component. Standard deviation was 1.318 for this component.

Based on the physical sensations sections of the survey, we found that the common side effects from anxiety-inducing components in our experience were heart rate increase, body tensing, and sweating (figs. 6 and 7).

As Figure 6 shows, we had 60% of users report feelings of heart rate increase. Figure 7 shows one user's heart rate data, collected via Arduino and pulse sensor, displayed overtime as they completed the experience. In each participant's data we consistently see an increase in heart rate from the

Figure 4: Results from one-way ANOVA between subjects to compare self-rated anxiety levels in participants (N = 28). Chart compares each condition and listed the given p-value.

	Spaceship Break Down	Puzzle	Maze	Extending Hallway	Heart Rate Door
Spaceship Break Down	1	0.9996	**0.0184**	0.4385	1
Puzzle	0.9996	1	**0.0348**	0.5749	0.9996
Maze	**0.0184**	**0.0348**	1	0.6436	**0.0184**
Extending Hallway	0.4385	0.5749	0.6436	1	0.4385
Heart Rate Door	1	0.9996	**0.0184**	0.4385	1

	M	SD
Spaceship Break Down	2.75	1.3229
Puzzle	2.8214	1.2781
Maze	3.7857	1.0666
Extending Hallway	3.3214	1.2188
Heart Rate Door	2.75	1.4044

Figure 5: Mean and Standard Deviation for level of anxiety felt for each condition given by self-reporting participants.

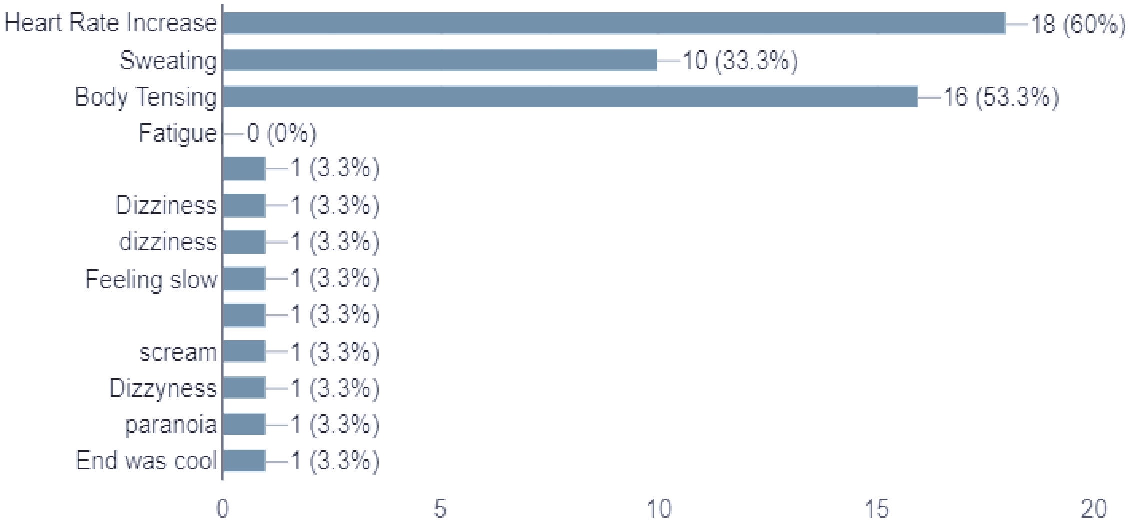


Figure 6: Physical Sensations experienced during VR experience. Users were asked to check the physical sensations they experienced as well as any additional ones.

user's given resting heart rate. During the initial "Crash" portion of the experience, users had an average heart rate of 96.67, during the "Puzzle" portion users had the lowest average heart rate of 96.39, during the "Maze" portion users had an average heart rate of 97.12, and during the "Hallway" portion users had the highest average heart rate of 99.07.

To validate the heart rate data obtained by the Arduino and PulseSensor we also used a Fitbit Versa in initial testing. After comparing the Arduino and PulseSensor data, we found the PulseSensor had some issues with sudden movements that caused spikes in the data and so made certain portions of the data unreliable. As seen in Figure 7, the heart rate begins to read over 200 BPM at the end of the sample. This was due to the participant's removal of the PulseSensor after the credits started to roll. Although there were outlying spikes in the data, the overall trend of each participant's data seems to reflect how they actually felt while in the experience.

CONCLUSIONS AND FUTURE WORK

We have presented the results of the project with a focus on exploring user anxiety and stress during task completion in VR. The system specifically simulates the mildly stressful environment of a malfunctioning spaceship and asks the user to complete simple tasks to escape the ship. A user study measuring experience indicates that users experienced the most anxiety during the "Maze" component of the experience. We conclude that when designing a future VR experience, a "Maze" component or some of the individual components included in the "Other Experience" section could potentially be used to induce anxiety in their

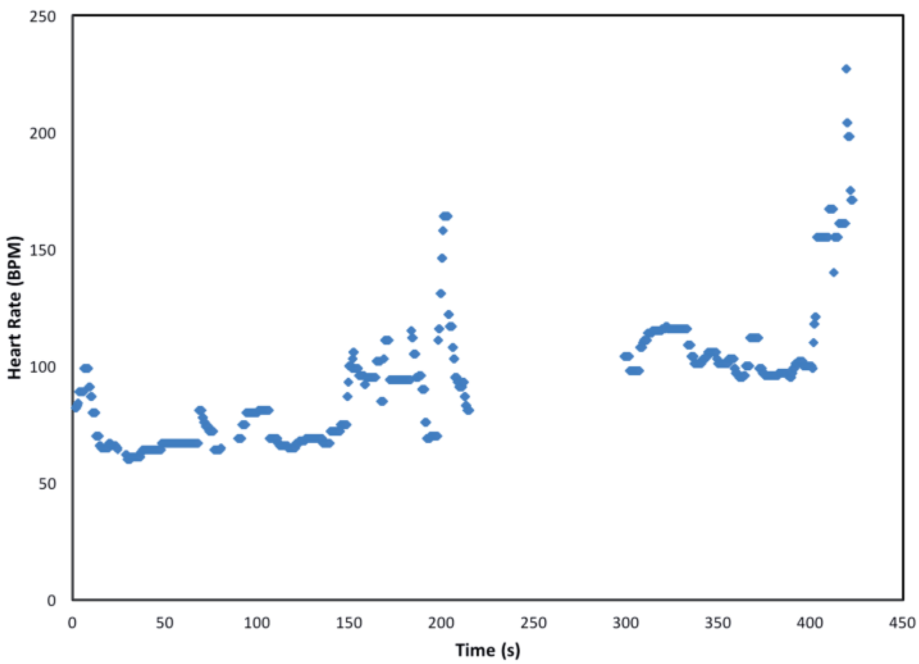


Figure 7: Heart rate feedback recorded from the PulseSensor for a single participant throughout the whole experience. The momentary break in the flow of data is due to the scene change in between the end of the maze and the start of the extending hallway.

participants.

The system included a heart rate monitor which fed data back into the VR HUD for the user. As noted, this was one of the "Other Experiences" that contributed to user anxiety. We would like to continue to explore the use of biofeedback to better understand the role this particular component played in user experience and how it could be used to help users manage anxiety.

The ultimate goal of the project is to develop an experience that allows users to manage their stress and anxiety. Future work includes developing a better understanding of the role that the heart rate monitor plays in the user experience.[6]

BIOS

Chanelle Mosquera is a student at the California Polytechnic State University pursuing her Bachelor's and Master's degrees in Computer Science with a minor in Computing for Interactive Arts.

Ross De Vito is a California Polytechnic State University graduate with a B.S. in Mechanical Engineering and a minor in Computing for Interactive Arts. He is excited about the world of game development and cannot wait to see what it has to offer.

Enrica Costello is a faculty at the California Polytechnic State University, San Luis Obispo, teaching digital media in Art and Design. She holds master’s degrees in architecture (University of Genova, Italy), and in media arts and technology (University of California in Santa Barbara). She is currently pursuing a PhD.

As a Professor in the Department of Computer Science at Cal Poly, Zoë Wood’s projects unite visual arts, mathematics and computer science. She engages students, via her NSF funded research projects, advising senior projects and master’s theses. She is committed to increasing the number of underrepresented students in her field.

NOTES

1. Rian Dutra da Cunha, Rodrigo Luis de Souza da Silva, "Virtual Reality as an Assistive Technology to Support the Cognitive Development of People With Intellectual and Multiple Disabilities," presented at VI Congresso Brasileiro de Informática na Educação, 2017; M.J. Smith, E. J. Ginger, K. Wright, M.A. Wright, J. L. Taylor, L.B. Humm, D. E. Olsen, M.D. Bell, and M. F. Fleming, "Virtual reality job interview training in adults with autism spectrum disorder," *J Autism Dev Disord* 44, no. 10 (October 2014), 2450–63; Dorothy Strickland, "Virtual reality for the treatment of autism," *Studies in Health Technology and Informatics* 44 (1997), 81–86.
2. Davor Gasparevic, "Improving Productivity with VR Meditation Apps," *Productivity Bytes,* December 18, 2017, https://productivitybytes.com/vr-meditation-apps/; Kayla Matthews, "How Virtual Reality is Improving Care for Mental Health Disorders," *The Doctors Weigh In,* November 18, 2018, https://thedoctorweighsin.com/vr-mental-health/; Maria Temming, Virtual reality therapy has real-life benefits for some mental disorders," *ScienceNews*, November 1, 2018, https://www.sciencenews.org/article/virtual-reality-therapy-has-real-life-benefits-some-mental-disorders.
3. Jaehee Cho, Yeongmin Won, Atit Kothari, Stephanie Fawaz, Zixu Ding, and Xu Cheng, "INJUSTICE: Interactive Live Action Virtual Reality Experience," in *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts* (New York: ACM, 2016), 33–37.
4. Yury Gitman, "About Us," *PulseSensor*, n.d., https://pulsesensor.com/pages/about-us.
5. "Wrmhl," n.d., https://github.com/relativty/wrmhl.
6. Laura Parker, "Depressed and Anxious? These Video Games Want to Help," *New York Times*, March 24, 2019, https://www.nytimes.com/2019/03/24/technology/personaltech/depression-anxiety-video-games.html?smid=nytcore-ios-share.

Figure 1: Screenshot of a large cat shooting lasers at players in Team Fortress 2 achievement trap map "achievement_all_v4" from YouTube video "Team Fortress 2 - Laser Death Cat" by user jasonrawr, uploaded June 25, 2010.

Dancing About Architecture:
On Zero Player Level Design

Robert Yang Assistant Arts Professor
New York University Game Center

ABSTRACT

In racing games like *TrackMania* and platforming games like *Super Mario Maker,* users have built complex "Press Forward" and "Auto-Mario" levels that propel the player's game character to perform dazzling feats of acrobatic virtuosity, but with trivial or minimal player input. I argue this type of "zero player level design" complicates typical ideas of gameplay and players: these zero player levels are playful design objects that play with not-playing and emphasize the virtuosity of architectural choreography.

ON LOCAL LEVEL DESIGN

Many contemporary video games feature robust built-in editor tools that let players build new levels without the need for any specialized professional software or hardware. The accessibility and immediacy of these tools often attracts people who do not usually consider themselves to be game designers, and new design patterns often emerge organically out of these casual player-designer communities. These passionate amateurs use level design very differently from the industrial developer's canonical design patterns, constituting a practice that I call "local level design."

Local level design often happens in a very specific context and community to the original game. For example, in 2010, Valve changed how players unlocked items in its first-person multiplayer shooter Team Fortress 2; to earn new upgrades, players suddenly had to grind achievement goals, such as killing a certain number of enemies using a specific weapon. In response, the player community quickly established achievement grinding servers with specially designed achievement farming maps so that players could easily fulfill these achievement goals and acquire these upgrades more quickly than during typical unfocused play. Officially, Valve strongly disapproved of these new achievement servers and maps, arguing that it was tantamount to a cheat or an exploit.

As a rebuttal to this moral crisis, a user named The303 made "achievement_all_v4", a novel achievement trap map where everything seems like a normal achievement farming map for a few minutes, until a giant monstrous invincible cat erupts from the ground and attacks every player with powerful laser beams and cannons. At the end, any surviving players on the entire server are wiped-out via nuclear detonation. The goal of the map was clearly to trick players into thinking they were going to play on an achievement grinding map, but then punish them in a highly visible and humorous way.

Both the achievement grinding map and the ensuing achievement trap are clear examples of local level design as a form of discourse, and in this case,

Figure 2: Screenshot of the player's car driving through typical-looking stock track "D02-Race" in Trackmania Nations Forever, from YouTube video "Free Steam Games - #2 TrackMania Nations Forever (+ Gameplay)" by user TechmsTutorials, uploaded August 4, 2012.

these maps acted as a moral dialogue reflecting on the community's actions. It exemplifies how users frequently invent new ways of understanding the game's core building blocks and assumptions, thus discovering entirely new ways to use the game's design language.

In this essay I explore a form of local level design that has emerged across several different games and genres: what I call the "zero player level" that paradoxically calls for minimal or trivial player input to complete successfully. These levels complicate typical ideas of player agency in games and center the player-designer as an elegant choreographer rather than a wry commentator or skilled performer. This phenomenon intersects with how these communities develop (or debate) a game's design language, how they understand the boundaries of their various design practices, and how the player community maintains its identity.

PRESS FORWARDS IN *TRACKMANIA*

Nadeo's *TrackMania* racing games feature a built-in track editor prominently featured in the game's advertising materials and main menu. To facilitate file sharing, the game client automatically downloads new user-made custom tracks when the player connects to a multiplayer server. A community-run database called *TrackMania* Exchange also lets players upload and archive their track files and serves as a social hub for players to discuss and analyze favorite tracks and building techniques.

TrackMania games usually feature a hundred or more tracks built by Nadeo that gradually increase in length, complexity, and difficulty. Early *TrackMania* games even forced players to race these tracks to earn currency, which they could use to unlock new blocks for the track editor. For new players, these pre-built included tracks establish a design norm of commonly accepted track design patterns, and many custom player-built tracks rely on these stock tracks to tutorial-ize certain skills and driving maneuvers. Later *TrackMania* games have since formalized track design into three game modes / genres: Race (tracks that emphasize competition with other cars), Platform (tracks that emphasize tricky jumps and drops), and Puzzle (tracks where progression and checkpoints are unclear).

Notably, Nadeo does not include the community favorite "press forward" (PF) tracks in its taxonomy. Instead of challenging players to hone their reflexes and wits on the track, the PF beckons the player to simply hold down the "forward" button and watch what happens as a more passive spectator. Through no skill of their own, the player's car executes amazing stunts and maneuvers based on the track's delicate Rube Goldberg-like orchestra of serendipitous aerodynamics -- a car might spin 1080 degrees in the air before barely grazing a ramp in just-the-right-way to land perfectly on the track below. Paradoxically, if the player makes any kind of choice like letting go of the "forward" key, or (god forbid) turning left by 0.1 degrees, any miniscule deviance leads to a disastrous crash.

The only way to fail a PF track is to play it and to make an actual choice. Successfully completing a PF requires the player essentially to give up their agency in the game world. In this sense, it is clear why Nadeo has sought to suppress the press forward. This is a radical design practice that resists the intended mode of playing *TrackMania*. It is a surprisingly existential video game world that basically punishes players for trying to wield any agency or control, and furthermore trivializes the achievements of skilled drivers who race on "normal" tracks. When virtuosity is guaranteed, it is no longer virtuous! The PF strikes at the heart of a local level design community and asks us, what makes a strong player community -- players or designers?

The "play" and "skill" of the press forward is less about its performance, and more about its construction. The most widely acclaimed PFs seem to focus on sheer size with complex level-over-level intersections and unanticipated improvisation of non-standard track pieces. For instance, ThunderClap's PF "Hyperion's Wrath" somehow directs the player's vehicle to hit the track at a strange angle, drive across it seemingly sub-optimally, fall off, spin erratically, skid along a decorative chrome statue, and then land perfectly on a half-pipe. The design goal is to create a sort of uncanny performance that a human player could probably never achieve on a track that is otherwise invisible and illegible to humans. Instead, the track performs itself, and human players are merely its instrument.

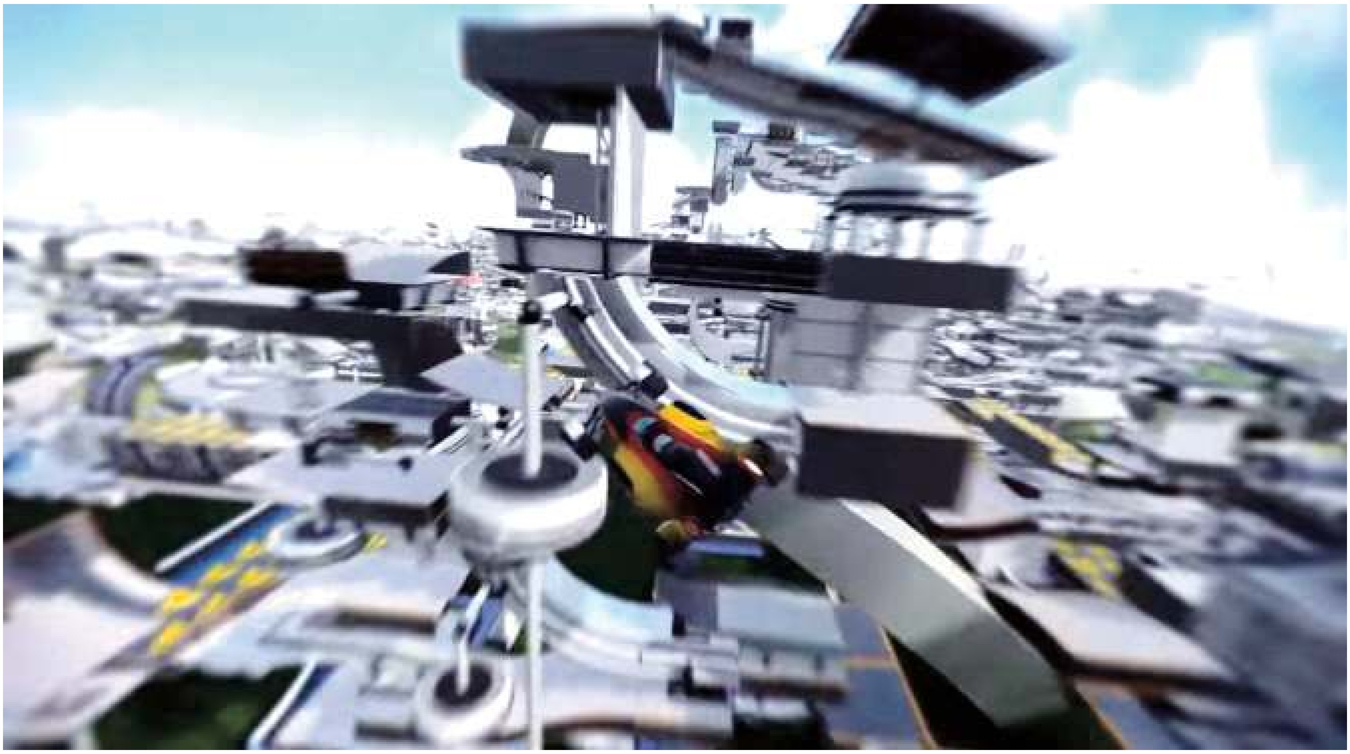


Figure 3: Screenshot of the player's car careening through elaborate PF "Hyperion's Wrath" in Trackmania Nations Forever, from YouTube video "Trackmania [PF] - Hyperion's Wrath | PRESS FORWARD" by user L4Bomb4, uploaded April 18, 2016.

There is a notable variant on the PF: the "press nothing" (PN) which requires players to press absolutely nothing on the keyboard. These track designs accelerate the player's car without any player input, usually with a path of "boost pad" track blocks placed directly after the player start. However, PNs are much less popular than PFs, perhaps because cars gradually lose velocity as they proceed through the track; to maintain sufficient speed, a PN designer must dedicate substantial space to boost pads at regular intervals, which compares unfavorably to the PF's aesthetic of surreal immediacy.

AUTO-MARIOS IN *SUPER MARIO MAKER*

Nintendo's Super Mario Maker series (SMM) features a user community actively managed (sometimes too much or too little, depending on who you ask) by its developers, inviting its users to build new courses using the common building blocks and platformer tropes of the popular Super Mario games. In a big departure from typical Mario games, the SMM games de-emphasize completing a pre-made sequence of levels. While Super Mario Maker 2 does feature a humorous story mode where Mario must rebuild an accidentally demolished castle by completing various "jobs" (pre-made example courses built by Nintendo), these one-off job courses are clearly meant to demo various game mechanics, powerups, and building patterns, and do not feature any coherent storyline or progression like other Super Mario games. SMM 2's main menu button layout echoes this shift away from playing Nintendo's stock courses: the first option at the top is "Course Maker", followed by "Story Mode", "Course World" (to browse other users' levels), and finally "CourseBot" (a utilitarian menu to manage your existing course files and downloads).

The demo courses are just one of Nintendo's attempts to develop a shared design language and/or impose design norms upon the SMM community. SMM2 in particular features a new community "tag" system, where users can attach labels to describe courses. When a course evokes the common platformer gameplay of the Mario series, users are

Figure 4: Screenshot of a user building a simple incomplete track in the Trackmania Nations Forever editor, from YouTube video "Trackmania Nations Forever Editor Tratě [CZ]" by user LuccassCZ, uploaded September 14, 2012.

supposed to tag it as "Standard". Meanwhile, courses about slow methodical deduction with minimal screen scrolling garner a "Puzzle-Solving" tag, and a "Speedrun" tag implies heavy use of quick skillful continuous movement against a timer. Note the similarities to Nadeo's TrackMania track categories (Race, Platform, Puzzle). Also note that some course tags can be merely descriptive, such as the "Short and sweet" tag.

There are two crucial constraints to SMM2's tagging system: (1) a course can have a maximum of two primary tags to be displayed in the main browser menu, and (2) tags are predefined (and localized) by Nintendo, which means users cannot invent their own tags. Instead, grassroots community genres must be labeled directly in the course title, which limits the discoverability of these levels by the rest of the global SMM community.

The tagging limitations predictably lead to heated genre debates within the SMM2 community. What is allowed under a Standard tag, and would a Spanish-speaking player understand this tag differently when it is localized as "Tradicional"? Can a Standard course feature Puzzle-Solving and Speedrun sections as well? What does it mean when Nintendo refuses to bestow official tags upon certain community genres, such as the popular "Kaizo" courses that focus on comically unfair difficulty? Yes, SMM2's tags are a big improvement from SMM1's lack of filters and categorization, but certainly this new form of classification is not without its own problems.

Unlike the relative popularity of PFs / PNs in TrackMania, the SMM community heavily prefers its own variant of the PN, the "Automatic" level, or "Auto-Mario." Auto-Mario courses depart from typical Standard, Puzzle-Solving, or Speedrun frames, and require players to press absolutely nothing on the game controller. While such automatic levels were an unofficial genre devised by the SMM1 community, the existence and use of an official "Auto-Mario" tag in SMM2 now represents an official canonization, as well as hope that Nintendo actually pays attention to community output.

"Keep" courses are the SMM equivalent of the PF track in TrackMania. Keep Walks, Keep Runs, Keep As, Keep Bs, all encourage the player to continually hold right or to jump and continually build-up speed, and the tutorial is right there in the title. These courses are less popular than the typical no-input Auto-Mario course, likely because there are already a wide variety of ways for SMM designers to maintain velocity and to redirect the player -- so requiring the player to hold down a button is not as necessary as in TrackMania.

Auto levels often rely heavily on the trampoline object, which can either push the player left / right or up / down. The most common way to begin the Auto level is to place a trampoline a few tiles above the player's start position, so that when the trampoline falls due to gravity, it instantaneously propels the player to the right. The trampoline's versatility leads many users to design trampoline-themed Auto-Mario courses, like in user So yo mogi's (そよもぎ) course "Bane-darake no zenjidōmario ritānzu" (バネだらけの全自動マリオリターンズ) ("The Return of Auto Mario: Full of Springs") where they use dozens of trampolines to create complex emergent behaviors, like cannons that shoot trampolines that bounce on other trampolines, with a perfectly timed trampoline that pushes Mario narrowly through the split-second gap between the dozen bouncing trampolines.

This interest in complex simulations has culminated in Auto "RNG" (Random Number Generator) courses that players can complete only if a very rare randomly-generated object or interaction occurs, such as with user Phenotype's course "Lucky Draw" which relies on a 1-in-7.5-million chance for a series of magikoopa enemies to all randomly conjure coins instead of any other object. As the course title suggests, Lucky Draw is basically a slot machine built with Mario blocks, and at this time of writing only 35 plays (of 18,000,000+ attempts around the world) have been successful. The only

Figure 5: Screenshot of main menu sidebar with Course Maker button listed before Story Mode in Super Mario Maker 2, from "Super Mario Maker 2 - Gameplay Walkthrough Part 1 - Story Mode and Course World! (Nintendo Switch)" by user ZackScott-Games, uploaded June 28, 2019.

Figure 6: Screenshot of course browsing interface in Super Mario Maker 2 with each thumbnail image, course title, play stats, and tags from "Super Mario Maker 2 - Gameplay Walkthrough Part 1 - Story Mode and Course World! (Nintendo Switch)" by user ZackScottGames, uploaded June 28, 2019.

viable player strategy is to leave the Nintendo Switch console running the course constantly, overnight, as if it were mining for cryptocurrency or training a machine learning network. In this way, the Auto RNG completely negates the typical Mario player's platforming skills in favor of exposing the game engine's machinations. It is a digital brutalism.

DANCING ABOUT ARCHITECTURE

Unlike most levels which center the player's performance, the zero player level's very shape and geometry is the performance. That performance is a dance, a movement in space to a rhythm. Some Auto-Mario levels even double as "Music" tagged courses, which use special music note blocks that can sequentially activate to play a song. In Media Molecule's game Little Big Planet, creators also used the in-game editor to create no-input "music levels" roller coaster rides with long walls of motion-activated music. Metanet Software's indie platformer N also featured fan-made Don't Do Anything (DDA) levels—and Metanet's advertising for the sequel N++ focused on photographs of dancers. Players, player-designers, and developers of these games all seem to relate this activity to music and dance.

Which makes sense, because so much of zero player level design involves embodied intuition. Every *TrackMania* PF designer must reach a minimum fluency with *TrackMania*'s car physics and handling, for how else can they predict how a certain ramp will cause the car to turn at a certain rate and launch at a certain speed? The only way is through trial and error, practice, and patience. Every PF represents at least days of painstaking playtesting and repeated rehearsal—if not weeks, months, or in the case of the PF Hyperion's Wrath, two years of work.

The Super Mario Maker course builder helpfully visualizes the player's jumps and trajectory, helping course creators fine-tune their object placement. This type of debug feature is surprisingly rare for most level editors; common first person

Figure 7: Screenshot of Mario traversing a busy screen full of koopa paratroopas and trampolines in Auto-Mario course “Bane-darake no zenjidōmario ritānzu” in Super Mario Maker 2, from YouTube video “バネだらけの全自動マリオ リターンズ by そよもぎ - Super Mario Maker 2 - No Commentary” by user NinThumbWorldArchive, uploaded July 4, 2019.

Figure 8: Screenshot of Mario trapped within a track beneath three sets of magikoopas and question blocks in the Auto-RNG course “Lucky Draw” in Super Mario Maker 2, from YouTube video “5K special: Super Mario Maker - Lucky Draw - 12F5-0000-03C4-2811” by user Linkums, uploaded May 28, 2019.

shooter level design tools like Hammer or Radiant do not readily visualize dimensions or the player's capabilities, and common industry level design practice places the onus on the level designer to memorize and measure the exact distances (or "metrics") for how far the player can jump or move in the space.

The musician Martin Mull famously argued that "writing about music is like dancing about architecture." I argue that zero player level design is essentially a form of "dancing about architecture" because the game's virtual architecture is performing and activating itself, channeled through the player-designer's phenomenological experience of the game feel.

In level design we often argue that we must learn more from architecture as a field. When industry practitioners invoke this refrain, they usually mean the formal rigor of architectural drafting and planning processes. However, level design already shares workflows and techniques with architectural visualization and CAD practices. The architecture we need in games is the architecture that understands itself as the intersection of social theory, economics, and art. Level design is more than just a 3D blockout for a AAA shooter franchise. Level design is also the history of design trends in *TrackMania* or Super Mario Maker; level design is how casual designers wield casual creator tools and understand themselves as artists; level design is the rich study of how we experience and inhabit spaces. So yes, let's learn more from architecture. See you on the dance floor. →

BIO

Robert Yang makes games about gay culture and intimacy. He is most known for his historical bathroom sex simulator The Tearoom and his male shower simulator Rinse and Repeat. He is currently an Assistant Arts Professor at NYU Game Center, and lives and works in Brooklyn. https://debacle.us/

"There's a long line behind* Grand Theft Auto III. *Forget everything that came before, this was the game that put the PS2 on the map, and made open-world freeform gaming the best kind."

—Bethesda Softwork's Todd Howard[1]

Grand Theft Antecedents

Bobby Schweizer Assistant Professor
Texas Tech University College of Media & Communication

INTRODUCTION

The success of DMA Design's Sony PlayStation 2 release *Grand Theft Auto III* (2001) ushered in an era of major commercial game development focusing on the construction of dense, sophisticated urban environments. Reflecting on the *Grand Theft Auto* franchise, game writer Eric Wolpaw described the draw of this openness: "*[GTA III]* sets you down in the middle of a detailed clockwork world, presents you with a physics model and a wide variety of interesting objects to interact with, and then gives you the freedom to smash them into each other and enjoy the resulting mayhem."[2] The open world urban environment of *GTA III*'s Liberty City greatly differed from the heavily scripted Yokosuka, Japan of *Shenmue* (Sega AM2, 1999) and *Deus Ex*'s (Ion Storm, 2000) dystopian, cyberpunk New York.

Though the openness of *GTA III* appeared radical to consumers at the time, its roots can be seen in earlier city games that established many of its patterns and conventions. In fact, game developers long recognized the opportunities urban environments affords for play: cities have the potential to be multi-genre spaces featuring diverse cultural narratives, where player expectations can be exploited or defied.3 The video game city is defined by the way it draws attention to the experience of moving through its landscape to understand its innerworkings.4 Video game cities afford the operationalization of ***circulating navigable environments***: in these "operational spaces," media scholar Michael Nitsche explains that games "encourage players to engage them, find their own identity in relation to them, develop a history with them, customize them."5

GTA III's innovation regarding operational space is that its open world urban environment provides unexpected outcomes to improvised game navigation, demonstrating how the massive undertaking of creating a "living" open world urban environment could be a commercially viable platform. Successfully outrunning the police through the streets of Liberty City means a *GTA*

III player needs to know police-chasing behaviors, how other vehicles serve as obstacles, and the fastest routes to a Pay and Spray shop to disguise your car with a new paint job. Players who previously focused their efforts on improving their shooting skills soon learn that nothing compares to the benefits of intimately knowing street layouts and how to take advantage of NPC patterns. Contrasting the *GTA III* operational space of *Shenmue*, game critic Gonzalo Frasca described the former as "goal-oriented towards the resolution of a mystery and/or problem," whereas the latter was like a "flight sim."6 As Adam Greenfield said of the conflict between inscribed design and the improvisational way people actually engage with it, "the city is here for you to use."7 *GTA III* players learn to use every pixel and polygon in its irreverently violent world, from browsing the variety of licensed music on in-game radio stations, to interacting with notable voice actors like Michael Madsen, Joe Pantoliano, Debi Mazar, and Michael Rappaport.

This historical survey of video game cities is an inquiry of design impulses, considering how rapidly changing platforms produce new digital urban forms and what forms persist regardless.[8] This history demonstrates the shift from the video game city as a mere backdrop to the creation of polygonal worlds of spatially situated embodiments and algorithmic infrastructures. This essay begins with a brief history of *GTA III* developer DMA Design and details the progression of two-dimensional, beat 'em up side-scrollers and platformers to the navigable, three-dimensional spaces of open world video games. From the perspective of the North American commercial game industry's impact on cultural memory of games, *Grand Theft Auto III* represents the culmination of these historical design impulses and helped usher in the modern era of open-world video game city design.

"A LIVING, BREATHING CITY"

The history of DMA Design—the Dundee, Scotland company that eventually became Rockstar North—is well known amongst developers.[9] Hobbyist computer programmer and high-school drop-out David Jones, along with a few friends, started DMA Design in 1987 and had its first major commercial hit with *Lemmings* in 1991. Emboldened by the sales of *Lemmings*, and, with a two-game contract with Nintendo, DMA purchased office space and impressive computing power to make their next game. The resources were important, wrote David Kushner, because "DMA needed the muscle power (of Silicon Graphics computers) to bring Jones's geekiest dream to life: 'a living, breathing city'."[10] Jones was working on an idea called *Race'n'Chase:* a linear, mission-based game of cops and robbers. Even at this early stage, Mike Dailly (a founding programmer at DMA) had originally imagined *Race'n'Chase* as a third person driving game with the virtual camera positioned behind the player's car.[11] Though the technology existed to accomplish this in the first entry in the franchise, the game's design was more conducive to the top-down perspective.

In the early stages, the concept was a technical challenge for Jones, who wanted to see "how alive and dynamic we could make the city from very little memory and very little processing speed."[12] The game, as senior producer Gary Penn explained in an interview, was influenced by the "open plan structure" of games such as *Syndicate* (Bullfrog, 1993) and *Mercenary* (Novagen, 1985). He described his perspective on the project as "basically *Elite* (Braben and Bell, 1984) in a city."[13] Yet, the project's sophistication led Penn to critique their game as a "fucking simulation," and the studio had to rethink the game's premise.[14] They soon reversed the cops and robbers' roles, allowing players to take illicit contracts, steal cars, and drive recklessly while mowing down pedestrians. The project's name was changed from the innocuous *Race'n'Chase* to the more evocative *Grand Theft Auto* and released for the PC in 1997. Progressing through Liberty City, Vice City, and San Andreas, the player took on a variety of challenges on foot and in automobiles, destroying property, and killing NPCs. The game was a massive commercial success on the PC version, and later for the PlayStation console. This allowed the studio to expand and release two sequels: *Grand Theft Auto: London 1969* and *Grand Theft Auto 2*.

"A LITTLE STREET CLEANING"

The video game city began its life as a backdrop, its setting providing no real significant interaction between player and game space.[15] As the media scholar Edward Dimendberg writes of film noir, the city is a setting that prompts tales of crime, corruption, violence, unrestrained desire, and the externalizations of the anxiety associated with these conflicts. The popular "beat 'em up" game genre referred to the American films of its era when concerns about the decaying urban center, street crime, and drug wars pervaded media.[16] In films such as *The Warriors* (1979), *Class of 1984* (1982), and *Surf Nazis Must Die* (1987), gangs war over territory, juvenile delinquency runs rampant, and violence pervades the American urban environment. Game historian Pierantonio Zanotti explained how *The Warriors* and *Streets of Fire* (1984) figured significantly in early game development because of their popularity with the Japanese game developers that produced early beat 'em up games for the US market.[17] In these early games, the themes of urban crime met the genre conventions of martial arts remediated into video game form.

> ***"Don't let the bully of the block push you around. Fight for your right to walk the street. [...] Spar with the computer bully or with a friend to see who's the toughest on the block."***
>
> **— *Urban Champion* instruction manual**

One of the earliest examples of this mode of development and marketing was *Urban Champion* (Nintendo R&D1, 1984 JP /1986 NA); the brawler player character walks along a street, engaging in a series of one-on-one fist fights while the buildings' residents drop objects from above to show their disapproval. However, the true progenitor of the beat 'em up genre was *Kunio-Kun* (Technos Japan, 1986). Known as *Renegade* in English-language markets, the video game drew obvious inspiration from the structure and visual style of *The Warriors*.[18] The game developer Technos went on to create *Double Dragon* (1987), *River City Ransom* (1989), and *Crash 'n' the Boys: Street Challenge* (1992). The beat 'em up genre most often rendered the city as street and sidewalk in the foreground with the facades of buildings forming a wall in the background, as seen in examples of *Teenage Mutant Ninja Turtles: The Arcade Game* (Konami, 1989), *Final Fight* (Capcom, 1989) and *Streets of Rage* (SEGA-AM7, 1992). This established clear boundaries of action. By representing the stereotypes that characterized media representation of the American city of the 1980s, these early games were able to capitalize on the spectacularized themes of crime and conflict in urban environments.

Batman Returns (Sega CD, Malibu, 1993) took the beat 'em up formula of the previous Batman games and added platforming traversal, where architectural features of buildings—ledges, balconies, and rooftops—serve as platforms for jumping and grappling. In addition, transitions to other evocative locations—barrooms, factories, sewers, and subway stations—provide additional immediate interaction with the architectural features of the environment, but the processes of the city remain dormant.

However, two-dimensional video game genres were not the only game titles to use the city as a thematic backdrop. *Spider-Man*, for the original PlayStation (Neversoft, 2000), tasked the player with swinging and slinging Spider-Man across polygonal 3D rooftops, though they are laid out in such a way that there is only a single path through the level. Down on the ground, city streets provided a frequent backdrop for racing games such as *Metropolis Street Racer* (Bizarre Creations, 2000), which portrayed well-known cities in detail; however, like *Spider-Man*, its courses only traversed specific paths. *Duke Nukem 3D* (3D Realms, 1996) incorporated a minimal number of abstracted, geometric components—grey streets and sidewalks adjacent to tall building facades representing only a few different store fronts—to make its game space feel like a city. Similarly, *Max Payne* (Remedy, 2001) provides a linear experience of New York City, where players are restricted from roaming the space or revisiting locations. Most of the levels in *Max Payne* are based in interior spaces, exteriors only serving as transitions between buildings. The player travels through a labyrinth that joins noir and crime drama locations, from seedy motels, dive bars, noisy clubs, liquor stores, and pawn shops, its contiguous portrayal of moving through the city as a series of game levels.

In considering the video game city as more than a thematic backdrop, game developers had to consider both the technical and experiential challenges of allowing a player character to move around urban streets freely. The structure of the open world city involves the embodied avatar circulating through space, either by foot or vehicle. Rather than proceeding from a starting point to a designated end, players are not confined to a prescribed path through the game space. When executed well, moving across the city becomes as important to gameplay as what happens when you arrive at your destination.[19] Players may repeat journeys and revisit places across the city while game elements—traffic, pedestrians—move about autonomously, maintaining the illusion of a "living" urban world. These provide *spaces of flows* that urban theorist Manuel Castells describes as composed of "purposeful, repetitive, programmable sequences of exchange and interaction between physically disjointed positions held by social actors."[20] The significance of these spaces is not just their physical arrangement, but how interrelated processes enable spatial relationships. The 3D environments of the *Grand Theft Auto* games illustrate the circulating navigable city, by not only presenting a city open for exploration, but also expecting that the player, as well as the traffic, police, and pedestrians that populate the game world, will retrace their paths.

At first blush, this sort of exploration seems like it was only enabled through the advanced computer processing and graphics technology of the late 1990s. However, this same pattern was cleverly used 15 years prior in 2D games like *N.Y.C. The Big Apple* (Synapse Software, 1983), *Turbo Esprit* (Durell Software, 1986) and River City Ransom (Technos Japan, 1989). Though *N.Y.C.: The Big Apple* is a Commodore 64 and Atari 8-bit video game about the experience of being a tourist, those who have played the top-down *Grand Theft Auto* games will recognize the familiar overhead perspective of player vehicles. While driving, players look for specific destinations, where they exit their vehicle to embark on a mission. As a tourist, this means seeking out Manhattan landmarks like Central Park or the Empire State Building and complete mini-games inside of them. The world of this game even has a day/night cycle: players must drive around the city to identify which landmarks are "open" at any given time. But like all tourists, the player must face a set of restraints in pursuit of their comprehensive visit to the city. The game constrains movement through a budget: gas costs money, being towed comes with a hefty fine, and, when struck by a vehicle, the player must pay a significant hospital bill. Through procedural rhetoric, the game made the claim that the New York City tourist experience is not necessarily a pleasant one.

Turbo Esprit also provides an early example of the open world structure from an on-street perspective. The player is placed behind the wheel of a car on the streets of an unnamed city to track down and destroy drug-smuggling automobiles. The game's interface takes a "cockpit" view, with the dashboard and steering wheel taking up half of the screen. The player is represented on the streets as a car that is differently colored from the others—red—with both directions of traffic on screen at the same time. The streets are lined with narrow sidewalks and solid walls of buildings (and some versions, running on more powerful hardware, even show pedestrians on the sidewalks). The current cardinal direction displays along the bottom of the screen, helping the player maintain orientation. By using the controller or keyboard, the player can weave through lanes to out-maneuver traffic while chasing down criminals. The player can even pull up a map of the city that indicates where the target is located—though they must be careful because the game does not pause while it occludes the view of the streets.

A number of other games at this time made use of similar design decisions, both in structure and action. *River City Ransom* (Technos Japan, 1989), in contrast to linear level-based beat 'em ups, experimented with the player circulating through the city, returning to the same places multiple times. Its "town" districts were populated with the kinds of shops one expects to see in the city, their wares tied to the game's character progression system. Non-combative pedestrians fill the spaces of River City to provide contrast to the hostile outlying areas. Much like *N.Y.C.: The Big Apple*, in *Dick Tracy* (Realtime Associates, Inc., 1990) players drive around a city from a top-down perspective to find the entrances of buildings to progress to the side-scrolling action portions of the game. The streets of the game were mostly empty, save for the occasional car passing by or thugs on a rooftop shooting at Tracy; however, the experience of driving around looking for clues while visiting various locations reinforces the idea of a navigable, contiguous city. The legacy of these games, some of which have faded from popular memory, is apparent in the *Grand Theft Auto* franchise. They illustrate the progression of how early video game developers thought about the city not just as a setting, but as the subject of the game itself.

In early attempts at the 3D navigable city, buildings looked a lot like wooden blocks standing on a play mat, as the polygonal, rectilinear shapes of buildings were easier for graphics engines to render. *The Killing Cloud* (Vektor Grafix, 1991) features the player character patrolling San Francisco in a futuristic hoverbike. The visual perspective treats the city not like a corridor of tall buildings, but a vast landscape of distinct structures. Though flying above the poisonous clouds that hang over San Francisco reveals crudely rendered buildings, this depiction makes narrative sense: only certain buildings would be tall enough to be visible as they poke through the gaseous clouds that have blanketed the city. Though most buildings are simple rectangular placeholders, major landmarks were modeled after their real-world counterparts, such as Coit Tower and the Transamerica Pyramid. In addition to lending their specificity to this representation of San Francisco, these buildings serve the original purpose of the landmark as an orienting marker for navigation. Like many of history's most interesting games, *The Killing Cloud* was ambitious, but difficult to play. Similarly, Bethesda Softwork's *The Terminator* (1991) rendered a 3D perspective of Los Angeles. Though the street layout of the area represented—from West Hollywood to Downtown—was relatively accurate, most of the abstracted space is sparse, and with the exception of a handful of landmarks, its buildings remain generic. A few years after *SimCity 2000*'s release, Maxis explored the idea of inhabiting the cities players had built. *SimCopter* (1996) took user-created maps and made them navigable as 3D environments. *Streets of SimCity* (Maxis 1997) made use of the established urban shapes of *SimCity 2000* in a similar way, allowing players to race and crash in polygonal streets a full year before DMA Design released the first *Grand Theft Auto* to PC. The much-maligned *Superman: The New Adventures of Superman* for the Nintendo 64 (Titus Interactive, 1999) allows players to fly through an attempt at a polygonal imagining of Metropolis. Though the city in many of these early 3D games function more as a backdrop than later open world titles, their ambitious proposition of open navigation greatly influenced the specific treatment of open-world games introduced by *GTA III.*

Though inventive game programmers and designers found ways to create unique video game city experiences without hi-resolution, 3D capability, these eventual advancements in computer graphics allowed meaningful differences in the experience of dense, detailed game spaces. DMA Design gained experience designing for these kinds of worlds with their polygonal third-person action game *Body Harvest* (1998) for the Nintendo 64. As moving through a video game space necessitates that world be graphically rendered by the hardware during play, spaces full of elaborate polygons were a challenge for console game hardware in the late 1990s due to memory limitations. The slow load times of optical drives on consoles such as the PlayStation and Dreamcast meant that it was difficult to constantly stream data stored on disk into memory.[21] Even computers with robust graphics cards and processors had hard drive read/write and memory speeds limiting the draw distances of the city. This did not deter game developers but rather challenged them to manipulate technology and establish new design conventions around these constraints.

Driving games proved productive outlets for experimenting with city spaces because its confined gameplay, where the road and camera could be fixed behind the vehicle. For example, while DMA was working on *GTA III*, Angel Studio (eventually Rockstar San Diego) released the open-world racing game *Midtown Madness* (1999). Unlike the traditional track and circuit-based racing games, players are free to drive around expansive spaces of a 3D Chicago populated with traffic, pedestrians, and even changing weather conditions. Building are textured with bitmaps to provide detail. In its sequel, released a year later, players race across even more detailed versions of London and San Francisco. Angel Studios went on to design the open-world city game *Midnight Club Street Racing* (2000), featuring cities based on Los Angeles, Paris, and Tokyo, with landmarks suggesting the urban environments of these real spaces. *Driver* (Reflections Interactive, 1999), another long-running series, also adapted real cities for a variety of vehicular activities. *GameSpot*'s editor Ryan MacDonald described the game as "a 3D *Grand Theft Auto* except you're never on foot."[22] Like *Turbo Espirit* and *Streets of Sim City* before them, these driving were a natural media reflection on automobile-centric culture of American cities.[23]

The technology that made the polygonal 3D era of *Grand Theft Auto* games possible was British developer Criterion Software's RenderWare, a cross-platform 3D graphics rendering engine adopted by many PlayStation 2 developers. Early games like Criterion's own *Burnout* (Criterion Software, 2001) and *City Crisis* (Syscom, 2000) established the viability of RenderWare for building polygonal cities. DMA, under the new leadership of Sam and Dan Houser, explored the capabilities of RenderWare, constructing 3D models, modelling physics, and even scripting behavior.[24] Before it became *Grand Theft Auto, Race 'n' Chase* was modeled in this way as a living city, and DMA considered using 3D for both *GTA* and *GTA 2*.[25] However, by the time the series had morphed into *GTA III*, this simulation was a definitive characteristic of the design. Like urban historian Lewis Mumford's concept of the "Invisible City," *GTA III*'s Liberty City placed buildings and streets as an urban skeleton, but pedestrian movement, traffic patterns, and cycles of day and night were its nervous system; these flows demonstrated how the urban life of the game actually functioned around the player.[26] *GTA III* and subsequent open-world video game urban environments demonstrate how their component systems—everything from player statistics and controller input to programmed behaviors and scripted events—feed back into the circulating systems of code that build the city, making experiencing the invisible city significant for the player.

GRAND THEFT AFTERWARDS

By the time *GTA III* went into development, DMA had changed significantly. Following a series of acquisitions and mergers, BMG Interactive—the London-based office of the German music publishing group that released the first *Grand Theft Auto*—became a publishing subsidiary of a company called Take-Two, and under the leadership of Sam and Dan Houser, was renamed Rockstar. David Jones left the company and took with him many of its programmers. With the advent of the PlayStation 2 and its more advanced processing, graphics, and physics rendering capabilities, DMA licensed Criterion's RenderWare, jettisoned the game engine written in-house, and set about building what became *GTA III*'s open world environment.[27]

Under the production direction of Sam and Dan Houser, *GTA III* would focus more closely on cinematic narrative design than its predecessors. On the surface, the tightly scripted narrative supported by a cast of Hollywood actors seemed at odds with the "freedom" that was a core design tenant of the 2D series.[28] However, its combination of narrative, open world design, and a major marketing push immediately launched the game into financial success. *GTA III* marked the moment that brought the living city to popular consciousness as the first major commercial success in North America that supported open exploration. It occurred at a moment in which a confluence of events—technological, economic, and cultural—made it seem completely novel, despite its many aforementioned antecedents.

After *GTA III*'s successful release, 3D open world cities became an extraordinarily popular setting for video games, influencing a variety of clones that adhered to or strayed significantly from its model. *The Getaway* (SCEE Studio SOHO, 2002), for example, is notable for two reasons: its developers recreated a sizeable portion of central London, unlike GTA III's fictitious Liberty City pastiche of New York.[29] *The Getaway* opted for a more cinematic portrayal of the player character, focusing in a linear story through an open space by reducing the number of non-diegetic user interface (UI) elements. Blinking turn signals on the car are used for navigation instead of a mini-map, health is portrayed by the player's limp, prescribed paths are clearly marked to keep players from getting lost. *Mafia* (Illusion Softworks, 2002) was notable for taking the open-world city structure and implementing rigid laws similar to our own: don't run stop signs, don't draw weapons in public, and do not get caught engaging in criminal activity or risk a severe penalty. These strict rules were interpreted as a downside by many who preferred the lack of consequences afforded by *GTA III*'s permissible mayhem. *True Crime: Streets of L.A.* (Luxoflux, 2003) expanded the scope of the re-created city by mapping their space to a huge street-for-street swath of Los Angeles. *Spider-Man 2* (Treyarch, 2004) introduced an entire city as navigable geometry to web-swing on. While DMA, newly rebranded as Rockstar North, set about quickly iterating on the formula for *Grand Theft Auto: Vice City* (2002) by refining the technology to accommodate navigation of a larger geographic space, their subsequent *Grand Theft Auto: San Andreas* (Rockstar North, 2004) increased the scope of open-world games significantly with its enormous virtual landmass and increasingly believable pedestrian and vehicular flows.[30]

Nobody at DMA Design could have predicted the success of *GTA III* nor its transformation of the video game industry. Producer Gary Penn stated that "the plan was to build an entire city system. A sort of generic city system that we would use for all sorts of different types of things, all sorts of different types of games," but the success of the formula meant that the business homogenized toward one product.[31] *GTA III* served as a blueprint for other game makers building their own circulating navigable cities. Dan Houser explained in an interview that in *GTA III* "there aren't that many single things that you can do in the game that you couldn't do in another game in terms of the actions you do" but what made the game significant was the way that they were linked "in a living 3D world."[32] *GTA III* stands as a testament to how video games have the ability to simulate the flows of a believable, living city in service to a unique, embodied play experience, fundamentally transforming how we approach and produce cultural narratives.

NOTES

1. Michael Thomsen, "The Game Industry Remembers Grand Theft Auto III - IGN," October 17, 2011, https://www.ign.com/articles/2011/10/17/the-game-industry-remembers-grand-theft-auto-iii.
2. Ibid.
3. Zach Whalen, "Cruising in San Andreas: Ludic Space and Urban Aesthetics in Grand Theft Auto," in *The Meaning and Culture of Grand Theft Auto*, ed. Nathan Garrelts (Jefferson, NC: McFarland Press, 2006), 147.
4. Bobby Schweizer, "Moving Through Videogame Cities," *Mediascape* (Fall 2013), http://www.tft.ucla.edu/mediascape/Fall2013_MovingThroughCities.html.
5. Michael Nitsche, *Video Game Spaces: Image, Play, and Structure in 3D Game Worlds* (Cambridge, Mass.: MIT Press, 2008), 195.
6. Gonzalo Frasca, "Sim Sin City - Some Thoughts about Grand Theft Auto 3," *Game Studies 3*, no. 2 (December 2003), http://www.gamestudies.org/0302/frasca/.
7. Adam Greenfield and Nurri Kim, Against the Smart City (The City Is Here for You to Use Book 1 - Kindle Edition (Do projects, 2013), accessed October 8, 2019, https://www.amazon.com/Against-smart-city-here-Book-ebook/dp/B00FHQ5DBS.
8. Author's note: this is explicitly not a story of progress and causality and, though it attempts to catalog a variety of videogame cities by conventions, there will always be games that are overlooked or missed.
9. See, in particular, David Kushner's *Jacked: The Unauthorized behind-the-Scenes Story of Grand Theft Auto* (HarperCollins Publishers, 2012).
10. Ibid., 25.
11. Nostalgia Nerd, "From DMA to GTA: The Story of DMA Design - Nostalgia Nerd," accessed October 8, 2019, http://www.nostalgianerd.com/from-dma-to-gta-the-story-of-dma-design.
12. Kushner, *Jacked, 25*.
13. Tristain Donovan, "The Replay Interviews: Gary Penn," *Gamasutra* (blog), January 31, 2011, https://www.gamasutra.com/view/feature/134644/the_replay_interviews_gary_penn.php?print=1.
14. Kushner, *Jacked*.
15. Georgia Leigh McGregor, "Situations of Play: Patterns of Spatial Use in Videogames," in *Situated Play, Proceedings of DiGRA 2007 Conference,* 2007.
16. Pierantonio Zanotti, "Playing the (International) Movie: Intermediality and the Appropriation of Symbolic Capital in Final Fight and the Beat 'em up Genre," *Eludamos. Journal for Computer Game Culture 9*, no. 1 (September 19, 2018): 51.
17. Pierantonio Zanotti, *The Warriors* and *Streets of Fire* (1984), 53–54.
18. Zanotti, "Playing the (International) Movie."
19. Frasca, "Sim Sin City—Some Thoughts about Grand Theft Auto 3."
20. Manuel Castells, "Flows, Networks, Identities," in *Critical Education in the New Information Age* (Lanham: Rowman & Littlefield Publishers, 1999), 57.

21. Andrew Williams, *History of Digital Games: Developments in Art, Design and Interaction* (CRC Press, 2017), 203.
22. Ryan MacDonald, "Driver Review," *GameSpot*, July 9, 1999, http://www.gamespot.com/reviews/driver-review/1900-2545944/.
23. Schweizer, "Moving Through Videogame Cities"; Mimi Sheller and John Urry, "The City and the Car," in *The City Cultures Reader*, ed. Malcolm Miles, Tim Hall, and Iain Borden (London: Routledge, 2004), 202–19.
24. Kushner, *Jacked*, 84.
25. Donovan, "The Replay Interviews: Gary Penn."
26. Lewis Mumford, *The City in History: Its Origins, Its Transformations, and Its Prospects* (New York City, NY: Harcourt Brace Jovanovich, 1961), 563.
27. Donovan, "The Replay Interviews: Gary Penn."
28. Kushner, *Jacked*, 34.
29. Ian Bogost and Dan Klainbaum, "Experiencing Place in Los Santos and Vice City," in *The Meaning and Culture of Grand Theft Auto*, ed. Nathan Garrelts (Jefferson, NC: McFarland Press, 2006), 162.
30. Kushner, 114.
31. Donovan, "The Replay Interviews: Gary Penn."
32. Douglass C. Perry, "Rockstar's Sam Houser Mouths Off," IGN (blog), September 10, 2001, https://www.ign.com/articles/2001/09/10/rockstars-sam-houser-mouths-off.

Using Pattern Language in Learning and Practicing Game Design

Christopher Barney Lecturer
Northeastern University

ABSTRACT

In order to better instruct students in a variety of game design courses, I have employed the structure of a pattern language as conceived of by architect Christopher Alexander. To better use his ideas both in the classroom setting and in professional game design, I have developed a number of concrete exercises to derive patterns, link them together into a pattern language, and use the resulting language to execute practical design tasks. This paper describes these techniques.

INTRODUCTION

As professional game designers, we face the problem of organizing our understanding of the art and science of game design in a way that will both allow us to execute on our design intentions, and also communicate them to our colleagues. As instructors, we face the problem of imparting both our understanding of specific game design techniques, and the more generalizable skills of understanding how those techniques were arrived at, how they can be effectively applied and how they can be extended into new areas of design that are not covered in our time-limited curricula. The concept of a pattern language, famously explored by Christopher Alexander in his 1977 architecture and urban planning text *A Pattern Language*, provides one solution to this problem. Alexander states:

> *Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.*[1]

Alexander's architectural concept of pattern languages poses patterns as physical designs responding to human relationships. The appeal of this idea has spread to disciplines outside of architecture, most notably into the field of computer science in the form of programming patterns.[2] Staffan Bjork and Jussi Holopainen attempt to apply pattern thinking to game design in their book *Patterns in Game Design*.[3] However, while this book provides a useful set of patterns, it does not link them together into a language or focus on what problems each pattern existed to solve.

In order to be useful as a design tool and as a pedagogical approach, both the connectedness and purpose of a pattern language are necessary. I decided to structure a course I was scheduled to teach at Northeastern University around these ideas. Each week of the course, I require students to complete one or more pattern derivation exercises based on the ideas presented in the course textbook[4] and in lecture, and then to apply those patterns in practical exercises assigned as homework. The rest of this paper will look at a selection of these exercises, the patterns they generated, and the assignments in detail.

An Overview of the Pattern Derivation, Sample Pattern, and Design Process

PATTERN EXERCISES

Alexander, with his colleagues and students, took many years to generate the 253 patterns found in A Pattern Language. It is likely that a similarly exhaustive effort will be required before anything resembling a stable defensible language for game design can be asserted. However, for the purpose of an individual designer or a student, a more modest process can yield useful results.

The first exercise that I present to students is the derivation of a high-level pattern. I present the students with the following 4 questions:

1. *Name a formal game design element.*
2. *Name 10 games that use that element. The more differently each uses it, the better.*
3. *What design problems do those games use the element to solve?*
4. *Is there a pattern in the way the elements are used?*

The students must write up the pattern that they discover using a "pattern template" that will be described in the following section. The following is a sample of a student's work for this exercise:

Question 1: *Name a formal game design element.*

Life Force/Health

Question 2: *Name 10 games that use that element.*

Half-Life 2, Team Fortress 2, Super Metroid, Borderlands 2, Super Smash Bros. Ultimate, Super Mario World, Pokémon Blue, Tekken, Dragon Ball Fighter Z, Halo

Question 3: *What design problems do those games use the element to solve?*

Players require barriers that must be overcome in order for victory to hold meaning, so having health or a way of measuring how many mistakes you can make is necessary. Giving players a life force of some type provides this barrier.

Question 4: *Is there a pattern in the way the elements are used?*

Health is a resource associated with a specific entity, and if that is the player, they are prevented from progressing unless the resource is recovered.

SAMPLE STUDENT PATTERN

This work was then formatted into a pattern template generating the following formal pattern:

Pattern Name: Life Force

Authors: Eric Crawford, Peter Manning, Matteus van der Wilden, Namhoon Kim

Design Problem: In a game without the possibility of failure, objectives often feel meaningless and uninteresting.

Pattern Description: Players are given a resource that represents how close they are to failure.

Pattern Seed: Many games have health systems

Child Patterns: Flexible Health Systems

Games that use this pattern and how:

Half-Life 2: Players have a health bar and an energy bar, when both are empty players lose

Team Fortress 2: Players have a health bar that can be overfilled and slowly ticks down to base health given enough time. This is called "overheal."

Super Metroid: Players have energy tanks that can be increased throughout the game

Borderlands 2: Players have a health bar that refills once they die, but they must get a kill to stop it from rapidly draining in this state.

Super Smash Bros. Ultimate: Players have a % that determines how far they get thrown when hit, higher % is worse for survivability.

Super Mario World: Player gains items that allow them to take additional hits and have additional abilities.

Pokémon Blue: Players have teams of up to 6 Pokémon, when all lose health the player loses.

Tekken: Players have a health bar. When it reaches zero, they die.

Dragon Ball Fighter Z: Players take special blue damage that heals over time unless the enemy can break it, giving players additional chances after normally game-ending hits

Halo: Limited health with regenerating shields.

In the preceding example, the student goes from a high-level design observation ("Many games have health systems") to a useful, if widely understood game design principle ("Players are given a resource that represents how close they are to failure"). This descriptive response structure is common. I ask students to reword pattern descriptions to be prescriptive as discussed in the "Common Problems" section below.

Many of the initial patterns derived during this exercise can be seen as painfully obvious or simplistic by experienced designers, but it is important to note that these students have not been taught these basic principles of design: here, they have observed them, understood what they have seen, and articulated their observations clearly. As the class progresses and more difficult exercises are presented to the students, the complexity and nuance of the patterns they generate increases, as will be seen in the example patterns section.

The last two lines of this first sample pattern are also important. After a significant number of patterns have been generated by a class, I begin to require students to examine the collection of patterns and identify any patterns that might be connected to theirs. There are several types of possible connections. The first, referred to as *parent patterns* in the above exercise, are either more general, higher level versions of the current pattern, or are patterns required for the current pattern to function optimally. The last line of the example pattern— "Child pattern"— is the opposite: patterns suggested by the current pattern, or smaller, more granular patterns that are necessitated by or benefit the current pattern. Not shown in the example are *sibling* or *related patterns*. These are patterns that share parents or children with the current pattern and may complement it but sit at the same level of granularity within the larger collection.

It is these connections that make a pattern language rather than just a collection of patterns. As Alexander describes, when a sufficiently rich set of interconnected patterns is present, a designer may identify the set of problems that they wish to solve with their design, then identify the subset of patterns suited to provide solutions to those problems, and thus begin actual design ideation. This process builds a framework allowing the designer to assess the chances that any combination of architectural components, mechanics, narrative elements, and aesthetic choices have of functioning coherently as a game capable of producing the intended experience in its players.

The scene design that resulted from the application of this pattern is included in the "SAMPLE STUDENT DESIGN" section at the end of this paper.

DESIGN PROCESS

For students or designers to clearly understand whether the patterns they are generating are useful, they must attempt to apply them. Each week in my architecture and level design courses, I assign students a practical scene implementation project based on the contents of that week's reading and lecture, but they must apply the pattern they derived that week in their design. In my Exploratory Game Design course, students work on four major projects as teams over the course of the semester, but I use pattern derivation exercises at the beginning of each section of the course, and they must apply the patterns they derive in each project.

The process of constructing a pattern exercise that will result in students deriving patterns that relate well to the intended topics of a course is not trivial, but it is well within the reasonable capacity of an experienced design instructor. I have included a selection of different pattern exercises in the "ADDITIONAL PATTERN EXERCISES" section of this paper and a selection of the resulting patterns in the "STUDENT PATTERN EXAMPLES" section.

COMMON PROBLEMS

Students' early attempts at stating the patterns they see are often poorly formed. Many students state their pattern circularly, and merely reword the initial design problem they are investigating. For example:

> *"In order to create horror in a game, a designer may include horrific elements."*

In these cases, I instruct students to be more specific and enumerate those elements and how they are used, then to look for patterns in the detailed descriptions they have generated.

Another common error is to state a pattern as a passive description rather than a prescriptive course of action. In the case of the sample pattern above, "Players are given a resource that represents how close they are to failure" should be revised to "In order to make game objectives meaningful and interesting designers may be given a resource that represents how close they are to failure."

ADDITIONAL PATTERN EXERCISES

A. Formal and Functional Patterns

In an early class in the Game Design and Architecture course, I describe two aspects of game design as being its formal and functional elements. Formal elements being the things in a game, or its nouns; the functional elements are the mechanics of a game, or its verbs. These two patterns exercises help students to apply that idea:

A Formal Pattern

1. *Name a formal game design element.*
2. *Name 10 games that use that element. The more different ways they use it the better.*
3. *What design problems do those games use the element to solve?*
4. *Is there a pattern in the way the elements are used?*

A Functional Pattern

1. *Name a functional game design element.*
2. *Name 10 games that use that element. The more different ways they use it the better.*
3. *What design problems do those games use the element to solve?*
4. *Is there a pattern in the way the elements are used?*

The challenge with this exercise is that any given element may exist in one game to solve several design problems, or in several games to solve different problems. Generating a list of only 10 games makes seeing the totality of patterns related to an element difficult. However, strong patterns are often evident with this small sample set, making it appropriate for students. Experienced designers may find examining much larger sets of games productive.

B. Higher and Lower Level Pattern Exercise

The first exercise attempts to generate a pattern at a higher level of abstraction or generality. The second looks for lower level patterns suggested by a previous pattern.

A High Level Pattern

1. *Name a high level design element.*
2. *What problem does that design element solve?*
3. *Name 10 games that also solve that problem. The more different ways they solve it the better.*
4. *For each game describe their solution.*
5. *What do those solutions have in common?*
6. *Is there a pattern?*

A Lower Level Pattern

1. *Pick one of the patterns in the previous two exercises. If you were making a game with that pattern, what problems/questions would you have?*
2. *List 10 games that have solved that problem.*
3. *For each game describe their solution.*
4. *What do those solutions have in common?*
5. *Is there a pattern?*

Targeted Pattern Exercises

Later in pattern-based courses, I introduce exercises intended to create patterns that explore a specific topic such as Emotional Patterns, Theme Patterns, or Boss Fight Patterns.

An Emotional Pattern

1. *Pick an emotional effect.*
2. *Think of 10 other games that create that emotional effect.*
3. *For each game list and describe the techniques used.*
4. *Is there a pattern?*

A Theme Pattern

1. *Pick a theme.*
2. *Select 10 games that feature that theme.*
3. *For each game list and describe the techniques used to support or create their theme.*
4. *Is there a pattern?*

A Boss Pattern

1. *Pick a game boss encounter from a game*

you know well.
2. *Describe the player experience that encounter creates.*
3. *Think of 10 other games that create that effect with a boss encounter.*
4. *For each game, list and describe the techniques used to create that effect.*
5. *Is there a pattern?*

As you can see, the format of these prompts are similar, and generating new prompts is trivial. Creating prompts that generate the insights you are looking for consistently is more challenging.

STUDENT PATTERN EXAMPLES

The following examples were created by students at Northeastern University during the Spring 2018, Fall 2018, and Spring 2019 semesters in the Exploratory Game Design, Spatial and Temporal Design, and Architecture and Level Design courses. They are reproduced here unaltered. They are imperfect and not intended to represent "valid" or "correct" patterns that should be adopted for use by the larger game design community. I include them as examples of how the process of following the above pattern generation exercises can cause students to recognize and articulate various principles of game design.

FORMAL PATTERN

Pattern Name: Temporally available move space
Author: Zhihui Chang
Design Problem: How to increase the potential available spaces for players?
Pattern Description: The temporally available move space has different functionalities. For a moving platform, players have to jump on it within a regular timing period, which increases the difficulty and practices players' skills. For a moving guard with limited view, a rock that blocks the view is an ideal hidden place for players to avoid being caught, which allows players to guide their moving path accordingly. Also, as these spaces are temporally available to players, the shift changes players' perception of space, making the game play more compelling. Also, there are temporally unavailable move space, which is the opposite side of temporally available moves pace, like space in front of a moving bullet. It has the same effect to limit players' move space and express useful information.
Games that use this pattern and how:
- ***Assassin's Creed:*** Dynamic enemy movement creates a constantly shifting set of available space for undetected character movement.
- ***Thief:*** See Above
- ***Dishonored:*** See Above
- ***Super Mario Brothers:*** Moving platforms create temporary paths for player traversal. This is seen in almost all platformer games.
- ***The Secret World:*** Turret and enemy placements that cause instant death or fail states create complex spaces the player must decode in order to traverse.

FUNCTIONAL PATTERN

Pattern Name: Different Play styles
Author: Eric Crawford
Design Problem: Different users have different approaches to game play challenges.
Pattern Description: Provide tools (when appropriate) to allow for various approaches to game play challenges.
Games that use this pattern and how:
- ***Octopath Traveler:*** 8 playable characters, of which the player chooses 4 to use at a time
- ***Team Fortress 2:*** 9 different classes with specific team roles, each with dozens of weapons that allow players to tailor them to their play style
- ***Skyrim:*** Different races, classes, and types of weapons
- ***Pokémon:*** Hundreds of Pokémon to choose from for your team of 6

Zelda: Breath of the Wild: Different weapons, gear, and progression paths to suit everyone

Overwatch: Many different heroes that fit into three categories

Splatoon 2: Distinct weapon types that have different play styles, with multiple types of each of these weapons

Borderlands 2: Multiple classes with distinct roles, randomly generated weapons and mix-and-match grenades and shields

Mario Kart 8 Deluxe: Different characters with different weights, different car parts to choose

Rubble Trouble: Given a variety of tools at the start of each stage, not all will be used

Pattern Seed: Many games different playable characters

Parent Patterns: Unique Experiences

Child Patterns: Balancing to Inspire Strategies, The Game Must Offer Autonomy, Combos Within Combos, Transhumanism as Game play.

EMOTIONAL PATTERN

Pattern Name: Engagement through Sorrow

Authors: Justin Dunn, Savion Mercedes, Scott Thompson, Ian Wolff

Design Problem: Players need a way to become emotionally invested and engaged in the game's narrative

Pattern Description: Somber or sad experiences that might negatively remove or change an aspect of the narrative so that the character and/or the player must reflect upon and connect to what has happened thus far.

Games that use this pattern and how:

The Last of Us: Kill off the Protagonist's daughter

The Walking Dead: Telltale Series: The death of Lee (the main character of the game)

Hellblade: Senua's Sacrifice: Protagonist's mother and boyfriend are both killed

Metal Gear Solid 3: Having to kill villains with sad backstories. Some villains turn out to not be villains for added sadness

Pokémon Mystery Dungeon: Explorers of Time: Entire Main story end game is a suicide mission and the player is made very aware of it, and then has to watch the results of that

Mass Effect 3: Several major characters sacrifice themselves.

Persona 4: Players spends 7+ months in game with the protagonist's family, and then one is hospitalized and the other is killed.

Nier Automata: Protagonist learns of upsetting truths and engages in somber actions

A Way Out: Players are forced to actively fight each other and be the cause of the other's downfall and death

Child Patterns: My Battery is Low, and it's Getting Dark.

SAMPLE STUDENT DESIGN: ***Simple Platforming Section***

For our first group project, we created a simple section of a 3D platforming level. The scene consists of a hexagonal room with several irregularly shaped pillars arranged in a spiral around a central platform. The bright color of the platforms in the center stands out against the dark colors of the walls, and the shapes being placed around a center gives the room a sense of rhythm and balance. The platforms themselves have a repeating pattern of the number of sides they have; 5-6-5-6-5. This also echoes the shape of the room. The pattern of vertical pillars is further accentuated with a few more vertical pillars in the corners of the room.

A hallway leads into the room at ground level, and a second one leads away from the upper platform. This entrance hallway is designed to lead the players' eyes to the center platform through the use of line and shape, and the rise of the platforms also draws attention to the upper exit.

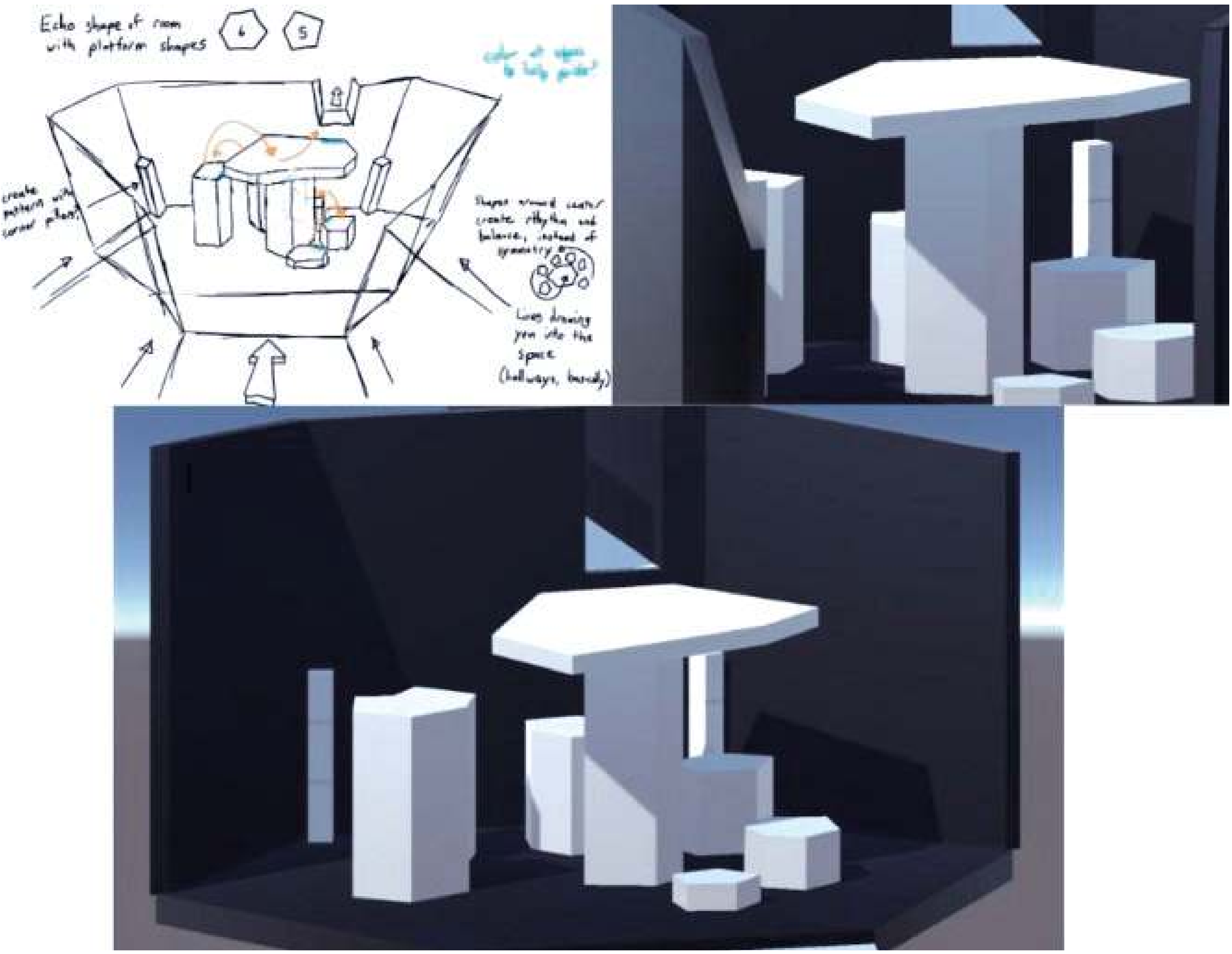
Our theoretical game has fall damage, but only from falling from high enough. In this room, only the last two pillars and the final platform would be high enough to cause damage by falling off of them. This is to allow players to get used to the idea of jumping between platforms, and the shape of the room and nature of the pillars allows the player to simply circle around and try again if they fall.

CONCLUSION

Using pattern derivation and language construction as a pedagogical framework for teaching game design has made two things clear. First, that creating a functionally complete pattern language is beyond the scope of a single course—probably beyond the scope of the life's work of a single designer. Second, even a partial language proves useful in generating good design. The process of pattern derivation yielded good results as an instructional exercise. Students produced patterns that related to the intended basic design concepts taught by the course. The patterns created showed that students understood not just the techniques they were describing, but the design principles that gave rise to them, and further, the reasons behind those principles. Students understood the why of design, not just the how.

A more complete discussion of pattern-based design and learning process is forthcoming from CRC Press under the title Pattern Language for Game Design. It includes a robust set of pattern derivation exercises. Publication is expected in the spring of 2021.

Figure 1: Design sketch and white box models



NOTES

1. Christopher Alexander, et al., "A Pattern Language," in A Pattern Language (Oxford: Oxford University Press, 1977), 1–1171.
2. Erich Gamma et al., "Design Patterns: Elements of Reusable Object-Oriented Software," in Design Patterns: Elements of Reusable Object-Oriented Software (Boston: Addison-Wesley Professional, 1995), 1–395.
3. Staffan Bjork and Jussi Holopainen, "Patterns in Game Design," in Patterns in Game Design (Boston: Charles River Media, 2004), 1–423.
4. Christopher Totten, "An Architectural Approach to Level Design," in An Architectural Approach to Level Design (Boca Raton, FL: A K Peters/CRC Press, 2014).

ACKNOWLEDGEMENTS

Christopher Alexander, without whom none of this could exist.

All of the students who worked so hard to refine this process and shaped it through their insightful feedback.

Jason Wiser, who developed the initial course materials for the Spatial and Temporal Design course that these techniques were first applied to.

Christopher Totten, whose encouragement to share this process directly resulted in this paper.

J Li for her work on Pattern Language for LARP Design, available at http://www.larppatterns.org. Her work was an early inspiration for my interest in using patterns in game design.

Jessie Schell for his talk, "The Nature of Order in Game Narrative," given at GDC 2018 and available at https://www.gdcvault.com/play/1025006/The-Nature-of-Order-in.