Resynthesizing Reality: Driving Vivid Virtual Environments from Sensor Networks

by

Don Derek Haddad

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning
in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2018

Author .... Signature redacted ....
Program in Media Arts and Sciences, School of Architecture and Planning
November 14, 2017

Certified by.. Signature redacted ....
Joseph A. Paradiso
Alexander W Dreyfoos (1954) Professor
Program in Media Arts and Sciences
Thesis Supervisor

Accepted by .... Signature redacted ....
Pattie Maes
Academic Head
Program in Media Arts and Sciences

Resynthesizing Reality: Driving Vivid Virtual Environments from Sensor Networks

by

Don Derek Haddad

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning
on November 14, 2017, in partial fulfillment of the
requirements for the degree of
Master of Science in Media Arts and Sciences

Abstract

The rise of ubiquitous sensing enables the harvesting of massive amounts of data from the physical world. This data is often used to drive the behavior of devices, but when presented to users, it is most commonly visualized quantitatively, as graphs and charts. Another approach for the representation of sensor network data presents the data within a rich, virtual environment. This thesis introduces the concept of *Resynthesizing Reality* through the construction of *Doppelmarsh*, the virtual counterpart of a real marsh located in Plymouth Massachusetts, where the Responsive Environments Group has deployed and maintained a network of environmental sensors. By freely exploring such environments, users gain a vivid, multi-modal, and experiential perspective into large, multi-dimensional datasets. We present a variety of approaches to manifesting data in "avatar landscape", including landscapes generated off live video, tinting frames in correspondence with temperature, or representing sensor history in the appearance and behavior of animals. The concept of virtual lenses is also introduced, which makes it easy to dynamically switch sensor-to-reality mapping from within virtual environments. In this thesis, we describe the implementation and design of *Doppelmarsh*, present techniques to visualize sensor data within virtual environments, and discuss potential applications for *Resynthesizing Reality.*

Thesis Supervisor: Joseph A. Paradiso
Title: Alexander W Dreyfoos (1954) Professor
Program in Media Arts and Sciences

This masters thesis has been examined by the following reader.

Professor Glorianna Davenport Signature redacted

Thesis reader

Director of the Living Observatory (LO).

This masters thesis has been examined by the following reader.

Signature redacted

Professor Ken Perlin..

Thesis reader
Professor of Computer Science,
Director of the NYU Media Research Lab

Acknowledgments

This work was brewed with love and with the help and support of the following individuals whom I'd like to thank in this note:

To my advisor, Professor **Joe Paradiso**, for his trust and humble spirit. For his exquisite taste in music that influences my sonic exploration. To my thesis reader previously director of the Interactive Cinema group at the Media Lab, Professor **Gloarianna Davenport**, for her energy and detail oriented mind. To my thesis reader Professor **Ken Perlin**, for his creative critical mind that came up with the name Resynthesizing Reality. For his generous, friendly and humble spirit.

To my mother **Lida Simonides** and to my grandfather **Elia Socrates Simonides**, for their love and iron will. Thank you for motivating me all along. To my family members, my sister **Bruna**, my brother **Ryan** and father **Bechara Haddad**. Thank you for your support and love.

To students and alumni from the Responsive Environments Group: **Gershon Dublon**, **Brian Mayton**, **Spencer Russell**, **Clement Duhart**, **Jie Qie**, **Evan Lynch**, **Artem Dementyev**, **Pragun Goyal**, **Vasant Ramasubramanian**, **Nan Zhao**, **Nan-Wei Gong**, **Irmandy Wicaksono** and **Amna Carreiro**. Thank you, your presence and support did a big impact on my life. To **Mark Feldmeier**, for his mentorship, wisdom and hacktivist mind.

To friends who helped me craft and proof read this document: **Mike Aoun**, for his scientific personality and critical mind. **Elie Mahfouz** for his eagle eye, and wisdom. **Holly Haney**, for her kind helpful energy. To friends who helped me with the user study: **Joe Francis** and **Allan Kwan**. Thank you fellow gamers.

To **Marvin Minsky**, to his wife **Gloria Rudisch Minsky** and beautiful family, thank you for your support, love, and welcoming energy. To **Edith Ackerman**, for her genuine smile, for her imagination, and bricoleur spirit.

To **Xiao Xiao** and to her family, for their love, support and kindness.

Contents

List of Figures

Chapter 1

Introduction

"You see, Dad, Professor McLuhan says the environment that man creates becomes his medium for defining his role in it. The invention of type created linear, or sequential, thought, separating thought from action. Now, with TV and folk singing, thought and action are closer and social involvement is greater. We again live in a village. Get it?"

Drawing by Alan Dunn, © 1966 The New Yorker Magazine, Inc.

Figure 1-1: A cartoon from the New Yorker Magazine featuring McLuhan's philosophy on the future of media.

In the digital age, we live not one, but many parallel lives. While the physical world continues to ground our daily life, a multitude of virtual worlds increasingly occupy our minds. From websites to video games, these worlds range from predominantly 2D interfaces to massively multi-player 3D fantasy environments. These spaces continue to emulate the senses with the goal of invoking "true" sensations in the player's body and mind; continuing and extending effects that have been explored over the centuries in theater, and for the past 100 years in film. Happiness, boredom, anger, rage; With immersive graphical and audio-computer mediated environments, these feelings/emotions become ever more real because the player/explorer is summoned and must act. As Marshall McLuhan says in his masterpiece "Understanding media: the Extension of Man" [42] – The game only begins when the player consents to becoming a puppet for an intended amount of time . With this invisible agreement signed, the player is ready for the experience.

1.1 Motivation

"We usually say that one must first understand simpler things. But what if feelings and viewpoints are the simpler things?" [44] - Marvin Minsky

Virtual environments are like prosthetics for the imagination. They hijack one's auditory and visual apparatus and provide an alternative to our physical environment with a resolution determined by technology. The era of ubiquitous sensing provides new digital pipelines, bringing us few steps closer to entering distant, evocative environments. Ongoing research at the Responsive Environments group includes state of the art sensor technologies to augment and mediate perceptual experiences in physical environments as well as in virtual worlds. The group has developed *Cross Reality* [33] exploring the link between the physical and the virtual. Within *Cross Reality*, sensor networks can "tunnel" dense real-world information into virtual worlds, and interactions flowing from within these virtual worlds can incarnate back into the physical through displays and actuators. This work on *Resynthesizing Reality* extends the concept of *Cross Reality* by examining how virtual explorations of rich environments can give the user a qualitative sense for complex streams of both live and cached sensor data by engaging humans' innate abilities and desires to interpret sensory cues of the physical world in a resynthesized environment. Our sandbox for *Resynthesizing Reality* is called Doppelmarsh, a virtual clone of a large-scale wetland restoration site in Plymouth, Massachusetts called Tidmarsh. Environmental changes at Tidmarsh are tracked over time by dozens of sensor nodes deployed across its *Networked Sensory Landscape* [41]. The data generated from these sensors are used to drive the appearance of Doppelmarsh from the articulation of its weather to the evolution of its animals.

Figure 1-2: The virtual world of Doppelmarsh mirrors the physical world of Tidmarsh.

1.2 Thesis Goal

This thesis presents the concept of *Resynthesizing Reality* which inherits core ideas from *Cross Reality* such as mediating the physical world and the virtual via sensor networks, with adding major modifications on the creation of such environments and the art that help convey this illusion of life. We propose to limit the scope in which the data is being mapped while emphasizing *Telepresence* [45] through well crafted virtual environments that contain visual cues. If taken too far, the art of mapping can sometimes "dilute" the meaning behind the numbers, resulting in an artistic and expressive yet obscure transmogrification of the data. A “resynthesized” virtual environment will resemble its real counterpart perceptually, cognitively, and emotionally. This thesis describes current and evolving techniques used to build virtual replicas of real existing sites, while animating elements of their weather and their wildlife. Finally, the core of this work extends concepts of enhanced and ubiquitous *Presence* [15], using game design elements in non-gaming contexts.

1.3 Chapter Summaries

Chapter 1: Introduces the subject from a broader perspective, while laying out the outline of this thesis.

Chapter 2: Introduces the Tidmarsh Living Observatory community while expanding on the background, the literature, and the related work, Serious Games, *Cross Reality*, Skeuomorphism, and Computer Mediated Reality.

Chapter 3: Covers the synthesis of real outdoor environments into virtual environments using the Unity game engine [17] extending prior work done on *Cross Reality* building, "Doppellab" [16], while featuring the construction of the avatar wetland "Doppelmarsh" and its progress since 2014.

Chapter 4: Covers the implementation of dynamically synthesized virtual environments, driven by real-time sensor data. This chapter also, presents "MiddleMarsh", a middleware server that connects to Chain-API [62], and many other APIs to conjure and cache frames used to drive the weather conditions of Doppelmarsh.

Chapter 5: Covers various interfaces used to aid human interactors/explorers as they traverse the virtual terrain in Doppelmarsh.

Chapter 6: Covers the evaluation of the assumptions related to mapping the weather in Doppelmarsh using both a user study, and a visual approach.

Chapter 7: Concludes this thesis with a summary listing the main contributions presented in this work.

Chapter 2

Background and Literature

"Our purpose is to imagine and design learning environments in which the children themselves can imagine and design." - Edith Ackerman

2.1 Introduction

This chapter begins with an introduction to Tidmarsh Farms[1], the mission of the Living Observatory (LO) and the contribution of the Responsive Environments group (ResEnv) to the endeavor. We then introduce Doppelmarsh as a virtual clone of the Tidmarsh Farms landscape. Doppelmarsh is a successor to previous experiments implemented by students of ResEnv, namely Doppellab and ShadowLab. Both of these virtual environments are characterized in the ResEnv literature as "Cross Reality" [32] [16]. We then describe the traditional and older graphical techniques used to visualize network diagrams, as well as a range of contemporary examples spread across several industries. Core concepts in the proposed work converge to a field of research known as *Serious Games.* Section 2.2.4 covers a brief history and the present situation of *Serious Games.* The last section in this chapter talks about *Computer Mediated Reality* and its connection to both *Cross Reality* and to *Resynthesizing Reality.*

---

[1] LivingObservatory (LO) seeks to tell the long-term story of the Tidmarsh Farms Wetland Restoration and to advance scientific knowledge and public understanding of wetland ecology. Founded in 2011 by Glorianna Davenport, Alex Hackman and Hyun Yeul Lee, LO was incorporated and became a charitable 501(c)3 organization in 2016. Website: http://livingobservatory.org

2.1.1 Background

Figure 2-1: Picture taken pre-restoration (picture on the left) and post-restoration (picture on the right) from the same location on Tidmarsh. Such a visual indicator shows how nature started to heal itself. Photo-credit: AHackman.

Doppelmarsh resynthesizes areas of Tidmarsh Farms, a 610 acre former cranberry farm in Plymouth Massachusetts. In 1989 this farm produced 1% of Ocean Spray's harvest. In 2010, the land owners, with the help of federal and state agencies, committed to transition the cranberry bogs of the farm to more natural wetlands. Today, earthwork on 225 acres of Tidmarsh East has been completed, and is currently the largest fresh-water wetland restoration in Massachusetts. In October 2017, the property owners sold this site to Mass Audubon who plan to create the Mass Audubon Tidmarsh Wildlife Sanctuary.

In parallel, in 2010 the owners of Tidmarsh Farms created Living Observatory (LO), a non-profit organization whose mission is to tell the long term story of this landscape in transition and to develop experiences that will allow the general public to better understand ecological processes. With Doppellab as a precursor, ResEnv decided to participate in LO in order to explore how one might build a low power sensor network and a virtual environment in a unconstrained physical landscape. Hence the design of the sensor nodes and Doppelmarsh (the example of resythesized reality described in this thesis) began in 2012, 3 years before excavators would move across the Tidmarsh Restoration site to physically remove landscape stressors to insure a future wetland trajectory. Of 8 earthen dams and their attendant water-control structures; the filling of edge and lateral ditches; the digging of a sinuous 2.5 mile channel; micro-topography and the positioning of large wood across the site. Over 30,000 native trees, shrubs and herbaceous perennials were grown on site and planted. We describe this active phase as time zero. Last summer, the farm turned into a public wildlife sanctuary managed by Massachusetts Audubon Society[2], and maintained by the LO community [39].

Figure 2-2: A broad range of native plant species are present post-restoration. Photo credit: Tristan Spinski for The New York Times

2.1.2 Tidmarsh Living Observatory

The Responsive Environments group considered this as an opportunity to conduct experiential research that centers heavily around extrasensory perception, *Telepresence*, and creative applications. Many graduate students from the group joined the Living Observatory team to build and install the low-power sensor network at Tidmarsh and incorporate the data that is being gathered into various projects. These projects interweave under the *Networked Sensory Landscape* [41] and mentioning them briefly in this document is essential for clarifying the meaning and novelty of this contribution.

Sensing the landscape

A network of custom low-power wireless sensor nodes, a small network of wired cameras, and a wired network of microphones and hydrophones were deployed by

---

[2] MassAudubon, Currently the largest landowner in Massachusetts, MassAudubon serves as a leader and catalyst for conservation by acting directly to protect the nature of Massachusetts, and by stimulating individual and institutional action through conservation, education and advocacy. http://massaudubon.org

Figure 2-3: Topographic map (1942) of Manomet Village, Plymouth MA. showing footprint of proposed Tidmarsh Farms Wetland Restoration (c. 225 acres)

researchers from the Responsive Environments group at several sites on Tidmarsh Farms as shown in figure 2-4. These networks broadcast sensor data, live images and real-time audio to the Internet via onsite solar-powered or wired gateways with IP links. Each sensor device includes an ambient light sensor, an atmospheric pressure sensor, an accelerometer, a temperature sensor and a humidity sensor as default sensing modules. The sensor nodes are also equipped with several 12-bit analog inputs ready to accept streams of data from external sensors, extending their sensing capabilities. The first generation of nodes is powered by three AA batteries whereas the second generation will be powered with a small solar panel and a rechargeable battery for sustainability. Several cameras are installed on site capturing and sending frequent images from locations of interest. The live audio streams harvested by the array of microphones and hydrophones is mixed in a rackmount 32-channel mixer connected to an X86 computer with Internet connection [41].

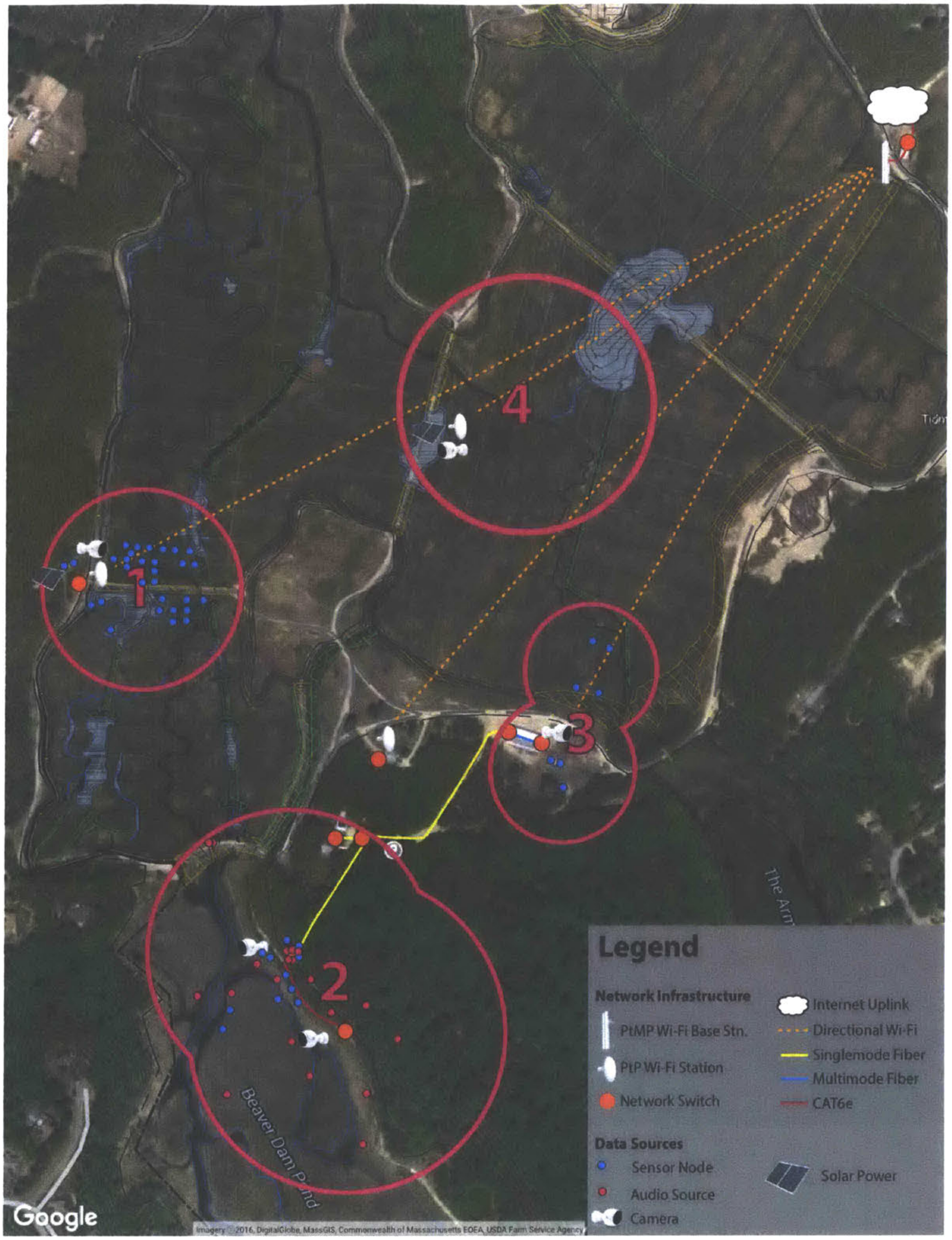


Figure 2-4: Tidmarsh deployment map as of May 2017 by Brian Mayton

Linking the landscape:

The *Networked Sensory Landscape* is designed to serve multiple end-user applications. The diversity of the incoming streams of data and the flexibility of the sensor nodes present challenges that Chain-API was built to address [62]. Chain-API is a hypermedia web service that adopts the REST [19] architecture implemented with Python libraries Flask [60] and Django [12]. It accepts JSON-encoded data via HTTP requests, stores them in a PostgresQL [56] database and serves augmented JSON+HAL [28] via HTPP and WebSockets with ZeroMQ on demand from modern application and custom-crawlers. The multichannel audio stream is published using the Icecast protocol, a free and common Internet streaming service, and then cached on the same machine under both Ogg and MP3 audio formats. Finally the images captured from the on-site cameras are stored in a web-directory served by Apache [41].

Figure 2-5: A picture from a camera deployed on Tidmarsh at zone 4.

Experiencing the landscape:

The Responsive Environments group built a range of user interfaces to augment human perception via auditory and graphical experiences. These experiences are crafted for both onsite and remote visits. Onsite explorers can examine the sensor data from each node using Google Glass. They can also try augmented auditory experiences with the project HearThere [61] which relies on the array of microphones to extend the wearer's hearing via a pair of custom bone-conduction headphones. Remote visitors can access the data through an interactive web-interface showing the sensor data visualization over time [31], explore a virtual replica of the site from a bird's eye view under a HoloLens, or get immersed in Doppelmarsh and see real-time sensor data in a 3D virtual replica of the landscape built with the Unity game engine.

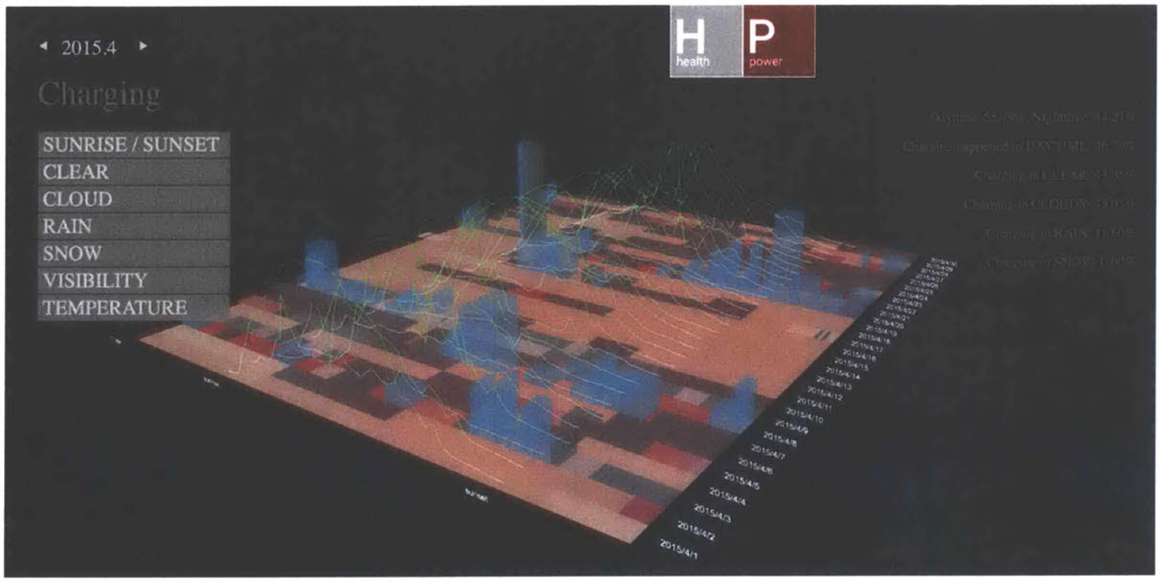


Figure 2-6: Online web-based wireless sensor networks visualization. Pulled from tidmarsh.media.mit.edu/viz

2.2 Related Work

2.2.1 Cross Reality

Doppelmarsh grew out of two prior projects in the Responsive Environments Group, ShadowLab and Doppellab. Both feature virtual environments that connect to sensors deployed within the physical building of the MIT Media Lab, building E14/E15. ShadowLab [32] employed temperature and infrared motion sensors embedded in physical devices called "data ponds" deployed in thirty locations around building E15. A virtual version of each data pond can be found in a stylized depiction of the Media Lab within the shared virtual environment Second Life [34]. The appearance of each virtual data pond changes based on the state of its physical counterpart.

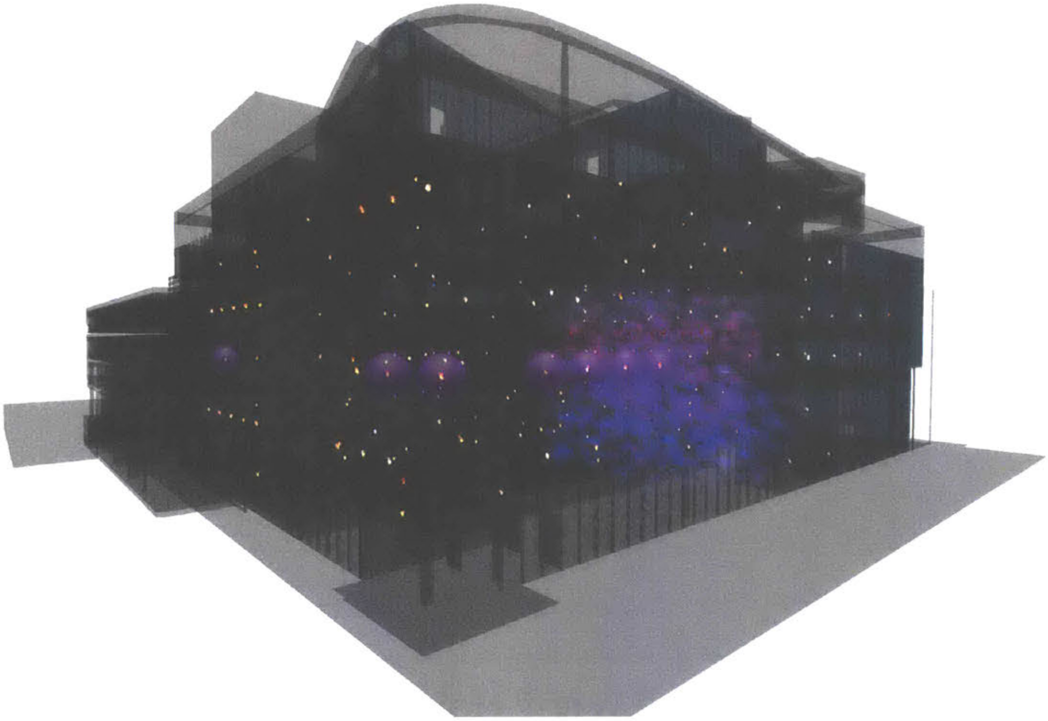


Figure 2-7: Doppellab — an immersive multimodal indoor sensor browser installed at the Media Lab E14 building. Pulled from doppellab.media.mit.edu

Doppellab, shown in figure 2-7, additionally tracked many parameters like temperature, humidity, motion sensors and audio via microphones that are located across the E14 building, as well as RFID readers, placed on each floor, to sense the presence of people with tags during special events [16]. A virtual version of the Media Lab was built using the Unity game engine. In this environment, colored spheres and flames were used to represent the state of the environmental sensors and boxes texture-mapped with photos represented people. Doppellab also connects to social media platforms by embedding a stream of real-time *Tweets* related to the social activities happening around the Media Lab. Doppelmarsh extends its predecessors in both the quantity, quality, and types of sensors, as well as in the setting. Its approximately 400 nodes are scattered across the landscape. These nodes include not only measurements of temperature and humidity, but also pressure, wind, rainfall, soil moisture, ambient and UV light. This landscape is augmented further with additional sensors such as cameras, microphones, and hydrophones [41]. Unlike Doppellab, Doppelmarsh's sensors are deployed outdoors in the wild rather than inside a building, resulting in more variability and unpredictability in the data. The challenge was to take this corpus of data and package it within the virtual environment Doppelmarsh in a way that made intuitive sense to the explorer without prior knowledge of the system.

2.2.2 Wireless Sensor Networks Visualization

Historically, "node-and-link" diagrams are essential for modeling and describing communication networks. Often forming a "mental map", these images can help the user internalize a cognitive model of a given system [46]. Due to their pervasiveness in many environments, wireless sensor networks benefit from such representations. However, these mental maps can be easily distorted or destroyed when the visual representation is mixed or shuffled, i.e. when nodes are added or removed from the system [46]. Animating these diagrams as they change state using Gestalt principles of organization has been shown to prevent the distortion of these mental maps [52].

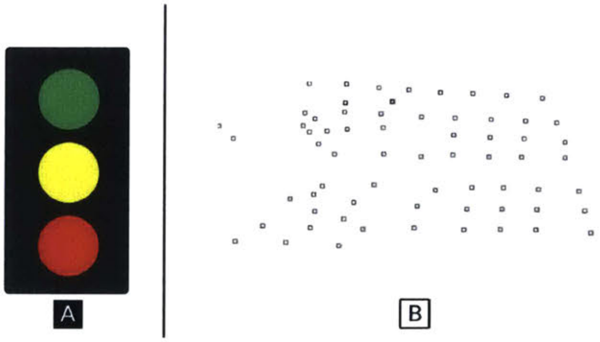


Figure 2-8: "Elements tend to be perceived according to an observer's past experience." [73] — Two examples of the Gestalt Law of Past Experiences, where picture A is almost recognizable by everyone whereas picture B might only be identified by some of the Tidmarsh Living Observatory team.

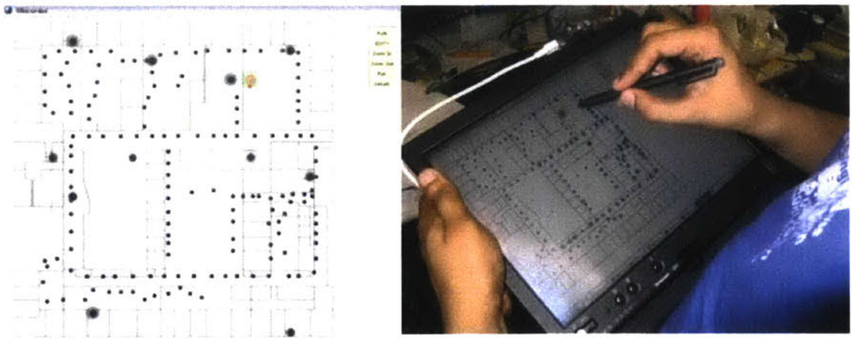


Figure 2-9: The Ubicorder is a mobile, location and orientation aware device that enables users to browse and interact with real-time data from sensor networks [47].

Sensor networks are also widely visualized on top of geographical maps [27] or on top of floor plans of buildings [83, 47, 63], in outdoor and indoor sensing applications respectively. Many of these systems have demonstrated the usefulness of such visualizations in comparison to tabular based representations [81], with a range of applications including environment and habitat monitoring [71], industrial process monitoring and automation [22, 8], machine status monitoring [5], healthcare, home automation and car traffic congestion [21, 29], to name a few.

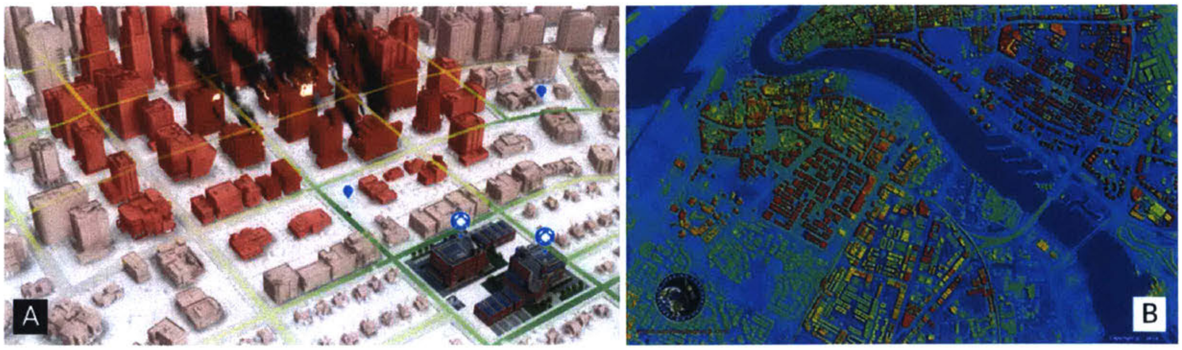


Figure 2-10: Picture A shows a heat-map representation overlayed on top of a fictional 3D model of a city in the game Sim City, and Picture B features a processed stereo satellite image of the city of Dubai, UAE. Copyright ©AIRBUS DEFENCE & SPACE and processed by Satellite Imaging Corporation

Heat-maps are also well known for representing data flowing out of sensor networks. They translate a corpus of numbers into colorful gradient images, usually overlaid onto existing maps or floor plans, giving the user a stronger insight on the spatial distribution of the data. This has been shown to be an efficient way of visualizing sensor data in a range of fields, disciplines, and applications [26, 24]. For instance, in the field of meteorology, the heat-map of periodic sensor data incoming from a combination of weather radars and weather stations can be overlaid on top of 3D terrains, often called *3D geovisualization* [50, 49].

The light is shining on immersive and interactive sensor-driven-monitoring systems in the industry. With the increasing amount of farming production, agriculture's traditional two-dimensional grain storage monitoring systems using graphs and spreadsheets have become less efficient. New, immersive virtual tools are being developed to address such problems, improving observability, operationality, and interactivity of grain monitoring [82].

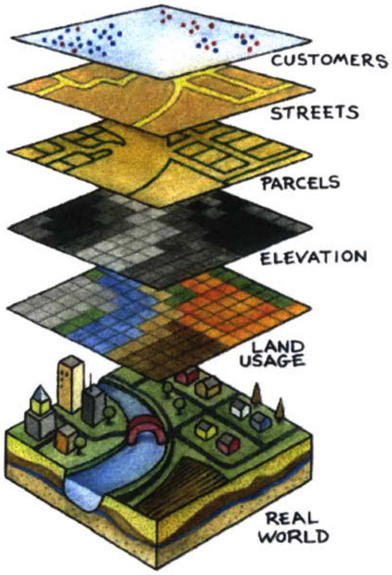


Figure 2-11: Map Overlay in a geographic information system (GIS) [6] inspires the creation of the needed layers to resynthesize any given environment.

2.2.3 Skeuomorphism

By definition skeuomorphism derives from objects that borrow ornamental design cues from other, real-world objects. For instance, annotated diagrams can be used to communicate how carvings of Lapith and the Centaur are used to adorn the pillars (see figure 2-12). Historically, the word "skeuomorph" comes from the Greek words

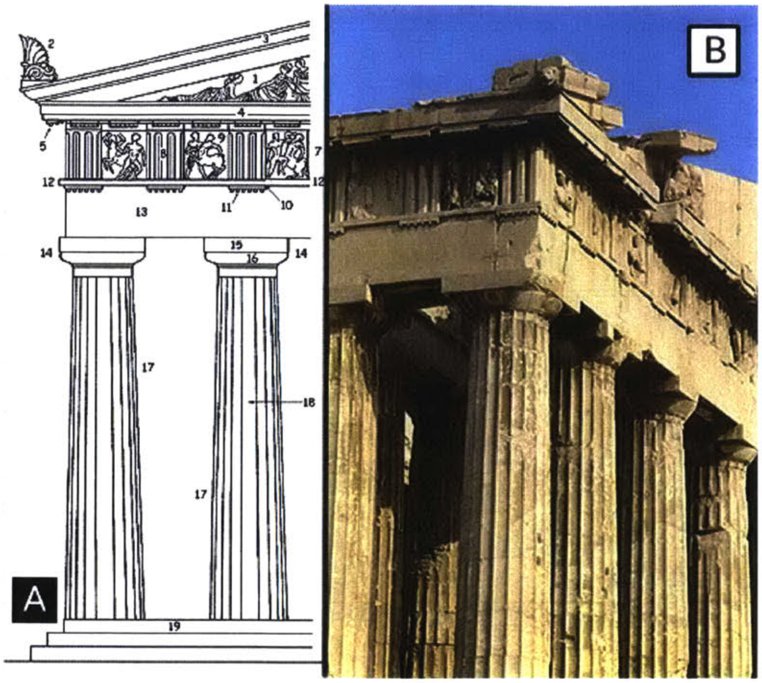


Figure 2-12: Skeuomorphism in Greek architecture. Figure A shows an annotated line drawing of a standard Greek building where the Metopes (2) and Tympanum (1) show sculptures or drawings of scenes appropriate to this building, i.e. the Parthenon, shown in picture B, features a long battle between Lapith and Centaur.

“skeuos,” meaning “container”, and “morphe,” meaning “shape.” It has been applied as visual texture on objects since 1890 [70]. Today, skeuomorphism is widely adopted by computer and mobile interfaces as a tool to guide the user on how to operate a given application or software. For example, audio tools like Reaktor [25] and Reason [58] still rely on skeuomorphism in their modular interface design, bridging the gap between generations of audio engineers and composers operating from within the “box”. Decades from today, one can imagine the reaction of the next generation of users who might stumble upon old fossils–like floppy disks–and wonder about the purpose of these “save buttons,” only recognizing their skeumorphism. Via interface metaphors, these design cues sometimes are adopted as common names that indicate generic features in a system [51]. A good example of that is the Desktop metaphor, which is adopted by default on almost every operating system. Today, the availability and mass deployment of virtual reality raises questions on which metaphors might guide the design of interfaces in virtual spaces.

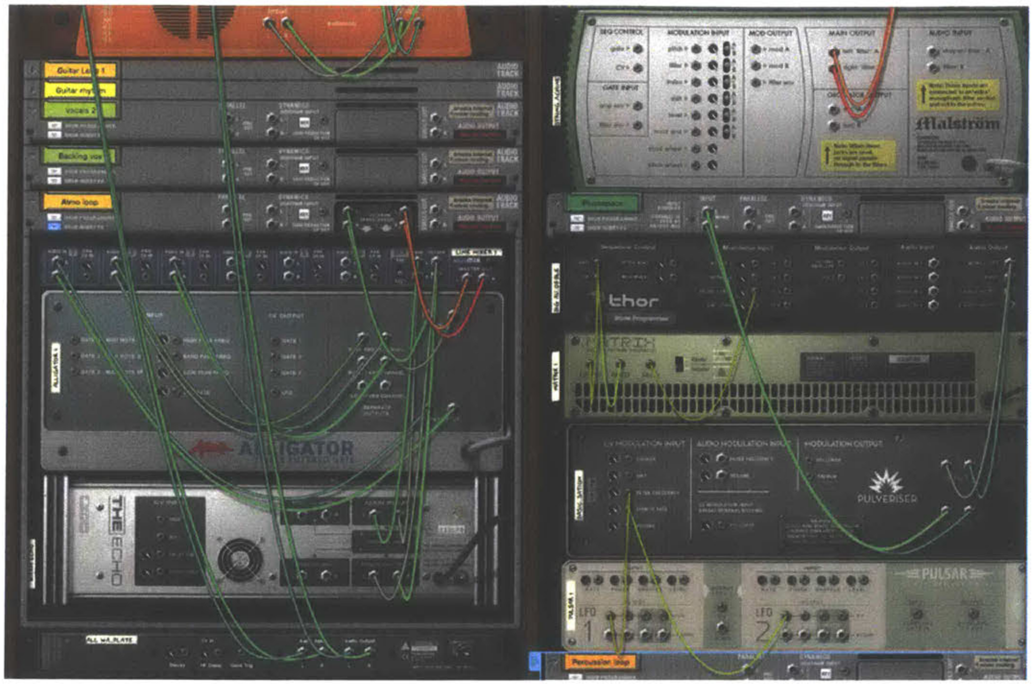
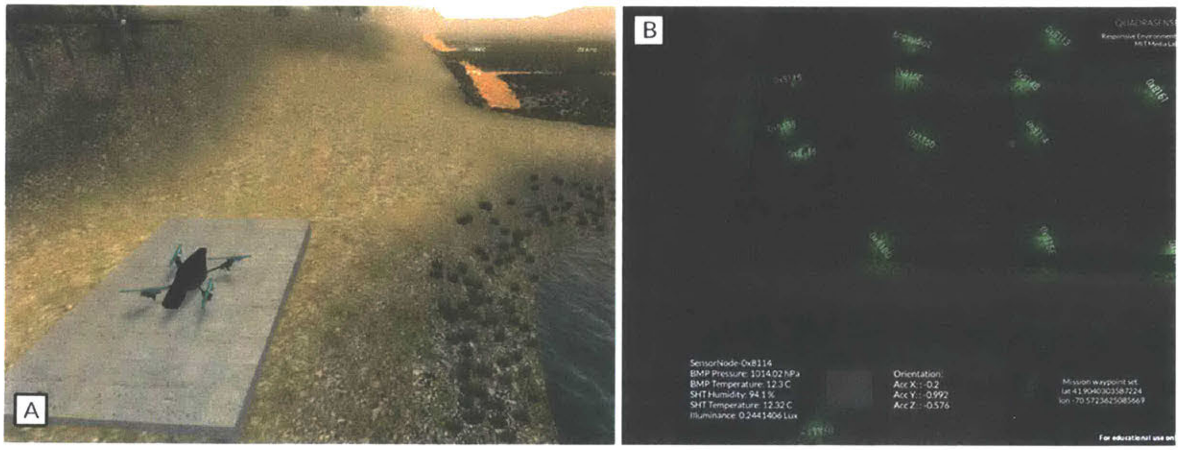


Figure 2-13: Reason9 - Audio tool using a skeuomorphic interface showing a rack of audio plugins or modules.

2.2.4 Serious Games

Serious Games as a concept goes all the way back to the days of Plato and his philosophy on the purpose of play in education [10]. As a term "Serious Games" was coined in the mid of the 20th century by Clark Abt, an engineer, environmentalist, entrepreneur, educator and social scientist. *Serious Games* refer to games designed with a primary purpose that is not entertainment [2, 80]. The pre-digital era of gaming was empowered at first by military ideas, most notably the Chaturanga, the precursor of Chess [55]. Throughout history, board games, or analogue games, were also used as contemporary warning sirens. For instance, the game Landlord, which preceded Monopoly, was intended to show the dangers of the capitalist approaches to land taxes and property management [55].

The early digital age of gaming was also fueled by the military. In the early 80s, the Bradley Trainer game was developed by Atari in collaboration with the US Military, and was used by soldiers as a training simulator [67]. Two decades after his original writings, Abt revisits his famous book on *Serious Games* with disappointment [2]. He was displeased to see that the evolution of video games emphasized only entertainment. He felt that the invisible hand of marketing and advertising shadowed centuries of gaming evolution. For instance, the game *Pepsi Invaders*, made in 1983, replaces the evil invaders in Atari's famous game *Space Invaders* with the letters P-E-P-S-I. The game was intended to be played by sales employees of Coca-Cola and their fan base, as a means of fostering company's morale against their competitor [13]. That same year, the gaming industry crashed due to the ubiquity and poor quality of the games being made. Without any seriousness or application, games became uninteresting and purposeless. Many thinkers, psychologists and educators talked about the



SensorNode-0x8114
BMP Pressure: 1014.02 hPa
BMP Temperature: 12.3 C
SHT Humidity: 94.1 %
SHT Temperature: 12.32 C
Illuminance: 0.2441406 Lux

Orientation:
Acc X: : -0.2
Acc Y: : -0.992
Acc Z: : -0.576

Mission waypoint set
lat 41.90403035587224
lon -70.5723625085669

Figure 2-14: Picutre A and B are both featuring the Quadrasense interface that allows a user to control a real UAV from within Doppelmarsh.

purpose of playfulness in learning [9]: Jean-Jacques Rousseau, Jean Piaget, Seymour Papert [53] and Edith Ackermann [3] to name a few. Their ideas unknowingly took on the non-entertainment focused approach of *Serious Games*, but here, the focus was education. It took *Serious Games* some serious time to become a legitimate field of research in academia [1] due to its interdisciplinary nature, and to some opposing views calling it an oxymoron [20]. Until 2002, Ben Sawyer discussed the potential of "instructional" games, arguing that there should be a stronger connection between commercial games and contextual applications [64]. Today, *Serious Games* as a concept is being applied to a wide range of disciplines, from teaching computer science concepts like recursion [7], to labeling images [79], to fostering social skills and leadership development [4], to healthcare and wellbeing [78]. Gamification is now playing an important role in the design of user interfaces and user experiences by using game-design elements in non-gaming contexts [11]. Like in the project Quadrasense [59], shown in Figure 2-14, that enables the user to control a remote UAV from within a video game interface in the avatar landscape Doppelmarsh. It is also notable to mention the many different fields that emerged from *Serious Games* like *Alternate Reality* [72], *Pervasive Gaming* [48] and *Game-based Learning* [57].

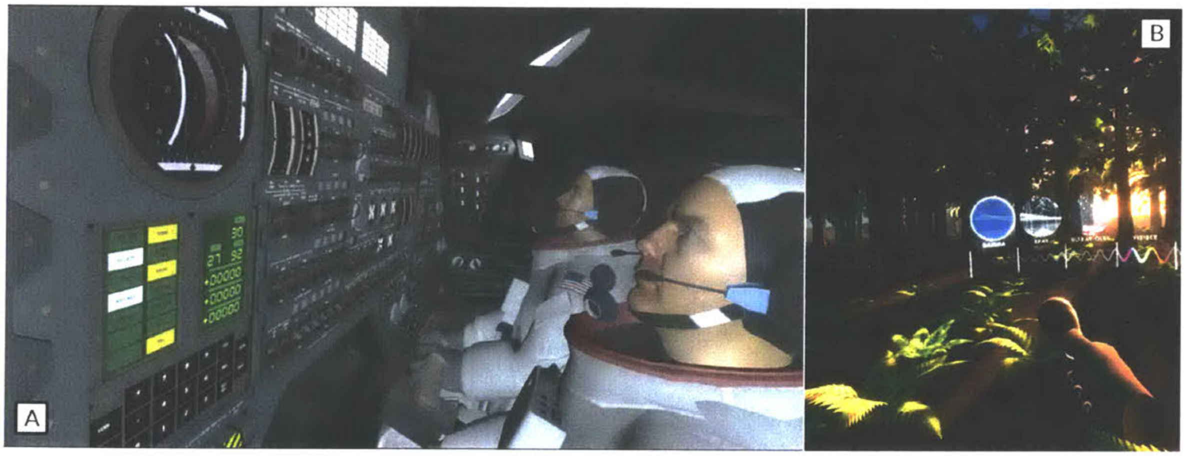


Figure 2-15: Picture A is a VR experience called Apollo-11 where the user experiences the historical mission to the moon. Picture B is an educational VR game that takes the user on a journey across the electromagnetic spectrum.

2.2.5 Computer Mediated Reality

Any intervention on humans' normal perception of reality could be conceived as "mediating" reality. Thus, various forms of "mediating" realities existed far before the invention of the modern computer. Sensory channels define reality as we perceive it, with the visual apparatus serving as a primary channel through which humans receive information about the world. Studies on the visual perception date back to G.M. Stratton at the end of the 19th century, who inspired generations of eye-glasses based "reality-mediators"[68]. In the 1960s, Ivan Sutherland and his students pioneered the now-familiar concept of the head-mounted displays (HMDs), calling it the *Ultimate Display* [69]. *Computer Mediated Reality* thus highlights the role of the computer in generating or synthesizing these images. Steve Mann first proposed computer mediated reality [37], as well as being one of the earliest ideators of wearable computing at the MIT Media Lab [38]. *Augmented Reality*, *Mixed Reality* and *Virtual Reality* (presented by the different modern HMDs and goggles) provide experiences within the reality-virtuality-continuum, ranging from *Augmented Reality* to *Augmented Virtuality* [37]. In 1989, Jaron Lanier coined the term *Virtual Reality* [66] to encapsulate the many different projects occurring at VPL research at that time. In 1994, Paul Milgram and Fumio Kishino proposed the taxonomy *Mixed Reality* [43], distinguishing it from both *Virtual Reality* and *Augmented Reality.* Steven Feiner defines *Augmented Reality* as displays that add virtual information to a user's sensory perceptions [18]. In contrast with this nomenclature, Ken Perlin considers anything that mediates perception as just part of our reality. For instance, Ken Perlin and his students have been working with VR headsets to prototype the next generation AR interactions [40]. As these fields are constantly blending into one another, it would be really difficult to distinguish between their "types". In other terms, whether augmented, virtual, extended, diminished or mixed; what we experience remains part of our own reality.

These realities usually include the displays and the set of equipment that comes with them–a pair of instrumented gloves for instance. We proposed another axis to these experiences orthogonal to the *reality-virtuality-continuum.* The production side of these experiences includes an axis ranging from representative of existing environments to fictional, new environments. Doppelmarsh is a somewhere in the middle of this axis, representing an existing marsh, yet with distortions and/or augmentations, making it a space unique to itself. We also suggest that these processes co-exist under the realm of *Computer Mediated Realities.* For instance, *Diminished Reality* introduces the concept of removing elements from any of these given realities presented in the continuum [23]. Similarly, the work proposed within *Cross Reality* [33, 54], also known as *Dual Reality* [35], by Josh Lifton and Joseph Paradiso in 2007, introduces the convergence of ubiquitous sensing and virtual environments. Thus, in *Resynthesizing Reality*, a direct descendant of *Cross Reality*, we propose the usage of outdoor

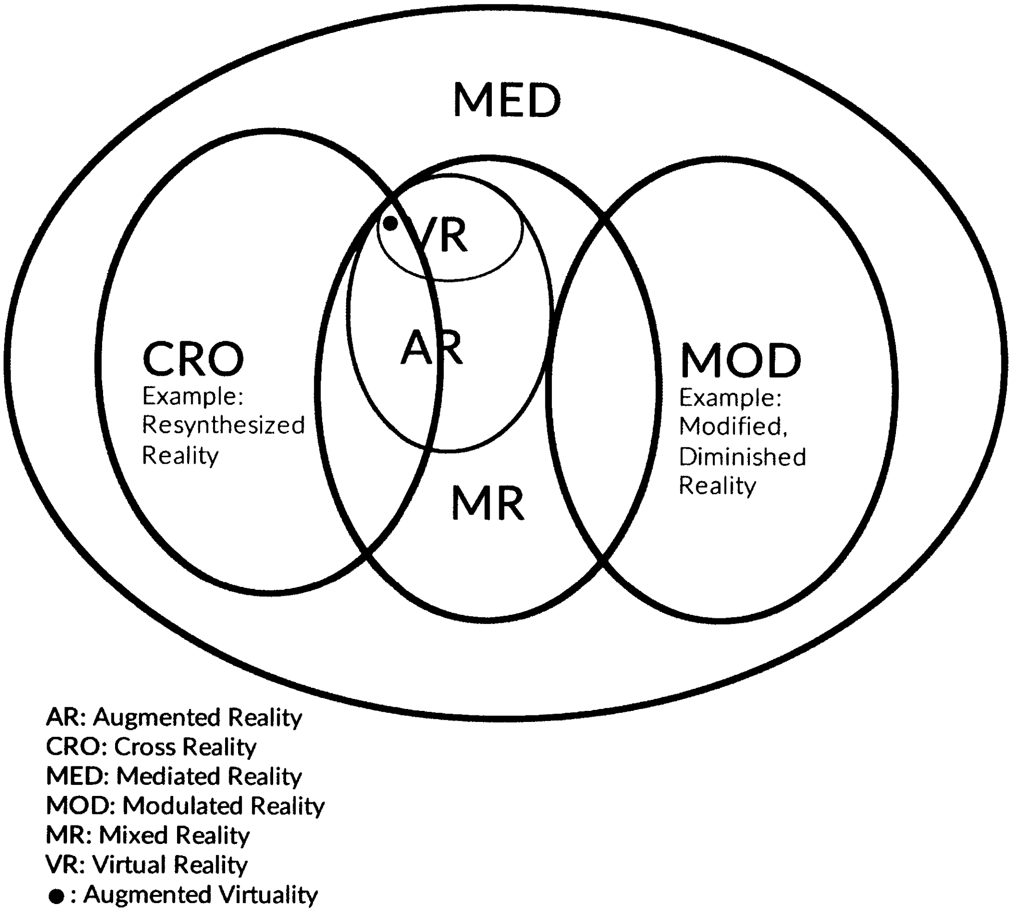


Figure 2-16: A suggestive addition to the Computer Mediated Reality framework. This updated venn diagram features the inclusion of Cross Reality within the framework.

sensor networks to drive the vividness of virtual environments. This helps to teleport the virtual explorers into a more-or-less realistic replica of such environments, and ultimately guides their imagination in filling any sensory gaps [65].

Chapter 3

Synthesizing Virtual Environments

3.1 Introduction

This chapter covers the resynthesis of real environments from the ground up. First, we present a brief history of Doppelmarsh's evolution since 2014. We establish the primary goals of the project. How does this work fit into the Living Observatory research agenda? Who is the ultimate player/explorer of the landscape? What is the over-arching strategy that allows us to make aesthetic decisions? What are the most appropriate techniques to realize our goals. From there we present some of the techniques used to build 3D virtual clones of existing environments. Then, we will expand on the concept of synthesizing such environments by covering the process through which we built Doppelmarsh, and detail the many optimization and rendering techniques that we had to consider to shape overall performance and aesthetics. In particular an aesthetic goal might be to make the environment inviting, evocative, familiar, and vivid inviting in that it provides an expanse with enough happening so as to generate curiosity on the part of the player/explorer. Evocative in that it is suggestive of the physical environment on which it is modeled. Vivid in that it produces powerful feelings or strong emotions in the player/explorers mind. This chapter will conclude with future ideas related to automating the processes that finally led to resynthesizing Tidmarsh. Chapter 4 explain how we dynamically update these synthesized environments form real images and sensor data.

3.2 The World of Doppelmarsh

3.2.1 A Brief History

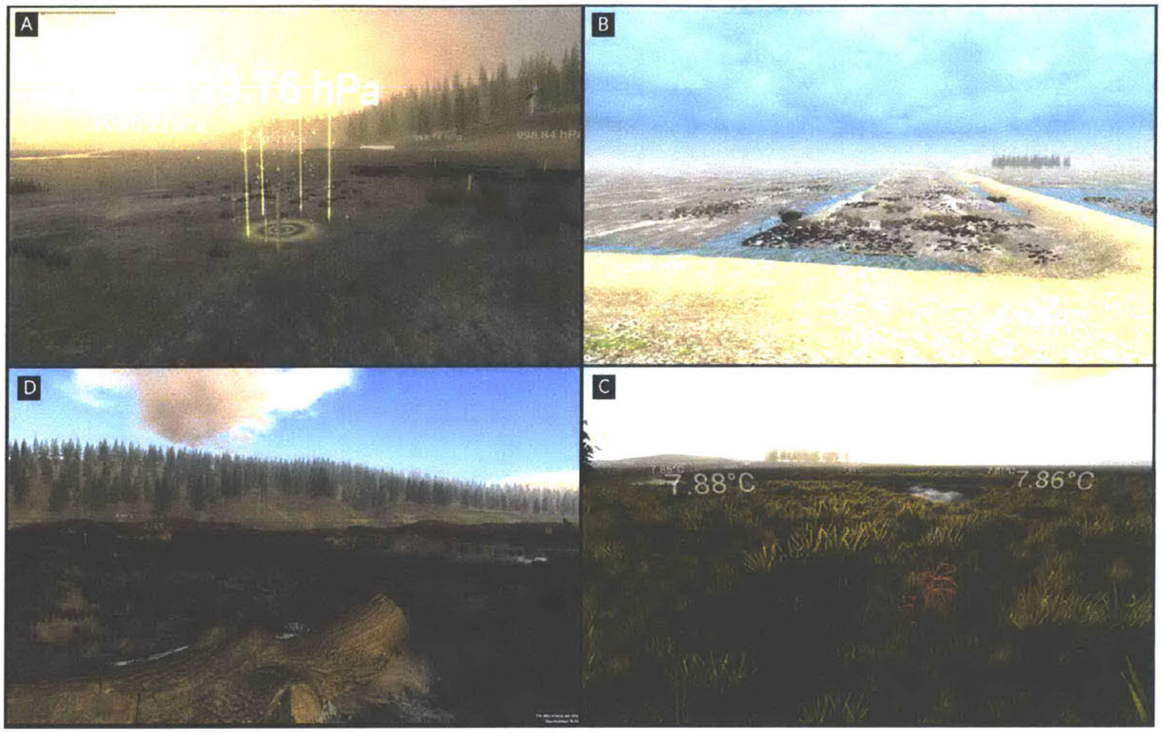


Figure 3-1: Evolution of Doppelmarsh since 2014. Picture A and B both features Cell3,4 that show the first prototype of Doppelmarsh done using Unity 3.2. Picture C and D include a boost in the overall graphics of that same site, built respectively in 2015 and 2016, and reflect every topographical change due to restoration activity

The work on Doppelmarsh started in 2014 after the great success of Doppellab [15]. Connecting sensor networks to game-engines proved to be extremely useful for building interactive applications—Essentially physical environments instrumented with sensors and then rendered in game engines such that the sensor data drives visualizations within the virtual environment. Using the Unity game-engine showed to be more flexible than previous attempts using the Second Life platform.

The first area of Tidmarsh to be synthesized is called *Cell3, 4*. The virtual terrain was constructed by LIDAR (Light Detection and Ranging) measurements collected in 2012 by the United States Geological Survey (USGS) from the real site. This physical site was actively restored in 2015 when excavators removed an earthen dike/dam, dug a new sinuous stream channel, broke up the flat cranberry mat to create microtopography, and placed large wood across the site. Parts of this area were then planted with 2 year old Atlantic White Cedar trees. These changes required us to adapt the topography of the virtual rendering which was edited to reflect these landscape scale changes. High-resolution photographs of the physical surface are used to paint the different textures of the virtual terrain, and the vegetation is selected from models of trees and herbs that resemble those at the actual site. The geo-tagged sensor nodes are represented in the virtual environment at their location that corresponds to their location in the physical world. An animation is triggered each time a node transmits an update to LivingObservatory servers that sit at the Media Lab. The audio, captured from onsite microphones, is streamed into Doppelmarsh and mixed with a generative musical system fed by real-time sensor data [36]. Doppelmarsh includes other basic components like the sky-box, the player and its motion, and the water system. The latter relies on purchased assets then assembled in the Unity editor. The second location to be synthesized using the same technique is called the *Impoundment*, (a.k.a. Beaver Dam Pond); the reservoir was drained in 2010 as one of the first restoration actions, leaving behind a braided stream channel running through a marsh. Here, another sensor network and a camera are deployed. Figure 3-1 shows the evolution of Dopplemarsh that correlate with phases of the restoration happening on Tidmarsh.

More recently, many techniques were used to enhance Doppelmarsh's visual experience. The goal of these enhancements was to make the environment inviting, evocative, familiar and vivid for the player/explorer while insuring that the experience could run smoothly on a range of machines. Let us unpack these aesthetic goals with some questions; we can then relate particular techniques used to these aesthetic goals. Inviting: we want to create an environment that reflects the physical so as to be familiar and accurate in the presentation of sensor data. The virtual expanse need to provide the player with a sense of scale. It also needs to have enough happening so as to generate curiosity on the part of the player/explorer. The virtual space should be evocative such that the player/explorer can image and believe that they are in the particular physical environment that is Tidmarsh. This physical environment has a particular topography including a stream, particular plant life. Also weather and seasons. Finally we want the rendering to be vivid, to be able to be retained in memory even as it reflects temporal change. These aesthetic goals allow us to use the digital tools that are at our disposal to create the virtual environment.

Figure 3-2: A screen shot of the latest Doppelmarsh. The shock waves that surrounds few of the sensor nodes are there to notify the user of incoming data.

New high resolution textures were taken from different locations on Tidmarsh and then mapped onto the virtual environments. A top view image of the site stitched from several images taken by drones helped guide the painting process. Trees, grass and a selection of birds were purchased from Unity's asset store to add to the ambiance, and the water material was tweaked to emulate the natural flow of the physical stream bed. The expanding bubble animation occurs each time a sensor updates. These were designed using Unity's Particle System, and fine-tuned to blast at a slow pace, adding dynamics intended to enrich the player/explorer's experience of variable environmental dynamics, in particular the awareness of micro-climates across the terrain.

3.3 Implementation

3.3.1 Terrain

In this section we discuss some technical processes used to synthesize real outdoor environments using the Unity game engine. One of the major features in Unity is its terrain engine. It allows the user to add and optimize large landscapes with great flexibility. Game designers can directly import their textures (surfaces, grass..etc) and models (trees, rocks...etc) directly into the terrain panel. They can then start paint and modify the terrain, even at runtime, with a UI toolkit that offers many brushes and options. The terrain panel is initialized upon creating a new terrain Gameobject from within the Unity editor. Unity will automatically create an empty terrain and attach it to the Gameobject that was just added. The user has the option to also import raw 16-bit height-maps that automatically generate a 3D mesh of the terrain.

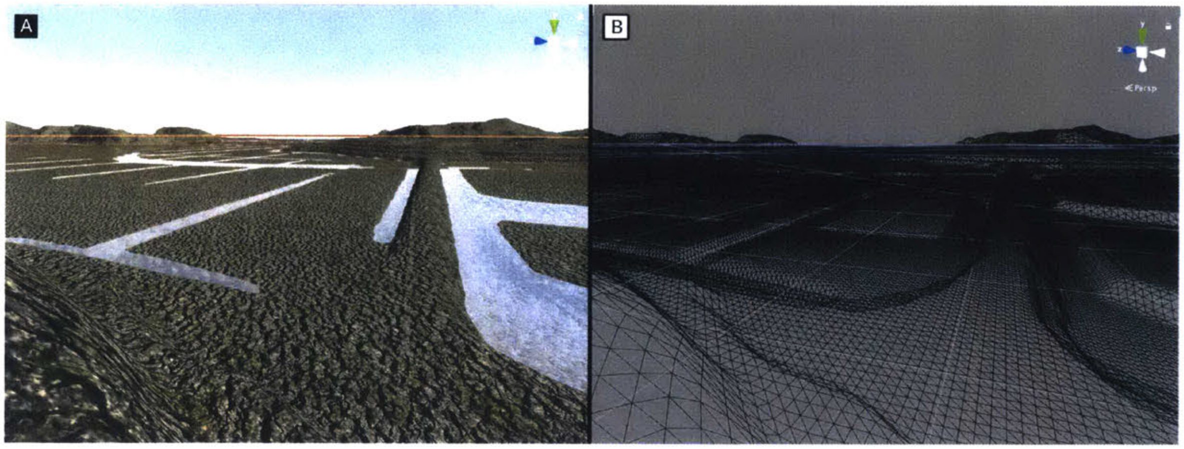
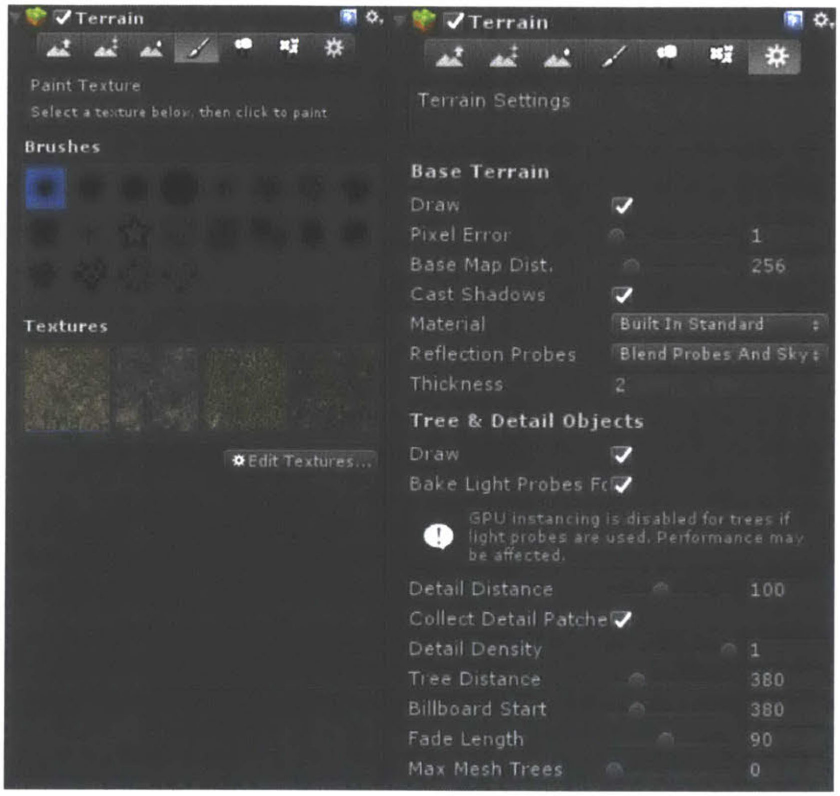


Figure 3-3: Doppelmarsh's terrain is generated form a height-map. Figure B shows an un-textured freshly imported terrain in wire-frames of Cell 3,4 whereas figure A highlights the stock texture, sky-box and water shipped with Unity5.

The terrain in Doppelmarsh is generated from a height-map of Tidmarsh Farms provided by the United States Geological Survey (USGS) using LIDAR data collected in 2012. A lower resolution height-map could be pulled from Google Maps, crushed and blurred with image editing tools like Photoshop or GIMP, and then imported back into Unity. This method might work well in areas where Google Maps can provide more resolution. Rendering large terrains can be computationally expensive. Unity's terrain panel provides the option to do basic optimization to boost the overall performance for vast landscapes. For instance, the level of details (LOD) on the trees and the grass can be modified to scale fidelity in rendering.



Select a texture below, then click to paint

Base Terrain

Draw ☑
Pixel Error 1
Base Map Dist. 256
Cast Shadows ☑
Material Built In Standard
Reflection Probes Blend Probes And Sky
Thickness 2

Tree & Detail Objects

Draw ☑
Bake Light Probes Fc ☑

GPU instancing is disabled for trees if light probes are used. Performance may be affected.

Detail Distance 100
Collect Detail Patche ☑
Detail Density 1
Tree Distance 380
Billboard Start 380
Fade Length 90
Max Mesh Trees 0

Figure 3-4: Terrain panel in the Unity Editor. The panel on the left shows the different textures and brushes used to paint Doppelmarsh. Whereas, the panel on the right shows the terrain optimization panel.

3.3.2 Models

Many external 3D models were imported into Doppelmarsh, and can be divided into three categories. The first, and most notable, category includes the base station and the sensor node. Sensor node Gameobjects get instantiated at runtime at their corresponding geo-matching location using a script that also syncs to Chain-API. Upon each update, each node will change the sensor value rendered on-top of it. The second category of 3D models includes aesthetics used to develop a stronger sense of presence. The sky-box, water prefab, and wooden logs are examples. The sky-box in Doppelmarsh was purchased from the Unity Asset Store. It features high resolution textures of clouds with control over their density, speed, and time of day. The third and last category is called non-player characters or NPC. This includes the different animals that coexist in Doppelmarsh. Some of these animals, like the birds, are recognized using the array of microphones and then visualized around the closest triggered microphone. Other animals, like the deer, hold valuable information about the historical sensor data of the zones they wander around based on their appearance. Chapter 4 will cover more about mapping sensor data and the animals.

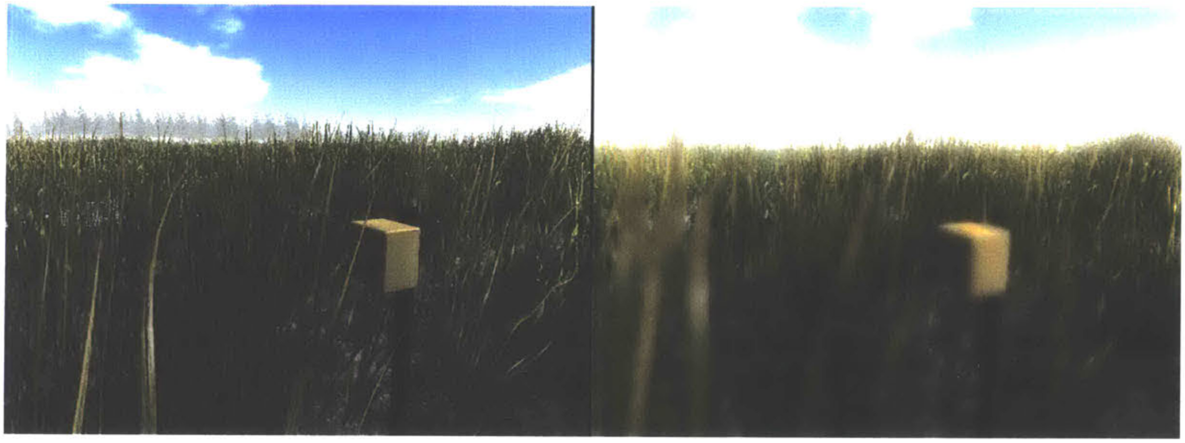
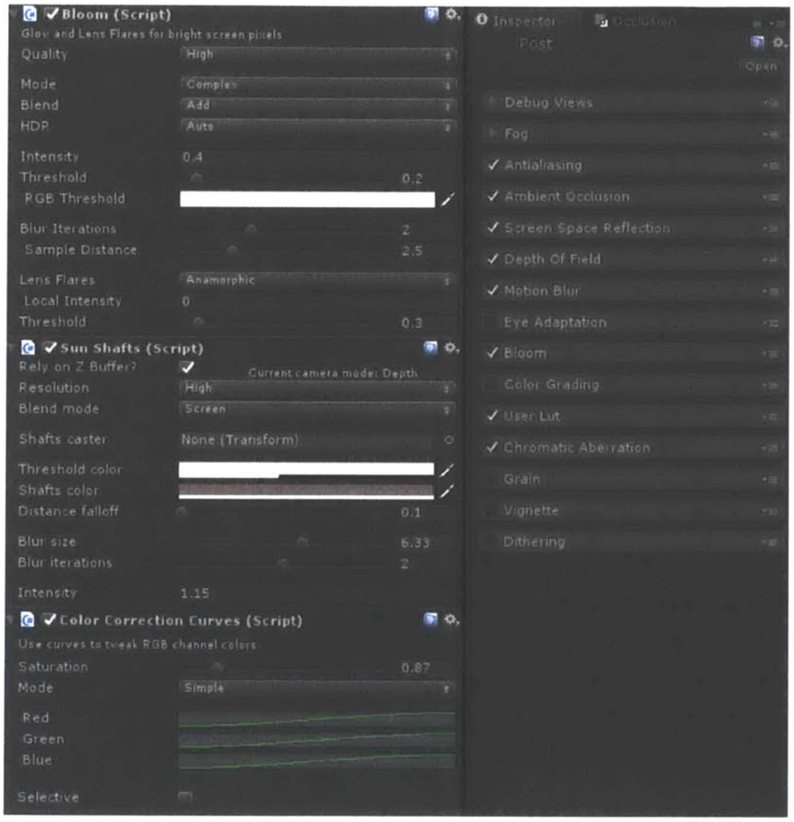


Figure 3-5: The figure on the left is a rendering without post-processing filtering, whereas the one on the right includes bloom, blur, anti-aliasing and more.

3.3.3 Camera

A simple first person controller prefab is included in Unity's standard packages and was used to enable basic player movement in Doppelmarsh. The camera is attached to the player's head, and thus follows the player's movement within the virtual environment. Chapter 5 covers more about controllers and user experience. The camera script also includes the option for high dynamic range (HDR) rendering. When combined with Unity's post-processing stack, image effects like bloom and blur may appear even more magical and dreamy, at the expense of computing performance.

Other effects like anti-aliasing were used to give a smoother appearance to graphics, especially on older computers. Figure 3-6 below shows most of Unity's post-processing effects [77] used in Doppelmarsh and some of their parameters.



✓ Bloom (Script)

Glow and Lens Flares for bright screen pixels

Quality	High
Mode	Complex
Blend	Add
HDR	Auto
Intensity	0.4
Threshold	0.2
RGB Threshold	
Blur Iterations	2
Sample Distance	2.5
Lens Flares	Anamorphic
Local Intensity	0
Threshold	0.3

✓ Sun Shafts (Script)

Rely on Z Buffer?	✓ Current camera mode: Depth
Resolution	High
Blend mode	Screen
Shafts caster	None (Transform)
Threshold color	
Shafts color	
Distance falloff	0.1
Blur size	6.33
Blur iterations	2
Intensity	1.15

✓ Color Correction Curves (Script)

Use curves to tweak RGB channel colors

Saturation	0.87
Mode	Simple
Red	
Green	
Blue	
Selective	☐

Debug Views
Fog
✓ Antialiasing
✓ Ambient Occlusion
✓ Screen Space Reflection
✓ Depth Of Field
✓ Motion Blur
Eye Adaptation
✓ Bloom
Color Grading
✓ User Lut
✓ Chromatic Aberration
Grain
Vignette
Dithering

Figure 3-6: Prior to Unity 5.6, post-processing scripts were attached to the main camera Gameobject as indicated by the panels on the left. At the time of this writeup, Unity's post-processing stack is included as an external asset that could be downloaded and imported for free into the Unity editor, and is now manageable via a profiling tab as indicated by the panel on the right

Anti-aliasing

The Unity Anti-aliasing filter gives a smoother appearance to the overall graphics. Aliasing is an effect where lines appear to be jagged that occurs when the resolution needs exceed the power of the graphics card. Anti-aliasing helps reduce the prominence of these jagged lines, at the expense of making them blurrier.

Ambient Occlusion

Ambient occlusion relies on shading techniques to compute the exposure of each point in the scene in relation to ambient lighting. The Unity ambient occlusion post-processing effect darkens creases, holes, intersections, and surfaces are in front of other surfaces in the scene. This avoids extensive rendering, adding realism, but slowing overall performance. It should be noted that we have optimized here for aesthetic experience over total performance.

Screen Space Reflection

Screen Space Reflection is a technique that reuses onscreen data to calculate reflections. It is commonly used to create more ambient reflections, such as the water reflection in Doppelmarsh. This technique is computationally expensive.

Depth of Field

Depth of Field is a common post-processing filter that simulates the real life properties of a camera focus. By blurring objects at different distances, this technique creates realism and gives visual cues about the objects' distances.

Motion Blur

Motion Blur is another common post-processing effect that simulates the blurring of an image when filmed objects are moving. This can be perceived by rapidly moving objects in a given scene or by long exposure time to simulate a slow motion effect. Motion Blur is used in Doppelmarsh and is more noticeable upon increasing the player's movement speed.

Bloom

Bloom is an effect used to replicate an imaging concept of real-world cameras. This effect produces fringes of light shining around the borders of bright objects in a scene, contributing to the illusion of an extremely bright light which overwhelms a person's visual perception in the real-world. In Doppelmarsh, blooming is specifically noticeable when looking at the virtual sun in the sky-box.

Chromatic Aberration

The Chromatic Aberration effect is used to replicate a camera defect resulting from a lens's failure to converge all colors onto the same point. It is often used in conjunction with artistic effects, such as intoxication effects. The Unity's post-processing stack provides support for red/blue and green/purple fringing, as well as for user-defined color fringing via an input texture. In Doppelmarsh, this effect is used to make an overall environment feel hotter or colder by modifying the intensity of the red/blue fringes. More on mapping strategies will be covered in chapter 4.

3.4 Optimization

Dealing with high resolution textures, high polygonal meshes, large terrains, and post-processing effects can be quite computationally expensive. An optimization strategy must exist to be able to run Doppelmarsh on a variety of platforms with all of these graphical enhancements. This section covers all the optimization techniques that allow Doppelmarsh to run smoothly on a variety of platforms. A more detailed compilation of optimization techniques could be found on the Unity's documentation page [75].

Occlusion Culling

Occlusion Culling is a feature in Unity that prevents the rendering of objects that are not visible to the player's camera. The Unity editor features an automated baking procedure that divides the scene into subdivisions called cells. Upon runtime, a camera is used by the occlusion culling processor to identify what cells are visible and what aren't. Equipped with this information, Unity will ensure only visible objects get sent to be rendered, which improves the performance dramatically [74].

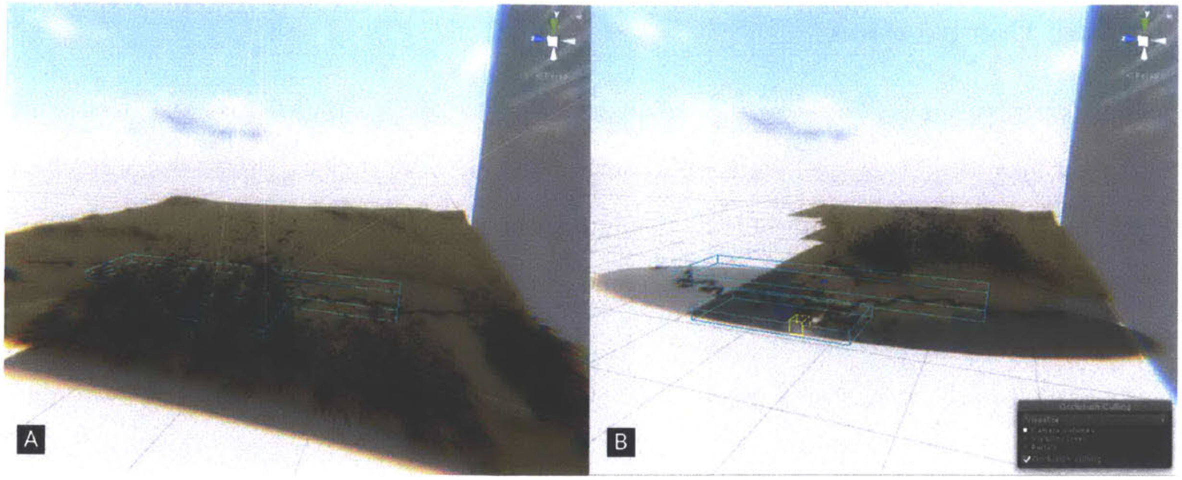


Figure 3-7: The effect of Occlusion Culling on Doppelmarsh. Figure A features the scene without occlusion culling, whereas figure B shows a fully baked scene with a cell size equals to 1.

Precomputed Lighting

Two types of Global Illumination (GI) are offered by Unity. By default the Unity Light in the scene is computed in real-time, which is also computationally expensive. Unity also offers the option to automatically pre-compute the lighting related to static objects in the scene. The GI in Unity can be accessed through a global panel, but it's also important to tweak the light properties for individual light sources. The default mode for these light sources is set to be dynamic, and therefore handled by the Real-time GI, but it can also be changed to a static option to be handled by the Baked GI system [76]. Baked GI is usually better than real-time GI for the overall performance. The most flexible way to use the lighting system is to use both Baked GI and Real-time GI in combination. However, this is also the most computationally demanding option and should be avoided when dealing with large landscapes.

Level of Detail (LOD)

Another technique that can boosts performance marginally is called Level of Detail (LOD). The distances between all of the Gameobjects and the camera are computed. This data affects the level of detail that is rendered. In other words, when the distance between any Gameobject and the camera increases less detail is shown.

Texture Compression

Using compressed textures results in faster load times, a smaller memory footprint, and dramatically increased rendering performance. Compressed textures only use a fraction of the memory bandwidth needed for uncompressed 32-bit RGBA textures. Unity accepts many common image formats as textures. By default, the Unity Editor automatically converts these textures to the most appropriate format that matches the target build. It's also advisable to generate mipmaps for textures. A mipmap texture enables the GPU to use a lower resolution texture for smaller triangles. This is a similar strategy to the use of texture compression to limit the amount of texture data transferred when the GPU is rendering [75].

Figure 3-8: Facebook's new x24 and x6 Surround 360 cameras approaches virtual reality. Photo-credit: Courtesy of Facebook.

3.5 Future Work

Doppelmarsh is built with the Unity game engine; however it is our intention as a group to explore the use of other game engines and advanced graphics tools like Cinema4D in the future construction of resynthesized environments, and ultimately, building a custom game engine that encapsulates the processes used to resynthesize real/virtual environments. The textures in Doppelmarsh are currently painted manually, but we also would like to explore automating that arduous task of painting by incorporating via satellite or drone-based imaging to create the foundation terrain automatically. Another explored, experimental path utilizes point-clouds to generate static 3D environments. Point-clouds synthesize many 2D images taken from a real site with poly-lens cameras as shown in Figure 4-1. Unlike the LIDAR technique, a point-cloud preserves the textures it captures. This technique captures the real world as is and turns it into a static, immersive 3D rendering. This technique presents future challenges, specifically in smoothly animating pieces of these environments. Nevertheless, in future attempts to resynthesizing Tidmarsh, similar techniques could be used to replace the LIDAR/texture mapping in generating the virtual environment of Doppelmarsh.

Chapter 4

Sensor Driven Realism

4.1 Introduction

In the physical world, an outdoor environment cannot be decoupled from its weather. As human beings, we evolved to learn about the environment through experiential perception combined with identifying repetitive patterns within the physical environment such as seasons, time of day, leaves of a plant from a distance. Wind is a physical force that we associate with other physical activity such as spread of cloud movement, or movement of leaves in trees, or swaying of grass. In short, wind enhances our sense of temporality in the environment. By experiencing and observing the wind's behavior in a given environment, we learn to recognize how it affects the movement of the trees and their leaves. By identifying whether it is windy out or not, we make decisions about how and what we want to do in the environment. Our decisions are in part a function of past experience. (this is true of other animals as well although reactions may be instinctual rather than learned i.e, in heavy wind birds can hunker down and become quite). In other words, our senses are like uncalibrated sensors which get calibrated through experiences. In this thesis we hypothesize that temporal development is as essential to the virtual environment as it is to our lives. By identifying a dynamic that reference salient temporal cues and patterns that we have learned to identify in the physical world, and representing them in our "resynthesized" virtual world, we can make the virtual world more vivid and evocative, thus giving the player/explorer a greater "sense of being there" [30]. This goes both ways. For example, even if you've never been in zero gravity, you might still be able to understand how it would affect your body if you had seen the affect of a person in zero gravity or experienced it inside a video game. This chapter describes mapping strategies that integrate real time sensor data and machine learning in Doppelmarsh in order to create a more vivid experience. This chapter also details the expansion of visual techniques, animations, and effects in the environment, and how they are integrated in a back end server in a concept we called Sensor Fusion as a Service (SFaaS).

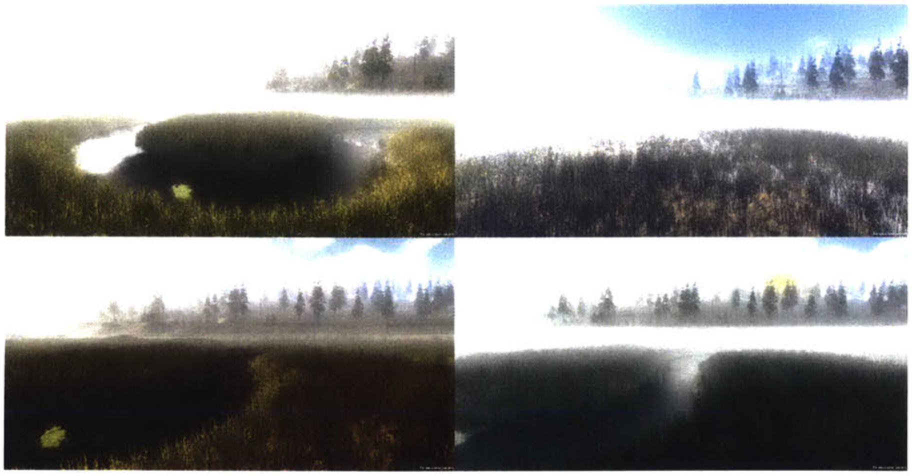


Figure 4-1: Different scenes from Doppelmarsh featuring the weather enhancements.

4.1.1 Sensor Fusion as a Service

The deployment of cloud computing in the world today brought with it metaphors and nomenclatures that are now familiar to the broader community of web-developers and software engineers. Cloud computing as a service provides as a stack divided into 3 main layers: IaaS (Infrastructure as a Service), Paas (Platform as a Service), and Saas (System as a Service). For example Amazon EC2 is an IaaS provided by AWS (Amazon Web Services). The back-end server that drives Doppelmarsh is a middleware server and is called MiddleMarsh. It is a good example of what a Sensor Fusion as a Service (SFaaS) offers. MiddleMarsh connects to multiple services that offer real-time streams of various types of data. Sensor data flow from Chain-API, while the stream of images is provided by a simple WebSocket interface serving a link to an image with some meta-data (location and timestamp). The images incoming from the cameras are analyzed with Google's Vision API that encapsulates powerful machine learning models to quickly classifying images and detecting features. MiddleMarsh synthesizes all of this information and stores them as "frames" in a MongoDB instance. These virtual frames of reality hold fragmented knowledge needed to generate visual effects and animations in the virtual environment. The next section describes the building blocks of these "frames," and the way they map to different visual effects within Doppelmarsh.

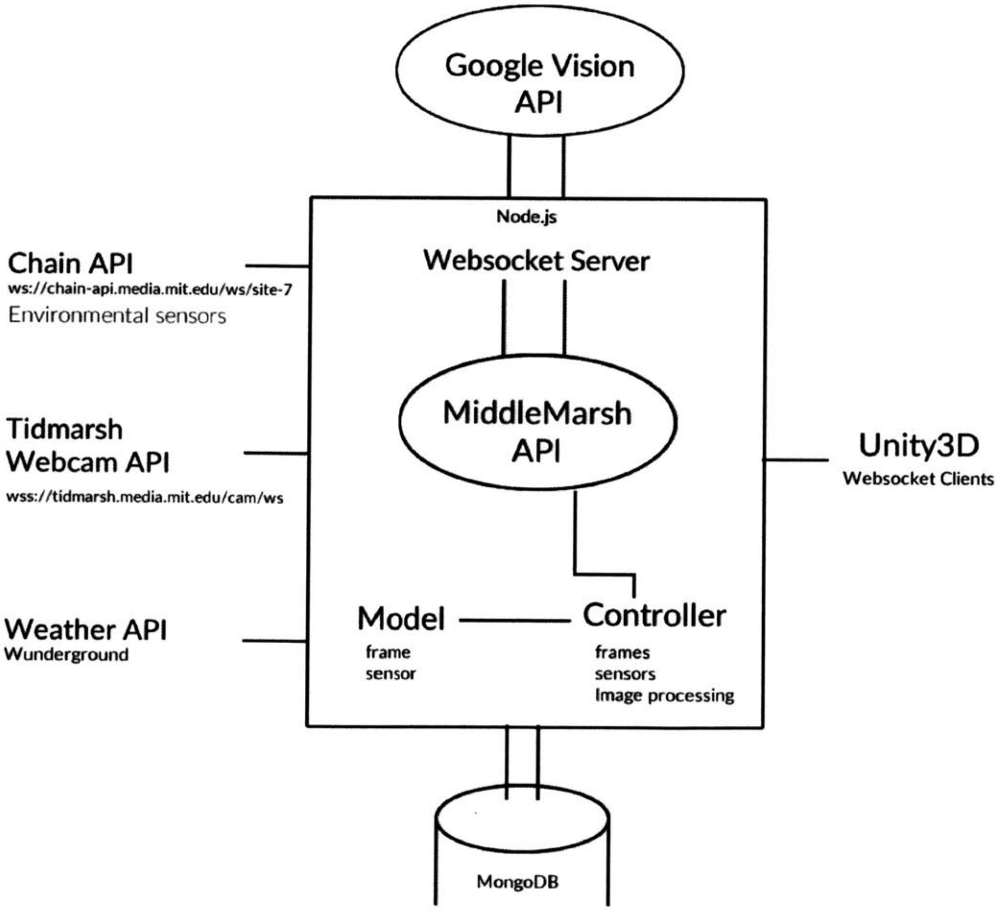


Figure 4-2: MiddleMarsh's system architecture diagram.

4.2 Mapping the Weather

Elements of the physical weather at Tidmarsh are mirrored by the virtual weather in Doppelmarsh through the use of environmental knowledge fused from different sources. The stream of data coming from the WSN contains real-time temperature, humidity, illuminance, and pressure of the mirco-climates that surround each sensor node. The base-station nodes are loaded with extra local weather stations measuring wind and precipitation, as well as capturing footages from various 1080p-HD webcams. The challenge in transforming real-time data into visual experiences lies in mapping. Good mapping usually emanates from metaphors.

4.2.1 From Wireless Sensor Networks

Temperature

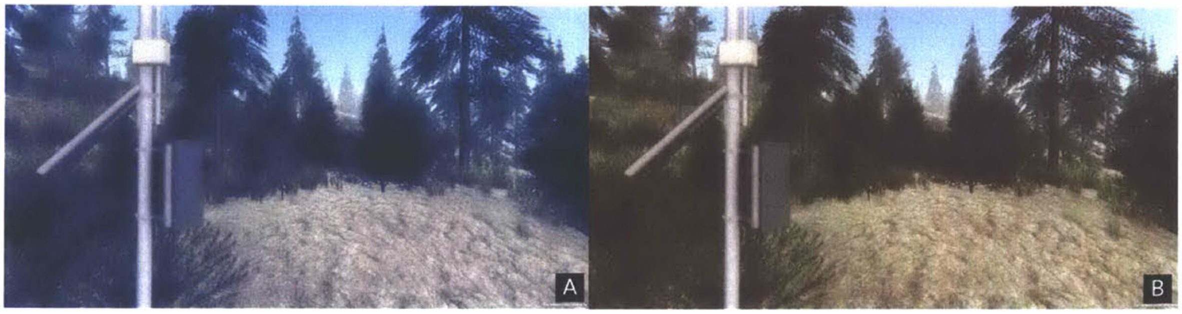


Figure 4-3: Screen capture from Doppelmarsh, showing the RGB filter mapping. Picture A shows a cold day on Tidmarsh (-12°C), whereas picture B shows the domination of the red color to show a hot summer day (29.4°C).

The first metaphor that comes to mind when thinking about visualizing temperature is color — red means hot and blue means cold. Almost every sink in the world relies on such a metaphor to convey temperature. It is difficult to track down the origin of this ubiquitous metaphor. We know from physics that when a metal is heated up, it turns red. Later, at very high temperatures, the metal becomes blue. So in reality, blue indicates hotter temperatures, at least in the case of metals. Despite this opposition with reality, the metaphor of cold=blue/red=hot sticks in people's minds and is widely used. In Doppelmarsh, the average temperature harvested from the site modulates the red and blue channels in the RGB filter found in Unity's post-processing stack, as shown in figure 4-3 and equations 4.1 and 4.2.

$$C = (10 - T)/67 \qquad \forall C > 0 \tag{4.1}$$

$$C = (T - 10)/67 \qquad \forall C > 0 \tag{4.2}$$

T: Temperature in °C incoming from live sensor data
C: The influence of the red or blue channel within the overall Unity's
post-processing color mixer

In the virtual environment, the shadows projected onto the snow appear to be more blue. The cooler the temperature, the more blue in tone the scene appears. The local-temperature also affects both the red and blue channels depending on the player's distance from any given node. A threshold sets the minimum distance that determines the nearest temperature sensor that affects the RGB filter. The change of color resulting from a player moving around the sensor network is very subtle, except for when passing near faulty sensors that probably require calibration or replacement.

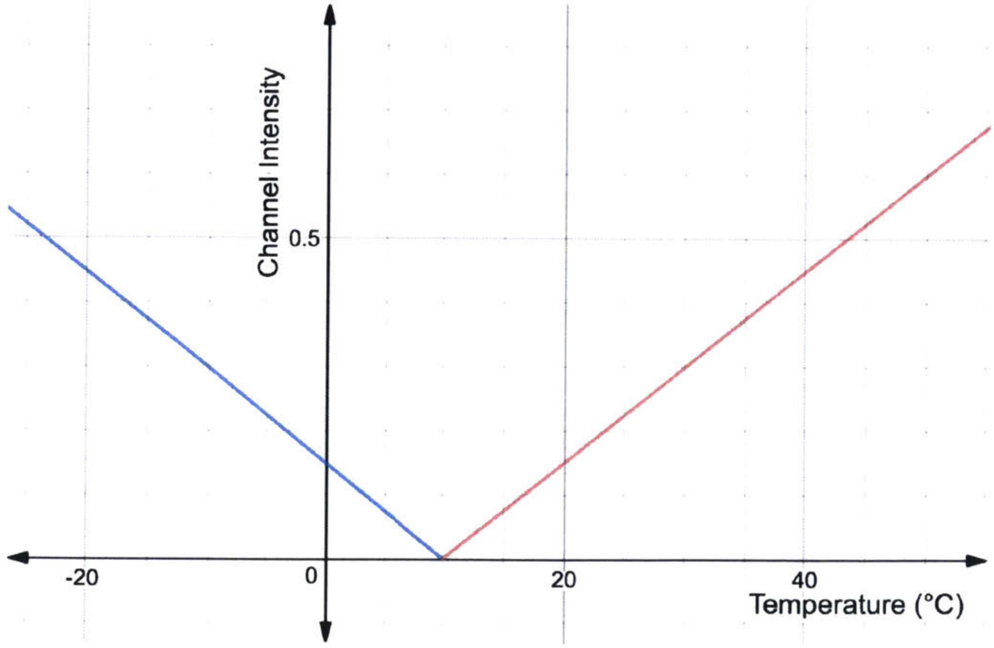
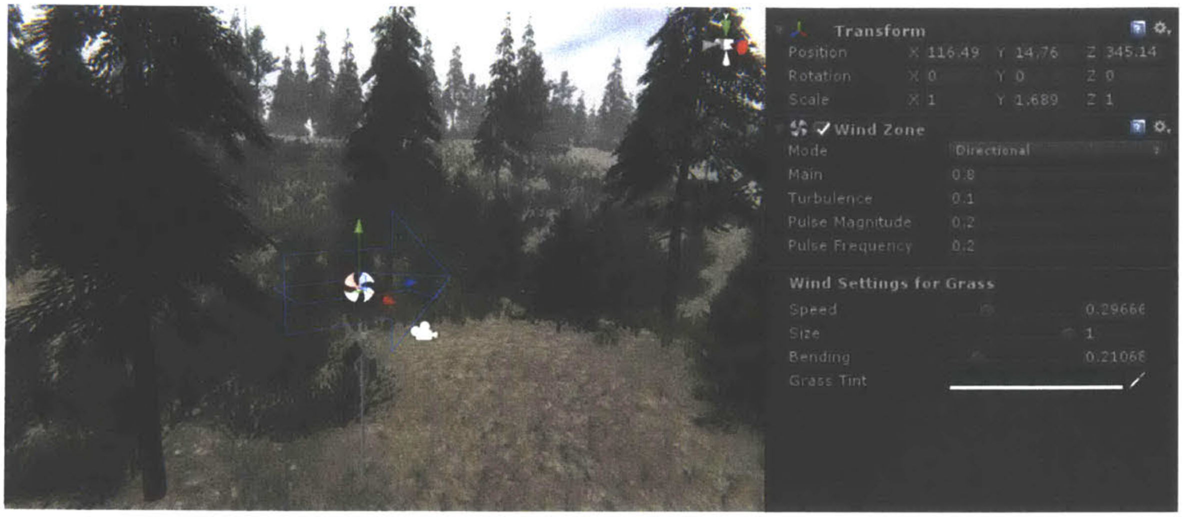


Figure 4-4: Red and Blue channels plot in function of the temperature.

Wind, Rain



Transform

Position X 116.49 Y 14.76 Z 345.14

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1.689 Z 1

☑ **Wind Zone**

Mode Directional

Main 0.8

Turbulence 0.1

Pulse Magnitude 0.2

Pulse Frequency 0.2

Wind Settings for Grass

Speed 0.29666

Size 1

Bending 0.21068

Grass Tint

Figure 4-5: Two main wind-zones in governs Doppelmarsh. The first is responsible for simulating the wind's effect on trees and objects, and the second controls the wind's effect on the grass. The reason to this redundancy is because the grass in Unity is rendered as a texture, whereas the trees are rendered as a mesh model.

The wind sensor is used to animate the virtual wind-zone which affects the global wind in the scene and affects the vegetation, trees, and clouds. The wind sensor gives two readings: wind speed and wind direction. Wind direction is mapped to the rotation of the wind-zone Gameobject and affects the orientation of the trees in the scene. Wind speed coming from local weather stations installed on each of the base-stations in Figure 2-4 is used to calculate the wind load and then is mapped to both the global wind zone and to the grass-specific wind zone. A separate zone was made to model the motion due to wind for the grass specifically using equation 4.3 and 4.4. The turbulence, on the other hand, is set to fluctuate randomly due to lack of a wind-gust measurements. The wind-zones parameters, shown in figure 4-5, are mapped to Tidmarsh's wind sensors using the generic formula 4.3 for wind load on specific objects [14].

$$F = A * P * Cd \tag{4.3}$$

$$P = 0.00256 * V^2 \tag{4.4}$$

A: The surface area exposed to the wind. Two areas were computed, one for a generic tree model and the other for a single medium length grass (ft^2)
P: The wind's pressure (psf) calculated by formula 4.4 using the real-time wind-speed
Cd: drag coefficient 1.2 for trees, since they resemble long cylinder, and 0.8 for grasses since they resemble short cylinders
V: wind speed from the wind-sensor (mph)

The generic formulas in 4.3 and 4.4 help calculate the wind-load as a function of the wind-speed as shown in graphs 4-7 and 4-6 with a standard size for a single grass and for a single tree trunk, respectively, in equations 4.5 and 4.6

$$y = (2ft * 0.2in/12ft)(0.00256 * x^2) * 0.8 \tag{4.5}$$

$$y = (6ft * 2in/12ft)(0.00256 * x^2) * 1.2 \tag{4.6}$$

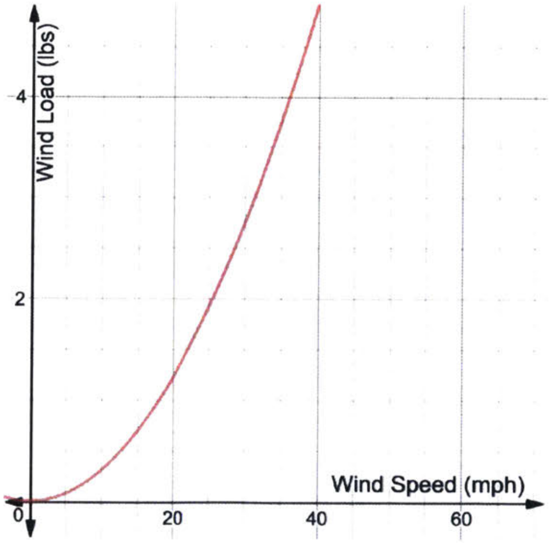


Figure 4-6: Plot showing the wind-speed's effect on the trees in Doppelmarsh.

Moreover, the local weather stations are equipped with rainfall sensors. The real-time readings of these sensors maps to the virtual rain in the scene in four discrete ways: no rain, light rain, rain, and heavy rain.

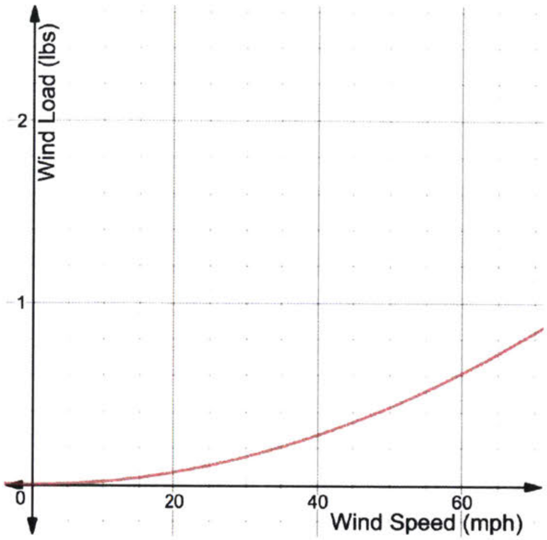


Figure 4-7: Plot showing the wind-speed's effect on the grass in Doppelmarsh.

4.2.2 From Cameras

Snow, Fog, Grass Color

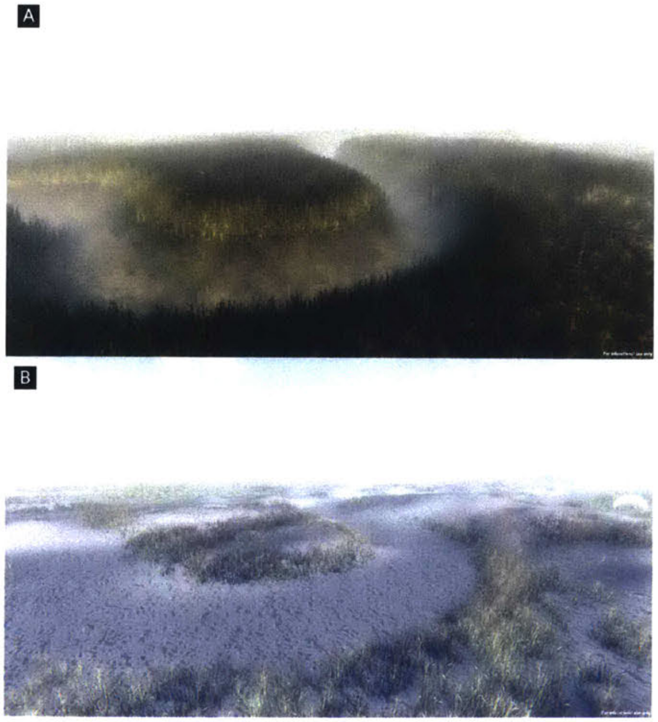


Figure 4-8: Screen shots from Doppelmarsh taken from the same angle. Picture A features a rainy, whereas Picture B shows the snow that covers the river during the extreme winters.

Each base station on Tidmarsh is equipped with a 1080p HD camera which gets triggered once every 10sec, or whenever movement is detected, to capture and transmit an image. These images are stored in an online directory, and a WebSocket server is developed to push this data to connected clients like MiddleMarsh. To analyze these images, MiddleMarsh connects to Google's Vision, a cloud-based API that makes it simple for developers to understand the content of an image by encapsulating powerful machine learning algorithms into a service. The rate is set to analyze an incoming image from Tidmarsh once every 10 minutes due to the limitations of Google's Vision zero-cost license. The Vision API returns keywords with probabilities, as well as the color palette of the image and other meta data. This information is then stored as frames in a MongoDB collection, along with other data fetched from the local weather station via Chain-API. Tidmarsh's WSN has been active since November 2014, but has experienced multiple cut-outs due to several issues ranging from severe weather conditions to battery replacements to Internet connectivity problems. To ensure a continuous virtual experience, MiddleMarsh fills in the blanks by connecting to Wunderground's Weather API and storing wind speed, wind direction, temperature, humidity, and the overall weather condition of Plymouth, Massachusetts. The conditions detected by both Google's Vision API and Wunderground's API are used to render the snow and the fog. The vegetations color palette's most dominant color is mapped to the color tint of the grass in Doppelmarsh.

4.3 Mapping Under Virtual Lenses

This section examines the effects of mapping wireless sensor data using the metaphor of virtual lenses. This metaphor allows a Doppelmarsh user to dynamically switch sensor-to-reality mapping from within the virtual environment. It also allows the creative freedom to explore and to experiment with additional artistic visuals and animations driven by the sensor network. Virtual lenses are used to render 3D multimodal data visualization in virtual environments. They can be toggled allowing the user to "see" extra information in a given environment. Two approaches are imple-

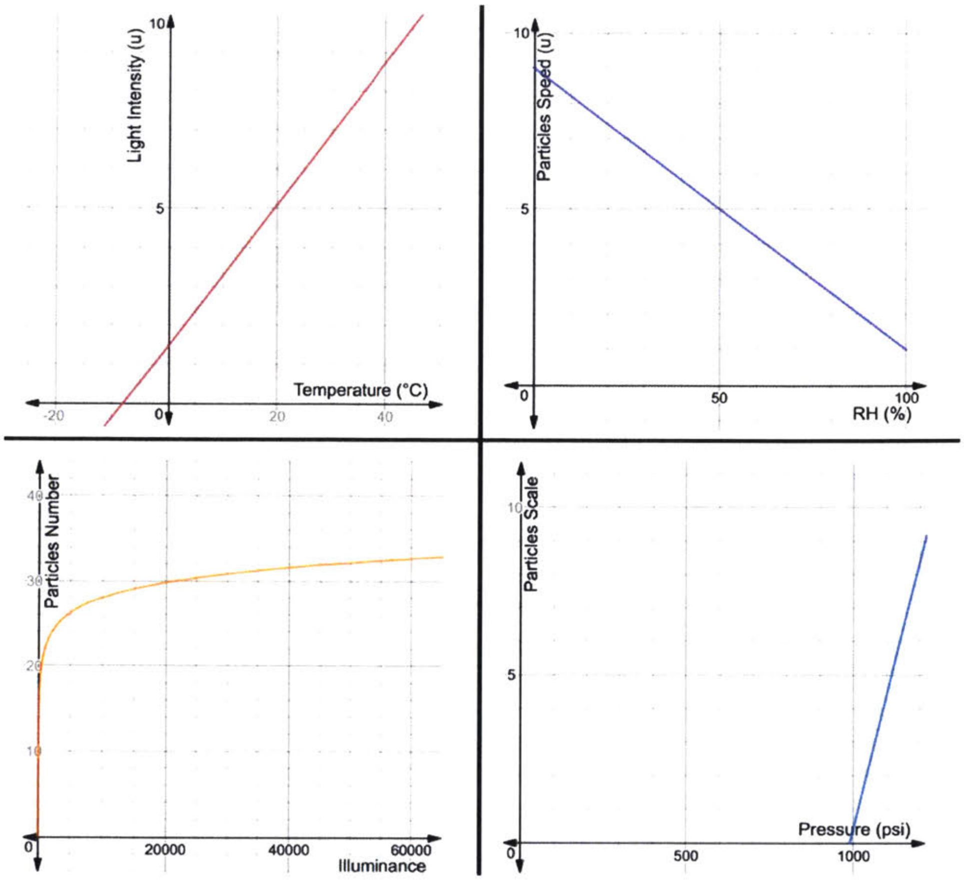


Figure 4-9: Four plots showing Sensor Vision's mapping done using a Light Object and a Particle System added to each sensor node in Doppelmarsh.

mented and tested in Doppelmarsh. The first, Sensor Vision, is used to animate the real-time data. The second, Synthetic Menagerie, visualizes historical sensor data which is embedded in the morphology of virtual creatures.

4.3.1 Sensor Vision

Sensor Vision is a multi-modal sensor browser in Doppelmarsh inspired by thermal imaging. A light source is added to every node in the scene, and, when activated, the temperature is mapped to the intensity of the light. A thermal-shader is then used to turn the player's camera into a thermal-like vision that shows heatmaps projected underneath each sensor node. Illuminance, pressure, and humidity are visualized using Unity's Particle System which renders falling bubbles on top of each node. The Illuminance is mapped to the number of particles, the humidity is mapped to the

Figure 4-10: Doppelmarsh at night with Sensor Vision mode. A live multi-modal sensor animation showing the real-time temperature, humidity, illuminance and pressure.

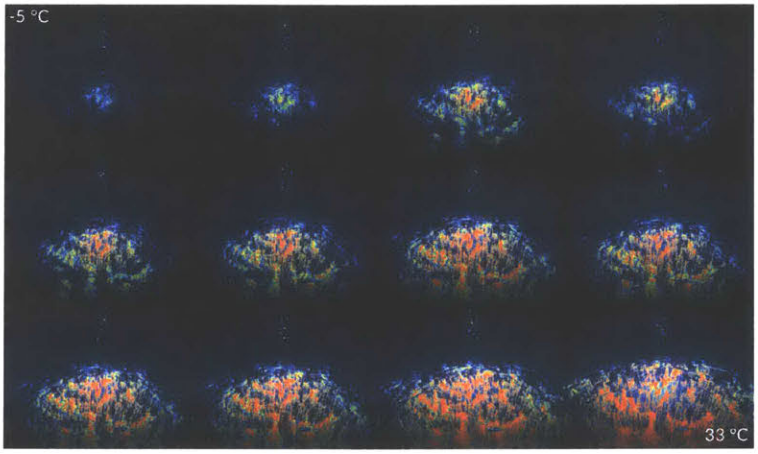


Figure 4-11: This figure shows different screenshots taken at different temperatures on a single node as viewed under the Sensor Vision lens in Doppelmarsh.

particles' speed, and the pressure defines the size of the particles. For instance, at dusk with high humidity few particles would be rendered and they would be animated at a slower pace.

4.3.2 Synthetic Menagerie

An important goal of Doppelmarsh is to make the environment evocative that stimulates the imagination. By mapping an energy metaphor to moving imaginary creatures that visualize state we intend to visualize longer-term changes in the environment embodied in the morphology of this "synthetic menagerie". At first, nearby sensor nodes are grouped into different zones that form areas in which creatures can

Figure 4-12: A synthetic creature spotted during the night, reflecting the average illuminance (LUX) aggregated from nearby sensors.

wander. The sensor data averaged from each zone affects the looks of a corresponding virtual animal. For instance, the average temperature and humidity affects the animal's fur. Average temperature sets fur length, and humidity controls its viscosity. The pressure is mapped to the animal's texture color, and the illuminance is mapped to the transparency of that color. The historical sensor data is collected using Chain-API, parsed and averaged by MiddleMarsh, and then stored in a MongoDB collection called Menagerie. This lens is used only to render animals roaming around the *Cell3, 4* site and was never deployed in other scenes of Doppelmarsh.

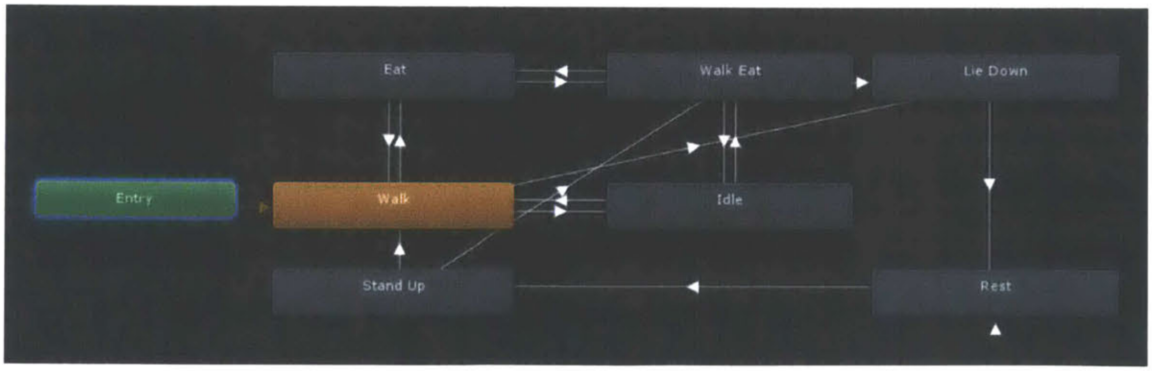


Figure 4-13: Showing the finite state machine diagram that controls the behavior of the virtual creatures in Doppelmarsh.

Morphology and Behavior

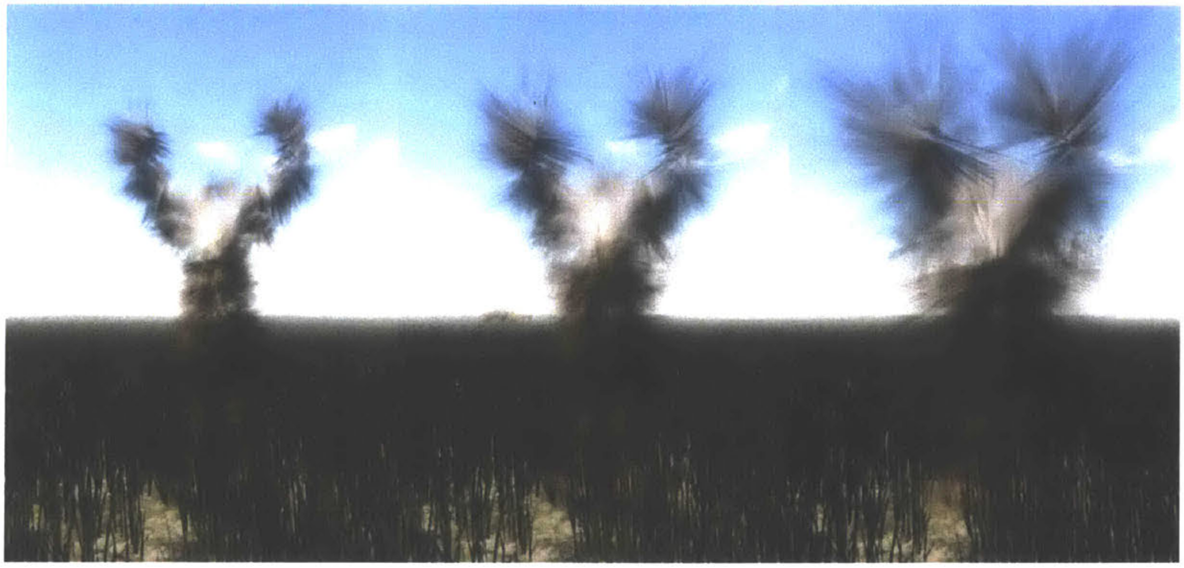


Figure 4-14: Mapping the temperature to the fur's length. Going from left to right shows a decrease in the averaged temperature.

Every creature has a finite state machine that dictates its behavior as shown in Figure 4-13. Creatures starts by patrolling between way-points, toggling randomly between states, in their designated zones that enclose neighborhoods of sensor devices. Each creature lives by consuming incoming sensor data from its own zone, and reflect real-time averaged data to generate changes in its fur's RGB color and transparency, length, and viscosity. Over time, the creatures morphology will be affected by the sensor data for the purposes of giving users of Doppelmarsh a feel for historical sensor data. The fur asset was downloaded from the Unity's assets store and tweaked to warp rigged 3D-models of deers and stags.

Creature's Vocalization

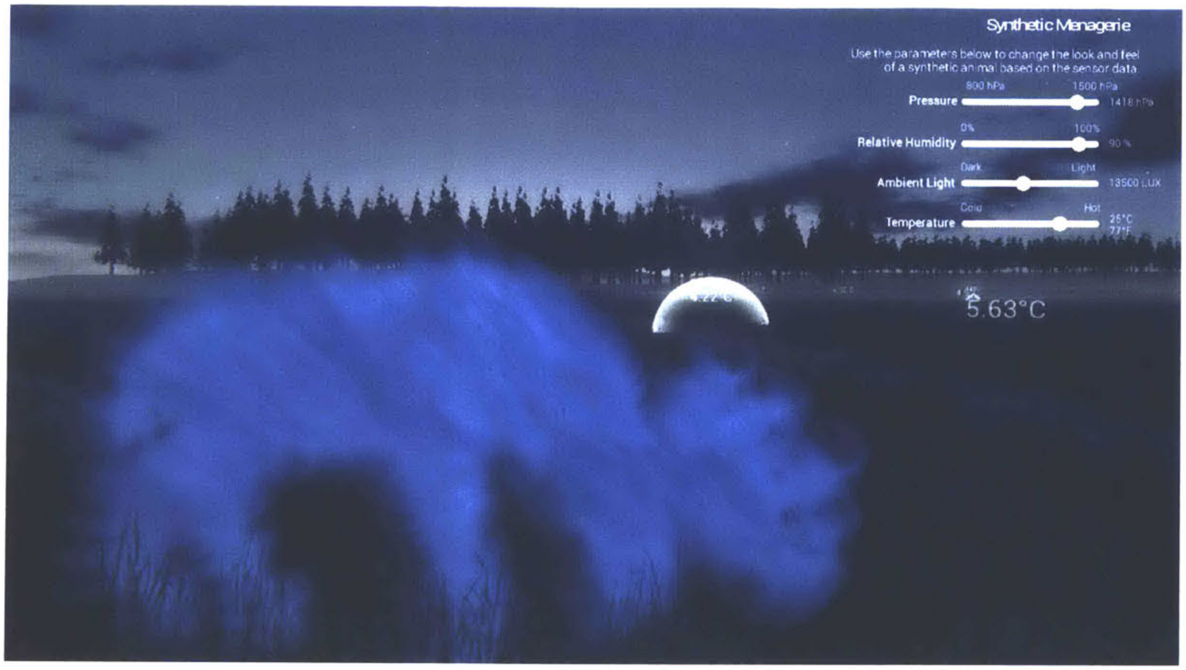
Doppelmarsh features a dynamic and spatial sound installation driven by the sensor nodes [36]. Sound samples generated from the creatures roaming Doppelmarsh add to its sonic space and in interesting ways. The music generated by the menagerie is mapped to each creature's movement in the scene. Whenever there's a collision between the creature's feet and the ground, a pitch shifted audio sample is played.

Figure 4-15: The look and feel of each creature signals historical sensor data to the explorer. Picture A, B and C contrast the look and feel under changes in pressure, humidity and temperature as seen by changes in fur color, stiffness, and viscosity. Picture D is taken form within the guts of a nocturnal creature.

Different samples are also triggered following the creature's internal state machine as figure 4-13 shows. For instance, when the creature is resting, a different sample gets played than when it's eating or walking. This finite state machine is also affected by the incoming stream of sensor data that define the behavior of the menagerie.

4.3.3 Summary

This section summarizes in Table 4.3.3 a list of the mapping done on the virtual weather in Doppelmarsh, as well as on the virtual-lenses Sensor Vision, and Synthetic Menagerie.



Use the parameters below to change the look and feel of a synthetic animal based on the sensor data

Pressure 1418 hPa

Relative Humidity 90 %

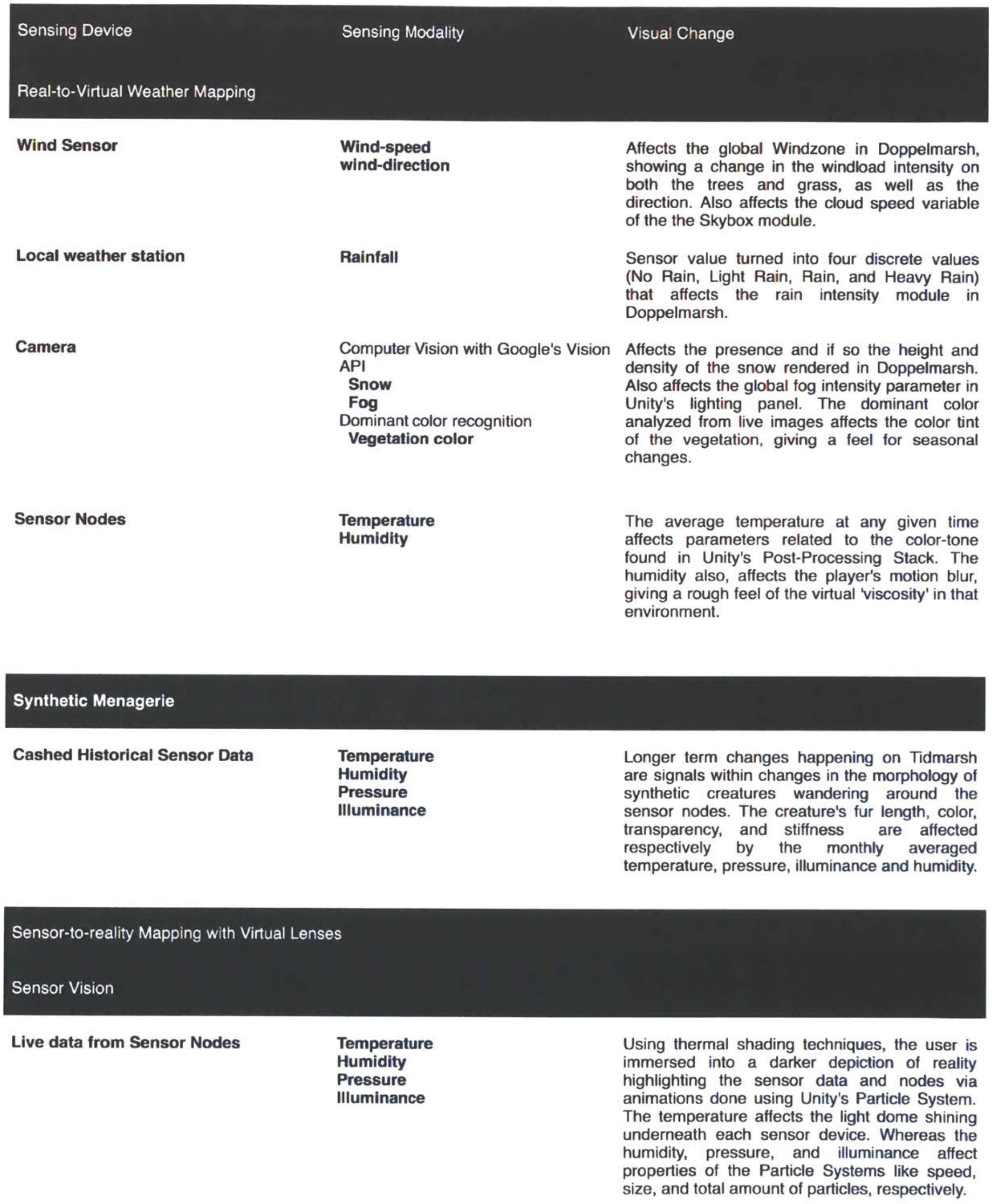
Ambient Light 13500 LUX

Temperature 25°C 77°F

Figure 4-16: A preview of the synthetic simulator showing a visible change in the creature's fur opacity, color length and stiffness in response to the change of sensor data.

Future Work

The work presented here develops an example of resynthesized reality and can be augment or extended through the integration of new sensing modalities and /or new visual mapping. For instance, a more focused mapping of seasonal changes using computer vision techniques of the phrenology might give us a more precise/integrate way to represent seasonal changes as well as the growth of many plant species. With dedicated effort to design custom computer vision tools using deep learning, such features could be detected. Another, and potentially more potent direction might be to distinguish visible changes in the pattern of each of the creature's skin texture, this could lead to a more aesthetic to a more visually attractive experience and a more in depth representation of historical data.



Sensing Device	Sensing Modality	Visual Change
Real-to-Virtual Weather Mapping		
Wind Sensor	**Wind-speed** **wind-direction**	Affects the global Windzone in Doppelmarsh, showing a change in the windload intensity on both the trees and grass, as well as the direction. Also affects the cloud speed variable of the the Skybox module.
Local weather station	**Rainfall**	Sensor value turned into four discrete values (No Rain, Light Rain, Rain, and Heavy Rain) that affects the rain intensity module in Doppelmarsh.
Camera	Computer Vision with Google's Vision API **Snow** **Fog** Dominant color recognition **Vegetation color**	Affects the presence and if so the height and density of the snow rendered in Doppelmarsh. Also affects the global fog intensity parameter in Unity's lighting panel. The dominant color analyzed from live images affects the color tint of the vegetation, giving a feel for seasonal changes.
Sensor Nodes	**Temperature** **Humidity**	The average temperature at any given time affects parameters related to the color-tone found in Unity's Post-Processing Stack. The humidity also, affects the player's motion blur, giving a rough feel of the virtual 'viscosity' in that environment.
Synthetic Menagerie		
Cashed Historical Sensor Data	**Temperature** **Humidity** **Pressure** **Illuminance**	Longer term changes happening on Tidmarsh are signals within changes in the morphology of synthetic creatures wandering around the sensor nodes. The creature's fur length, color, transparency, and stiffness are affected respectively by the monthly averaged temperature, pressure, illuminance and humidity.
Sensor-to-reality Mapping with Virtual Lenses Sensor Vision		
Live data from Sensor Nodes	**Temperature** **Humidity** **Pressure** **Illuminance**	Using thermal shading techniques, the user is immersed into a darker depiction of reality highlighting the sensor data and nodes via animations done using Unity's Particle System. The temperature affects the light dome shining underneath each sensor device. Whereas the humidity, pressure, and illuminance affect properties of the Particle Systems like speed, size, and total amount of particles, respectively.

Figure 4-17: Mapping sensors-to-reality Table that summarizes the link between the real and the virtual environment.

Chapter 5

User Interface & Experience

5.1 Desktop UI

5.1.1 First Person View

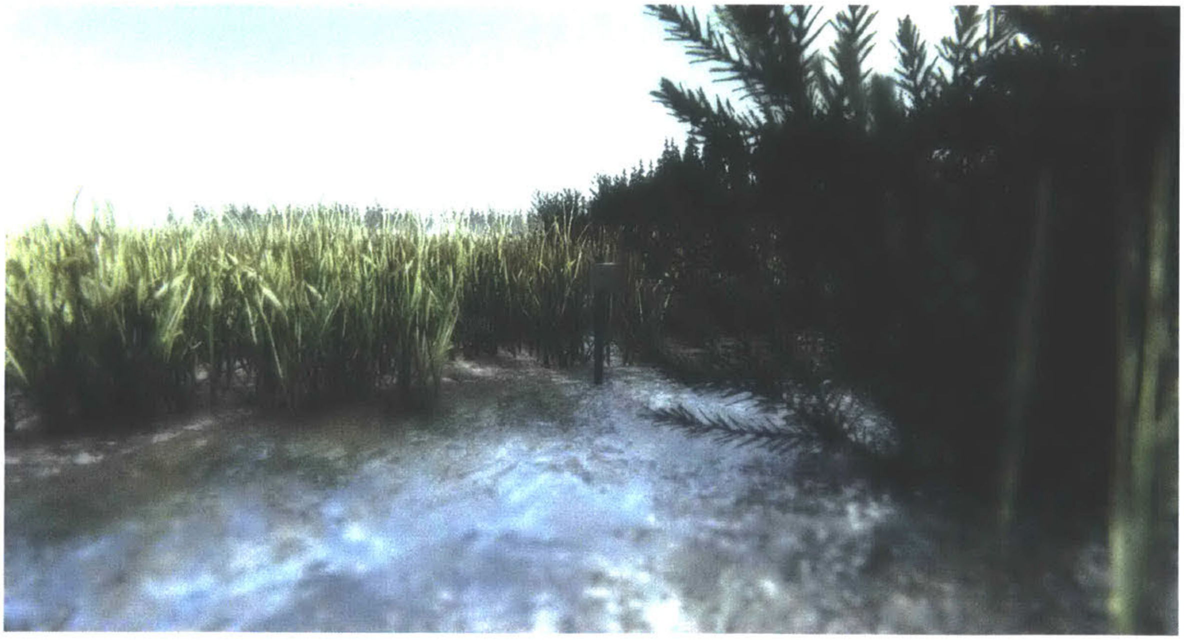
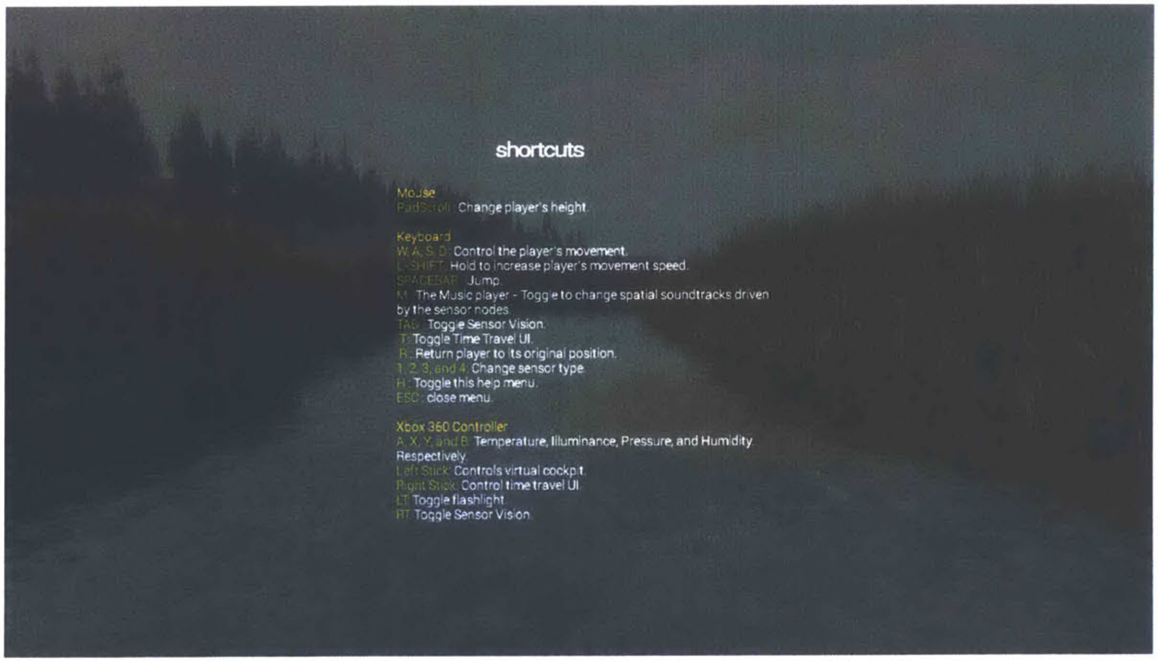


Figure 5-1: Showing Doppelmarsh from a first person view perspective.

Doppelmarsh is represented to the player/explorer in various ways. The first-person perspective is adopted to give the user control over a player/explorer in the scene. The player/explorer can use a keyboard and a mouse or a gaming joystick to navigate in the space. Unity's standard first-person-control asset provides basic player/explorer control in a 3D scene. The default keys on the keyboard that can be used to control the player's movement are the arrow keys, and the player's head position is controlled by the mouse. Similarly, on a gaming joystick, the two analogue sticks are used, one for the player's movement and the other for the player's head position.

Figure 5-2: Doppelmarsh's Timeline UI features two sliders. The first is used to change the day since October 2014 until the present moment, whereas the second is used to specify the time of the day.



shortcuts

Mouse
PadScroll: Change player's height.

Keyboard
W, A, S, D: Control the player's movement.
L SHIFT: Hold to increase player's movement speed.
SPACEBAR: Jump.
M: The Music player - Toggle to change spatial soundtracks driven by the sensor nodes.
TAB: Toggle Sensor Vision.
T: Toggle Time Travel UI.
R: Return player to its original position.
1, 2, 3, and 4: Change sensor type.
H: Toggle this help menu.
ESC: close menu.

Xbox 360 Controller
A, X, Y, and B: Temperature, Illuminance, Pressure, and Humidity Respectively.
Left Stick: Controls virtual cockpit.
Right Stick: Control time travel UI.
LT: Toggle flashlight.
RT: Toggle Sensor Vision.

Figure 5-3: Shortcuts menu showing the different key binding to both the keyboard, and the Xbox 360 game-controller.

A user can also explore Doppelmarsh in a temporal fashion. The keyboard shortcut "T" toggles the timeline UI, allowing a user to go back in time and experience the sensor data that drives the virtual environment. The user can also toggle between the different lenses mentioned in Chapter 4 using the "TAB" key on the keyboard, or the "RT" key on the gaming controller. Figure 5-3, shows a screenshot of the help menu in Doppelmarsh that features the rest of the key-bindings.

5.1.2 Real-Time-Strategy (RTS) View

A player/explorer traversing Doppelmarsh can also change the camera's perspective using the mouse's scroll-pad. Doppelmarsh in bird's-eye-view gives the user the ability to zoom out and look at the state of the whole network at once, whereas the first-person view provide a closer inspection of the data. When coupled with the sensor vision lens, introduced in chapter 4, this view informs the player/explorer about the overall distribution of the data. This perspective could also usefully control actuators or robots present on site. For instance, in the project Quadrasense the Doppelmarsh in RTS was used to control an unmanned air vehicle (UAV) on site.

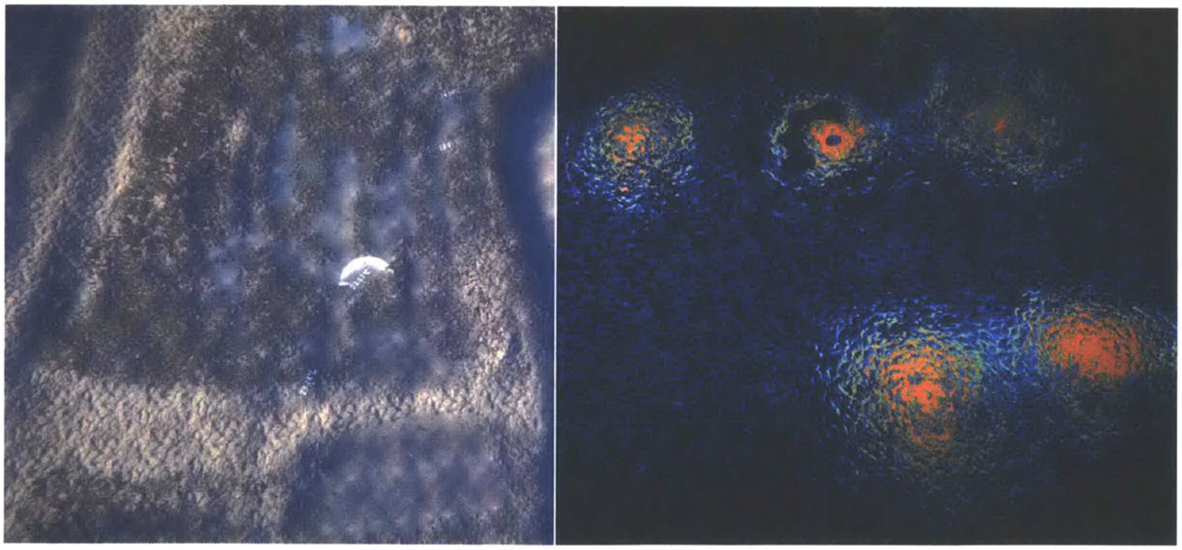


Figure 5-4: Doppelmarsh from a bird's eye view, with and without Sensor Vision.

5.2 VR UI

5.2.1 Virtual cockpit

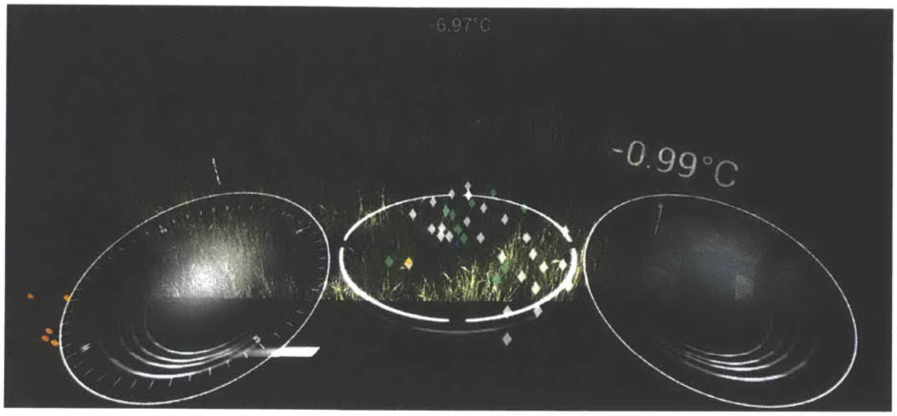


Figure 5-5: A view from within the virtual vehicle. The center mini-map shows online (green) and offline (gray) sensor nodes, whereas the minimap on the left shows the virtual creatures roaming Doppelmarsh.

Doppelmarsh is also optimized to run as a VR experience on both the Oculus Rift and the HTC Vive VR headsets. A cockpit model is used instead of the default first person view to allow more comfortable interactions with the system in VR. By placing players/explorers inside virtual cockpits, and giving them the ability to drive around, a smooth virtual navigation was achieved, without inducing any motion sickness. This vehicle also comes with a dashboards that shows different mini-maps of all of the sensor nodes as well as the player's position at its center as shown in figure 5-5. Other miscellaneous features were also added to the vehicle, like headlights for nocturnal explorations, and a perspective controller. The player/explorer can drive this virtual vehicle around the space to browse the sensor network, use the VR controllers to toggle between types of sensors, go into sensor-vision mode, or go back in time. The player/explorer controls this vehicle using either the Xbox 360 controller, or the default joysticks shipped with each VR system. The vehicle moves at a slow and constant speed to minimize motion sickness, as usually induced by poorly crafted VR experiences.

5.2.2 Sensor Vision Mode in VR

The metaphor of virtual lenses, introduced in Chapter 4, is also available in Doppelmarh's VR mode as shown in picture 5-6. Virtual explorers can toggle Sensor Vision mode with a key binded to either the trigger on a gaming joystick, or the button on a HMD controller.

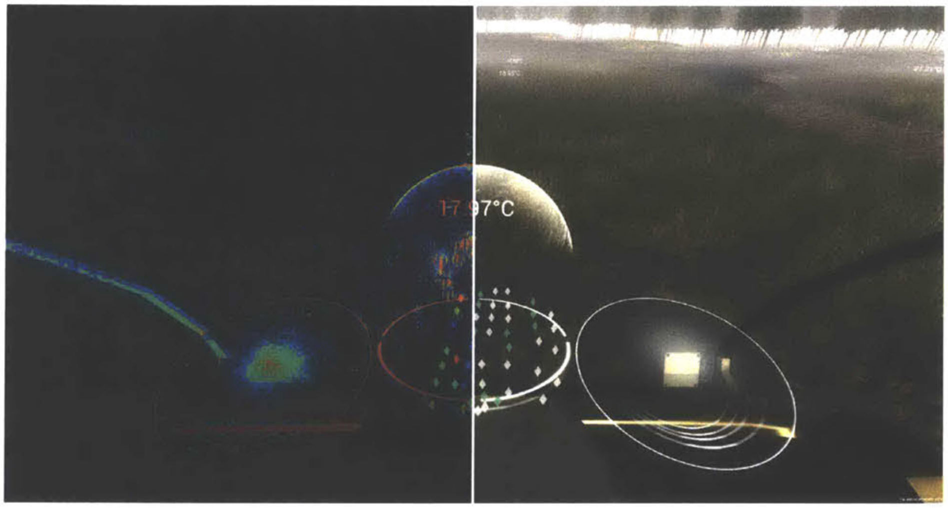


Figure 5-6: Another view from within the virtual vehicle showing a view with and without sensor-vision.

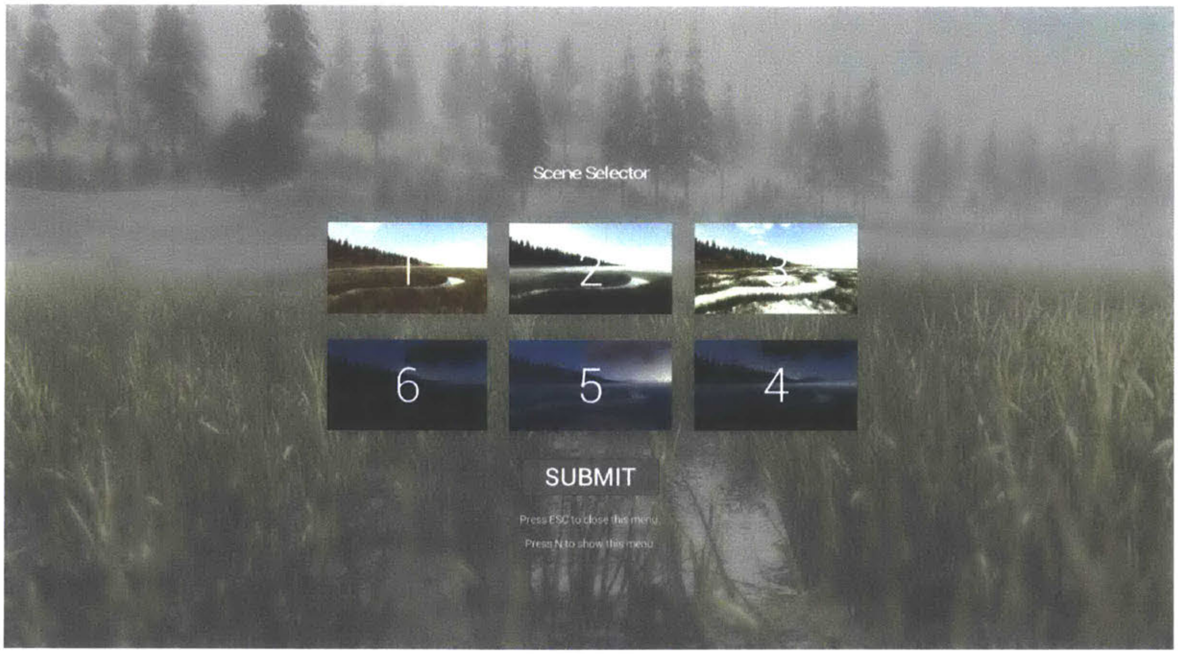
Chapter 6

Evaluation

6.1 Strategy for user study

6.1.1 Design & Implementation

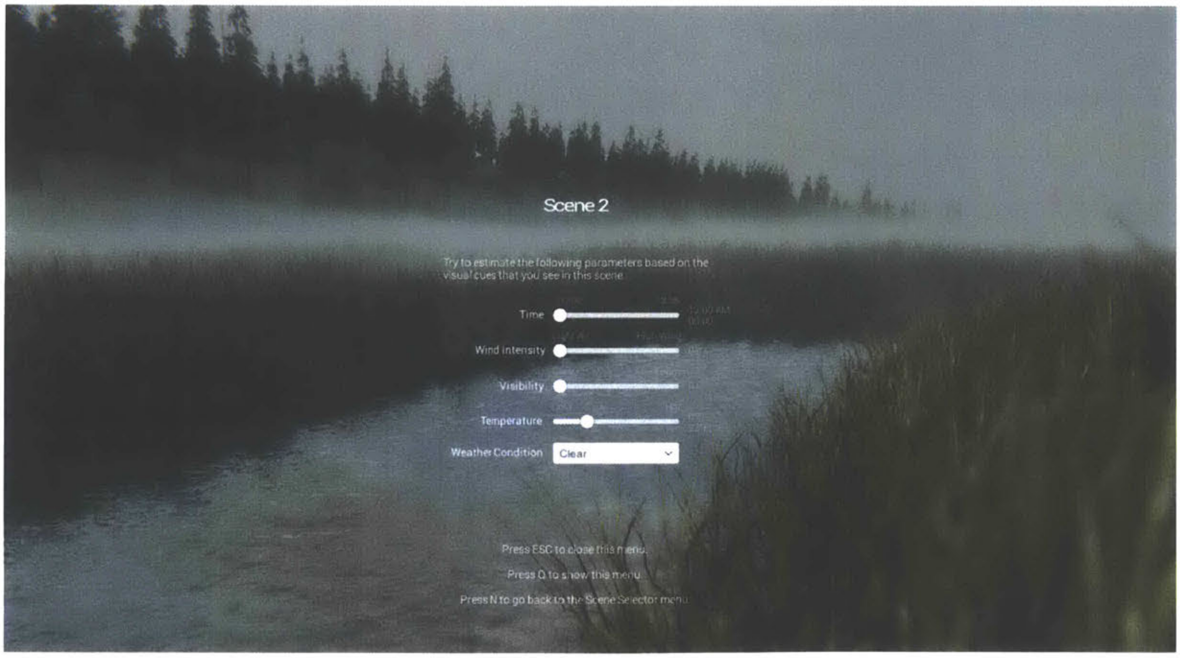
A user study was conducted to evaluate the mapping as realized in Doppelmarsh. The study opens with a welcome message that briefly introduces Tidmarsh, Doppelmarsh and the study. The user is then asked to input, using a slider, the average number of hours spent weekly on video games, as gamers would tend to have more experience in virtual environments and are likely to detect more information about this environment. A scene selector menu, shown in figure 6-1, pops in after getting the user's consent to take the study. The user is asked to browse six different scenes that shows Doppelmarsh in various weather conditions. The user is instructed to respond to five simple questions related to visual cues observed in each scene. After answering all of the questions, the submit button in the scene selector menu becomes active. The data generated from the study is sent via a POST request to a Google cloud form. This service also connects to Google spreadsheets, consolidating all the data, preparing it for analysis and graphing. A curated version of Doppelmarsh, running the user study, was built for computers running Windows, MacOS, and Linux. The study was distributed and taken by 10 users on their personal computers.



Press ESC to close this menu.

Press N to show this menu.

Figure 6-1: The user is asked to browser six different scenes found in Doppelmarsh.



Try to estimate the following parameters based on the visual cues that you see in this scene

Weather Condition: Clear

Press ESC to close this menu.

Press Q to show this menu.

Press N to go back to the Scene Selector menu.

Figure 6-2: The questions available in each scene. The user is asked to provide their opinion about the time of day, the wind intensity, the visibility, temperature, and the overall weather condition.

6.1.2 Results

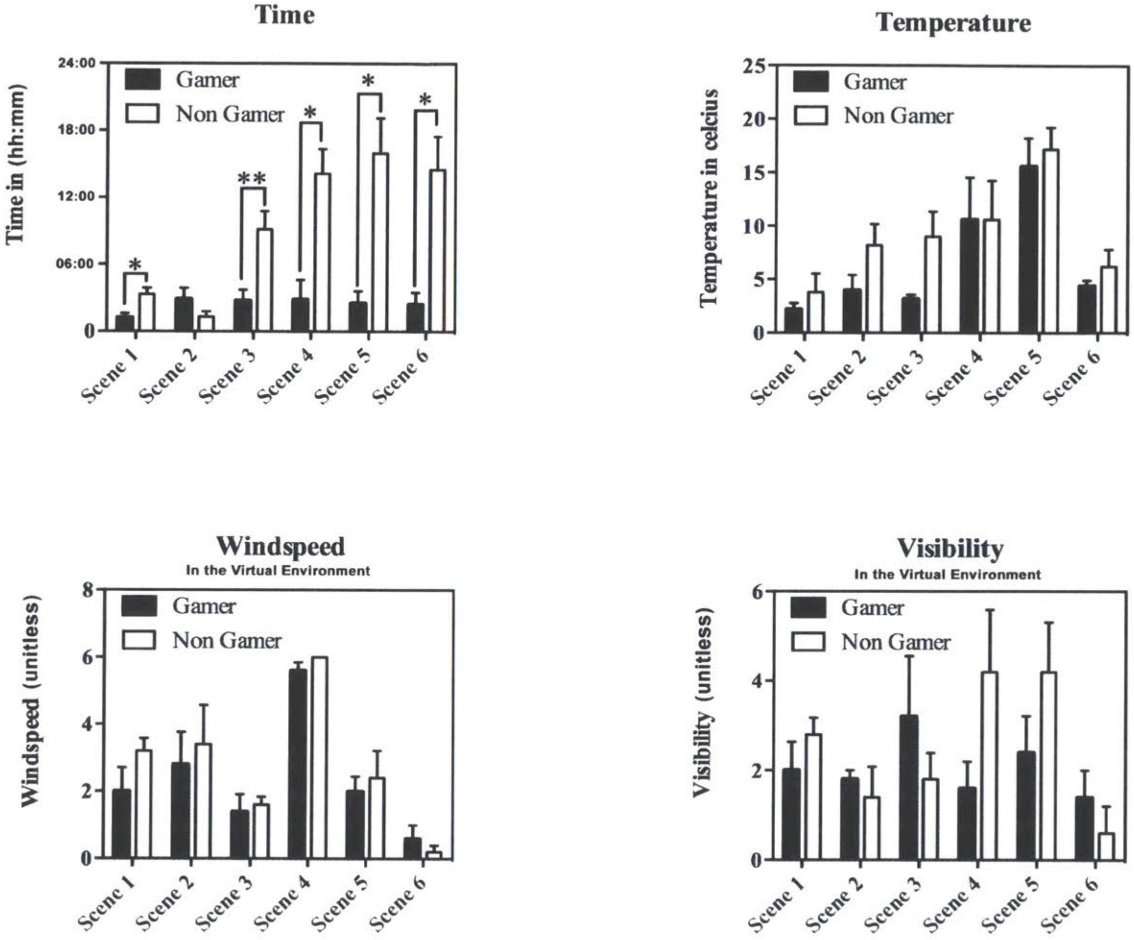


Figure 6-3: Showing 4 bar plots generated by the software Graphpad Prism.

The data generated from this user study was analyzed by Graphpad Prism. The sample was grouped into gamer or non-gamer. Group segregation was based on the amount of hours spent on video games per week. From the sample, five users turned out to be gamers, and five users turned out to be non-gamers. In the four bar plots shown in figure 6-3, the black bars represent the gamers, whereas the open bar represents the non-gamers.

Each bar represents the differential estimation (|Correct Value - Estimated Value|) of a variable (time of day, temperature, windspeed, condition) by both of these two groups. Different scenes with different conditions were tested and amounted to a total of six scenes. Gamers were noticeably better at estimating the correct time of the day, as shown in the first bar-plot in figure 6-3. Moreover, this was true in all of the scenes except for the second scene, where non-gamers were slightly better at answering the time. A similar trend is true for the temperature. In contrast, the wind speed and visibility revealed no differences between both groups as both were close enough to the correct answer.

6.1.3 Discussion

The striking difference ability of game players to identify time of day was a little bit surprising. For some reason, gamers have a better estimation of virtual time through graphical cues such as lighting in a given virtual environment. Time plays an important role as a core mechanic in many video-games. It could be that gamers have more flexibility in how they perceive time due to their practice. Another interesting view can be found by looking at the process that lead to these answers. In order to estimate the right time, a player/explorer needs to locate the position of the sun or the moon in the environment, and then try to estimate the time, based on life experiences. Gamers found a systematic approach to validate their guesses, whereas the average non-gamer lacked a way to solve this problem. Concerning the Temperature plot, the gamers seemed to be tricked in scene-3, and scene-4. What generated this confusion should be inspected carefully. The first scene conveyed a clear idea that it was hot. Both non-gamers and gamers provided similar answers that were reflective of the actual condition. Gamers responded with a little more accuracy though, but they were as off as non-gamers in providing the closest guess in scene-3 and scene-4. That might be due to some added ambiguity in the scenes with virtual snow, as it was poorly rendered. The windspeed, visibility and overall weather conditions were included in the study to make the players/explorers think about the aspects of environmental conditions that can help estimate a given temperature. The effect of wind and fog are easily perceived in any given outdoor environment, but the degree of accuracy in estimating the answers may have to do with the user's overall experience with physical environments. Also, it might be that these 2 variables are not closely correlated to whether a user is a gamer or not. Finally, one drawback of this study is the small sample size (n=10) that should be taken with a pinch of salt as it remains lucid and inconclusive.

6.2 Observation

Another way to evaluate Doppelmarsh's environment might focus on mere observation of the physical and virtual site from an identical perspective at an identical time. Looking at both the real and the virtual site from the same perspective can indeed tell a lot about the correctness of the mapping. The camera in Doppelmarsh was positioned at the same location as the onsite camera at Tidmarsh, the orientation of the camera was approximated to match the picture. Therefore six screenshots were captured from the virtual environment as shown in figure 6-4.

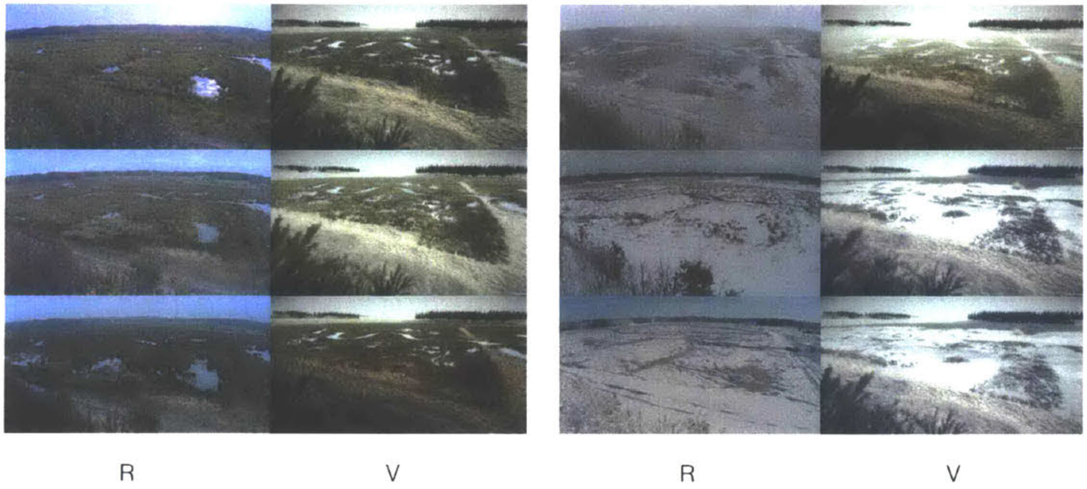


Figure 6-4: Showing six different images captured from Tidmarsh by the *Cell3, 4* camera, in parallel showing six scenes from Doppelmarsh from a similar angle.

Chapter 7

Conclusion

Figure 7-1: One of the mythical giraffe of Doppelmarsh that was later replaced by a mythical deer.

In this thesis we introduce the concept of "Resynthsized Reality" using the example environment, Doppelmarsh. This work draws from and extends earlier work in Cross Reality develops by students in the ResEnv. This work, showcased the processes needed to digitize clones of physical outdoor environments using remote sensing data such as LIDAR and deployed in the Unity game engine. When combined with real-time environmental sensor data, this static digitization of a physical landscape is transformed into a vast dynamic landscape which remote users can explore repeatedly and at a distance, unconstrained by the physics of human travel through time and space. To achieve this, a middleware agent called MiddleMarsh was built. This web service is responsible for tying the physical to the virtual. Multiple streams of sensor data and images flow through MiddleMarsh, where they are analyzed, processed and aggregated into frames. These fragments of reality are then filtered and cached in a NoSQL database. Clients running Doppelmarsh fetch real-time information from MiddleMarsh via WebSocket and via HTTP requests. This incoming stream of data is used to drive various animations and effects in Doppelmarsh.

A explorer/payer can explore Doppelmarsh in a range of styles borrowed from video-games like Real-Time-Strategy (RTS), First-Person-Shooter (FPS), and RPG (Role-Playing-Game), making Doppelmarsh a Serious Game that utilized game design concepts in non-gaming contexts. All of these perspectives are taken into account when presented to the explorer/player through different experiences. Doppelmarsh was used to run several applications that demonstrate the future role of ubiquitous sensing in virtual environments. One of these key-applications is *Telepresence* also known as *Presence.* Virtual explorers in Doppelmarsh can visit a real-site in Plymouth Massachusetts remotely, and explore it both spatially and temporally, while experiencing elements of its flora and fauna.

Doppelmarsh also introduced the idea of augmenting a user's visual apparatus in *Virtual Reality* with animations that are mapped to sensor data through the metaphor of virtual lenses. These lenses are designed to convey current or historical information of sensor data in an immersive and multi-modal fashion. This thesis also highlights the concept of a spatial sensor browser in *Virtual Reality* from within a virtual cockpit. The user drives a vehicle using the HMD-controllers, and watches the sensor network in action on the vehicle's dashboard and in the 3D environment. This vehicle moves at a constant speed with minimal acceleration to provide a smooth motion-sickness free experience.

To validate some of these implementations, a user study was conducted on a small sample of non-gamers and gamers. The study places users in different static scenes from Doppelmarsh and then asks them questions related to representation of the virtual weather. The findings from this study show that gamers were better at extracting the meaning of visual cues presented in the virtual environment, particularly in respect to their understanding of time. Although the sample size was small, the future potential of *Resynthesizing Reality* [1] as an approach to convey information about a physical landscape to avatars in virtual landscapes.

---

[1] Further documentation, source-code for both Doppelmarsh and MiddleMarsh, videos, and executables is found at *doppelmarsh.media.mit.edu*

Bibliography

[1] Espen Aarseth. Computer game studies, year one. *Game studies*, 1(1):1–15, 2001.

[2] Clark C Abt. *Serious games.* University Press of America, 1987.

[3] Edith K Ackermann. Programming for the natives: What is it? whats in it for the kids? *Tecnologias, Sociedade e Conhecimento*, 1(1):133–149, 2013.

[4] António Andrade, David Gouveia, Paula Escudeiro, and Carlos Vaz de Carvalho. Can sme managers learn from games? In *Proceedings of the XV International Conference on Human Computer Interaction*, Interacción '14, pages 96:1–96:5, New York, NY, USA, 2014. ACM.

[5] L. Barboni and M. Valle. Experimental analysis of wireless sensor nodes current consumption. In *2008 Second International Conference on Sensor Technologies and Applications (sensorcomm 2008)*, pages 401–406, Aug 2008.

[6] Jonathan E Campbell. *Geographic Information System Basics.* The Saylor Foundation, 2015.

[7] Amanda Chaffin, Katelyn Doran, Drew Hicks, and Tiffany Barnes. Experimental evaluation of teaching recursion in a video game. In *Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games*, pages 79–86. ACM, 2009.

[8] C. Chen, J. Yan, N. Lu, Y. Wang, X. Yang, and X. Guan. Ubiquitous monitoring for industrial cyber-physical systems over relay- assisted wireless sensor networks. *IEEE Transactions on Emerging Topics in Computing*, 3(3):352–362, Sept 2015.

[9] David Cohen. *The development of play.* Routledge, 2007.

[10] Armand D'Angour. Plato and play: Taking education seriously in ancient greece. *American Journal of Play*, 5(3):293, 2013.

[11] Sebastian Deterding, Miguel Sicart, Lennart Nacke, Kenton O'Hara, and Dan Dixon. Gamification. using game-design elements in non-gaming contexts. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, pages 2425–2428, New York, NY, USA, 2011. ACM.

[12] REST Django. Framework. *Dosegljivo: http://www. django-restframework. org*, 2016.

[13] Damien Djaouti, Julian Alvarez, Jean-Pierre Jessel, and Olivier Rampnoux. Origins of serious games. In *Serious games and edutainment applications*, pages 25–43. Springer, 2011.

[14] Hugh Latimer Dryden and George Cooper Hill. *Wind Pressures on Structures...* US Government Printing Office, 1926.

[15] Gershon Dublon and Joseph A Paradiso. extra sensory perceptıon. *Scientific american*, 2014.

[16] Gershon Dublon, Laurel S Pardue, Brian Mayton, Noah Swartz, Nicholas Joliat, Patrick Hurst, and Joseph A Paradiso. Doppellab: Tools for exploring and harnessing multimodal sensor network data. In *Proc. IEEE Sensors 2011*, pages 1612–1615, 2011.

[17] Unity Game Engine. Unity game engine-official site. *Online][Cited: October 9, 2008.] http://unity3d. com*, pages 1534–4320.

[18] Steven K Feiner. Augmented reality: A new way of seeing. *Scientific American*, 286(4):48–55, 2002.

[19] Roy Thomas Fielding. *Architectural styles and the design of network-based software architectures.* PhD thesis, University of California, Irvine, 2000.

[20] Rosemary Garris, Robert Ahlers, and James E Driskell. Games, motivation, and learning: A research and practice model. *Simulation & gaming*, 33(4):441–467, 2002.

[21] M. Gerla, E. K. Lee, G. Pau, and U. Lee. Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds. In *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pages 241–246, March 2014.

[22] V. C. Gungor and G. P. Hancke. Industrial wireless sensor networks: Challenges, design principles, and technical approaches. *IEEE Transactions on Industrial Electronics*, 56(10):4258–4265, Oct 2009.

[23] Jan Herling and Wolfgang Broll. Advanced self-contained object removal for realizing real-time diminished reality in unconstrained environments. In *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, pages 207–212. IEEE, 2010.

[24] D. H. Hoang, T. Strufe, Q. D. Le, P. T. Bui, T. N. Pham, N. T. Thai, T. D. Le, and I. Schweizer. Processing and visualizing traffic pollution data in hanoi city from a wireless sensor network. In *38th Annual IEEE Conference on Local Computer Networks - Workshops*, pages 48–55, Oct 2013.

[25] Native Instruments. Reaktor modular dsp software lab. *Web resource: https://www.native-instruments.com/en/products/komplete/synths/reaktor-6/*, 1996.

[26] Y. S. Jeong, Y. J. Chung, and J. H. Park. Visualisation of efficiency coverage and energy consumption of sensors in wireless sensor networks using heat map. *IET Communications*, 5(8):1129–1137, May 2011.

[27] A. Kansal, S. Nath, J. Liu, and F. Zhao. Senseweb: An infrastructure for shared sensing. *IEEE MultiMedia*, 14(4):8–13, Oct 2007.

[28] Mike Kelly. Json hypertext application language. 2016.

[29] S. W. Kim, C. W. Lee, W. H. Han, S. J. Baek, E. H. Song, and Y. S. Jeong. Real-time monitoring of multi mobile objects with usn and gml. In *2008 Second International Conference on Future Generation Communication and Networking*, volume 1, pages 110–115, Dec 2008.

[30] Richard Leacock. *The feeling of being there: a filmmaker's memoir.* Semeïon Editions, 2011.

[31] Qiansheng Li, Gershon Dublon, Brian Mayton, and Joseph A Paradiso. MarshVis : Visualizing Real-Time and Historical Ecological Data from a Wireless Sensor Network. In *IEEE VIS Arts Program (VISAP)*, Chicago, Illinois, 2015.

[32] Joshua Lifton, Mark Feldmeier, Yasuhiro Ono, Cameron Lewis, and Joseph A Paradiso. A platform for ubiquitous sensor deployment in occupational and domestic environments. In *IPSN2007*, pages 119–127. ACM, 2007.

[33] Joshua Lifton, Mathew Laibowitz, Drew Harry, Nan-Wei Gong, Manas Mittal, and Joseph A Paradiso. Metaphor and manifestation: Cross-reality with ubiquitous sensor/actuator networks. *IEEE Pervasive Computing*, 8(3), 2009.

[34] Joshua Lifton and Joseph A Paradiso. Dual reality: Merging the real and virtual. In *International Conference on Facets of Virtual Environments*, pages 12–28. Springer, 2009.

[35] Joshua Harlan Lifton. *Dual reality: an emerging medium.* PhD thesis, Massachusetts Institute of Technology, 2007.

[36] Evan F Lynch. *SensorChimes: musical mapping for sensor networks toward augmented acoustic ecosystem.* PhD thesis, Massachusetts Institute of Technology, 2016.

[37] Steve Mann. Mediated reality with implementations for everyday life. *Presence Connect*, 1, 2002.

[38] Steve Mann et al. Wearable, tetherless computer–mediated reality: Wearcam as a wearable face–recognizer, and other applications for the disabled. In *Presentation at the American Association of Artificial Intelligence, 1996 Symposium. Retrieved January*, volume 21, page 2002, 1996.

[39] Massachussetts Audubon Society. Discover Tidmarsh: Take a Walk on the Wildflower Side. *Mass Audubon Explore Newsletter*, March 2017.

[40] Terrence Masson, Ken Perlin, et al. Holo-doodle: an adaptation and expansion of collaborative holojam virtual reality. In *ACM SIGGRAPH 2017 VR Village*, page 9. ACM, 2017.

[41] Brian Mayton, Gershon Dublon, Spencer Russell, Evan F. Lynch, Don Derek Haddad, Vasant Ramasubramanian, Clement Duhart, Glorianna Davenport, and Joseph A. Paradiso. The networked sensory landscape: Capturing and experiencing ecological change across scales. MIT Press, 2017.

[42] Marshall McLuhan. *Understanding media: The extensions of man.* MIT press, 1994.

[43] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.

[44] Marvin Minsky. K-lines: A theory of memory. *Cognitive science*, 4(2):117–133, 1980.

[45] Marvin Minsky. Telepresence. 1980.

[46] Kazuo Misue, Peter Eades, Wei Lai, and Kozo Sugiyama. Layout adjustment and the mental map. *Journal of Visual Languages & Computing*, 6(2):183–210, 1995.

[47] M. Mittal and J. A. Paradiso. Ubicorder: A mobile device for situated interactions with sensor networks. *IEEE Sensors Journal*, 11(3):818–828, March 2011.

[48] Markus Montola, Jaakko Stenros, and Annika Waern. *Pervasive games: theory and design.* Morgan Kaufmann Publishers Inc., 2009.

[49] A. Moreno, A. G. A. Mujika, and . Segura. Visual analytics of multi-sensor weather information georeferenciation of doppler weather radar and weather stations. In *2014 International Conference on Information Visualization Theory and Applications (IVAPP)*, pages 329–336, Jan 2014.

[50] K. T. Murata, K. Muranaga, K. Yamamoto, Y. Nagaya, P. Pavarangkoon, S. Satoh, T. Mizuhara, E. Kimura, O. Tatebe, M. Tanaka, and S. Kawahara. Real-time 3d visualization of phased array weather radar data via concurrent processing in science cloud. In *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 1–7, Oct 2016.

[51] Dennis C Neale and John M Carroll. The role of metaphors in user interface design. *Handbook of human-computer interaction*, 2:441–462, 1997.

[52] Keith V Nesbitt and Carsten Friedrich. Applying gestalt principles to animated visualizations of network data. In *Information Visualisation, 2002. Proceedings. Sixth International Conference on*, pages 737–743. IEEE, 2002.

[53] Seymour Papert. Does easy do it? children, games, and learning. *Game developer magazine*, 1988.

[54] Joseph A Paradiso and James A Landay. Guest editors' introduction: Cross-reality environments. *IEEE Pervasive Computing*, 8(3), 2009.

[55] David Sidney Parlett. *The Oxford history of board games.* Oxford University Press, USA, 1999.

[56] Behandelt PostgreSQL. Postgresql. *Web resource: http://www. PostgreSQL. org/about*, 2010.

[57] Marc Prensky and Mark Prensky. *Digital game-based learning*, volume 1. Paragon house St. Paul, MN, 2007.

[58] Propellerheads. Reason dsp audio software. 2000.

[59] Vasant Ramasubramanian. *Quadrasense: immersive UAV-based cross-reality environmental sensor networks.* PhD thesis, Massachusetts Institute of Technology, 2015.

[60] Armin Ronacher. Flask (a python microframework), 2015.

[61] Spencer Russell, Gershon Dublon, and Joseph A. Paradiso. HearThere: Networked Sensory Prosthetics Through Auditory Augmented Reality. In *proceedings of the ACM Augmented Human International Conference*, 2016.

[62] Spencer Russell and Joseph A Paradiso. Hypermedia APIs for sensor data: a pragmatic approach to the web of things. pages 30–39, 2014.

[63] F. Salim, M. D. Pena, Y. Petrov, N. Sony, B. Wu, and A. A. Saad. Envis tag, scan, view: A location-based app for visualizing spatio-temporal data from sensor cloud. In *2014 IEEE 15th International Conference on Mobile Data Management*, volume 1, pages 329–332, July 2014.

[64] Ben Sawyer and David Rejeski. Serious games: Improving public policy through game-based learning and simulation, 2002.

[65] Christopher Stapleton and Jim Davies. Imagination: The third reality to the virtuality continuum. In *Mixed and Augmented Reality-Arts, Media, and Humanities (ISMAR-AMH), 2011 IEEE International Symposium On*, pages 53–60. IEEE, 2011.

[66] Jonathan Steuer. Defining virtual reality: Dimensions determining telepresence. *Journal of communication*, 42(4):73–93, 1992.

[67] Robert J Stone. Serious gamingvirtual realitys saviour. In *Proceedings of Virtual Systems and MultiMedia annual conference, VSMM*, pages 773–786, 2005.

[68] George M Stratton. Some preliminary experiments on vision without inversion of the retinal image. *Psychological review*, 3(6):611, 1896.

[69] Ivan E Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 757–764. ACM, 1968.

[70] Charles William Sutton. *Transactions of the Lancashire and Cheshire Antiquarian Society*, volume 1. 1893.

[71] Robert Szewczyk, Eric Osterweil, Joseph Polastre, Michael Hamilton, Alan Mainwaring, and Deborah Estrin. Habitat monitoring with sensor networks. *Communications of the ACM*, 47(6):34–40, 2004.

[72] Dave Szulborski. *This is not a game: A guide to alternate reality gaming.* Incunabula, 2005.

[73] Dejan Todorovic. Gestalt principles. *Scholarpedia*, 3(12):5345, 2008.

[74] Unity. Occlusion culling in unity.
URL: https://docs.unity3d.com/Manual/OcclusionCulling.html, 2017.

[75] Unity. Performance optimization checklist in unity 5.6.
URL: https://docs.unity3d.com/Manual/OptimizingGraphicsPerformance.html, 2017.

[76] Unity. Unity 5.6 global illumination documentation.
URL: https://docs.unity3d.com/Manual/GIIntro.html, 2017.

[77] Unity. Unity 5.6 post-processing documentation.
URL: https://docs.unity3d.com/Manual/PostProcessing-Stack.html, 2017.

[78] Gary Ushaw, Richard Davison, Janet Eyre, and Graham Morgan. Adopting best practices from the games industry in development of serious games for health. In *Proceedings of the 5th International Conference on Digital Health 2015*, DH '15, pages 1–8, New York, NY, USA, 2015. ACM.

[79] Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 319–326, New York, NY, USA, 2004. ACM.

[80] Phil Wilkinson. A brief history of serious games. In *Entertainment Computing and Serious Games*, pages 17–41. Springer, 2016.

[81] Alec Woo, Siddharth Seth, Tim Olson, Jie Liu, and Feng Zhao. A spreadsheet approach to programming and managing sensor networks. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*, IPSN '06, pages 424–431, New York, NY, USA, 2006. ACM.

[82] H. Zhou, K. Han, and X. Wei. Research on and realization of interactive wireless monitoring and management system of processed grain based on web3d. In *2015 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration*, pages 194–200, Dec 2015.

[83] Jiangwei Zhou, Yu Chen, Ben Leong, and Pratibha Sundar Sundaramoorthy. Practical 3d geographic routing for wireless sensor networks. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 337–350, New York, NY, USA, 2010. ACM.