An Immersive Virtual Reality Navigational Tool for Diagnosing and Treating Neurodegeneration

by

Paul J. F. White

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfilment of the requirements of the degree of

MASTER OF SCIENCE

Department of Biomedical Engineering

University of Manitoba

Winnipeg

Abstract

One of the earliest symptoms of Alzheimer's Disease (AD) is a loss of spatial navigation. In this work, we improved an existing screening test for AD that analyzed a patient's spatial navigation ability. The existing screening test was made more immersive, and therefore more reliable, by integrating support for a leading-edge consumer-targeted Head-Mounted Display (HMD). This integration brought some technical and usability challenges, that were addressed. Furthermore, we investigated the rehabilitative potential of Virtual Reality Navigational (VRN) activities in two case studies: an Early Stage AD (ESA) participant and a Late Stage AD (LSA) participant. We found that the ESA participant was able to significantly improve his navigation skills, and we observed some qualitative improvements in memory and navigation in his personal life. The LSA participant did not improve noticeably at the VRN tasks, but his mood improved after participating in the treatment sessions. These case studies suggested that VRN treatment may be beneficial for people with AD, especially at the onset stage.

Acknowledgements

There are a large, but countable, number of people to whom I owe a debt of gratitude for helping me over the course of my research. Since it would be impractical to list all of them here, I will limit my thanks to those who I feel contributed directly.

Professionally, I wish to thank my advisor, Dr. Zahra Moussavi, for encouraging me to pursue this Master's Degree, even though it was not my original plan.

I wish to thank Riverview Health Centre for allowing us to use the Day Hospital common room for the experiments described in Chapter III.

I wish to thank Ahmad Byagowi; your wonder and curiosity of the world around you and your insatiable desire to hack everything you touch to work just a little bit better is an inspiration to me and your fellow engineers.

I also wish to thank the other VR/Unity trailblazers in the BME lab, Kazu, Cassandra. You remind me that no work worth doing is done in a vacuum, and together, we ensured that the VRN Chair prefab and accompanying software would be understandable even for future students. I would also like to thank Omid, for his work with the statistical analysis comparing the HMD with conventional displays.

Finally, I wish to thank the rest of the staff at the University of Manitoba that have given me the chance to prove myself in challenging but hugely rewarding projects, either during summer work or during regular classes. Dr. Kinsner, Dr. Wu, Dr. Friesen, Dr. McLeod and once again, Dr. Moussavi.

On a personal level, I wish thank my parents and brother for supporting me materially; but much more importantly, emotionally and spiritually. You have given me a healthy home to grow up in, and by always encouraging me to do my best, have given me many chances to succeed.

I also wish to personally thank my friends, Jonathan, Thomas, and Kyle. You guys always challenge me to be better, in video games or in school or in work, by being 1 or 2 steps ahead of me.

Vanessa, thank you for serving as my most dedicated simulator sickness test subject. Know that the discomfort you felt was not passed on to others.

Finally, Courtney, you remind me that the arts are just as important to being human as the sciences, and your love and support encouraged me to see this beast of a project through to the end.

Dedication

To the reader, that you might peruse this manuscript and hopefully learn something new.

Table of Contents

List of Figures

List of Tables

List of Terms and Abbreviations

ABBREVIATION	TERM
VR	Virtual Reality
VRN	Virtual Reality Navigation
DK2	(Oculus Rift) Developer Kit 2
AD	Alzheimer's Disease
HMD	Head-Mounted Display
ESA	Early Stage Alzheimer's
LSA	Late Stage Alzheimer's
SSQ	Simulator Sickness Questionnaire

Chapter I: Background/Introduction

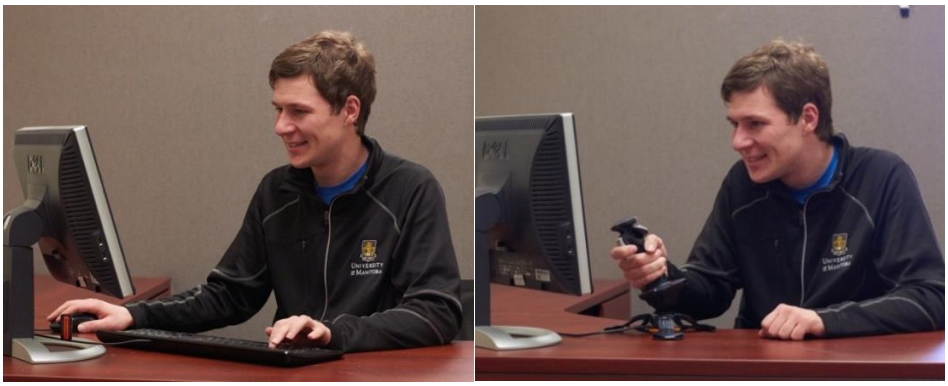
Since the 1990's, when immersive 3D Virtual Reality (VR) first began to emerge as an entertainment medium, researchers have been exploring its potential for use in more serious applications. Many authors speculated about a world where high-quality, computer-generated virtual worlds could be used to study human behaviour. Unfortunately, the technology had not advanced to the point where a fully immersive VR experience was possible, let alone beneficial [1]. The recent 'renaissance' of VR thanks to companies like Oculus, Sony, and HTC/Valve has rekindled interest in this field, and modern technology can provide truly immersive and comfortable experiences, within certain limits. In this chapter, we will discuss some common ways of interacting with VR, and highlight some applications of VR within research and clinical settings. In particular, we are interested in those applications that apply VR to the study of human spatial navigation, and aging. As the motivation for this study was rooted in offering a new paradigm for Alzheimer's disease (AD) neuro-cognitive rehabilitation, we will introduce the basic neuroscientific principles of AD, and discuss how it is currently treated and detected. Although treatment of AD was the motivation of this study, the developed VR platforms developed in this study could be used for neuro-cognitive rehabilitation of any type of dementia or mild cognitive impairment.

1.1 Interacting with Virtual Environments

Nearly all VR systems in the literature use a standard gaming setup of a joystick (or keyboard input) and desktop display [2], but these interaction paradigms have been shown to baffle elderly people and people with cognitive impairment, so a more immersive and intuitive system ought to be realized [3]. The level of immersion of a VR system is a property that describes how fully it engages the user's senses. A VR system's level of immersion is determined by its

interaction paradigm, and by the VR software itself. That being said, high levels of immersion are simply not possible when the interaction paradigm is too decoupled from the system. A quantity that is related to immersion is *presence*. Presence is the user's sensation of being invested in the environment presented by a VR system, and believing it to be real. Higher levels of immersion lead to a greater sensation of presence. In recent years, the level of presence afforded by virtual environments has increased due to improvements in computers and rendering technology, as well as powerful new HMDs and other interface devices. According to Oculus's best practices [4], factors that are important to presence include lighting (especially shadows), texture (the virtual "paint" that gives virtual objects colour) and responsiveness. The gold standards for presence are so-called 'rubber-hand illusions' [5], [6], where a person is physically touched in the real world in a way that corresponds with a visualized touch in the virtual environment.

Garcia-Betances et al. [2] defines 3 different tiers of immersion that VR systems can have: (1) non-immersive; (2) semi-immersive; and (3) fully immersive, which are illustrated in Figure 1. Non-immersive systems and semi-immersive systems are controlled by means of a standard input device such as a keyboard/mouse, or a joystick or gamepad. Most importantly, they are viewed using a conventional desktop display. The chief distinction between non-immersive and semi-immersive systems is that semi-immersive systems may use larger displays and more photo-realistic 3D graphics than non-immersive systems. Fully immersive systems are viewed by means of a Head-Mounted Display (HMD) or another head-tracked system like a Large-Scale Immersive Display as shown in Figure 1E. These fully immersive systems may be controlled by sophisticated input devices such as hand-tracking hardware (e.g. the HTC Vive or PlayStation Move controller), or even full-body motion-capture (e.g. the Xsens MVN [7]). In Figure 1D, a motion controller tracks a user's hand, which provides an immersive experience when paired with an HMD.



A B

C D

E

Figure 1: VR interaction paradigms from least immersive to most immersive.
A) Keyboard/mouse (non-immersive); B) Joystick; C) Joystick/HMD; D) Motion-tracked controller/HMD E) Large-Scale Immersive Display [8] (fully immersive)

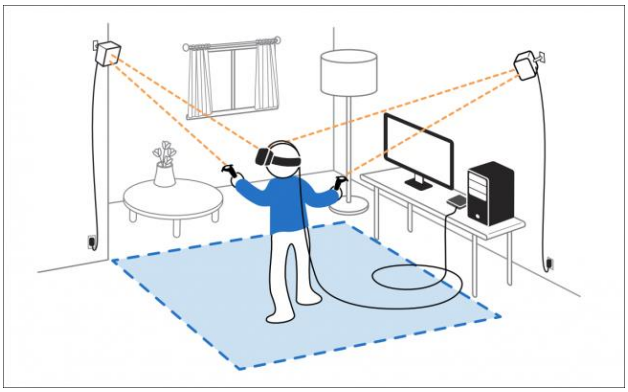
1.1.1 The Oculus Rift DK2

The Oculus Rift DK2 is a leading example of the new generation of low-cost, high performance, consumer-targeted HMDs. This new generation of devices improves upon previous HMDs in two key ways:

1. They use higher-resolution displays (>960x1080 pixels per eye) than HMDs used in the literature (which often feature 800x600 resolution per eye)
2. They use large lenses to expand the field of view and allow a user to focus on a display that is very close to their eyes, providing more optical immersion than previous HMDs [9], [10].

Another important feature of these devices is the way in which they track the user's head. Head tracking is a very important feature of an HMD, because it is what allows the user to 'look around' in VR. The Rift uses an 'Outside-In' position estimation algorithm to track the user (Figure 2A). This means that an external sensor, such as a camera, observes the movements of tracked objects within the tracking volume. This is in contrast to an 'Inside-Out' system, in which the tracked objects compute their own positions using sensors mounted on themselves to observe the outside environment (Figure 2B). Inside-Out systems can be harder to realize and may require extensive calibration to familiarize the tracked objects to the tracking volume, but they have the benefit of being easier to scale up to larger numbers of tracked objects. This is because each object contains the hardware necessary to position itself, rather than requiring the computer to compute positions for each object. Inside-out systems can also provide very accurate positioning information, but often at the expense of requiring the installation of standardized beacons, as shown in Figure 2B.

A



B[11, p. 24]

Figure 2: Two different tracking methods.

A) Oculus's Rift HMD contains LEDs that are tracked by an external camera (Outside-In).

B) An alternative 'Inside-Out' tracking strategy is implemented by the HTC Vive. In this system, sensors are mounted on the HMD and tracked objects (such as controllers) which allow them to calculate their positions within a pre-defined tracking volume. The volume is shown in blue, and is referenced by beacons, which are mounted on the walls.

The 'Outside-In' algorithm used by the Oculus Rift exchanges some scalability for simpler hardware that lends itself better to seated applications, as opposed to so-called 'ambulatory' applications (i.e. applications where the user physically moves about). In the case of the DK2, the HMD has many LEDs mounted on it that blink in different, specific patterns. By recognizing each particular blink pattern and comparing it with the known location of the corresponding LED on the HMD, the HMD's position can be calculated within the tracking frustum of the camera (Figure 2A). These blink patterns were analyzed by Oliver Kreylos as he attempted to develop a Linux-compatible driver for the Oculus Rift [12]. Unfortunately, while this computation can be done in

real-time, it cannot be done quickly enough to keep up with the framerate of the HMD, so intermediate poses are estimated using data from motion sensors called Inertial Measurement Units (IMUs) that are mounted within the HMD.

IMUs combine data from accelerometers (which measure acceleration and can detect the 'down' direction by detecting acceleration due to gravity), gyroscopes (which measure angular velocity), and magnetometers (compasses) to compute a tracked object's rotation in space. While IMUs can do a good job of quickly and accurately computing rotation, they cannot accurately compute position (displacement), since doing so requires double-integration of data from accelerometers. This double-integration results in a quadratic accumulation of displacement error, since linear acceleration measurements are often affected by a small DC offset due to imperfect removal of the gravity vector. Error accumulation of this magnitude quickly becomes unacceptable [13]. This can be corrected by 'fusing' IMU data with data from a drift-free position sensor, such as the tracking camera used by the Rift (Figure 3). Since the measurements from the camera come fairly quickly (at least 10 times per second), the acceleration measurements of the IMU can be integrated to find HMD displacement between camera frames. Since the amount of time between drift corrections will be very small, the amount of drift that could occur is acceptable and not noticeable. This combination of a camera and an IMU means that the Rift's tracking scheme is really a hybrid between the Outside-In and Inside-Out strategies.

An interesting feature of the Rift is that it can also operate in an 'IMU-only' mode, when the tracking camera is not available. This means that tracking can be done in a fully Inside-Out style, but sacrifices positional tracking for the reasons described above.

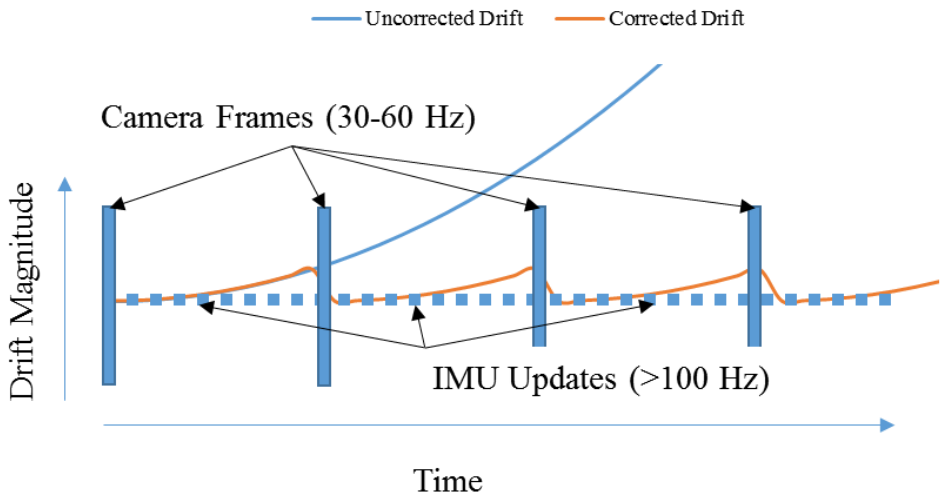


Figure 3: Pose estimation of the Oculus Rift DK2.
The tall ticks represent times when camera frames are processed, and the small, square ticks represent times when IMU samples are processed.

1.1.2 Simulator Sickness

A common side-effect of using any kind of VR system is simulator sickness. It can be described as a sensation of dizziness or upset stomach (similar to motion sickness) that appears during (or following) the use of a VR experience. This phenomenon was widely reported on in studies involving US Airforce cadets training in flight simulators, since simulator-based training is cheaper and therefore preferable to training in real aircraft. An excellent review is provided by Johnson [14]:

The 'sensory disconnect' theory, which is generally accepted as the root cause of simulator sickness, states that a user develops simulator sickness due to a disagreement between the visual system and the vestibular system. When a user views a moving virtual world through an HMD (or other display technology), they are shown an optical flow that is consistent with that virtual motion. However, since the user is actually stationary, the vestibular system reports that no actual motion is occurring. This sensory conflict is thought to trigger feelings of upset stomach and the gag/vomit reflex because these symptoms are consistent with hallucinogenic poisons, and the vomit reflex is

a response of the body attempting to quickly eject such poisons. A second theory, 'Postural Instability,' states that people experience motion sickness (or simulator sickness, as the case may be) in situations, where they are unable to maintain postural stability. These types of situations can occur when environmental motion (or perceived motion, as is the case in VR environments) interferes with the normal motion adjustments used to maintain balance, as is the case when a person spins around in tight circles. This theory is not widely accepted, in part because it predicts higher incidences of simulator sickness in the aging population (who have a greater difficulty keeping balance than the younger population), but data indicate *fewer* incidences of simulator sickness in the aging population [14].

Quantifying simulator sickness can be quite challenging. Currently, the most popular method is to use a questionnaire developed in the early 1990's by Kennedy, Lane, et al. [15]: the Simulator Sickness Questionnaire (SSQ). The SSQ scores a variety of possible symptoms in a participant by having the participant rate the experienced severity of various symptoms on a 4-point Likert scale. Although the SSQ is commonly used throughout the literature [16]–[21], it suffers from a large amount of variance between people, thus very large samples are needed to compare different interventions. Furthermore, simulator sickness symptoms can take several minutes to manifest themselves after a trigger event, and can take many hours to abate, so comparative studies need to take place over several days. Furthermore, some authors use modified SSQs with higher levels of granularity, or that omit certain symptoms [21], [22], further complicating comparisons.

Another means of quantifying simulator sickness is to measure stomach activity. Cevette et al. [21] used electrogastrography (EGG) and heart rate to detect simulator sickness in people using flight simulators. This technique was found to be even more sensitive to motion sensations

than the SSQ; Cevette [21] postulated that EGG might be a useful metric for predicting the onset of simulator sickness based upon physiological measurements, rather than subjective symptoms.

The Air Force studies [14] found the best way to treat simulator sickness was to simply desensitize participants, and let them adapt to the conditions. This was done by scheduling short sessions using the offending simulator to gradually accustom the user to the motion environment. There are also various medications such as dimenhydrinate or scopolamine [23] as well as home remedies such as ginger [24] that can reduce the symptoms. In terms of simulator design, simulator sickness can be mitigated by reducing opportunities for sensory mismatch. This may be done with external stimulation, such as Galvanic Vestibular Stimulation (GVS) [20], or with custom input devices capable of accurately capturing the full range of user motion. Furthermore, best practices for VR experiences suggest design strategies such as reducing latency in game loops (i.e. maintaining a high framerate), ensuring continuous responsiveness to user motion (i.e. never locking the camera view), and avoiding 'shaking, jerking, or bobbing the [perspective]' [25].

1.2 Clinical Applications of VR: State of the Art

The idea of testing for neurological anomalies using VR navigation activities is not new. The Morris Water Task is popular in spatial navigation experiments using rodents; it was first reported on in the 1980s [26]. The Morris Water Task is a navigation experiment in which the subject attempts to find an invisible target in a circular arena. In the original water task, the arena was a pool of water 1.3m in diameter. The target was realized as a single elevated platform upon which the rats could climb to avoid swimming in the somewhat cold water. Critically, the water was made opaque to prevent the rat from seeing the targets; the rats discover the target by swimming around and eventually bumping into it. The rat was able to see outside the arena and use distant objects (such as objects outside the arena in the room) as points of reference, so that

the location of the hidden platform may still be identified when the starting point is changed. Figure 4 illustrates the layout of the Morris Water Task.

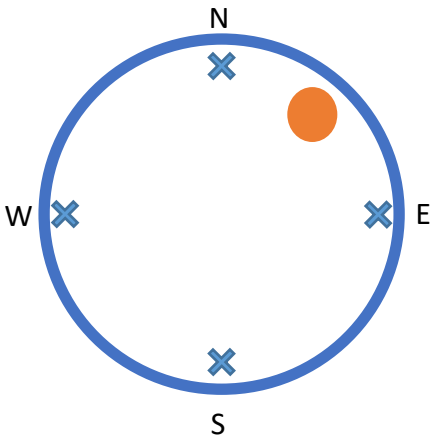


Figure 4: Morris Water Task.
The orange circle represents a target point where a hidden platform may be located, and the blue '**X**'s represent points where the rat may be placed into the arena.

The Morris Water Task has also been administered on humans as a means of classifying spatial navigation skills [27]–[31], and when combined with simple paper-based cognitive evaluations as in [31], VR exercises can be useful predictors of whether a person is developing dementia. Navigation assessments can also take place in more realistic settings, such as a house or a city block. In [22], Zakzanis et al. created a small virtual city called "Sunnybrook City". The input device for the Sunnybrook City VR environment was a standard gamepad (so the interaction paradigm resembled Figure 1C). The authors compared navigation performance between older and younger participants, and also highlighted a pair of participants diagnosed with probable AD. Participants wore an HMD, and were asked to retrace a path through the city. They were scored based upon their ability to quickly and correctly navigate between two points in the virtual city. The results showed that older participants had more difficulty navigating in the city compared to the younger participants . A custom questionnaire was used to assess simulator sickness both inside

the virtual environment, and after leaving the virtual environment. The simulator sickness results were not discussed in great detail, except to note that both younger and older participants experienced similar levels of simulator sickness [22].

In our previous work [3], human participants navigated inside a realistic virtual building using a custom-built wheelchair motion capture device (that work is described in greater detail in Chapter II). This experiment also bears many similarities to the Morris Water Task, in that the participants try to find a location that is not visible to them. The benefit of our highly immersive system is that it appears to greatly reduce instances of simulator sickness, and more importantly removes the bias that may be presented by participants struggling with otherwise non-immersive interaction paradigms.

Another popular application of VR is neuro-cognitive rehabilitation. VR environments can be used to teach patients activities of daily living such as crossing a street [32] or grocery shopping [33]. The Virtual Action-Planning Supermarket (VAPS) is a grocery shopping simulation in which participants find food items on a list in a virtual supermarket and bring their purchases to a virtual attendant at the checkout. The VAPS has been used to evaluate patients with stroke [34] and cognitive impairment [31], and according to clinicaltrials.gov, a randomized control trial was completed in November 2014 to evaluate its rehabilitative potential for stroke patients, although results were not available in the report [35].

Cushman et al. [36] compared navigation in a real-world environment with navigation in an equivalent non-immersive virtual environment (Figure 5). Participants were moved through the environment in an ‘on-rails’ style, and needed to choose whether to turn left, right or continue straight at intersections. The authors found that different participant groups performed similarly whether they were using a real environment or a virtual environment. In fact, relative performance

of each group relative to the others in the real environment was preserved in the virtual environment. In other words, this shows that virtual environments of suitable fidelity can act as stand-ins for real-world environments.

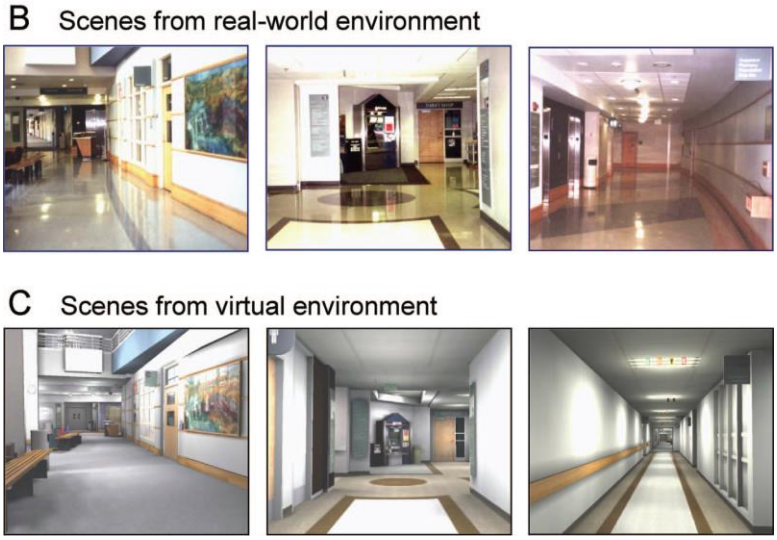


Figure 5: Comparison of real-world hospital and virtual hospital from [36].
This is an example of a non-immersive virtual environment, although it features highly realistic lighting and textures.

1.3 Alzheimer's Disease Diagnosis and Treatment: State of the Art

AD is a devastating condition that affects the mind by destroying the cells that make up the brain. In 1906, Dr. Alois Alzheimer described a case study of a patient suffering from severe dementia symptoms at less than 60 years of age. Dr. Alzheimer performed an autopsy and observed a reduction in neural gray matter (Figure 6), as well as the presence of plaques in the patient's brain. He linked these reductions in gray matter and *senile plaques* with dementia [37]. Another common observable biomarker called a *neurofibrillary tangle* has been identified [38, p. 750]. It is thought that neurofibrillary tangles occur as a result of genetic mutations that cause the proteins that hold microtubules together in neuronal cells to be incorrectly synthesized. These proteins are not able to maintain the microtubules, which then disintegrate. Since microtubules are important

for giving neurons structure, their disintegration causes neurons to fail, and leads to neurodegenerative diseases like AD [39].

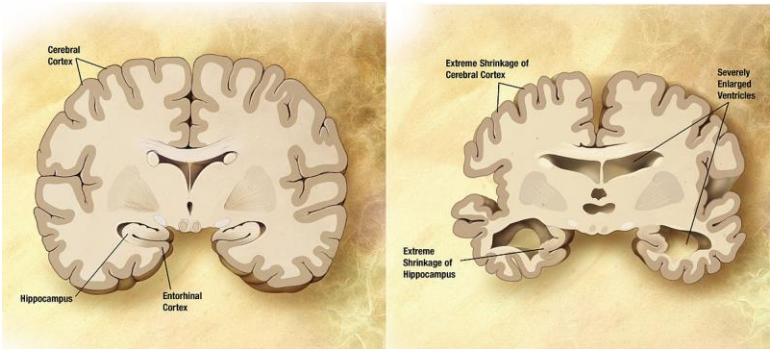


Figure 6: Comparison of a healthy brain and a brain affected by AD [40].
AD results in a loss of neurons, which leads to a reduction in grey matter and enlargement of normal cavities, such as the ventricles.

Typically, AD is treated with drugs such as Donepezil (Aricept) that enhance the firing rate of existing neurons. This can be likened to overclocking a CPU in a computer to improve its performance. These drugs do not replace or repair lost cells, but instead allow a person to make more use of remaining cells. The disease progression is not slowed though, so the drugs are most effective if the disease is diagnosed early [41].

Unfortunately, current early-stage diagnostic techniques are limited to observing and monitoring cognitive decline because senile plaques and neurofibrillary tangles, which are the main physiological symptoms of AD, are difficult to image at the onset stage. Typically, pencil-and-paper cognitive assessments such as the Mini-Mental State Examination (MMSE) [42] and Montreal Cognitive Assessment (MoCA) [43] are used to quantify a person's cognitive state, along with imaging techniques such as MRI to try and detect the tissue loss shown in Figure 6. Recently, radioactive indicators that bind to senile plaques have been discovered that can be imaged *in-vivo* using Positron Emission Tomography (PET) [44]. In-depth observational tools like the Clinical Dementia Rating (CDR) [45], [46] may be used to quantify cognitive state in greater detail.

The CDR consists of a semi-structured interview with specific questions that should be asked of the patient and an 'informant' who frequently interacts with the patient (usually a spouse or child). Since it has a very broad scope, the CDR can take more than an hour to administer. The interview proceedings are recorded on a worksheet that is later scored. In-depth, observational evaluations like the CDR are quite useful at staging dementia, and many studies have found them to be quite reliable. For example, Burke et al. [47] showed 25 video-taped CDR interviews to 5 independent raters (each rater was shown 10 tapes). The reviewers had a high degree of agreement with each other in their assessments of the patients. In other work, Morris et al. [48] trained 82 investigators from across the United States in scoring CDR worksheets, and found a high degree of agreement between the results determined by the investigators and the 'gold standard' set of scores that had been determined beforehand by the study leader.

Two other popular assessments are the MoCA and MMSE. Both the MoCA and MMSE are questionnaire-style assessments that probe language, memory, attention, orientation (i.e. knowledge of current location and current time), and visuo-spatial ability. Both assessments score participants out of a maximum of 30 points, with lower numbers indicating increasingly severe cognitive difficulty, and take less than 20 minutes to administer. Of the two, the MoCA appears to be more sensitive to subtle changes in cognition, in part due to the MMSE's heavier reliance on language ability. Lessig et al. [49] studied test results of 221 Parkinson's Disease patients, and found that the MoCA was more sensitive to early-stage dementia, but the MMSE was better at tracking changes in later-stage dementia.

One major drawback of evaluations such as the MoCA or CDR is that by the time they can detect AD, the person by necessity has already lost some cognition and is expressing dementia symptoms. It would be far better if we could detect AD symptoms that precede noticeable cognitive

impairment, for example using known biomarkers such as plaques and tangles, or a lower-level cognitive function. However, research has shown that in two situations the biomarkers may actually not correlate with impairment, specifically:

1. **High** levels of plaques and tangles sometimes correspond with **low** levels of cognitive impairment.
2. **Low** levels of plaques and tangles sometimes correspond with **high** levels of cognitive impairment.

The first situation is explored by Katzman et al. [50], who hypothesized that these inconsistencies occur when a person has a large number of redundant neural pathways or neurons, otherwise called a 'cognitive reserve'. It is thought that this cognitive reserve can be built up by performing intellectually stimulating tasks, such as staying abreast of current events, frequent socializing, and reading* [51]. The authors observed larger brains by volume and mass in post-mortem analyses of patients with large amounts of AD pathology but no observable metal decline. In related work, Snowdon et al. [52] analyzed a participant from the Nun Study in the United States whose brain did *not* have a large amount of extra tissue, had large amounts of AD pathology, and yet had no observed cognitive decline. The authors speculated that the type and location of the lesions in the case study might have been such that the particular patient was spared significant loss of cognitive functionality.

Katzman et al. [50] also alluded to the second situation (low plaque, high impairment) by referencing a paper illustrating cases of patients who experienced cognitive decline characteristic of AD, but upon autopsy, presented low levels of tangles or plaques. Those cases are reported by a rather old paper by Rothschild [53], which reports on 24 cases of patients with 'Senile

*In order to satisfy the last task, the author shamelessly recommends this very thesis, or any of his other works [62], [82]–[85].

Psychoses.' In that work, there are two cases where a relatively low amount of plaque is accompanied by a high level of intellectual deterioration (although the quantification of plaque levels is limited to a subjective 4-point scale). In addition to those, Rothschild made particular note of a particular patient due to her relatively high level of senile plaques, and relatively low level of intellectual impairment [53, p. 776]. This particular patient had an observed brain weight of 1360g, which is the second heaviest brain in the set. The high level of plaques and low level of impairment would corroborate Katzman's 'cognitive reserve' hypothesis.

Since the presence of plaques and tangles are not necessarily reliable predictors of whether a person will develop AD, one may consider lower-level subconscious cognitive function to detect the very early symptoms of AD or any other cognitive decline. Since the hippocampus is one of the first neural structures to suffer damage during the AD progression, it is widely held that one of the first brain functions to be affected by AD (indeed, in cases of cognitive impairment in general) is spatial navigation [22], [28], [36], [54], [55]. This phenomenon was also observed anecdotally by my advisor, Zahra Moussavi, as she watched and experienced her mother's development of AD. Dr. Moussavi noticed that her mother, who had previously known her entire city like the back of her hand, found herself needing directions to get around. However, as Dr. Moussavi's mother was a highly educated person, she showed no decline in questionnaire-style assessments such as the MoCA. In fact, she did not show any symptoms on these tests until much later on, at which point her memory had already begun to deteriorate rapidly. It is possible that she was able to perform so well on these tests due to an enhanced cognitive reserve, as was observed by Katzman et al.

In other works [56]–[58], we investigate whether we can predict the development of AD despite the presence of this cognitive reserve using spatial navigation assessments. As more data

is collected to continue those studies, a second question has been raised: How can we enhance this cognitive reserve? Billings et al. [59] showed that in lab mice, AD pathology could be decreased in the hippocampus by training in a navigation task. We wondered whether the reduced pathology found in their work could apply to humans as well.

In this thesis, I investigate whether VR navigation tasks can be used to treat AD at two stages in two case studies: a late stage, after severe deterioration has occurred; and at an early stage before there has been any noticeable decline. We anticipate that symptoms at both stages will improve after VR training, but the early-stage participant will have more pronounced results.

1.4 Summary and Thesis Organization

In this chapter, we introduced some of the basics of VR interaction. We showed how VR technology is used in other clinical applications in the literature, in particular spatial navigation and functional rehabilitation, and highlighted research showing that the data collected in a VR environment is comparable to data that would have been collected in a comparable, real-world environment. Finally, we briefly introduced the pathology of AD, and discussed how VRN training may be able to reduce the cognitive effects by aiding in the construction of cognitive reserves.

In Chapter II, our VRN assessment -the VRN Building- is described, along with how it was modified in this work to improve immersion for users. Chapter II also describes experiments that were used to ascertain any benefits of the viewing VRN Building with an HMD, as compared to a conventional display. Chapter III explores the application of VRN environments for treatment of neurodegeneration, and introduces a new environment, the VRN Home. In Chapter IV, the design of the software of the VRN Building and VRN Home is described in detail. Chapter V discusses the results of the experiments conducted in Chapter II and Chapter III. Finally, the work is concluded and briefly summarized in Chapter VI.

Chapter II: VR Navigation for Diagnosis – Methods and Results

Since we were interested in detecting spatial navigation deficits, we needed a way to study human spatial navigation. In earlier work [56], [58], we selected VR as a suitable medium to explore human spatial navigation and designed a VR maze, which took the form of a symmetric, virtual 3-storey building (Figure 7). In this chapter, we describe our VRN assessment: the VRN Building. We go on to describe changes that were made to the VRN Building in this work to make it more immersive, and some of the technical and usability issues that were overcome. Finally, we discuss the effects these changes had on participants.

2.1 The VRN Building Assessment

[60] defines two basic sub-classes of spatial navigation: Allocentric (also called geocentric) and Egocentric spatial navigation. When using allocentric navigation, a person mentally models the world as a static environment with themselves as a moving agent. They model their position as being relative to *objects* in the space. Egocentric navigation is slightly different, in that the person models the objects in the world relative to *themselves*. It is hypothesized that allocentric models are used for coarser navigation, (like the scale at which the Global Positioning System might be used) while egocentric models are preferred for fine-grain navigation, like guiding a car through a narrow street.

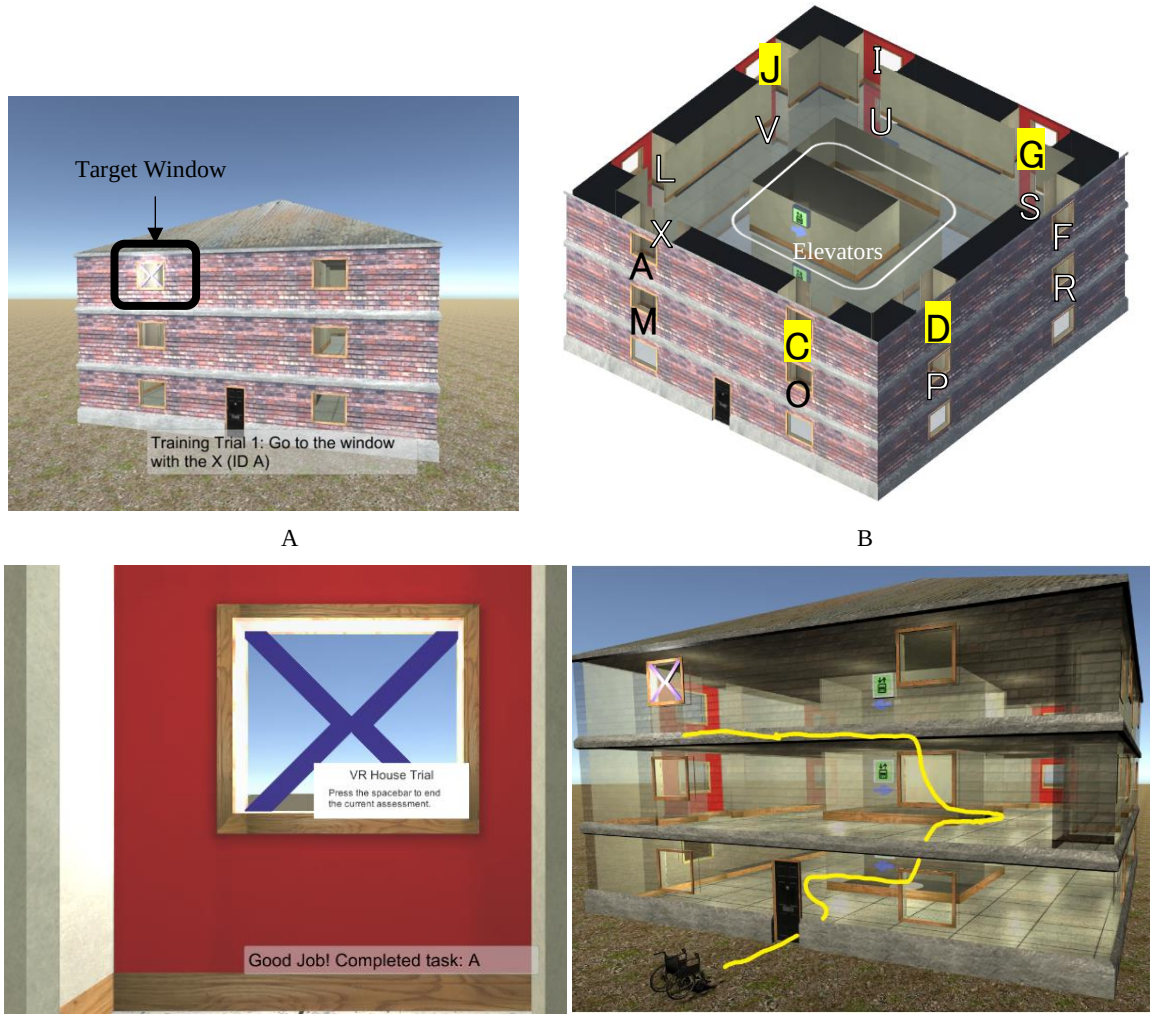


Figure 7: VRN Building navigation exercise from 3 different perspectives.

A) Exterior view. The target window is marked with an '**X**'. B) An orthographic view showing the interior layout, showing the IDs for each window. The third floor windows have IDs A-L, and the second floor windows have IDs M-X. C) Interior view of a target window, the '**X**' reappears when the participant enters the correct room to indicate that the trial is finished. D) The elevators to move between floors force participants to turn around. This perturbs participants' cognitive map.

Since place cells used in encoding spaces are located in the hippocampus, and the hippocampus is one of the first brain structures to be affected by AD, we hypothesized that people developing AD would develop difficulties encoding and navigating in spaces that require allocentric navigation strategies, and that these difficulties would develop early in the disease's progression. The VRN Building (also referred to in our previous work as the VRN House)

assessment is used to study human navigation in a landmark-less, symmetric 3-story building in a series of 8 trials [56]. At the beginning of each trial, the participant is shown an external view of the building, where a randomly selected window is marked with an **X** (See Figure 7A). The participant is instructed to enter the building and find the target window from the inside. When the participant enters the building, the **X** on the target window is made invisible to prevent participants from re-discovering the target window if they have forgotten its location or become disoriented. When the participant enters the correct room, the **X** reappears, the participant is congratulated, and the trial ends. The layout of the building intentionally forces participants to make turns to get to higher floors; this perturbs the participant's cognitive map and makes the assessment more challenging, especially for people with navigation impairments. As can be seen in Figure 7C and D, the participant needs to turn around 180 degrees to reach the third floor from the second floor, which adds an extra layer of complexity. Figure 7D illustrates an example of a path to get to a room on the third floor. We ensure that the participant recognized the target window by asking them to verbally identify the location of the '**X**' (e.g. back wall, third floor, left side), before they enter the building. To get accustomed to navigating in the virtual environment, participants have two un-scored training trials.

The building has a limited number of landmarks, specifically the elevators (or flight of stairs, in an older version). By building a mental model of the building, the participant can find their way to the target window. There are a total of 16 possible target windows: 8 on each floor and 2 on each wall (refer to Figure 7B). In each assessment, we perform 8 trials, each with a different, pseudo-randomly selected target window. This process is pseudo-random because each target window is selected such that over the course of 8 trials, the participant will be assigned two windows from each wall and one on each floor. Specifically, on the first trial, we randomly choose

between window M and window O; on the second trial we randomly choose between window D and window F; etc.

We can assess a participant's spatial navigation ability by characterizing the navigation errors they make (i.e. navigating to an incorrect window) and computing an error score for each trial. In the VRN Building, we look at three types of navigation errors:

1. **Wall errors**: selected window is on an incorrect wall (e.g. participant incorrectly chooses the North wall instead of the West wall). This type of error was observed more frequently than the others [58], and we believe that it indicates that the participant is disoriented by the floor transitions.
2. **Floor errors**: selected window is on an incorrect floor (e.g. third floor instead of second floor).
3. **Left/right errors**: selected window is on opposite side of the correct target (e.g. if the target window is Window A in Figure 7B, Window F would count as a left/right error because it is on the **right** half of its wall, while Window A is on the **left** half of *its* wall).

Any navigation error the participant makes will be some combination of these 3 types. The overall error score for the trajectory is computed using the classification scheme shown in Table 1 [58]. The error score for a particular trial is the sum of the scores for all the navigation errors for that trial; higher scores indicate increasing difficulty in spatial navigation. Each error type is weighted based upon the data that was analyzed in [58]. The best possible error score is 0 points (correct window on the first attempt) and the maximum possible error score for any one trial is capped at 10 points. A score of 10 points is assigned if the participant becomes 'Totally Lost,' which is defined as having 3 or more wall errors (Table 1). Finally, our scoring algorithm only counts Left/Right errors if the visited wall and floor are both correct.

Table 1: Navigation error types in the VRN Building.

ERROR TYPE	POINT PENALTY
Wall Error	4 points (if target is on 2nd floor) 3 points (if target is on 3rd floor)
Left/Right Error	2 points
Floor Error	1 point
Totally Lost (3 Wall Errors, or participant gives up)	10 points

As an example, consider the hypothetical case illustrated in Figure 8. The participant is supposed to go to Window A (refer to Figure 7B), but they become confused and go to Window R, then Window O, a finally Window A (the correct window). In this case, the participant would receive a total of 5 points (out of a maximum of 10).

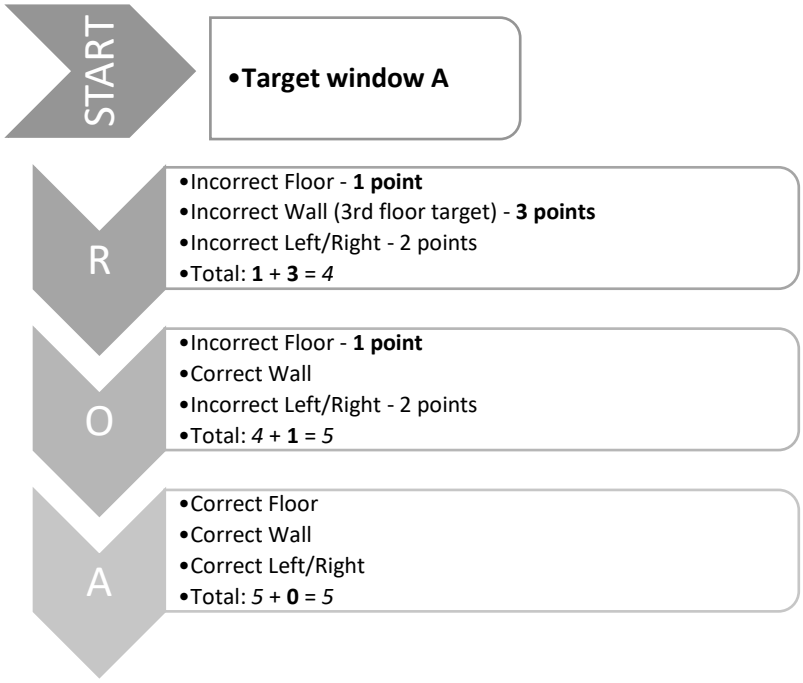
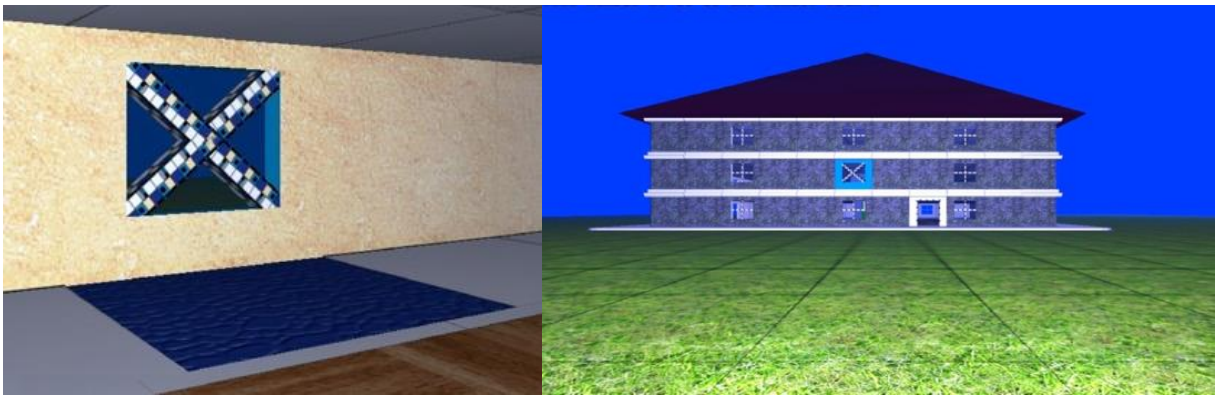


Figure 8: Scoring a set of window visits in the VRN Building.

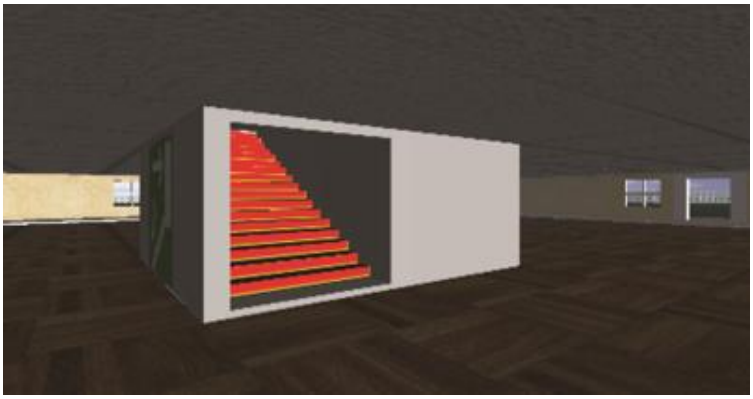
2.2 VRN Building 1.0

The first incarnation of the VRN Building was created in 2011 [56]. In that non-immersive implementation, participants navigated in the environment using a joystick and viewed it with a standard computer monitor (i.e. a laptop or desktop display). Version 1.0 of the VRN Building

was written entirely using Visual C++ for Windows, using only OpenGL calls and Nvidia PhysX. This implementation marked the genesis of an exploratory study aimed at characterizing the deterioration of spatial navigation in the aging population, and attempting to determine if it was possible to detect neurodegeneration using VRN exercises. Some screenshots of this early version of the VRN Building are illustrated in Figure 9.



A B



C

Figure 9: Screenshots of the first-generation VRN Building (Version 1.0).
A) Interior, showing a target window. B) exterior. C) Stairs leading from first floor to second floor.

As more people participated in the study, it became apparent that the system was causing simulator sickness, possibly due to the visual-vestibular mismatch discussed in Section 1.1.2. Approximately 30% of participants complained of simulator sickness effects when using the joystick; thus, a solution needed to be found to reduce the sensory mismatch. This problem was resolved by designing a custom input device that allowed the participant to move about naturally

in the real world [3]. This input device was implemented in the form of a specially instrumented wheelchair that captured a participant's real-world motion and translated it to the virtual world. We called this device the VRN Chair. The VRN Chair is shown in Figure 10.

Figure 10: VRN Chair.
The participant pushes the VRN Chair, which captures position and rotation data. This data is used to move the participant's virtual avatar in the VR environment.

In [3], the authors attempted to quantify the level of improvement provided by the VRN Chair in a study investigating healthy young people, but found no statistically significant difference in simulator sickness incidences when using a joystick input device as compared to the VRN Chair. It is possible that this particular sample (which was comprised exclusively of male, engineering graduate students) would have had more experience with video games and VR than average, and would have been desensitized to the effects of simulator sickness. The authors did, however, comment on a reduction in simulator sickness in participants of the longitudinal study. When using the joystick and a stationary system, approximately 30% of participants complained of simulator sickness symptoms, but out of the more than 370 participants that had used the VRN

Chair as of the publication date of [3], only approximately 2% complained of any simulator sickness symptoms.

2.3 VRN Building 2.0

At the start of my involvement with the VRN Building project, it had already gone through many modifications to improve its performance, and to ensure consistent experiences for participants. While some experiments had been conducted using the VRN Building with an Oculus Rift DK1 HMD [61], it was found that this particular HMD was not suitable for the longitudinal study due to 3 key performance limitations:

1. **Low Resolution**: The Rift DK1 suffered from a low resolution of 1280x800 pixels, which meant that lines between pixels were very clearly visible. This phenomenon is known as the "screen-door effect" and can be distracting to users. To ensure comfortable, immersive VR, a higher resolution is needed.
2. **Aliasing**: The first-generation VRN Building was scratch-built, and lacked graphical enhancements such as Anti-Aliasing (AA). This meant that some textures and geometry would appear to flicker when the participant would move about in the VR environment. This behaviour can be tolerated by participants on a standard desktop display or television, but when using an HMD, it becomes uncomfortable and can ruin presence.
3. **Head-bob**: Participants navigated in the first-generation VRN Building by walking and pushing the VRN Chair in front of themselves (as shown in Figure 10). As the participant walked, their head would sway from side to side relative to the wheelchair along with the natural sway of their gait. Since the Oculus Rift DK1 was incapable of tracking this head-swaying motion, participants would experience simulator sickness because they could feel their head swaying, but received no matching visual feedback.

The first two issues, being technical in nature, can be resolved by careful selection of hardware and a game engine. However, the third issue is a usability issue and must be solved by making changes to the way the VRN Chair is used.

2.3.1 Choice of Game Engine

During the summer of 2014, we obtained a Rift DK2 HMD, the successor to the DK1. The DK2 was the highest-quality commercially available consumer-ready HMD at the time, which is the primary reason we selected it. Compared to the DK1, the DK2 offered higher resolution, faster framerate and lower 'motion to photon' latency (i.e. the time between a user moving their head and the replication of that movement in the VR environment) as compared with the DK1. These advancements made the DK2 suitable for the longitudinal study by resolving Issue 1, but the VRN Building program would need heavy modification to make it compatible with this new HMD. Since the DK2 was still pre-release hardware, Oculus would change the driver software quite frequently. This meant that a great deal of valuable time would be required to even keep up with the API changes required for compatibility with the ever-improving Oculus Rift runtime. Since this would not be manageable for a small team such as ours, we elected to port the VRN Building to a commercial game engine to allow faster and more effective iteration. The onus of keeping up with API changes would fall to the company maintaining the game engine, which would be better equipped to do this. The selected commercial game engine would need to meet the following requirements:

- **Support the Oculus Rift platform** – this was a fairly obvious requirement, but beyond merely supporting the Rift at its current point, the selected game engine needed strong support going forward. If we had needed to implement many changes with each Oculus software update, we would not have gained any advantage over continuing to use the

custom engine upon which the original VRN Building was based. Therefore, either the company building the engine, or Oculus themselves had to have a strong commitment to support Oculus's HMDs into the future.

- **Ease of Use** – Since we intended to pursue other virtual reality applications, we preferred to choose a well-supported platform that could easily be learned by other students who would work with the software in the future. This went beyond support for HMDs, the engine features had to be well-documented and preferably have a strong community of developers. One problem with our existing application was that it had been handcrafted in the lab, so there would be no support available if its original creator left. Choosing a well-supported commercial engine would allow future students to be able to learn and effectively work with the tools on their own.
- **Broad Cross-Platform Support** – While the best HMD choice in summer of 2014 was the Oculus Rift, it would not do to have to re-program the application every time a platform shift was necessary. These types of platforms could include other PC operating systems (such as Mac OS X or Linux) or embedded consumer platforms (such as Sony PlayStation, Microsoft Xbox, or mobile solutions such as Galaxy Gear VR). Choosing a game engine with support for one or more of these alternative platforms would simplify future potential migrations.
- **Rich Graphical support** – In order to improve the participant’s sense of presence, certain graphical effects were necessary. Features like Anisotropic Filtering, Anti-Aliasing, and advanced lighting effects improve the 'realism' of a virtual environment. While we would have had to manually implement these effects ourselves with our original engine, many commercial game engines provide highly optimized and well-

tested support for them, and a programmer often simply needs to enable them. This easily addresses Issue 2.

Two commercial game engines were found that supported our requirements: Unity 4.5, and Unreal Engine 4 (UE4). Some of the basic features of these engines are enumerated in Table 2. Both platforms were evaluated and trialed by means of building simple projects and working through sample exercises over the course of approximately 2 months (one month for each engine). While both engines are similar, they have different strengths. The Unity engine is optimized to run on a wide spectrum of platforms, as can be seen in Table 2. Specifically, it has much stronger mobile support than UE4. Since Unity runs inside the .NET managed environment, it enjoys more robust compatibility and error reporting, although at the cost of more overhead, which translates to a slight reduction in performance as compared to native applications that can be developed with UE4. In many respects, these trade-offs are similar to the Java platform, which runs in a virtual machine.

On the other hand, UE4-based games run as natively compiled C++. This results in reduced overhead as compared to Unity-based games, but it also means that compiling the application is more complicated. When building a UE4 game, the entire engine must be compiled, which often takes a substantial amount of time. Furthermore, this process was found to be extremely sensitive to build settings, which can cause difficult-to-diagnose errors. These inconveniences that are inherent to UE4 bring the benefit of superior performance; when backed by powerful graphics hardware, UE4 is capable of producing incredibly rich visuals with high-quality textures, bumpmaps, shaders, and rich lighting effects. At the time this project began, UE4 also supported the Oculus hardware without need for additional plugins.

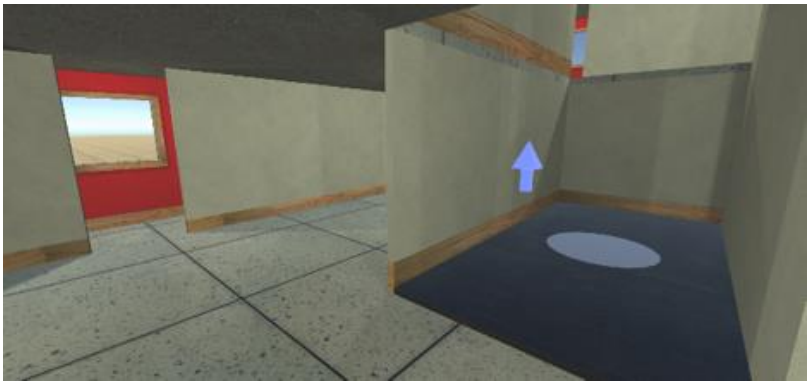
Table 2: Summary of game engines.

Two commercial game engines were considered for use in the second-generation VRN Building: Unity and Unreal.

	UNITY	UNREAL
LANGUAGE	C# (.NET/Mono runtime)	C++ (native runtime)
PLATFORM	Android, Apple TV, BlackBerry 10, iOS, Linux, Nintendo 3DS line, OS X, PlayStation 3, PlayStation 4, PlayStation Vita, Wii, Wii U, Web (Unity Web Player) Windows Phone 8, Windows (7, 8.x, 10, RT), Xbox 360, Xbox One	Android, iOS, Linux, OS X, PlayStation 4, Windows (7, 8.x, 10), Web (Through HTML 5), Xbox One
COST	Free for individuals, paid one-time subscription for Universities and other entities with revenues greater than $100 000	Free for use, 5% of revenues above $3000 per quarter.
SOURCE CODE	Closed source (open-source available at additional cost)	Open source
PARENT COMPANY	Unity Technologies	Epic Games
OCULUS RIFT SUPPORT	Oculus-supplied plugin (as of Unity 5.x, Unity supplies 1st-party integration)	Epic Games 1st-party integration

The Unity engine was ultimately selected mainly due to its superior ease of use. Even though UE4 is capable of producing superior visuals as compared to Unity 4.5, the added complexity of producing content was not worthwhile in our case. Additionally, Unity's broader and more robust cross-platform support was an important consideration for our lab. Figure 11 illustrates the current iteration of the VRN Building, running in the latest version of Unity (Unity 5.1.3) instead of the old, custom game engine.

A B



C

Figure 11: Second-generation VRN Building, implemented with Unity 3D.
A) External view of building. B) VRN Chair, with Oculus Rift DK2 Head-Mounted Display (HMD) C) Interior, showing elevators. Note the shadows and advanced lighting effects that were not available in the old game engine.

2.3.2 Usability and Interaction Design Decisions

We resolved the head-bob issue (Issue 3) by seating participants in the VRN Chair, rather than allowing them to push it around the test area. This was done to minimize head motion (specifically *translation*) relative to the VRN Chair. Even though the Rift is able to track the position of a participant's head in space, the algorithm relies upon the assumption that the tracking camera is stationary relative to the earth [12]. We found that the VRN Chair's motion would interfere with this position-tracking algorithm. Since the pose-estimation algorithm assumes the camera is stationary, the system will interpret the participant turning their head to the left (Figure

12B) exactly the same as the case when the participant looks straight ahead and rotates the wheelchair around them to the right (Figure 12A). This appears as a shuddering, jerky motion because the camera cannot update the pose estimation as quickly as the display (which is 75 Hz). Recall that the motion in between camera frames is estimated by on-board IMU sensors in the HMD. When the tracking camera is removed, the algorithm falls back to an IMU-only tracking scheme, which can only track rotation, but does not suffer from this aliasing problem. This is acceptable for our application since head translation is minimized by the participant's seated posture. Any rotational drift that occurs accumulates quite slowly, and can be manually zeroed by the investigator operating the experiment.

Additionally, we learned that it was important to calibrate the HMD to each participant's unique Interpupillary Distance (IPD). This calibration is recommended by Oculus as a means to reduce user discomfort, as it allows the HMD to display a stereo image more compatible with the participant's unique anatomy [4]. Some participants are very sensitive to this parameter, as we observed in other work [62].

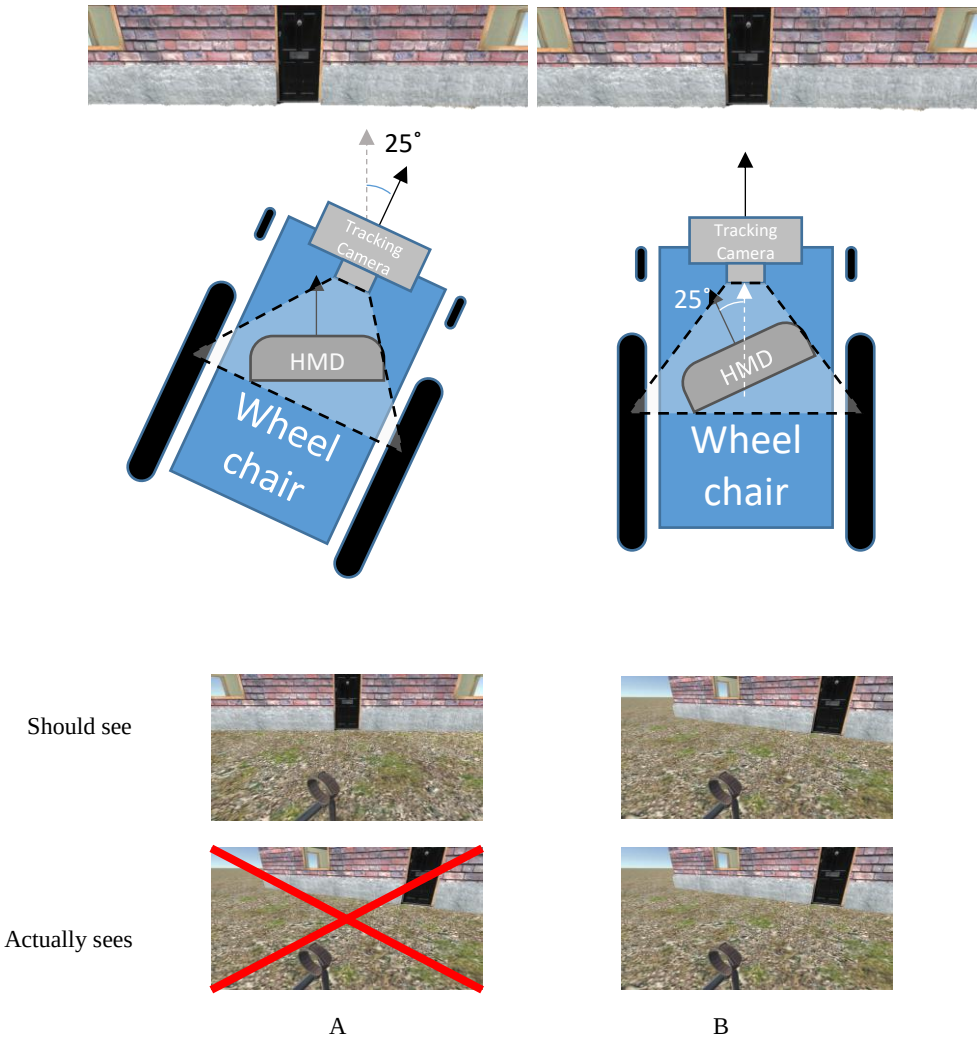
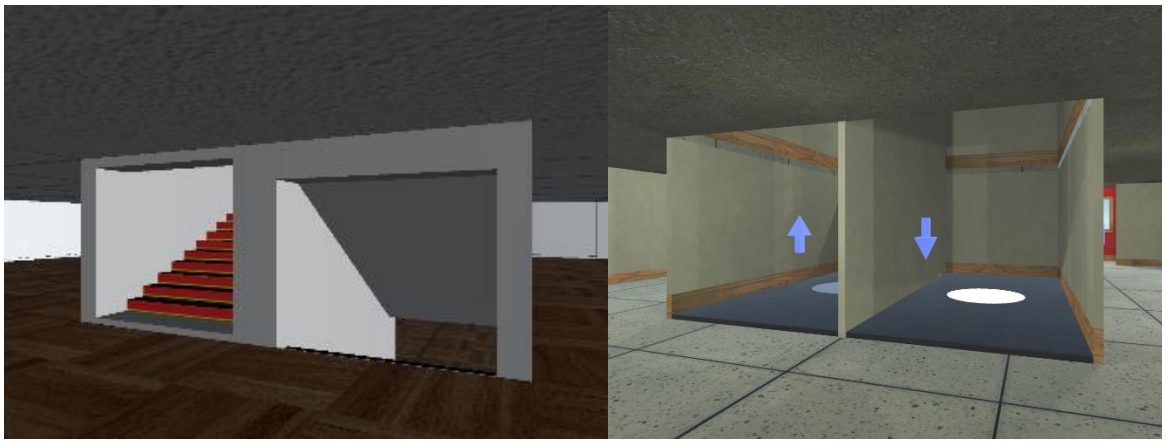


Figure 12: Limitations of using the Oculus tracking camera for localized positioning.
Since Oculus's pose estimation algorithm assumes that the webcam is stationary, it will interpret both the A and B cases in the same way for a particular camera sample. This means that rotating the chair while the participant just looks straight ahead will appear to the participant as though they are turning their head to the left, instead of looking straight ahead.

We also found that the VRN Chair itself was susceptible to rotational drifting, since the wheelchair wheels could sometimes slip along the ground. This was corrected by mounting an IMU to the chair's frame. While IMUs are susceptible to drift of their own, the drift they experience is very gradual over time, and is not noticeable over the course of a 3-5-minute trial like the VRN Building trials. Furthermore, since IMUs report real-world angle measurements that can be directly applied in the game world, they do not need sensitivity calibrations. We used a

BNO-055 chip from Bosch Sensortec, which is capable of providing rotational measurements at a rate of ~140 samples/second [63]. This is sufficient for our application, since the wheelchair's rotation needs to be calculated at a rate close to the DK2s framerate, which is 75 frames/second. Furthermore, the BNO055 performs sensor fusion computations on-chip, making it an appealing choice, since most other IMU chips do not.

Some important changes were made to the geometry of the virtual building; in particular, we reduced the size of the building and elected to use elevators in place of stairs to move between floors. We performed some simple usability tests with healthy young participants, and found that the ramps used in the original building were far too steep to be comfortable in VR; people reported feeling that the wheelchair would be at risk of plummeting down the virtual ramp. The new elevator paradigm removes this issue by presenting a method of moving between floors that more closely matches the user's physical experience, and removes the illusion of danger caused by the steep virtual ramp. Figure 13 shows a side-by-side comparison of the stairs in the first-generation VRN Building and the elevators in the second-generation VRN Building. In order to make the elevators as intuitive as possible, the new elevators feature a white floor-button and an indicator arrow instead of push-buttons (since our equipment lacks hand-tracking, wall-mounted push-buttons would have been complicated to interact with). When the participant moves onto the button, the elevator will begin moving in the direction indicated by the arrow. In this way, participants can freely navigate between all 3 floors, just as with stairs. Understanding that moving the participant in the virtual world without physically moving them in the real world could cause simulator sickness, we had to choose the elevator speed carefully. We chose the elevator speed to maximize participant comfort after iterating through several speeds with a few willing participants.



A B

Figure 13: Comparison of first and second-generation VRN Building interior
A) stairs in first-generation VRN Building B) elevators in second-generation VRN Building, from the same point of view on the second floor.

We also adjusted the dimensions of the virtual building. The first-generation building was nearly 30 virtual metres wide. To operate in a physically smaller tracking volume without colliding with real-world walls, the tracking sensitivity of the VRN Chair was set to be approximately 5:1 (1 real-world metre of movement = 5 virtual metres). We initially created this environment in Unity with those same scale and sensitivity settings, but participants felt as if they were moving far too quickly and experienced great discomfort when wearing the HMD. To address this issue, we scaled the building's footprint down by reducing the building dimensions without changing the relative positions of the windows and overall layout. The building was resized to fit an 18m x 18m footprint, which corresponded to the large room where we performed most of our tests. This meant that the VRN Chair's tracking sensitivity could be set to 1:1, which was far more natural and comfortable for participants.

Keeping in mind that the VRN Building might be used in an even smaller real-world environment, we knew that in some cases it might still be necessary to adjust the participant's movement speed with a sensitivity factor, as was done in the first-generation building. We found

that in large virtual environments with few local reference points (such as the VRN Building), participants handle a sensitivity ratio of 2:1 well (i.e. 1 metre of real-world motion corresponds to 2 metres of virtual motion). This balances small real-world room sizes against giving participants simulator sickness.

Another solution we use to work in size-constrained real-world environments is "decoupled" mode. When decoupled mode is enabled, the experimenter can reposition the participant in the real world without affecting the virtual avatar's position or rotation by disabling participant interaction. This allows large virtual environments to be navigated in smaller real world environments. When the participant approaches a wall or barrier in the real world, we can freeze their position and rotation in the virtual world and re-position them in the real world so as to avoid collisions, and maximize the distance we anticipate they will attempt to go. This decoupling technique is illustrated in Figure 14. To help prevent simulator sickness, decoupled mode allows limited interaction in two forms:

1. **Allow head tracking**: the virtual environment continues to respond to the participant's head motion, so they can continue to look around in the virtual world.
2. **Real-world video feed**: a live view of the real world from a webcam affixed to the front of the HMD is laid over the displayed virtual environment. This helps prevent motion sickness by providing an optical flow consistent with the rotations the participant may be experiencing (especially while the chair is rotated), while not fully disconnecting them from the virtual environment. This way, the participant will not forget where they were in the virtual environment after decoupled mode is disabled again.

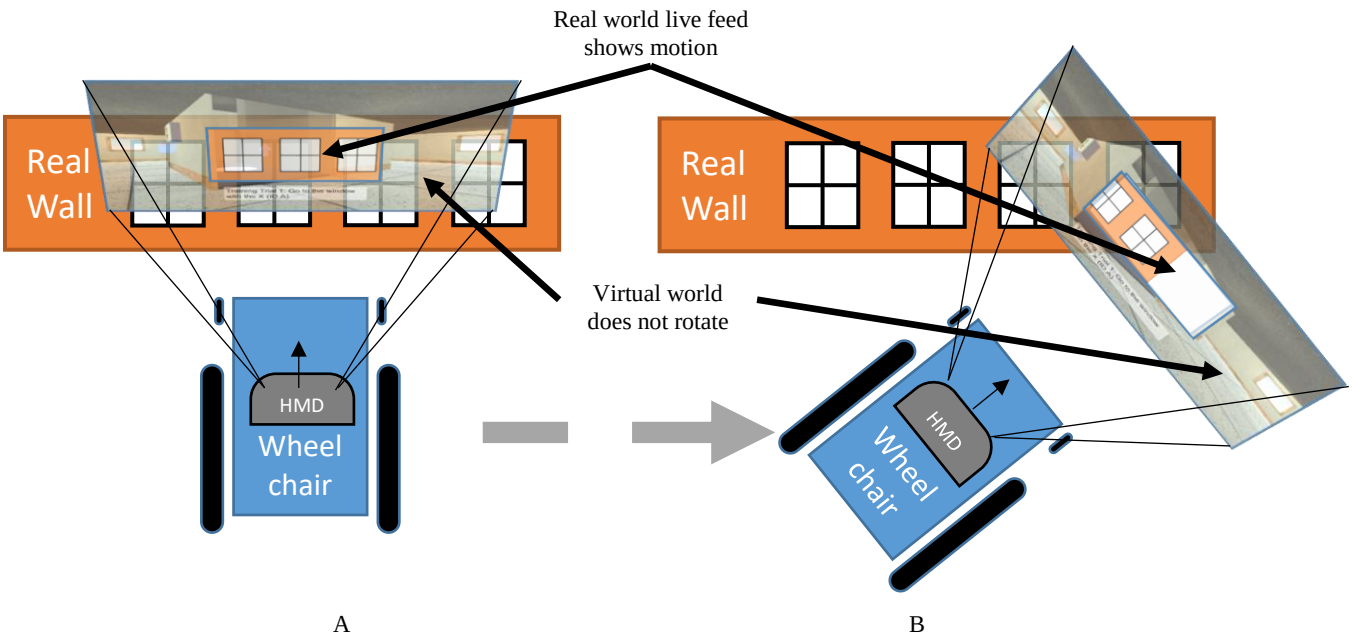


Figure 14: Operation of decoupled mode.

A) A possible situation where a participant has approached a wall in the real world, but still has a great deal of virtual space that they wish to traverse. B) Once the system is locked in decoupled mode, we can rotate the wheelchair to leave lots of room to move around. The participant can still look around, as illustrated by the superimposed point of view. Note that while the participant's body (represented by the wheelchair) is rotated in the real world, their virtual body does not rotate in the virtual world.

The original design for decoupled mode as implemented in version 1.0 of the VRN Building merely blanked the screen and ignored participant input during repositioning. We experimented with this method, but found it to be highly disorienting, as participants would need to spend some time recollecting their bearings once they resumed the VR experience. Maintaining interactivity during the decoupling operation as shown in Figure 14 made it easier for participants to resume navigation.

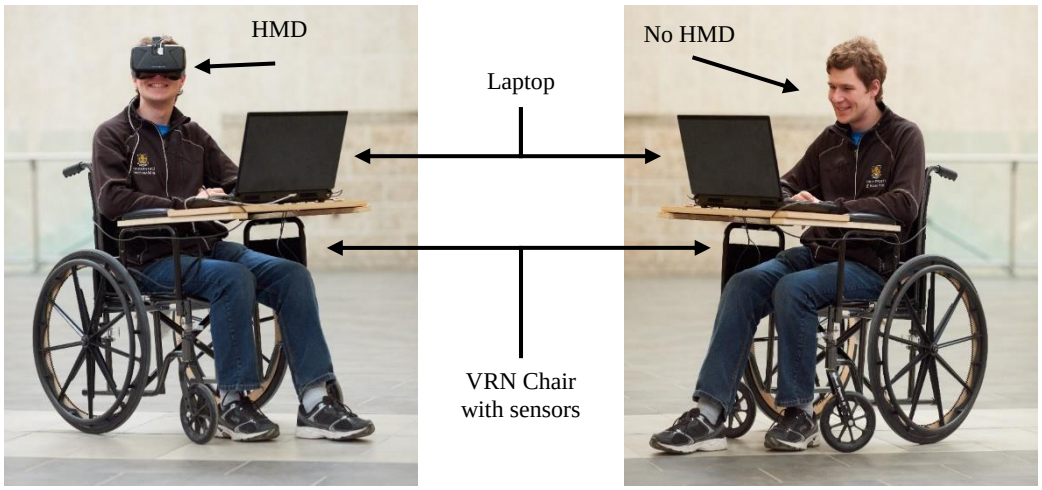
2.3.3 Testing VRN Building 2.0

The changes we made to the VRN Building by adding HMD support are not trivial, therefore we sought to evaluate whether or not the changes brought any benefits to our test participants. We investigated performance changes between the semi-immersive and fully immersive systems in two ways. The first was a study we conducted in Spring 2015 with 18 healthy

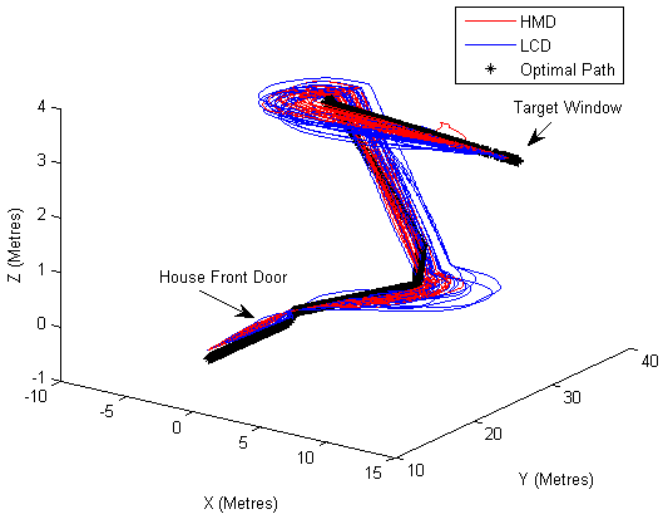
young participants [62]: we compared people's performance in the VRN Building environment using an HMD with their performance in the same environment without an HMD (i.e. viewing the environment with a conventional display, in this case the laptop's LCD). In this experiment, we were not interested in analyzing the spatial orientation or memory of participants, so we took measures to ensure that people would correctly remember how to get to the target window. We used the same target location for each of the four trials (Window P in Figure 7), and also gave participants one training trial to familiarize themselves with the system. Additionally, we gave participants verbal prompts to help them find the target window. We found that when wearing the HMD, participants navigated more accurately (i.e. the more closely tracked an optimal path to the target window) but more slowly, since there was no significant difference in their navigation time. The experimental conditions and results are illustrated in Figure 15, and a comparison of trajectories under the two different conditions is summarized in Table 3.

Table 3: Traversal distances and times in [62]

	HMD	LCD	SIGNIFICANCE
DISTANCE (METRES)	56.0±3.8	61.3±5.5	$p < 0.01$
TIME (SECONDS)	61.8±14.3	59.0±18.0	$p > 0.01$



A



B

Figure 15: Immersive VRN Building vs. Semi-immersive VRN Building.

A) Comparison conditions (HMD vs. laptop LCD). B) Recorded trajectories, all leading to the same target window. Participants navigated more precisely when using the HMD (red paths) than when using the conventional display (blue paths).

In that experiment, we found that while wearing the HMD, participant's paths were significantly shorter than while viewing the conventional display. However, navigation times were not significantly different. This suggested that increased immersion in a virtual environment results in an improvement in pathfinding, and more careful navigation. We found that less than 1/3 of participants experienced very minor simulator sickness, despite the vestibular stimulation offered by the VRN Chair. Based on feedback from participants, we concluded that this was likely due to

the ramp used to navigate between floors (See Figure 16). This resulted in us converting the ramps to elevators as discussed in Section 2.3.2.

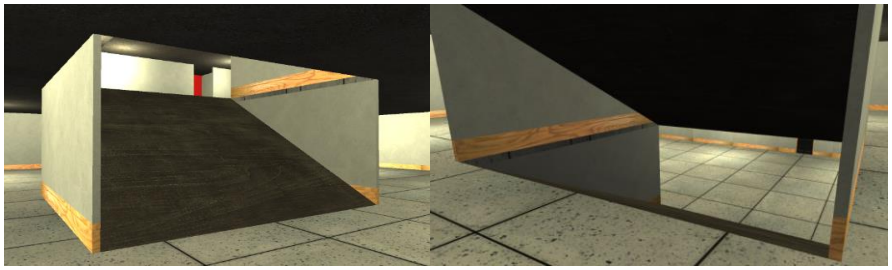


Figure 16: The ramp in the VRN Building (From [62]).

The ramp formed a 45-degree angle with the floor: this caused discomfort for some participants.

2.3.4 The Effects of HMD on VRN Building Assessment Performance

The second approach to assess the HMD as a display device was to see if normal assessment scores in the VRN Building were affected. During the course of the longitudinal study, we observed that 90% of returning participants show improved (i.e. reduced) error scores in subsequent assessments, so we could not simply compare matched trials for the same participant (i.e. first trial in first-generation VRN Building and 6 months later a second trial in the second-generation VRN Building). Therefore, we elected to compare people whose first VRN Building assessment used an HMD (the second-generation building) with people whose first VRN Building assessment did not use an HMD (first-generation VRN Building). By comparing naïve participants in this way, we avoided learning effects and prior experience bias (i.e. familiarity with the interface device after having used it previously).

Since we began using the HMD variant of the VRN Building, we have had several new participants whose first assessment involved the HMD. We selected 13 participants from this group that had completed an acceptable number of trials (≥ 4) and had a healthy MoCA score ($>$ 23), and compared their error scores with the first-time scores of 211 age- and MoCA-matched participants whose first experience had been in the first-generation VRN Building. More data on these participants is shown in Table 4. We found that naïve participants using the HMD had

significantly lower error scores than naïve participants using the conventional display. This suggests that the VRN Building assessment is easier to perform when viewed with an immersive HMD than when viewed with a conventional display. This may be because participants are not accustomed to mentally projecting themselves into the virtual world presented on the 2D computer screen, which increases the participant's cognitive load and results in overall lower navigation performance. This extra processing step is not present when wearing the HMD, thanks to its high level of immersion. This implies that the HMD-based assessment is more accurate for assessing spatial navigation than the conventional display, because participants can simply focus on navigating without struggling with projecting themselves into the virtual world shown on the conventional display.

Table 4: Aggregated data about participants used to investigate VRN Building performance.
The first column represents participants using the first-generation VRN Building assessment with a conventional display and the second column represents participants using the second-generation VRN Building assessment with an HMD.

	NAÏVE: FIRST-GENERATION BUILDING	NAÏVE: SECOND-GENERATION BUILDING
COUNT	211	12
AGE	60.65 ± 12.323 years	59.25 ± 18.709 years
MOCA (/30)	27.82 ± 1.697	27.75 ± 1.545
NAÏVE ERROR SCORE (/80)	23.77 ± 17.52	8 ± 8.676

This theory is supported by work conducted by Ruddle et al. [64] in 2009, that showed physically moving about in a virtual space is important to successfully complete navigation tasks. The authors [64] compared 3 interaction paradigms (and therefore levels of immersion) in a semi-complex navigation task:

1. **HMD + Walking:** users were able to naturally move about in the virtual environment, as they do with the VRN Chair in our work.

2. **HMD + Rotation Only:** users wore an HMD and were able to rotate in space, but were otherwise stationary. They could move forward/backward using a simple controller input device.
3. **Visual Only:** users used a conventional input strategy using a keyboard and mouse.

They found that users in the HMD + Walking condition were far more accurate than users under any of the other conditions.

2.4 Summary

This chapter discussed the operation and design of the first-generation VRN Building, and modifications that were made to make it compatible with a cutting-edge, consumer-targeted HMD: the Oculus Rift DK2. These changes included:

- Choosing a commercial game engine to improve graphical fidelity
- Selecting an appropriate participant posture to reduce simulator sickness
- Adding an IMU to the VRN Chair to reduce drift
- Re-sizing VRN Building geometry to make it more comfortable to navigate in

Finally, we investigated the effects these changes had on participants. We found that participants navigate more precisely in a VR environment when using the HMD than when using a conventional display, and that our assessment participants find the VRN Building assessment easier when using the HMD than when using the conventional display. We believe that this may be because some users have difficulty mentally projecting themselves into the virtual world shown on a conventional, 2D display, and that using an HMD makes this mental projection unnecessary. Physically moving about in the virtual space removes potential experience biases, and allows people to navigate more accurately.

Chapter III: VRN Navigation for Treatment – Methods and Results

In this chapter, we discuss how VRN tools can be used to treat AD and illustrate in two case studies: a participant with Early Stage AD (ESA) and a participant with Late-Stage AD (LSA). This preliminary investigation saw the development of the first version of a VRN rehabilitation program, and investigated what sort of metrics should be measured in a larger trial. In each case study, the participant was treated with a sequence of VRN exercises, with periodic evaluations to see if there were any observable cognitive improvements.

The VRN Building assessment has some limitations in terms of target demographic; in particular, participants past a certain degree of impairment are unable to even understand what is expected of them, let alone find the correct windows. With this in mind, the VRN Home treatment was designed as a substitute to the VRN Building for our LSA participant. Since the VRN Home treatment would be too easy for our ESA participant, we elected to treat him with a variant of the VRN Building assessment. Both participants interacted with their respective VR environments using the VRN Chair, as described in Chapter I, and had their overall cognitive impairment assessed by means of the MoCA. The treatment protocol for both participants was 3 training sessions (45 min/session) per week for 8 consecutive weeks. However, the ESA participant concluded the study early, after 7 weeks due to personal plans unrelated to the study.

3.1 ESA Participant: Method

Our first participant[*] was a retired, 74-year-old male, who lived with his wife. He had received a Master's degree in Social work, and had worked in that field throughout his career. In the time before our study, he was physically and socially active in his life, and capable of living independently. He was diagnosed with Mild Cognitive Impairment (MCI) with probable

[*] The results of the ESA case study were published in the Journal of Experimental Neuroscience [85].

development of AD. For this reason, we classified him as our ESA participant. He reported symptoms including short-term memory loss and an increased difficulty remembering directions while driving. Furthermore, he had scored 24 on the MoCA v7.1. In the time leading up to our treatment regimen, he was only comfortable driving his vehicle in familiar areas. He had a family history of AD on his father's side.

We recruited the participant from our ongoing VRN Building assessment study[56], which he had previously volunteered for twice within the previous 2 years. His first error score in 2013 was 66%, and his MoCA score was 28/30. Two years later, during his second assessment, his error score increased to 72% and his MoCA score dropped to 24/30. These data, collected 6 weeks prior to beginning treatment, serve as our baseline. It is important to note that we consider an initial error score of more than 50% in the VRN Building assessment combined with an increase in error score to be a warning sign of AD (normally, people's error scores *decrease* in subsequent assessments). Since the participant's MoCA score also decreased, we referred him to a neuropsychiatrist who upon further assessment diagnosed the participant with MCI, with probable development of AD. We have consistently found that the VRN Building assessment is very difficult for AD patients; in our experience, all participants diagnosed with AD fail to find any of the targets in the VRN Building. Since the ESA participant was still at a very early stage of the disease progression and was struggling with the VRN Building assessment, we decided to use it as a treatment for this work.

Since the VRN Building assessment was not originally structured to be used as a treatment method, we developed a treatment procedure that was a modification of the assessment procedure. The ESA participant's treatment was to take place over a period of 7 weeks, with three 45-minute treatments each week (Monday, Wednesday and Thursday). The treatment period was split into 2

phases: *Supported Training* and *Independent Training*. These phases differed simply in that the Supported Training phase was restricted in certain ways, while the Independent Training phase was repeated applications of the standard VRN Building assessment. The Supported Training phase was restricted in the following ways:

1. We restricted the targets to only those on the second floor to avoid the second rotational perturbation associated with going to the third floor.

2. Each window was visited twice each trial. The first time, we gave the participant hints and guidance during his navigation; the second time he was not assisted unless the examiner believed he had clearly become lost. As we progressed through the Supported Training phase, we observed that the treatments were no longer sufficiently challenging for the participant, so we decided to swap the ordering, so that the first window would be un-assisted and the second one would be assisted (if needed). During the third week, once the participant was more comfortable with the second floor windows, we began including target windows on the third floor.

The treatment advanced to the Independent Training phase when the participant demonstrated a high degree of mastery of the trials in the Supported Training phase. During the Independent Training phase, we performed the standard VRN Building assessment (i.e. pseudo-randomly selected windows on both the second and third floors with no repetitions) as many times as we could within our 45-minute session. This worked out to approximately 12-15 targets. Once the participant was able to find 8 targets *consecutively* with no errors, we decided to limit the training to 8 target windows to reduce the participant's time commitment (The regular VRN Building assessment uses 8 target windows). In order to track performance across the Independent Training phase and the Supported Training phase, we compare the number of errors, rather than

the error score, since the error score may not be applicable to data collected during the Supported Training phase. A benefit of the two-phase treatment system is that withholding the third-floor windows allows us to control for practice effects, and see if training solely on the second-floor allowed the participant to navigate more accurately on the third floor.

In addition to tracking navigation errors, we tracked the participant's progress using the MoCA. We performed 3 MoCA assessments during the course of treatment, and 2 follow-ups: once before starting, then during the program at the 4-week mark and at the end of the seventh week. We performed follow-up evaluations 5 weeks and 28 weeks after the end of the program. We also asked the ESA participant's wife to keep a log of his daily activities and driving performance.

In summary, we used four performance metrics to track spatial navigation and overall cognition in this work:

1. We tracked the ESA participant's navigation errors in the VRN Building over the duration of the treatment, and looked for changes in the overall number of errors, as well as particular types of errors (wall errors, floor errors, and left/right errors).
2. We tracked his ability to navigate to third floor windows before treatment, after training only on the second floor, and during two follow-up sessions.
3. We scored his overall cognition using the MoCA at various points during the treatment, and during two follow-up sessions.
4. The ESA participant's wife kept a journal commenting on the participant's navigation while driving and on his "real-world" cognitive health at home.

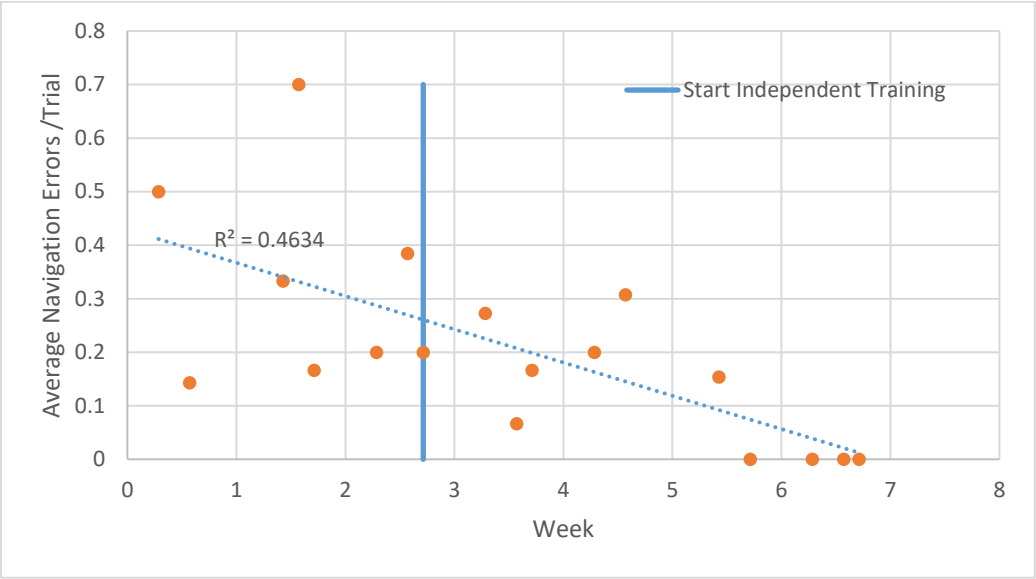
3.2 ESA Participant: Results

The ESA participant showed a substantial improvement in navigating in the VRN Building assessment. This can be clearly observed by comparing the baseline assessment (where the participant could not reach more than 1 target window without error) with the end of the training program, when he could reliably locate 8 randomly selected windows with no errors. Figure 17 shows the steady improvement of the participant's spatial navigation by plotting the number of navigation errors for each session. Since each session had a different number of trials, in Figure 17 the number of errors is plotted as the number of navigation errors divided by the total number of trials for that session. Figure 17A shows a plot of the total navigation errors, and Figure 17B specifically shows wall-type navigation errors.

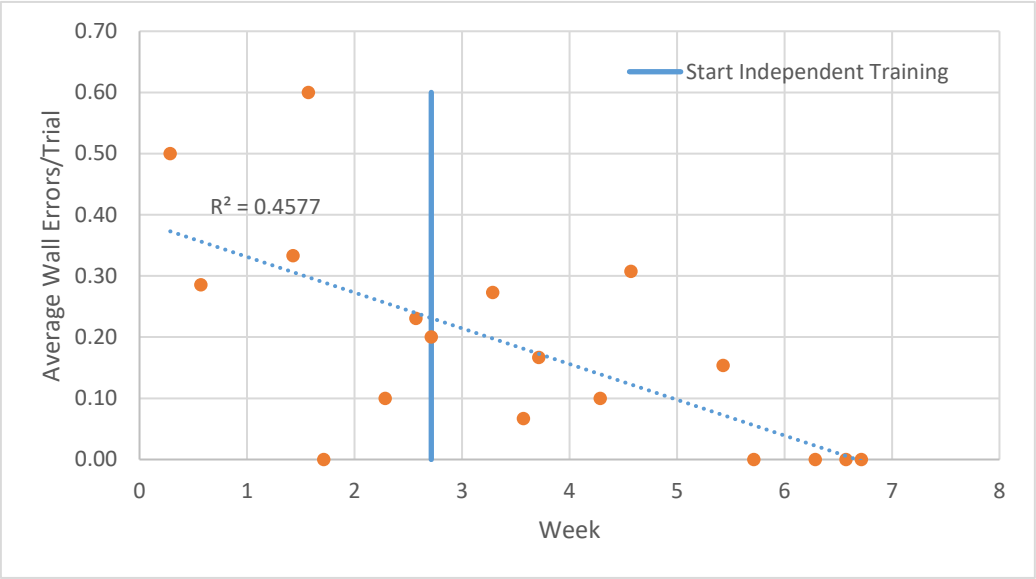
Error! Reference source not found. shows the participant's navigation errors at certain milestones, when navigating to 4 specific windows on the third floor of the VRN Building. These window IDs correspond to the window IDs shown in Figure 7B. After training on the second floor, the participant was able to correctly locate third floor windows that he had not navigated to since the baseline assessment, and a follow-up assessment showed that this effect persisted 5 weeks after ending the training sessions. However, this improvement appeared to have worn off 28 weeks after training.

Table 5: ESA participant's VRN Building navigation errors for 4 of the third floor windows.
Each column contains the number of navigation errors the participant made the first time he navigated to a window following a particular milestone. The 'Pre-Training' column contains the 4 third floor windows that the participant navigated to during the baseline assessment. We ensured that during the follow-up trials the same windows were visited as during the baseline assessment.

Window ID (Figure 7B)	Pre-Training (6 weeks prior to treatment)	After Second Floor Training (Middle of Week 3)	Independent Training (Final Day)	6 Weeks After Training	28 Weeks After Training
C	3	0	0	0	0
D	0	0	0	0	4
G	3	0	0	0	3
J	1	2	0	0	0



A: Total navigation errors



B: Incorrect wall-type navigation errors

Figure 17: The participant's navigation errors over time.

The number of errors is divided by the number of windows visits (as opposed to trials) during each session, because as the treatment progressed, the participant visited more windows within the 1-hour time slot. Note a ceiling effect towards the end of training, where the participant was able to find all target windows with no errors.

The ESA participant's MoCA scores remained relatively consistent during the treatment. This progression is illustrated in Table 6. The MoCA defines scores of 25 or less to be indicative

of cognitive impairment, and since the participant's scores were at or below the borderline of 25, he was classified as having MCI. Since the participant was still at a relatively high level of cognitive ability, most assessments were unable to capture any changes in him due to the ceiling effect. In fact, Table 6 shows that the only area where the participant had difficulty on the MoCA test was the 'Delayed Recall' (or 'Memory') section. We did not expect to see much improvement in that area because the focus of our training was mainly in spatial navigation. Therefore, in order to determine whether the treatment resulted in any real-life improvements, we asked the participant's wife to keep a log of her observations. The participant's wife noted improvements in his daily living functions, in particular in his orientation skills while driving, as well as his mood. Furthermore, we subjectively noted that during the earliest treatment session, the participant would make self-deprecating comments concerning his mental state, (e.g. "my dopey brain") but as treatment progressed and his scores improved, these comments ceased.

Table 6: ESA participant's MoCA scores.

The 'memory' related sub-question of the MoCA has been broken out to illustrate the ESA participant's ability to recall words over a period of time. Note that only the 'No Cue' column contributes to the overall MoCA score.

Week	MoCA Variant	Overall MoCA Score	Visuospatial/ Executive (/5)	Delayed Recall (/5)			All Other (/20)
				No Cue	Category Cue	Multiple Choice Cue	
Baseline (6 weeks prior to start)	7.1 Original	24	5	0	2	3	19
Week 4	7.1 Original	25	4	0	2	2	19
Week 7	7.1 Original	26	5	2	2	1	19
5-week follow up	7.2 Variant	25	5	0	3	2	20
28-week follow up	7.1 Original	23	5	1	1	3	17

3.3 ESA Participant: Discussion

The VRN Building treatment has shown some benefits for a person at an early stage of AD, but the benefits of such a treatment protocol cannot be generalized until they are confirmed in a number of other individuals at the early stages of the AD. The main benefit of the presented VRN system here as compared with other designs is that it addresses some limitations inherent to so-called 'ambulatory' VR systems.

The overall results of this case study suggest that people at early stages of AD can learn to navigate paths in a suitably immersive VR system, and the learned paths may translate to overall real-world spatial navigation skill, as indicated by the participant's wife. Furthermore, the results illustrated in **Error! Reference source not found.** suggest a training period as short as 4 weeks might be enough to achieve significant improvement, and Figure 17B shows that the ESA participant's wall-type errors decreased after the conclusion of the Supported Training phase. The participant was able to translate the orientation skills (i.e. correctly identify the different walls) he learned on the second floor to the third floor, which he had not navigated on since the baseline assessment nearly 2 months previously. That being said, 4 weeks may be overly aggressive, since it actually took 6 weeks until the participant could complete the assessment with 0 errors.

Although **Error! Reference source not found.** shows that the participant's spatial navigation performance had not deteriorated even 6 weeks after training, we observed decline in his spatial navigation performance at 28 weeks. It is worth noting that while the participant was completely lost when trying to find Window D, he navigated to the correct location when trying to find Window G, but on the wrong floor. He became lost while navigating from the second floor to the third floor. Performance could possibly be maintained by having periodic 'booster' sessions to maintain and reinforce any cognitive reserve improvements. Participants could be challenged

even further by navigating from one window to another, rather than always starting at the same starting point outside the building.

During the treatment, the participant's MoCA increased by 1-2 points; however we caution against considering this to be a significant improvement because it is normal for repeated evaluations to vary by small amounts [65], [66]. We suspect that this improvement could also be explained by learning effects, since the 3 consecutive MoCA tests in Table 6 used the same MoCA variant with the same memory words. In the first follow-up MoCA test, we used a different variant with different memory words and noted that the participant's score remained similar. Table 6 also shows that most of the points that the participant lost were in the 'Delayed Recall' section, which tests memory. We reiterate that this was not the focus of our training program; our focus was on spatial cognition, but the MoCA questions that assessed visuospatial reasoning showed a ceiling effect for the participant.

Due to a lack of other widely-accepted objective assessments that would not suffer from ceiling effects when measuring spatial cognition at the time of this study, we opted to document the various subjective benefits that our treatment may have caused. During the course of his treatment, the participant's wife kept a journal of his behavior at home, and noted that he appeared to be happier and more confident in his day-to-day activities, particularly in driving and remembering directions; and was no longer asking his wife for direction after beginning his treatments. His mood was also significantly improved to the extent that he adopted a healthier life style, e.g. reducing alcohol intake and practicing brain exercises [67]. We acknowledge that these lifestyle changes may also have contributed to his improvements, although the brain exercises do not target spatial navigation.

Since using an HMD with our VRN system removes the typical bias exhibited by elderly people inexperienced with using computer systems, it allows us to study spatial navigation more effectively than other VR designs. Our wheelchair-paradigm allows people to physically move about; that substantially reduces simulator sickness [3], especially in comparison to stationary VR systems [22], [31], [68], [69]. Indeed, the vestibular stimulation that our system provides is a key differentiator amongst other work, and this additional stimulation may contribute to path integration and environment encoding at a low level [70].

We have observed quantitative and qualitative benefits in this case study, which encourage further investigations with larger samples (in the literature, groups of 10 [71], and even 66 [72] or more participants are used). Future work should address learning effects by assessing spatial navigation in an additional navigation exercise, similar to what was done with the LSA participant (see 3.4.2 below). We explored the use of a virtual Morris Water Maze environment, but this was not ready in time to get a baseline score. Alternatively, a different navigation environment could be used for training, such as a virtual marketplace, and then the participant's navigation could be scored using the VRN Building.

3.4 LSA Participant: Method

The LSA participant was a 61-year-old male with advanced AD. At the time of our experiments, he was living at home with the assistance of his wife and another caregiver. He was unable to participate and understand the VRN Building assessment. He was using Aricept (donepezil) and Ebixa to manage his AD.

Since the LSA participant's cognitive ability had deteriorated to the point where he was unable to perform the standard VRN Building assessment, we needed to devise a different

treatment regime for him than for the ESA participant. This treatment regime consisted of three components:

1. Navigation training and assessment in a VRN environment modeled after a typical North American home (VRN Home)
2. A custom real-world navigation assessment
3. A VR driving simulation

As with the ESA participant, the LSA participant was treated with three 45 minute sessions per week for 8 weeks. We assessed the LSA participant's MoCA score once every 4 weeks, and performed the custom real-world navigation assessment on the first day of every week.

3.4.1 VRN Home Treatment

The VRN Home treatment takes place in an environment that is modeled after a typical North American bungalow house, complete with furniture and interior lighting. Some screenshots of the environment are shown in Figure 18. During each treatment session, there were two phases:

1. **Training Phase**: the participant was guided through the house by a virtual dog to two different rooms. This was repeated 3 times (increased to 4) before advancing to the assessment phase.
2. **Assessment Phase**: the participant was asked to reproduce the dog-guided path from memory.

This is a marked difference from the VRN Building treatment, where the ESA participant needed to figure out the route to the destination on their own. We used the virtual dog guide to compensate for the LSA participant's reduced comprehension; we noted that he could understand instructions like "follow the dog and learn the path" much better than complicated instructions such as "turn right, go to the bathroom, turn around to go to the living room and then go to the

backyard". In cases where the participant would become lost, the VRN Home was designed to display a series of arrows onto the ground to indicate the correct path to the next target room (Figure 18C). The time intervals when this 'hint' path is shown were logged by the software for later analysis.

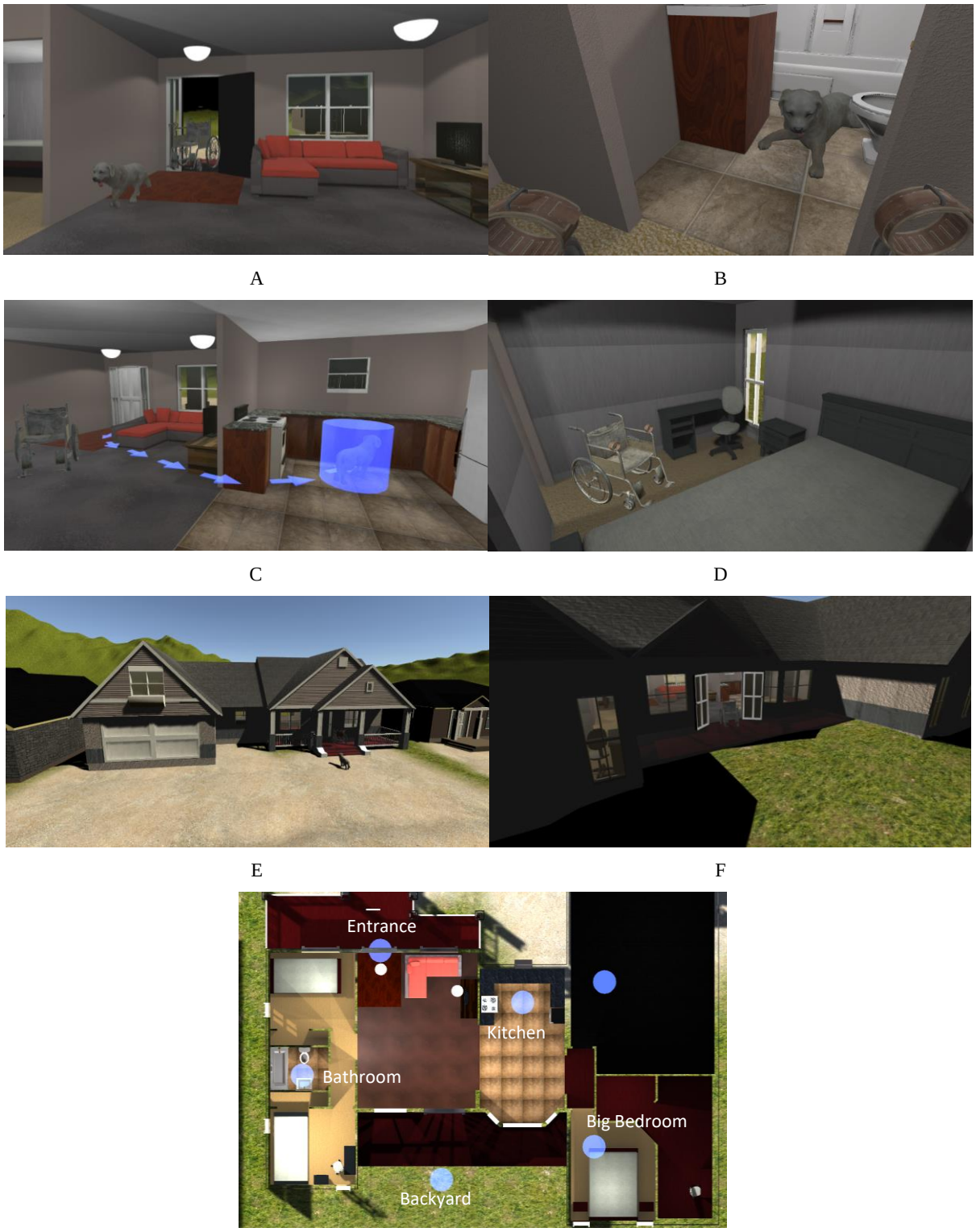


Figure 18: Various screenshots of the VRN Home environment illustrating different rooms.

The dog guide is shown in A) the living room and B) in the bathroom. C) The participant's avatar is shown in the kitchen, as well as the 'hint' arrows that prompt the participant. D) The participant's avatar is shown in a bedroom. The VRN Home is shown externally from E) the front and F) the back. G) shows a top-down view of the VRN Home with the important rooms labeled.

As the treatment progressed, we noted that the participant was not improving during the assessment tasks; he would become confused and search the area immediately around him for the dog, expecting to see it as he had in the training trials. Therefore, we reframed the task by verbally prompting him to look in each of the rooms he had been shown (e.g. “Look in the bathroom,” or “look in the backyard”). We did not give him any directional clues to find the room (e.g. “Turn left,” or “go straight”). During the training phase, the examiner would also verbally tell the participant which room he was currently navigating to (e.g. “Now we are going into the kitchen.”).

During each session, the LSA participant would navigate along one path pseudo-randomly chosen from those shown in Figure 19. These paths were created so that the participant would be exposed to different paths between waypoints in the VRN Home, as opposed to simply memorizing a single path, and allowed us to analyze how well he encoded the house. Since there were a large number of possible navigation segments, we evaluated how well the participant reproduced the path during the assessment phases on the three most frequently used path segments (e.g. kitchen-bathroom). We analyzed these path segments based on the following 3 parameters:

1. **Distance**: the distance covered in navigating between the start and end point of a particular path segment. This captures wandering behavior. We can compare this to the optimal distance, as shown to the participant by the dog.
2. **Navigation Time**: the time taken to navigate a particular path segment. This can also be compared to the optimal time, as shown by the dog.
3. **Hint Time**: the time spent viewing hint arrows. This is an indicator of disorientation, since the participant indicated that they required help.

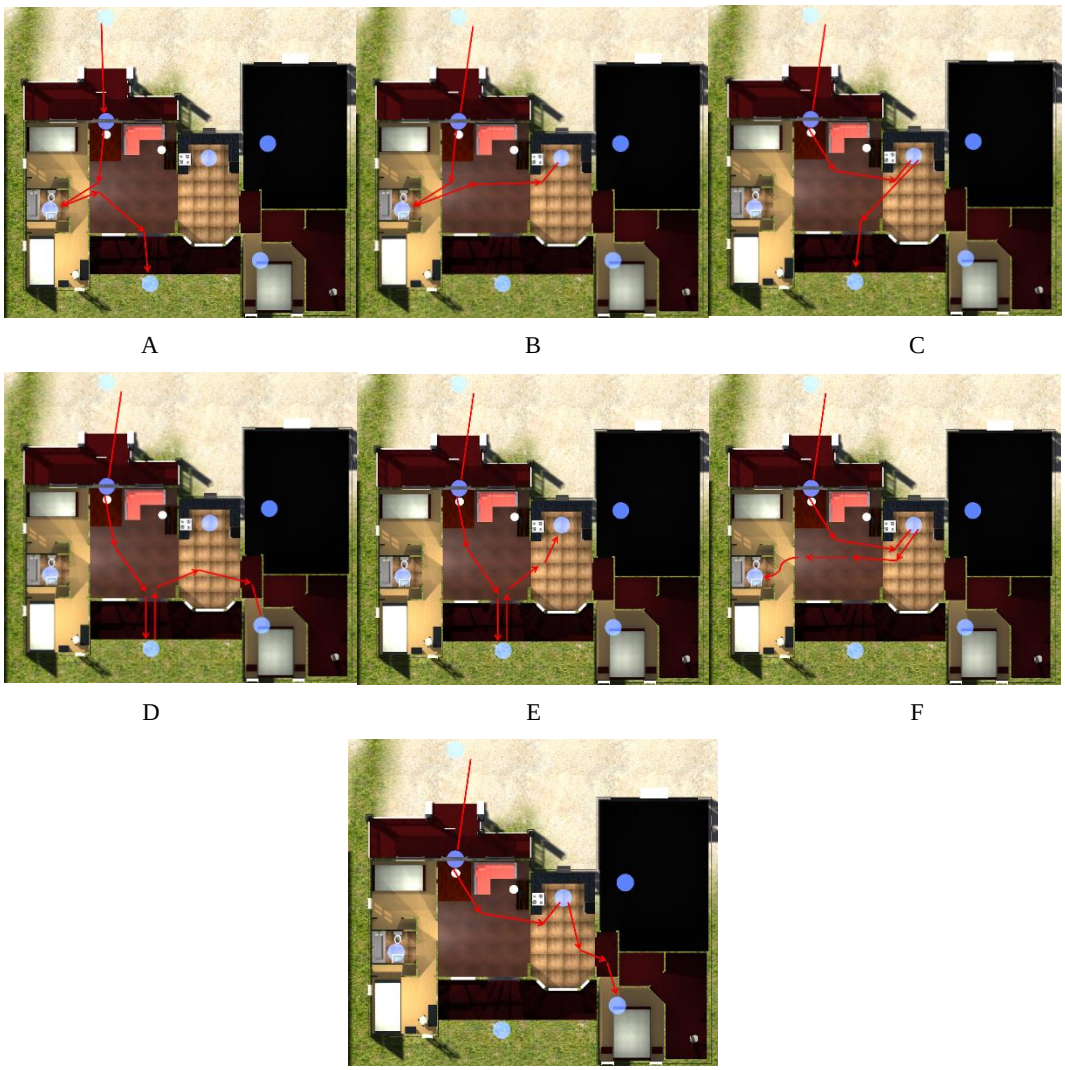


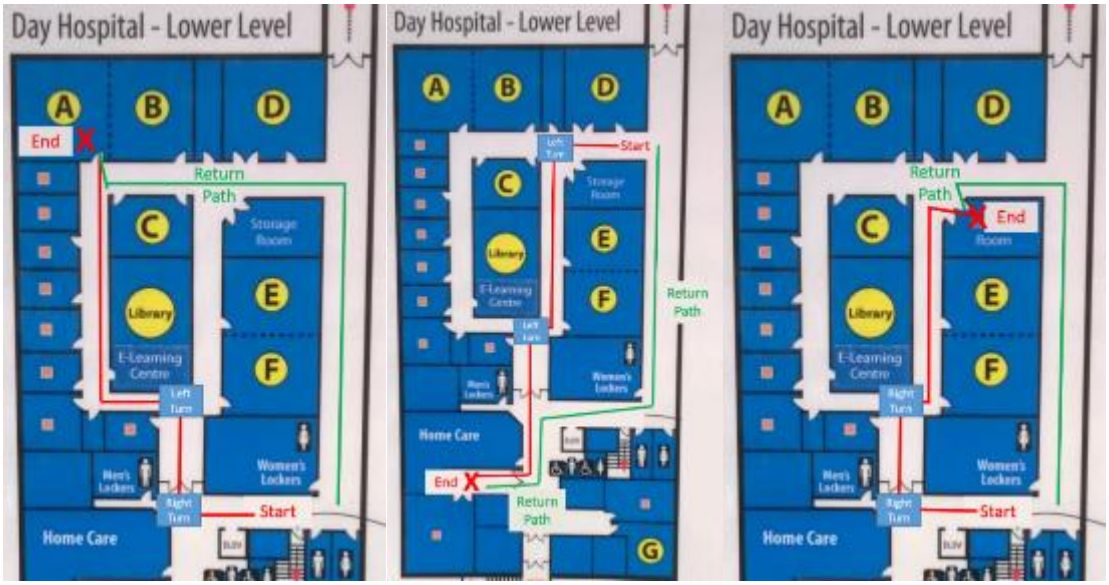
Figure 19: Path segments used to train and evaluate the LSA participant.

We used these 7 paths containing 10 different path segments to prevent the participant from navigating along the same path each session, and just learning that one path.

3.4.2 Real-world Navigation Evaluation

The real-world navigation evaluation was a custom-designed evaluation protocol intended to balance the VR evaluation described above. This performance evaluation was designed to detect translation of the LSA participant's VRN training into real-world navigation ability. The participant was led through a real-world environment (that was previously unknown to him) by an examiner and a familiar person (his wife) along a pre-defined path through the education area of the Riverview Health Centre Day Hospital (See Figure 20). At the end of the path, the familiar

leader (his wife) was left behind, and the participant was returned to the starting point by the examiner through a different path. The participant was then asked to walk through and reproduce the first path and find the leader (his wife). During each session, the above assessment was carried out 3 times with a randomly-selected path from Figure 20*. Each of these paths has only 2 turns; thus, they are comparable in terms of difficulty. The number of incorrect turns was recorded, and each trajectory was manually sketched on a map by the examiner in order to capture any major deviations.



A B C

Figure 20: Real-world navigation tasks for LSA participant.

This task was designed to evaluate whether the VR training had translated to the real world. Each path only has two turns, so they are comparable in terms of difficulty.

3.4.3 Driving

When we began working with the LSA participant, we learned that he had been an automotive enthusiast before his dementia. In an effort to incentivize him to come to the experiments, we developed a simple, immersive driving simulator by modifying the Unity Car

* We originally tried having him navigate all 3 paths in a randomized order, but found that he became confused and could not remember any of the paths

Tutorial [73] to be compatible with the Oculus Rift DK2. This simulation and equipment setup is illustrated in Figure 21. We tracked the participant's performance in the driving simulator by counting the number of checkpoints that he drove through in 7 minutes, and tracked the number of times he crashed the car and needed help to get it back on the road. During the course of this particular experiment, technical limitations of the Unity 4-based simulation forced us to migrate to a more stable platform (based upon the then newly-released Unity 5), therefore we had to re-start the data collection approximately halfway through. Since the two simulators use different car physics models, it is not practical to compare data between them as they have different 'feels'.

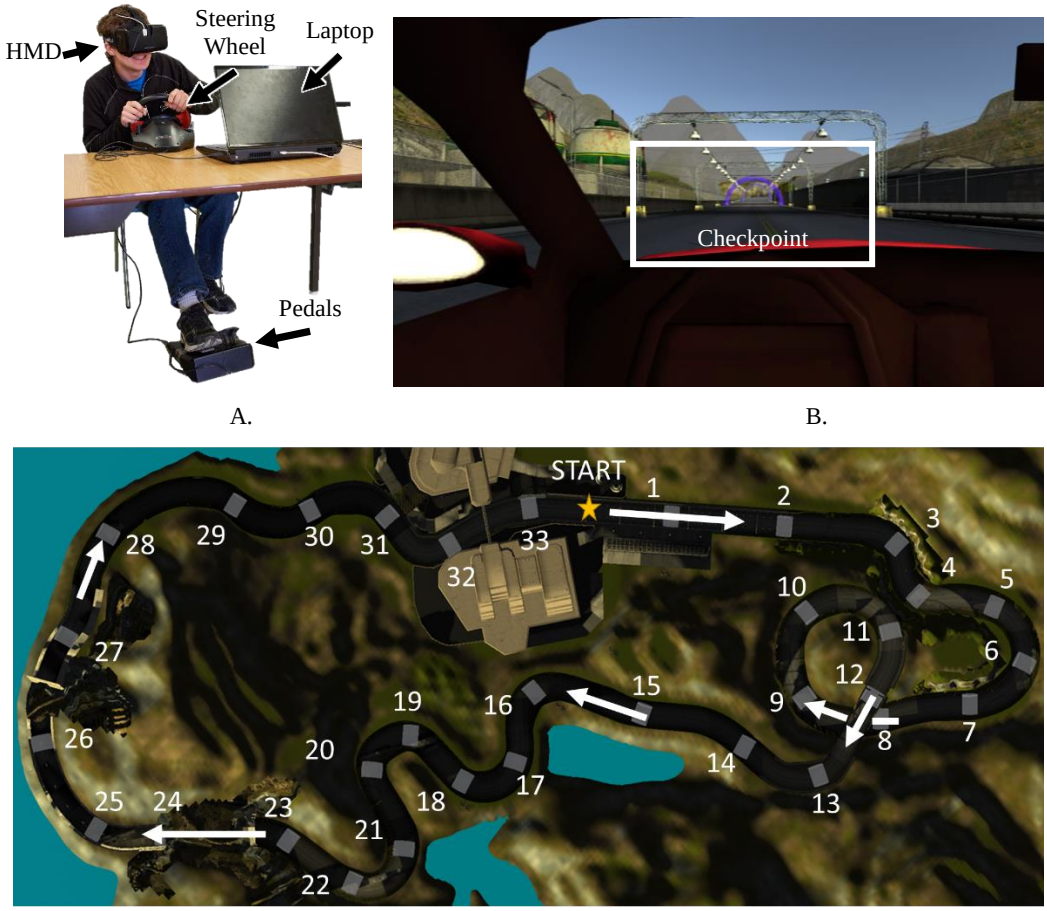


Figure 21: Virtual driving simulator.

A) the driving simulator setup, including a laptop to run the software, the steering wheel input device and a participant wearing an HMD. B) the driving simulator in first person. Checkpoints are represented by blue hoops. C) the track, to be navigated in a clockwise direction. The checkpoints are illustrated as gray boxes.

3.5 LSA Participant: Results

The LSA participant's MoCA score was 0 at the baseline, and it improved to 2 by the end of the training program. We observed modest improvement in his performance during the assessment phases of the VRN Home treatment, but we did not observe significant improvement in the real-world navigation evaluation. However, he showed significant improvement in the virtual driving task. The participant's wife's subjective assessment indicated that his mood improved, especially after treatment sessions, but she did not report any other noticeable improvement at home.

3.5.1 VRN Home Performance

We recorded the participant's trajectories in the VRN Home environment during both the training and assessment phases. The three most common path segments were Entrance-Bathroom, Entrance-Kitchen, Bathroom-Backyard, and Entrance-Backyard. In Figure 22, the hint time, navigation time, and navigation distance for the assessment phases for each of these three segments is plotted. The gaps between points would be filled by days when different paths from Figure 19 were used. The hint time indicates how much time the LSA participant viewed the hint arrows on that particular path segment, while the navigation time indicates how much time the participant spent navigating along that segment. We can see some modest improvement in A) and B) since the hint time, navigation time, and navigation distance decreased. In C), we can see that the first four sessions showed substantial improvement, but this trend did not continue in the fifth and sixth sessions.

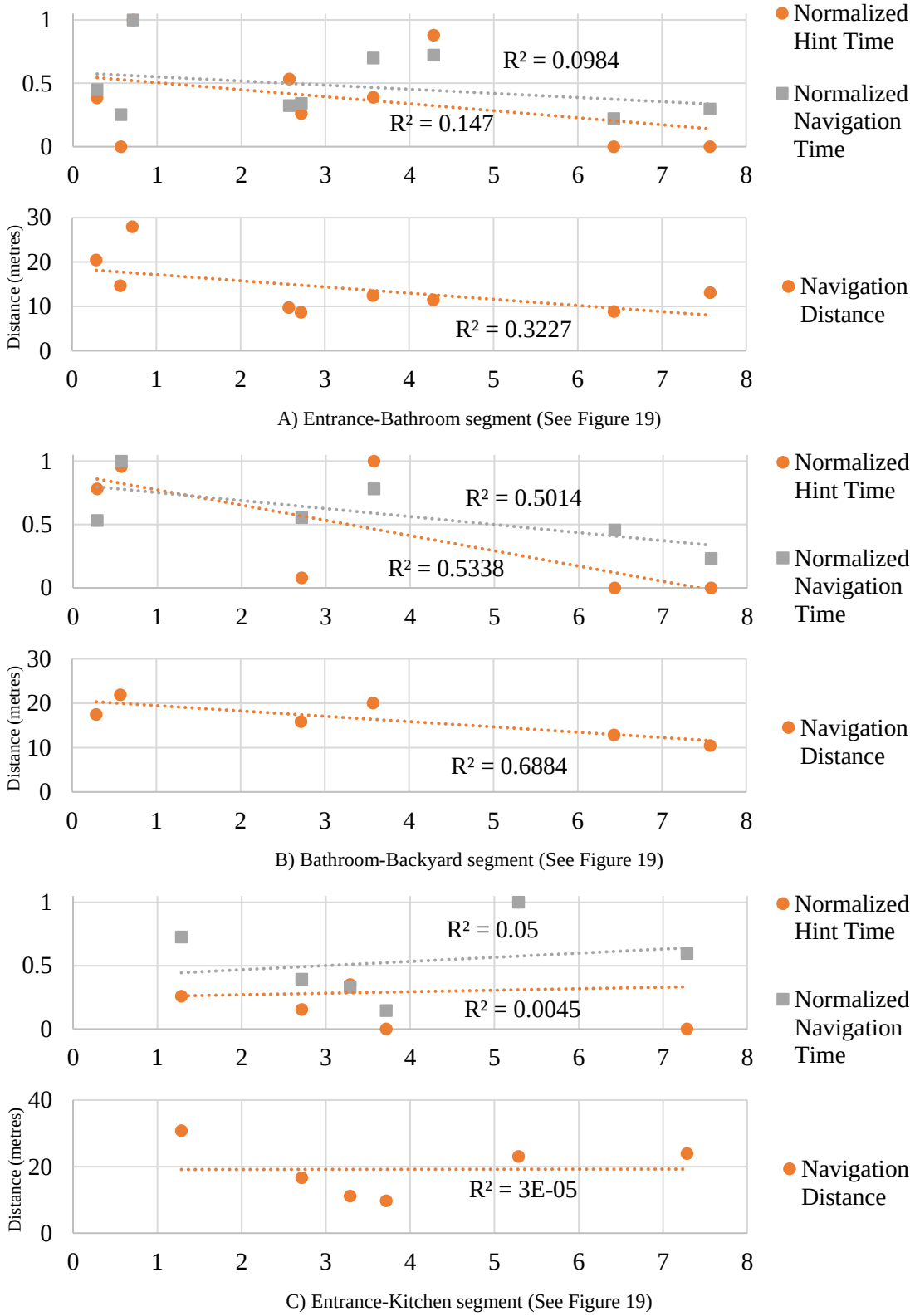
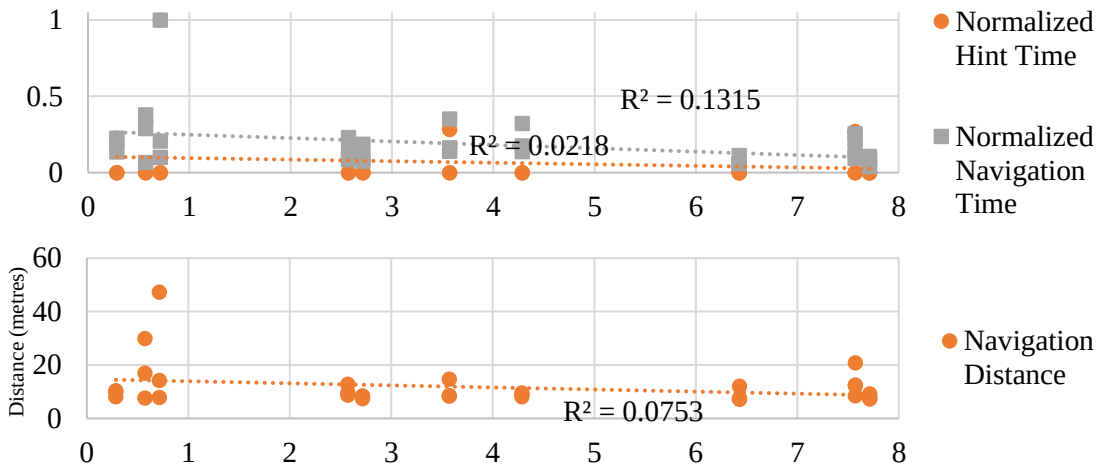


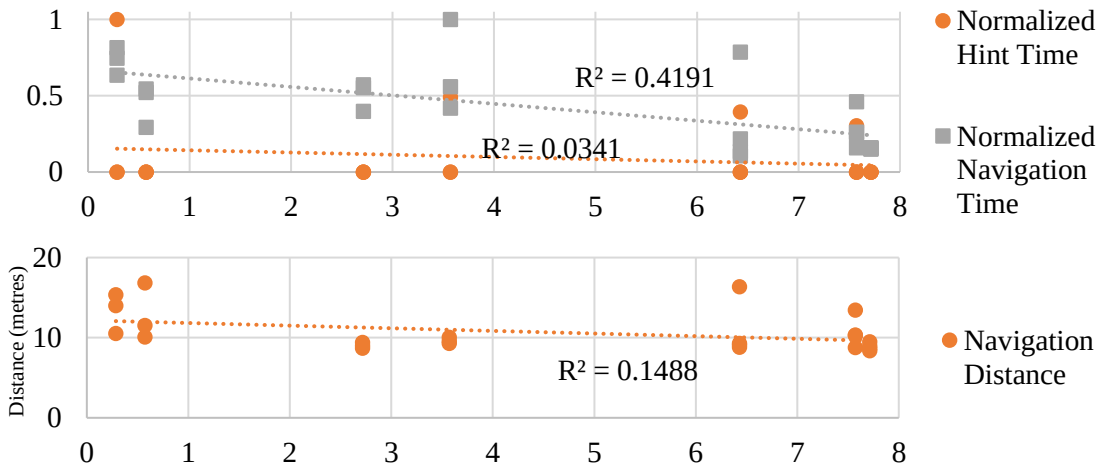
Figure 22: LSA participant navigation results without dog guide. (Assessment Phase)

For each segment, the upper plot illustrates Hint Time and Navigation Time. These times have been normalized (i.e. hint times have been normalized to maximum hint time, and navigation times have been normalized to maximum navigation time) so they can be plotted on the same graph. The lower plot illustrates the distance traversed to reach the destination. In all plots, the X-axis is the week number of the treatment.

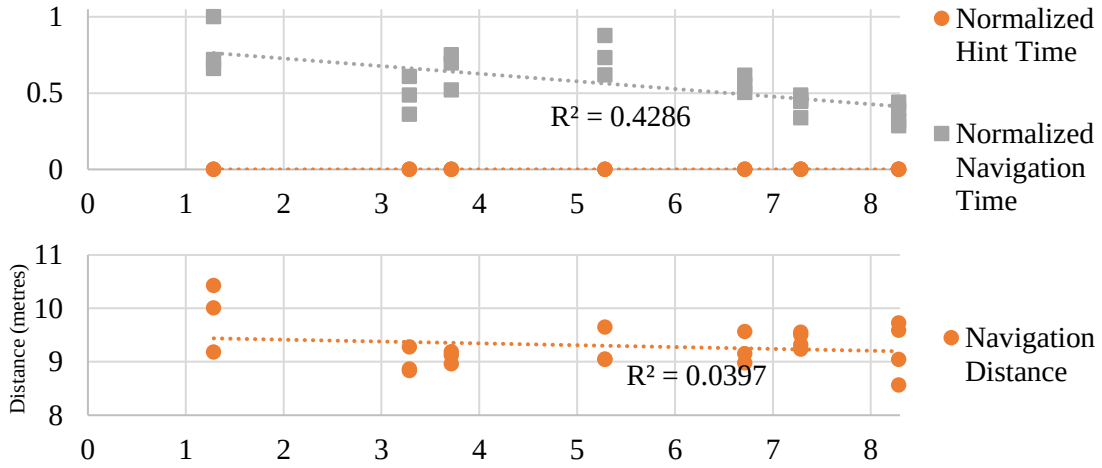
We were also interested in determining whether the LSA participant became more skilled at following the guide dog. We analyzed the participant's navigation ability as before on the same 3 path segments, but instead used the data collected during training phases while following the dog, as opposed to during the assessment phases. We observed a clear, but modest reduction in navigation time, but not in navigation distance. Hint time appeared to be largely unchanged, although the participant rarely required hint arrows during the training trials. Figure 23 shows the navigation times and distances during the training phases for the path segments, which correspond to the those shown in Figure 22.



A) Entrance-Bathroom segment (See Figure 19)



B) Bathroom-Backyard segment (See Figure 19)



C) Entrance-Kitchen segment (See Figure 19)

Figure 23: LSA participant navigation results with dog guide. (Training Phase)

Modest improvement can be observed in the LSA participant's ability to follow the dog guide. Note that he very rarely became lost when following the dog, and needed the hint markers only on rare occasions, therefore hint time is frequently 0. In all plots, the X-axis is the week number of the treatment.

The 3 path segments in Figure 22 include 4 waypoints: Entrance, Backyard, Bathroom, and Kitchen. In order to see if the participant encoded the house by forming a mental map of the relative layout of these waypoints, we considered alternative path segments between these 4 waypoints to see if navigation along them improved or not (See Figure 24). The changes in these alternative path segments are illustrated in Figure 25. A decrease in the navigation times and distances in these alternative segments would provide evidence that the environment was successfully encoded, but we did not observe much decrease in any of the alternative path segments; only the Backyard-Kitchen segment showed any improvement, but since those samples were taken close to each other, this could possibly be explained by learning effects. There are fewer data points than in Figure 22 because these path segments were chosen less frequently.

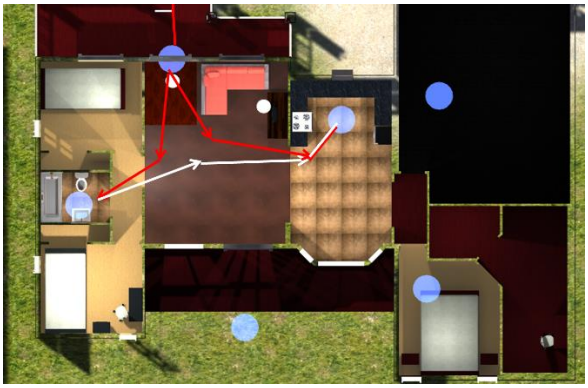


Figure 24: Example of an alternative path used in LSA VR path evaluation.
The Bathroom-Kitchen path can serve as an alternative to the Entrance-Kitchen/Entrance-Bathroom paths because it links their two endpoints.

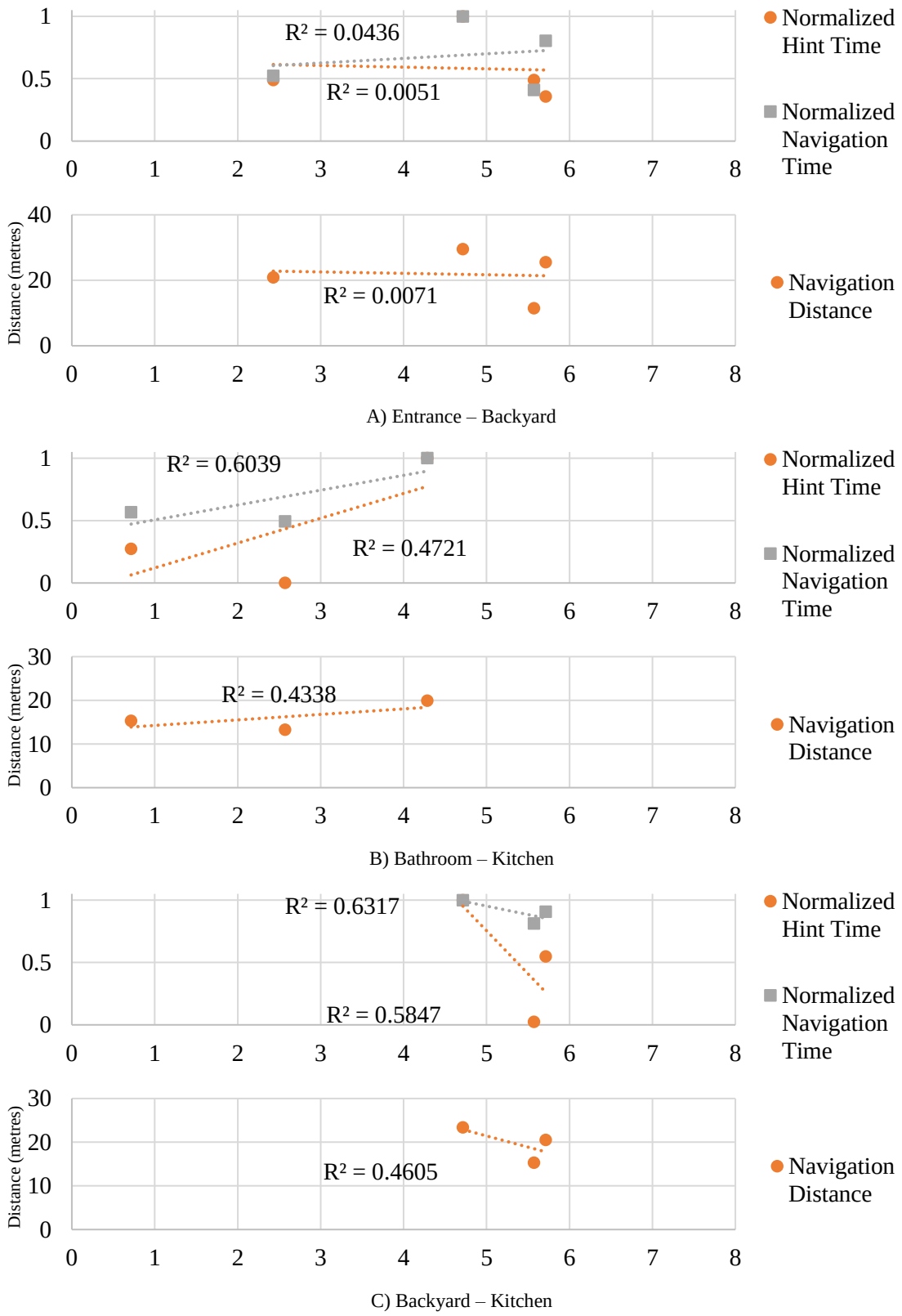


Figure 25: LSA participant's traversal results to check encoding.

For each segment, the upper plot illustrates Hint Time and Navigation Time and the lower plot illustrates the distance traversed to reach the destination.

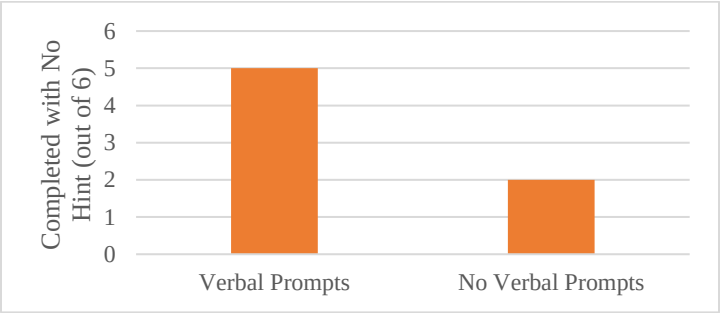


Figure 26: Comparison of LSA participant's navigation during the assessment phase with and without verbal prompts. This is a summary of the data in Table 7. The participant was more successful during the assessment phase when he was given verbal prompts, than when he was not.

Our original objective for the assessment phase was for the LSA participant to reproduce the entire path practiced during the training trails in the VRN home with no prompting at all. However, since we observed that he was having difficulties, we reframed the task by verbally prompting him to look in each of the rooms he had been shown (e.g. “Look in the bathroom,” or “look in the backyard”). We did *not* give directional prompts (e.g. “Turn left,” or “go straight”). We analyzed the effects of these prompts by comparing data from 12 different assessment phases where the participant had navigated once *with* verbal prompts, and once *without* verbal prompts (6 pairs of sessions, where both sessions occurred within one day of each other). This data is summarized in Figure 26, and shown in greater detail in Table 7. We found that in 5 of the 6 sessions with verbal reminders, the hint arrows were not required, but when room reminders were not given, the participant could only solve the segment without hint arrows 2 of the 6 times. Finally, we looked at changes in navigation distance and navigation time over the 6 pairs of sessions. In 2 pairs, we observed a substantially shorter navigation distance when verbal room reminders were given, and only 1 pair where the navigation distance was substantially longer (i.e. they differed by more than 10m). In the remaining 4 pairs, there was no substantial difference. In 2 pairs, the navigation time decreased by more than 2 minutes, and in 1 pair, it increased by more than 2

Table 7: LSA participant statistics with and without verbal prompts.

The participant was quicker and more efficient when verbal prompts were provided.

Path Segment	Date	Trial Number	With Prompts?	Hint Time	Hint Time Change	Navigation Time	Navigation Time Change	Navigation Distance	Distance Change
Entrance-Kitchen	2015-11-12	4	TRUE	171.5	108.21	274.19	132.87	40.01	13.17
	2015-11-12	9	FALSE	63.32		141.32		26.84	
Entrance-Bathroom	2015-11-19	5	TRUE	0	-47.87	32.34	-131.04	7.27	-20.61
	2015-11-18	9	FALSE	47.87		163.38		27.88	
Entrance-Backyard	2015-11-04	9	TRUE	0	-48.82	125.15	12.59	13.00	1.51
	2015-11-04	4	FALSE	48.82		112.56		11.49	
	2015-11-05	10	TRUE	0	-35.62	53.76	-166.35	11.96	-13.56
	2015-11-05	5	FALSE	35.62		220.11		25.52	
Bathroom-Backyard	2015-11-19	10	TRUE	0	0	127.728	49.837	12.79	0.47
	2015-11-18	9	FALSE	0		77.891		12.32	
	2015-11-19	5	TRUE	0	0	39.394	-29.994	9.81	-0.64
	2015-11-18	5	FALSE	0		69.388		10.46	

minutes. These changes in hint time and navigation distance suggest that verbal room reminders were helpful to the participant, since he only had to remember where the rooms were, not the order in which to visit them.

3.5.2 Evaluation of Translation to Real Life

In the experiments designed to evaluate the plausible translation of learning in VR to real life, we observed no improvement; the LSA participant was unable to locate his wife on any of the routes.

3.5.3 Driving Results

Over the course of the investigation period, we observed improvement in the LSA participant's driving ability, as measured by the farthest checkpoint he could reach within a 7-minute time trial. Since we needed to re-start this experiment when switching to the more stable simulator, (approximately half-way through) only the data collected with the improved simulator is presented below. We did not observe much change in the participant's crash rate.

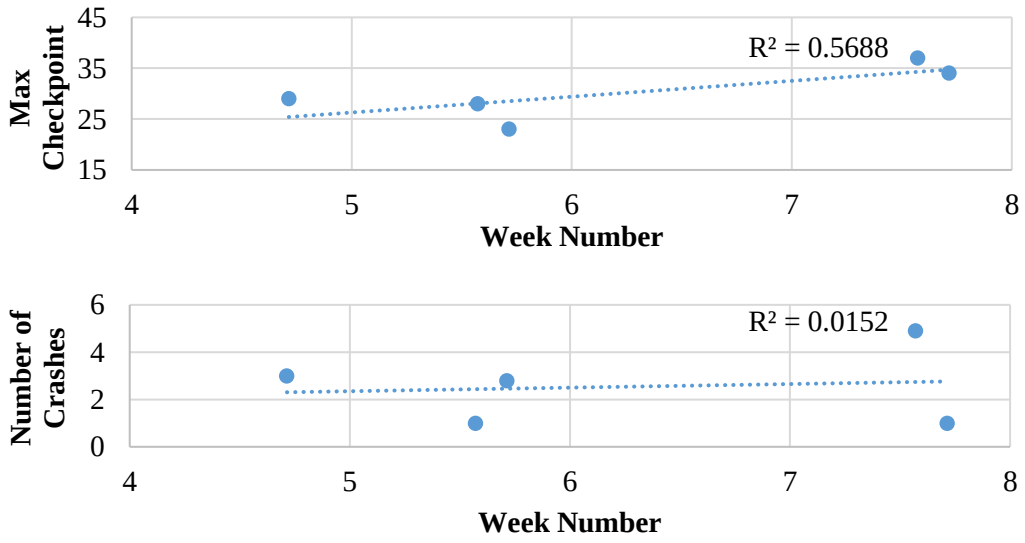


Figure 27: LSA participant's driving performance.

Checkpoints are numbered as in Figure 21C. In all 5 samples, the participant drove the car for 7 minutes.

At the end of the treatment, we attempted to perform a follow-up trial with the original "unstable" simulator. We were able to compare the participant's performance near the beginning of the treatment with his performance at the end of the treatment under similar circumstances. When the treatment first started, the participant was able to drive through 15 rings in 8 minutes, and at the end of the treatment, he managed to drive through 21 rings in 7 minutes. In other words, his average time per ring improved from 32 seconds to 20 seconds using the original simulator, after training in the new simulator. In the new simulator, he achieved a final average time of 12.35 seconds per ring, as shown in **Error! Reference source not found.**. We performed a follow-up trial 4 weeks after the final treatment, and found that the LSA participant's driving skill in the newer simulator had not deteriorated significantly; he reached checkpoint 33 in 7 minutes (12.72 seconds per ring) as compared with checkpoint 34 in 7 minutes (12.35 seconds per ring) in the final treatment trial.

Finally, we conducted 2 trials to compare the LSA participant's driving ability while using the HMD with his driving ability using a conventional laptop display. We observed a large improvement in his driving scores when using the HMD, making sure to control for order of presentation mode. See Figure 28.

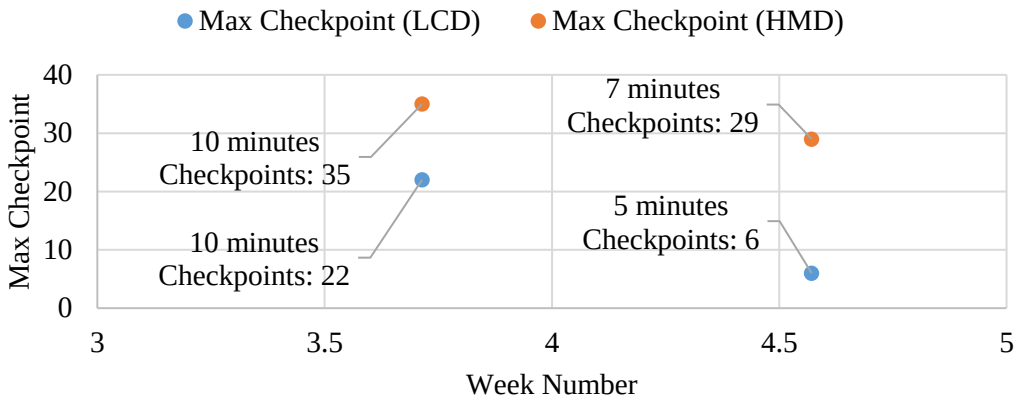


Figure 28: VR driving skill with HMD and conventional LCD.
In the first sample (October 22), the HMD trial occurred first, and in the second sample (October 28), the LCD trial occurred first. This was to account for learning effects.

3.6 LSA Participant: Discussion

The results from the second case study are somewhat less positive than the first, although the participant appeared to have experienced some subjective benefits. Figure 22 and Figure 23 show that there was very little improvement in the participant's ability to navigate through the VRN Home, as evidenced by the weak R^2 values. We hypothesize that some of his navigational difficulties stemmed from difficulties understanding what was expected of him during the assessment phase of each session. By providing him with verbal prompts of the room names, we suspect that we were able avoid confounding our results with the participant's language difficulties. We also observed that he performed the driving task more quickly and accurately with the HMD than with a conventional computer display.

We observed that the entire experience of VRN training was enjoyable for the LSA participant, and brought him a great deal of pleasure. Mood improvement is very important for people with AD; they often suffer from depression and so any activity that brings them pleasure and stimulates them is inherently of value. We believe that this experience was impactful for the LSA participant as he appeared to have formed long-term memories of his experiences; by the

final session, he would ask of his own volition if he would be able to “do the dog stuff.” While the participant could not always recall the names of objects/rooms in the house when prompted, on occasion he would spontaneously name a room or object he saw as he walked through the house. When the participant made an independent observation like this, it was always correct. We noted the dates when these observations occurred; they are plotted in Figure 29. The frequency of the participant’s spontaneous observations increased with time, possibly as he became more comfortable with the virtual environment and/or the tester. It is also possible that these observations increased in frequency due to the verbal prompting we gave him. This would likely have aided in the forming of word-place associations.

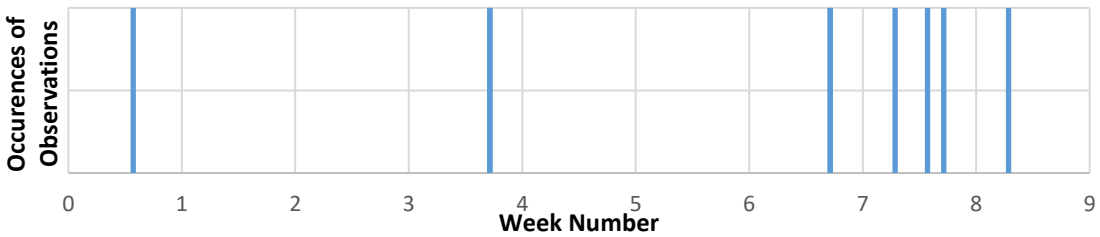


Figure 29: LSA participant’s spontaneous observations with respect to time.

As the participant spent more time in the VRN Home environment, he would vocalize observations about the rooms in the house or on the dog's behaviour, saying things like “The dog always goes to the bathroom” or, “now I'm in the kitchen”.

The LSA participant’s driving score changes are also of interest, as they show that using an HMD improved his ability to handle the virtual car. We suspect that since driving was a very important activity for the participant before he developed AD, the neural networks associated with driving might have had more redundancy than other networks in his brain. This is consistent with cognitive reserve theories [50], [52], as discussed in Section 1.3. We suspect that using the HMD made the experience immersive enough to allow the participant to activate those networks, but the experience conveyed by the conventional display was not immersive enough (it is also possible that since he was used to the HMD from the VRN Home treatment, shifting the presentation mode to a conventional display was confusing for him). This provides further evidence that by using an

HMD in the VRN Building assessment, we can more accurately capture the true spatial navigation characteristics of people, and reduce the dissociative effects of VR input devices. These effects can be dangerous confounding variables that distract the participant from navigating in the virtual environment, and skew results to be different from what they would be in a real, physical environment.

3.7 Summary

This chapter discussed the application of VRN applications within the context of a treatment program. Two participants at different stages of AD were trained in two different VRN – based cognitive treatment programs. The first participant was an individual at the onset stage of AD, and we trained him to navigate within in our VRN Building environment. The second participant had late-stage AD, and was trained to navigate in a virtual replica of a typical North American home. Our research goals were to find out whether individuals with Alzheimer's could learn and recall a new path in a VR environment, and if that training could translate to cognitive benefits in real life. The results showed that the early-stage AD participant was much better at learning a VR path than the late-stage AD participant. Furthermore, subjective feedback from the participants' primary caregivers indicated that the early-stage participant enjoyed cognitive improvement in his daily life at home.

Chapter IV: VRN Software Description

Since the VRN House was built using the Unity game engine, any software developer familiar with Unity should be able to work with it. The purpose of this chapter is to describe the custom software that we wrote to provide the necessary behaviours of the VRN environments described in Chapter II and Chapter III. We begin by describing each of the software modules that make the VRN environments work. This chapter may be read serially, but it is also intended to be used as a reference for a person seeking to understand the VRN Home and VRN Building software. To this end, keywords from the software will be stylized like `this`, to clarify to the reader that a component of the software is being referred to. For those readers who have never used the Unity platform before, a summary of its conventions is presented in Appendix B: Unity Overview.

4.1 Design of the VRN Building

The VRN Building is designed to be very modular and reusable. The most important element is the VRN Chair module. This module contains the scripts and components necessary for interfacing a VR environment with the VRN Chair. It includes a prototype `GameObject` (known in Unity as a 'prefab'), that can easily be inserted into scenes to allow them to be navigated with the VRN Chair. In addition to the VRN Chair module, there are two other modules: the VRN Building module, which contains scripts and assets specific to the VRN Building; and the VRN Environment module, which contains scripts and assets that can be used more generically in other VRN environments. All three of these modules are described in this section.

4.1.1 VRN Chair Prefab

The VRN Chair prefab is designed to allow other developers working with the VRN Chair to easily use it in their Unity projects. A screenshot of the prefab (with many of its associated components) in the Unity editor is illustrated in Figure 30.

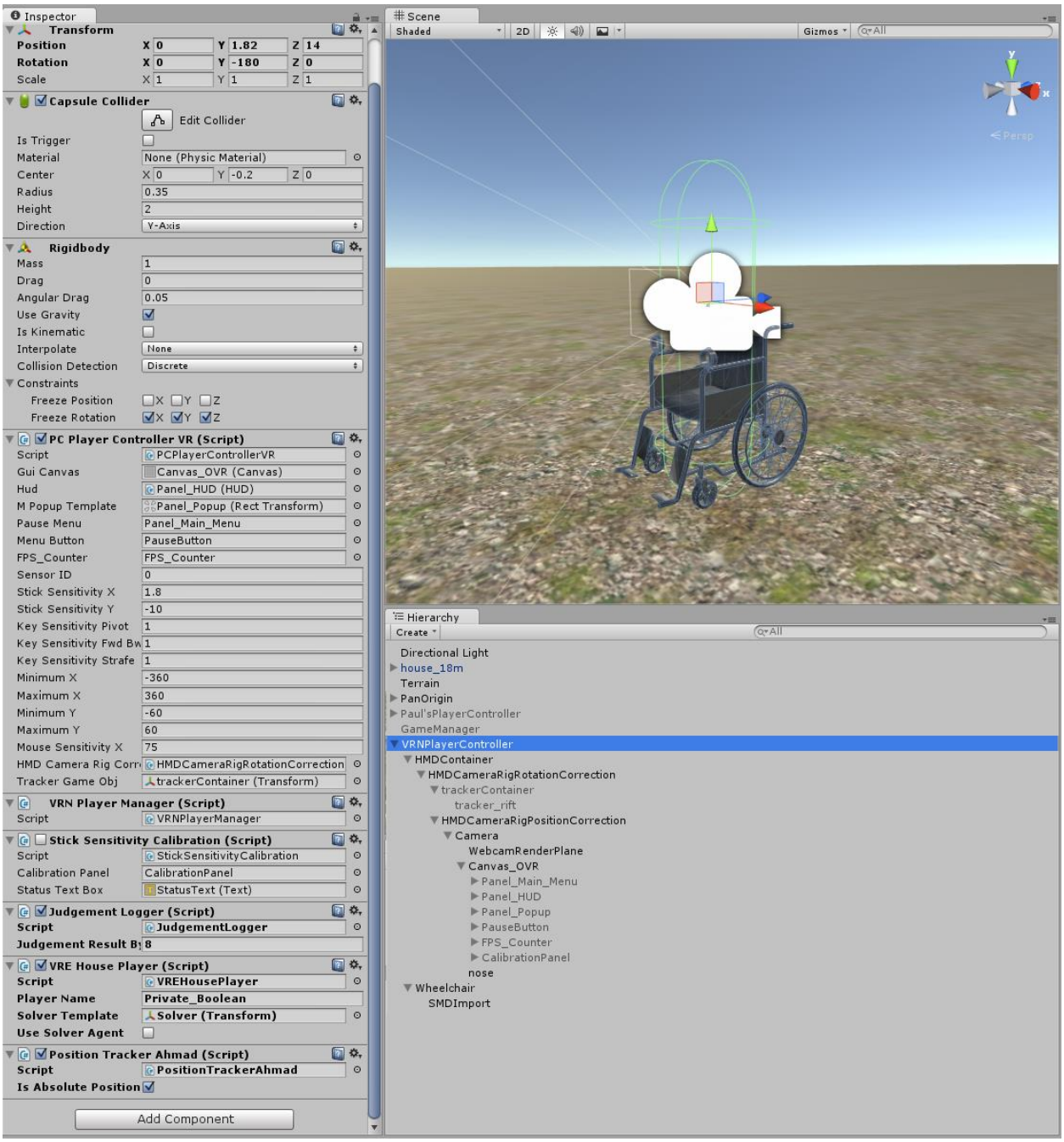
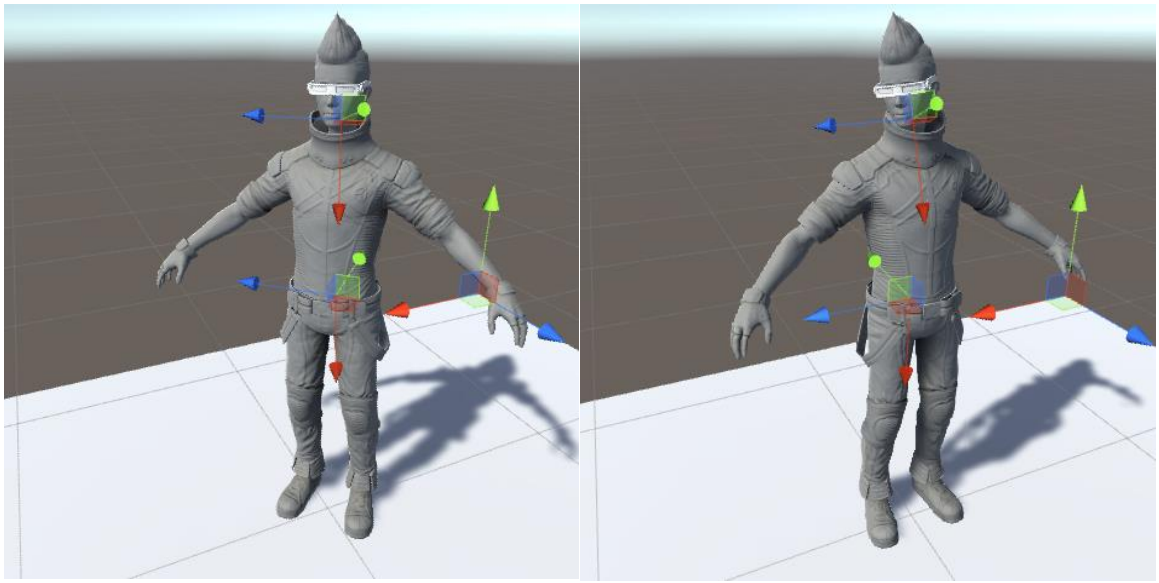


Figure 30: Unity prefab of VRN Chair

4.1.1.1 VRN Chair Kinematic Chain

The somewhat complicated `GameObject` hierarchy of the VRN Chair prefab is necessary in large part due to the cascading nature of the transformations that govern the behaviour of in-game entities. In the field of mechanics, this is called a 'kinematic chain.' If we model a human body as a kinematic chain, the head rotates in the coordinate space of the chest, and the chest

rotates in the coordinate space of the whole body, and the whole body rotates in the coordinate space of the world. This means that there are many combinations of rotations that will allow the head to move about in space; if you want to look to the left, you can turn your head to the left, rotate your entire body to the left, or perform some combination of the two, as in Figure 31. This is an example of an *over-actuated system*. In general, in an over-actuated system, a particular set of joint parameters can result in just one end effector state, but that end effector state may be achieved by a number of different sets of joint parameters [74].



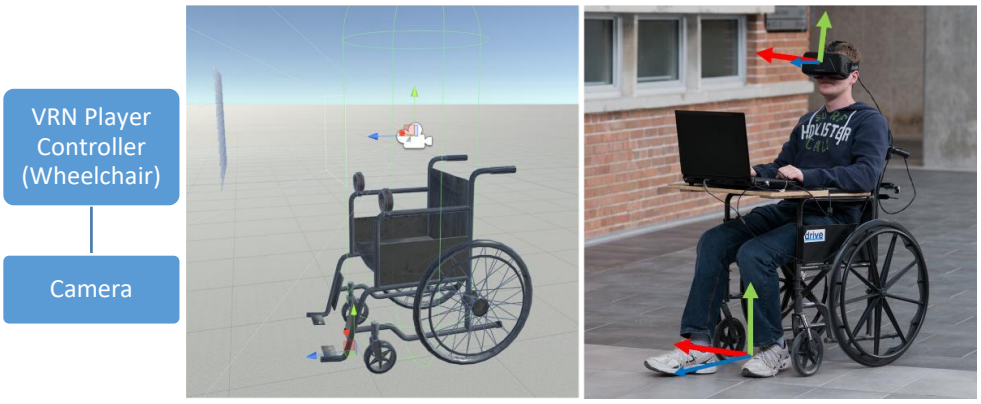
A B

Figure 31: Cascading transformations can combine in different ways to get the same net result.
A) the entire body has rotated 45 degrees to allow the person to look to the right. B) only the head has rotated. The net result is the same, but there were many ways to get there; this is an example of *over-actuation*.

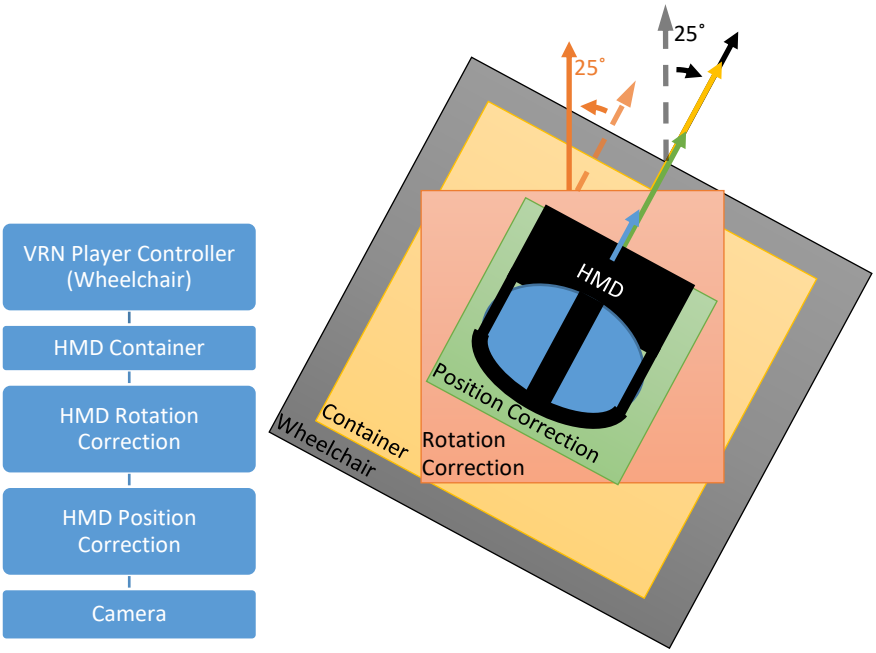
Kinematic chains work very well for animating objects in VR; by specifying translations or rotations at specific joints, we can affect the overall pose. This is called ‘forward kinematics’, because the system’s state is evaluated by “moving forward” from a given set of parameters. A possible design of the VRN Chair hierarchy using a forward-kinematical model is shown in Figure 32A. This model consists of an entity to represent the VRN Chair and the participant’s body, and a second entity to represent the participant’s head. Unfortunately, this scheme requires some

further modification in order to work properly. HMDs like the Oculus Rift track their rotation in absolute coordinates, relative to the real world. This means that if we were to calculate the virtual head rotation by adding the rotation reported by the HMD to the rotation reported by the VRN Chair, we would double the rotational contribution of the VRN Chair, since both the HMD *and* the VRN Chair detect it. We can solve this by adding an additional joint (i.e. `GameObject` in the hierarchy) between the virtual head and the virtual body that is made to rotate counter to the VRN Chair's rotation, effectively 'unwinding' it. We have actually added 3 extra joints to compensate for this and other similar issues, as illustrated in Figure 32B, and described below.

1. The **Rotation Correction** joint 'unwinds' the wheelchair's rotations, so that they are not counted twice. Additionally, to lock the rotation of the HMD (to implement decoupled mode, Section 2.3.2), we created a script component for the HMD Rotation Correction link called `HMDLock`. This script allows a programmer to specify which motion vectors to lock, preventing the joint from rotating (or translating) along those vectors, within its space.



A



B

Figure 32: Kinematic chain for virtual model of VRN Chair.

A) a simplified model that suffers from the 'double rotation' issue. There are two entities: the participant's body and the participant's head. B) the Rotation Correction joint is added to the kinematic chain to 'unwind' a duplicated transform. The Container and Position Correction links are also added to the kinematic chain for the developer's benefit

2. The **Position Correction** joint is a child of the Rotation Correction joint. It allows us to disable Oculus' neck model, by 'unwinding' translations that it introduces. In the absence of a video feed for optical-based position tracking, Oculus's algorithm falls back to an *inertial-only* system that models the user's head and neck. This neck model is not used when optically enhanced tracking is active. The neck model is designed assuming

that the user is using the HMD sitting in a stationary chair facing forward, and models subtle translations experienced by the HMD as the user moves their head. When a user rotates their head about their neck, the true axis of rotation of the HMD is about the user's neck (as opposed to the IMU in the HMD), and so the HMD translates in space by a small amount as the participant rotates their head. Since the IMU is unable to detect this translation, the neck model computes it based off of physical parameters of the user's head and neck. This neck model is illustrated in Figure 33. However, when the participant's head rotation is caused by rotation of the wheelchair, rather than rotation of the user's head about their neck, the translation computed by the neck model results in the participant experiencing peculiar head drifting, so we disable it. Disabling this correction could cause a small amount of discomfort when the participant actually *does* rotate their head about their neck, but it is less disorienting than having the point of view incorrectly translating about.

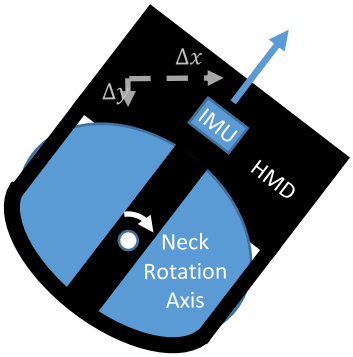


Figure 33: Oculus's neck model.

In the absence of a tracking camera, (see Figure 2) the pose-estimation algorithm falls back to an inertial-only model, that models a user's neck. Rotating the HMD will therefore also translate the camera's view, in order to account for small translations in space experienced by the user's eyes, which rotate about a point behind them (i.e. the neck rotation axis).

3. The **HMD Container** is a master joint in which the other joints (and camera) are encapsulated. By encapsulating the other joints, the entire assembly can be easily positioned in the scene, and the 'unwinding' movements can occur transparently to the developer. The HMD Container joint is necessary in cases where the developer wishes to change the camera view (e.g. in the VRN Building, we need to show the participant an external view of the rotating building). The original version of the Oculus plugin for Unity did not handle camera swaps very well, and it was more reliable to simply move the camera to a new position. Encapsulating the entire camera assembly in the HMD Container makes this more seamless.

4.1.1.2 The `PCPlayerControllerVR` Script

The code that actually makes the VRN Chair work is the `PCPlayerControllerVR` script component. This component extends a different script called `PCPlayerController`. The `PCPlayerController` script is responsible for generically handling inputs from the participant and translating them into interactions in the virtual environment (e.g. moving around). It also connects to UI elements such as popups and the Heads Up Display (HUD) and allows the software to display information to the participant (see below for more information). The `PCPlayerControllerVR` extends the `PCPlayerController` class by handling input specific to the VRN Chair. This includes processing the signals that represent forward/backward information, but also handling rotation updates from the VRN Chair's IMU. The IMU data is handled by the `IMUSensorReader` class, which communicates with the IMU over a USB serial port using a proprietary binary format. This binary format is described in detail in the IMU source code documentation (Appendix A). Figure 34 shows a class diagram of the `PCPlayerController` code.

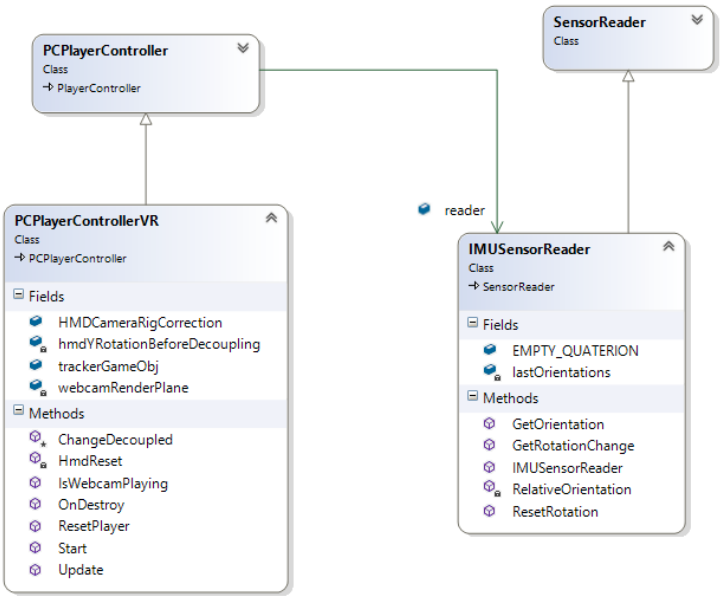


Figure 34: `PCPlayerController` class diagram.

The `PCPlayerControllerVR` inherits from `PCPlayerController`, which provides basic services such as joystick input handling and popup display. The `SensorReader` handles incoming packets from the IMU. `IMUSensorReader` extends the `SensorReader`, and buffers rotation data for use by the `PCPlayerController`.

The `PCPlayerControllerVR` script is configured by means of two configuration files:

1. **ChairSettings.xml** controls basic settings such as IMU configuration settings and decoupling options
2. **Controls.xml** specifies the sensitivities of the various possible input axes.

Further details of these files can be found in the documentation in the source code in Appendix A.

4.1.2 VRN Building Software Architecture

The software architecture of the second-generation VRN Building follows an object-oriented philosophy. It is designed around the basic principle of a 'Task Environment' that contains a participant (or player) that completes tasks. The environment can give the participant directions, wait for input, and track which tasks get completed. Overall, the system is split into 3 basic packages:

1. The **VRN Environment** package contains the abstract classes needed to define a task-based environment for Virtual Reality Navigation applications.
2. The **VRN Building** package contains the classes and prefabs necessary to implement functionality specific to the VRN Building. This includes the engine for choosing which tasks to activate/deactivate, as well as the "judgement logger," which classifies the rooms visited by the participant.
3. The **VRN Controller** package contains the classes and prefabs necessary to use the VRN Chair within the context of a Unity application.

4.1.2.1 VRN Environment Package

In order to encourage software re-use, we developed a simple framework to implement the generalized Virtual Reality Environment (VRE) task environment features in a re-usable way. This was done with an eye to the future, when other types of task environments may be created, such as the VRN Home, whose functionality was introduced in Section 3.4.1.

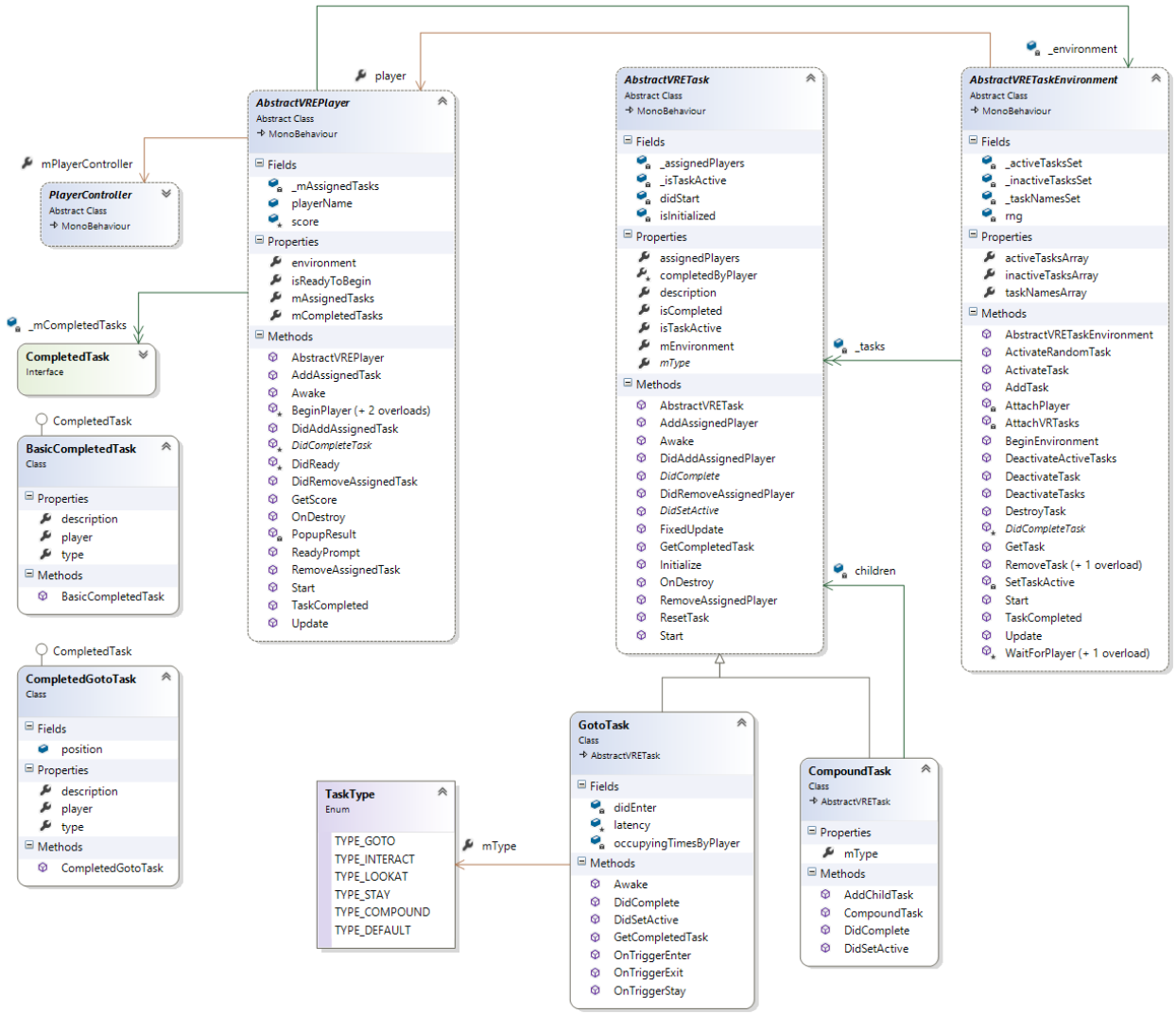


Figure 35: UML diagram of VRN Environment framework.

As can be seen in Figure 35, there are several classes in the VRN Environment package, but the most important ones are `AbstractVRETaskEnvironment`, `AbstractVRETask`, and `AbstractVREPlayer`. These abstract classes provide base behaviours for `GameObjects` in the virtual environment. They are extended by concrete subclasses, that actually implement behaviour *specific* to the application. The `AbstractVRETaskEnvironment` (or rather, its concrete subclass) manages the entire virtual environment, keeping track of the various objects that can be interacted with, and keeping track of where the player is and what they're doing. This

class is responsible for originating the event chains that occur in the virtual environment, specifically: the ‘Initialization’ event chain, and the ‘Completed Task’ event chain.

The ‘Initialization’ event chain is not actually triggered within the VRN Environment framework; but rather by one of the concrete subclasses that make it up. We did this because many VR experiences only start after the participant has been asked to confirm that they are ready to start. This method also gives the participant a chance to put on the HMD and get comfortable before launching into the navigation exercise. In the VRN Building program, this initialization is handled by the `VRE`**`House`**`TaskEnvironment` (which extends `AbstractVRETaskEnvironment`). An important part of the ‘Initialization’ event chain is the detection of all the task `GameObject`s, and the player `GameObject` in the Task Environment. These are detected automatically at start-up based on `GameObject` tags, so there is no need for a developer to manually assign references in the Unity editor; they only need to ensure that all tasks in the Unity editor are tagged with “Task” and the player `GameObject` is tagged with “Player.”

The ‘Task Completed’ event chain (Figure 36) begins when a concrete subclass of `AbstractVRETask` detects that it has been completed, i.e., when its `DidComplete()` function returns ‘True’. The `AbstractVRETask` communicates this to the `AbstractVRETaskEnvironment` by calling the `TaskCompleted()` callback. In this callback, a concrete subclass of `AbstractVRETaskEnvironment` tries to process the event. Depending on the application, this could lead to another task being spawned, or the trial ending, or some other behaviour. Finally, the `AbstractVREPlayer` object is notified that it successfully completed a task. These events can be passed on to a concrete subclass of `AbstractVREPlayer` and be handled in a way that is appropriate to the application. (e.g. show

a popup in the player's view and/or play an animation to indicate to the player that they did something well.)

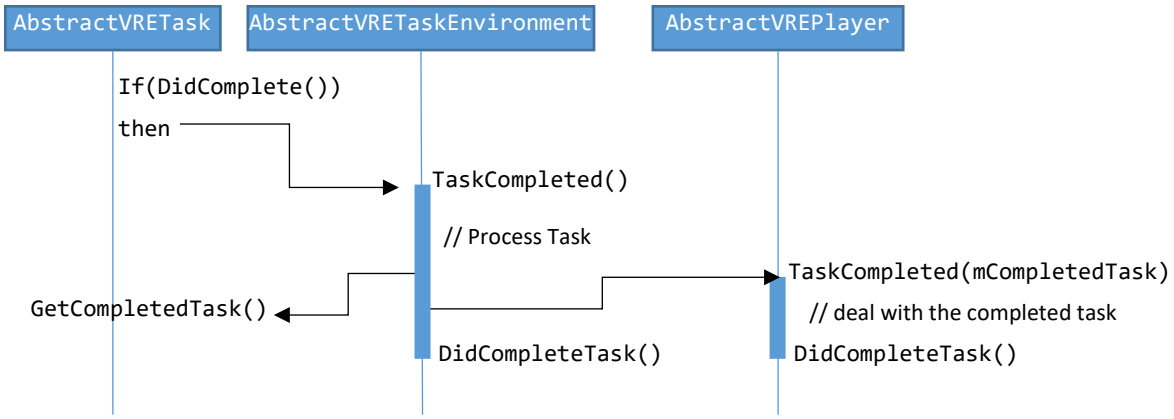


Figure 36: Event flow of the 'Task Complete' event chain in the VRN Environment framework.

The basic and re-usable options of the `AbstractVRETaskEnvironment` are controlled by the **AbstractTaskEnvSettings.xml** configuration file.

4.1.2.2 VRN Building Package

The VRN Building package provides the functionality that is unique and specific to the VRN Building application that cannot be generalized to other cases. A UML diagram of the various C# classes that make up this package is illustrated in Figure 37. The classes in Figure 37 have been categorized based on which part of the VRN Building they belong to.

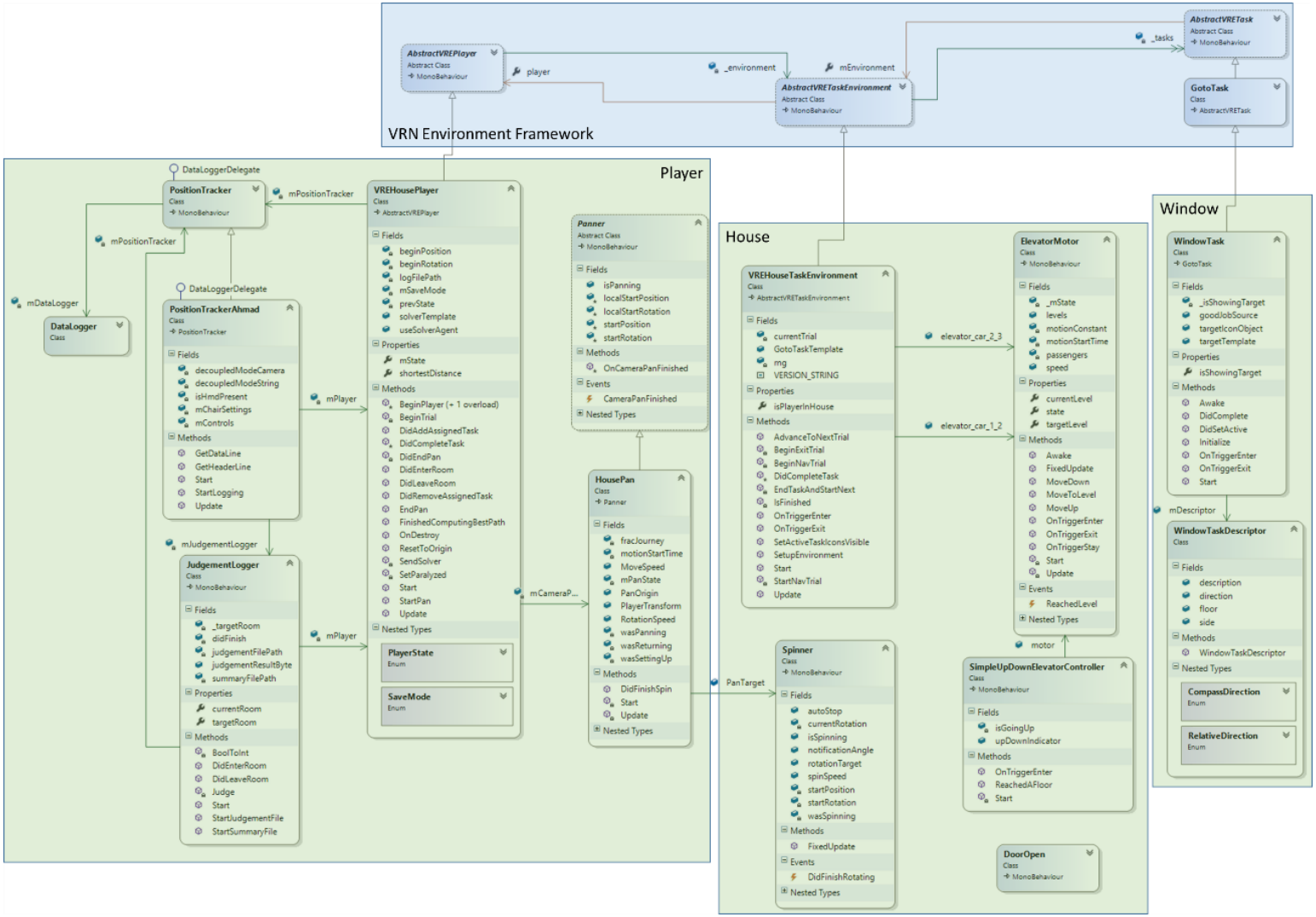


Figure 37: UML diagram of the VRN Building package.

A condensed illustration of the VRN Environment framework is highlighted in blue. The green-highlighted modules form the VRN Building package, and many of these classes extend the abstract base classes of the VRN Environment framework. The classes are grouped by functionality.

The VRN Building environment is realized in a single Unity scene, and is shown in Figure 38. There are three main entities in this virtual environment:

1. The **player** (VRNPlayerController): is realized as a `GameObject` that is based upon the VRN Chair prefab as discussed above. It contains a script component that extends `AbstractVREPlayer`, as well as some other script components that are described in greater detail below.
2. The **building** (house_18m): `GameObject` contains a script component that extends `AbstractVRETaskEnvironment`.
3. The **windows**: these `GameObjects` are children (or more appropriately, 'grandchildren') of the building `GameObject`, and are grouped beneath the Tasks `GameObject` in the Unity editor (See Figure 38). Each window `GameObject` contain a script component that extends `AbstractVRETask`.

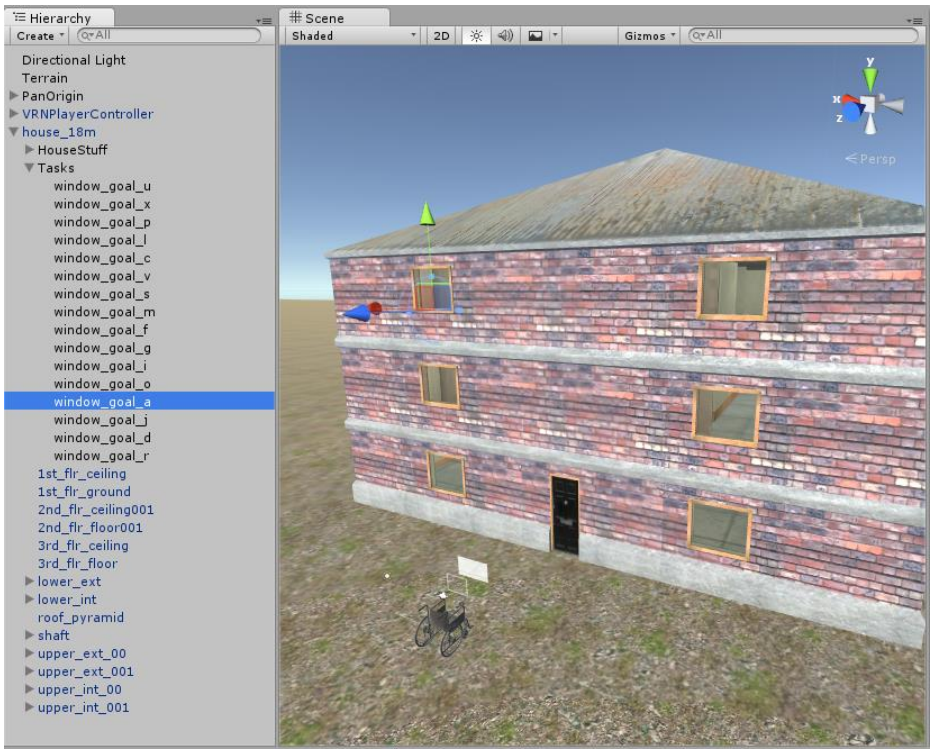


Figure 38: VRN Building package in Unity Editor.

The `GameObject` hierarchy is visible on the left.

Player

The base VRN Chair Prefab (Section 4.1.1) was augmented for use in the VRN Building environment by the addition of several script components. One of the most important functions of the player module is logging data. This functionality is provided by a variety of classes in order to produce data files that are backwards-compatible with the first-generation VRN Building (Section 2.2). The `PositionTracker` and its subclasses (`PositionTrackerAhmad` and `AdvancedPositionLogger`) are responsible for recording the player's movement in the virtual world. They spawn a separate thread by means of the `DataLogger` class, which polls the `PositionTracker` object for a data line to write out to a log file. The log data is written to a file in a separate thread, because file I/O operations can sometimes block. If this were to happen in the main run-loop (i.e. in the unity `Update()` function), it could delay the drawing of the next frame, which would negatively affect the simulation's framerate. The

`PositionTrackerAhmad` specifically provides lines of data that are compatible with the data logging conventions used in the first-generation VRN Building. It produces a comma-delimited text file containing the fields described in Table 8.

Table 8: Data fields logged by `PositionTrackerAhmad` in summary file.

QUANTITY	DESCRIPTION
POSITION 3-TUPLE (X, Y, Z)	Current position, relative to world origin in metres
ROTATION 3-TUPLE (X, Y, Z)	Wheelchair angle in degrees. (Only the Y-component is really useful, as it describes the yaw).
ELAPSED DISTANCE	Total distance traveled in metres (sum of Euclidean distances between all sampled positions)
EXPECTED DISTANCE	Minimum distance to target in metres (Calculated in advance)
ROOM CLASSIFICATION	The byte-representation of the room classification: BIT: 7-4 / 3 / 2 / 1 / 0 DESCRIPTION: Unused / In No Room / Correct Floor / Correct Wall / Correct Left/Right
TARGET CHARACTER/CURRENT CHARACTER	Decimal code representation of target window (ASCII, so A = 65) (for compatibility reasons). When not in a window, use the '-' character (ASCII 45)
TARGET LETTER/CURRENT LETTER	ASCII representation of target window and current window for readability. When not in a window, use the '-' character.
SENSITIVITY FACTOR	Sensitivity setting as described in VRN Chair .
DECOUPLED/DECOUPLED CAMERA	'Decoupled' shows whether the participant is in the decoupled state. In post-processing, we can subtract time spent decoupled as it inflates the measured navigation time. 'Decoupled Camera' describes which decoupled mode was used (See Section 4.1.1.2).
HMD	Is the HMD being used? The Unity-based VRN Building can also be used like the first-generation building, using a laptop display instead of the HMD.
VERSION STRING	Version of the building that created this log file.

The `PositionTrackerAhmad` is also responsible for managing a separate logging entity called the `JudgementLogger`. This script component handles a callback whenever the player enters a window zone, and logs whether the player chose a window on the correct floor, wall, or side (left/right). Since this script component only performs file I/O when the player enters or leaves a window zone, (and is therefore infrequent relative to the framerate) a separate logging thread is not needed as in the case of the `PositionTracker`. The `JudgementLogger` produces a second, formatted text file for each trial that summarizes visited windows, and a

summary file for the entire assessment. The summary file is updated only when the participant enters a room. The fields in the summary file are described in Table 9.

Table 9: Data fields logged by `JudgementLogger` in summary file.

QUANTITY	DESCRIPTION
TARGET CHARACTER/CURRENT CHARACTER	Decimal representation of target window (e.g. A = 65) (for compatibility reasons).
CORRECT FLOOR, CORRECT WALL, CORRECT LEFT/RIGHT	1 = true, 0 = false for each.
ELAPSED DISTANCE	Elapsed distance in metres.
ELAPSED TIME	Elapsed time since start in seconds. This includes the time spent in decoupled mode.

In addition to logging data, the player module also needs to perform some simple interactions with the environment. Specifically, the player's viewpoint must pan out to a point several meters away from the building, so the participant can watch the building spin in order to view the target window before the trial starts. After this, the viewpoint needs to return to the player module's body. This behaviour is handled by the `Panner` (specifically its subclass, `HousePan`) and `Spinner` script components. The `Panner` translates the viewpoint to the distant view and triggers the `Spinner` (which is a script component applied to the building itself, which spins the building). Once the spin is complete, the `Panner` returns the viewpoint to the player's body.

Windows

The windows are controlled mainly by the `WindowTask` script component, which extends `GotoTask`. Since the `WindowTask` inherits from `GotoTask`, it mainly implements aesthetic functionality such as handling the display of the '**X**' icon and playing the 'Good Job' notification when the participant completes their task. Each window is also assigned a single `WindowTaskDescriptor` that stores meta-data for the window, such as its ID and position information (i.e. floor, wall, and left/right coordinate).

Building

The building module is a somewhat complex construct that consists of both static and dynamic geometry. It has a script component attached to it called `VREHouseTaskEnvironment` that extends `AbstractVRETaskEnvironment`. This script component is responsible for setting up the initial conditions of the environment for the current trial, such as resetting the interior elevators to an appropriate height and deciding which `WindowTask` to mark as active. These conditions are determined by consulting the **HouseSettings.xml**, **TaskOrder.xml** and **HouseTemp.xml** files. **HouseSettings.xml** contains settings specific to the VRN Building, such as how many trials should be performed, what task the participant should be trained on, etc. **TaskOrder.xml** specifies what tasks should be presented to the participant and in what order (and whether or not the task order should be randomized). **HouseTemp.xml** temporarily stores the current trial number and current participant ID; this file is deleted at the end of the session, and its presence indicates an on-going experiment. When the player completes a `WindowTask` (by navigating to it), the `VREHouseTaskEnvironment` asks the player to confirm that they are finished the trial and ready to begin the next one (by calling the `WaitForPlayer()` co-routine defined in `AbstractVRETaskEnvironment`). Then the various temporary files that the software uses to store its state are prepared for the next trial, and the entire scene is re-loaded. Since the setup is done based on the parameters stored in the state files, the next trial will be loaded.

The building also has some script components to control the behaviour of various interactive features. These include l to automatically open/close the front door and logic to move the elevators up and down, as well as to hide the target '**X**' when the player enters the building.

4.2 Design of the VRN Home

The VRN Home is a different environment from the VRN Building. Since the VRN Home was targeted at later-stage AD patients, we needed to modify the experience compared with the VRN Building. To this end, it is implemented as a more realistic house, complete with furniture and appliances. It was designed to be more friendly and accessible than the VRN Building, which had no interior landmarks and therefore had a more austere aesthetic. Furthermore, the VRN Home includes a guide avatar, which teaches the participant routes in the house (and therefore the layout of the house). This guide takes the form of a virtual dog. In order to increase participant enjoyment and improve immersion, the virtual dog is animated and varies its behaviour based on what the participant is doing. Specifically, if the participant is taking too long, the dog will lie down to rest.

4.2.1 VRN Home Software Architecture

Similar to the VRN Building environment, the VRN Home environment is built around the idea of completing spatially separated tasks, and tracking participant motion over time. Therefore, the VRN Home is based off the VRN Environment framework, just like the VRN Building. The code and assets that are unique and specific to the VRN Home environment are contained in the VRN Home package.

The VRN Home Package contains the scene illustrated in Figure 39 (hood_5_1_2). The three most important assets in this scene are the house and furniture geometry, which are contained in `house_populated`, the virtual dog (`VRNLeader`), and the `VRECheckPointEnvironment`, which manages the different `GotoTask`s the player interacts with. Note also the `VRNPlayerController`, which is re-used from the VRN Building environment. Like the VRN Building, this scene makes use of static light maps to improve performance, but also uses light probes to improve illumination effects inside the house. The light probes appear as coloured spheres in Figure 39.

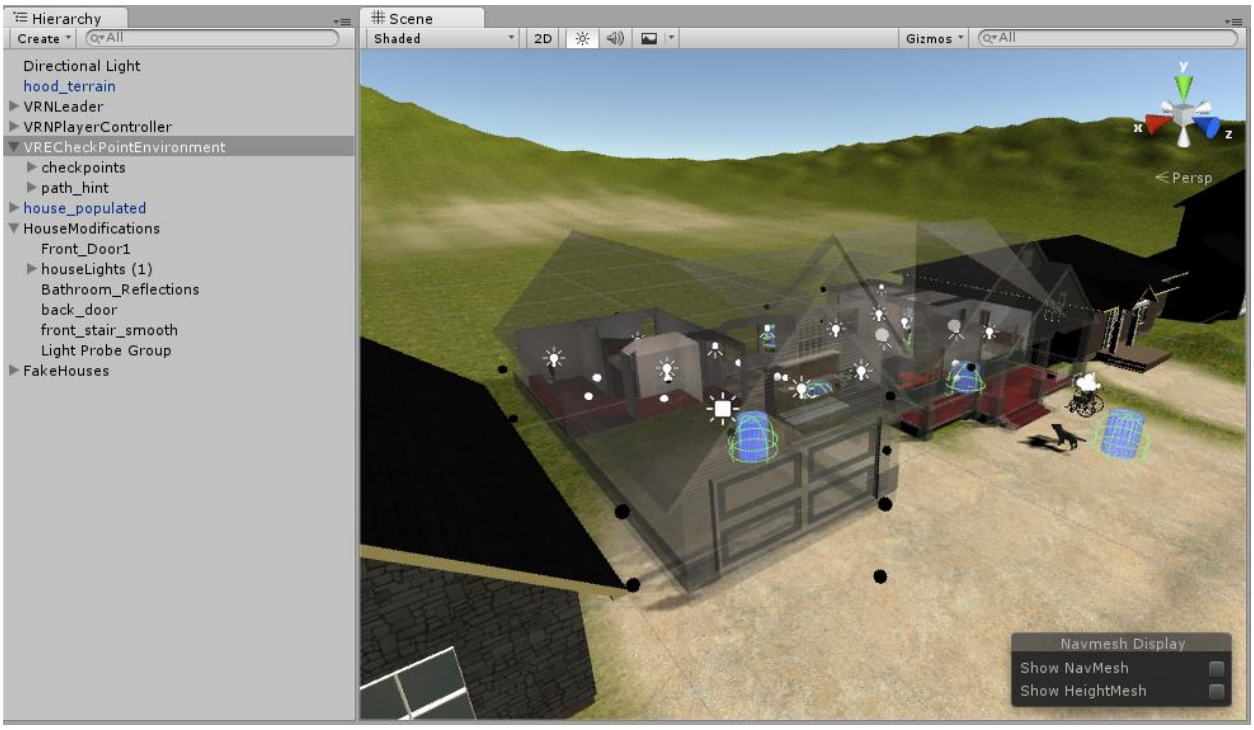


Figure 39: VRN Home in the Unity Editor.

The `GameObject` hierarchy is visible on the left. External walls of the house have been made transparent to show the interior. Note the use of light probes (black/white spheres).

Figure 40 contains the UML diagram of the script components included in the VRN Home package. This diagram is very similar to the one for the VRN Building, although it is somewhat simpler, since there was no need to pan the camera or adhere to the complex data logging requirements of the VRN Building. Note how the `VREHousePlayer` and `VREHouseTaskEnvironment` shown in Figure 37 are replaced by the `CheckpointPlayer` and `CheckpointTaskEnvironment` respectively, as shown in Figure 40.

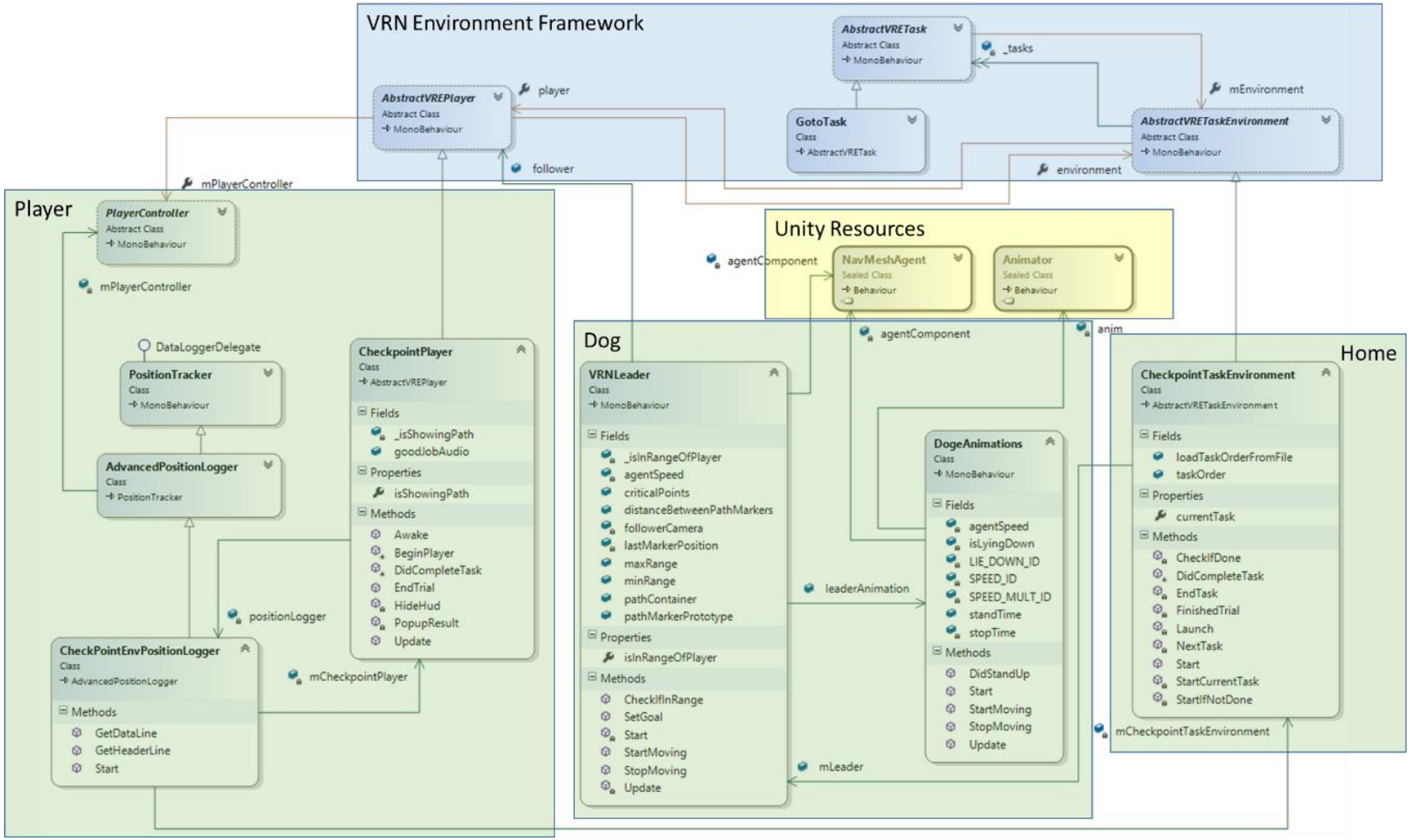


Figure 40: UML diagram of VRN Home software package.

The VRN Environment package is condensed and shown in blue. The green boxes represent the classes of the VRN Home package.

The VRN Home introduces the virtual dog, which leads the player through the house. The virtual dog is able to navigate thanks to Unity's `NavMeshAgent` component. To use the `NavMeshAgent`, a 'NavMesh' must be created for the scene in the Unity editor (See Figure 41). This mesh is created based on the static geometry in the scene (such as walls and ramps) and dictates where the `NavMeshAgent` is allowed to go. To send the dog to a particular target, the `NavMeshAgent` is assigned a destination position, and a maximum navigation speed. Once a non-zero speed is assigned, the `NavMeshAgent` accelerates towards the destination, staying on the NavMesh. In this way, it avoids the static geometry of the scene.

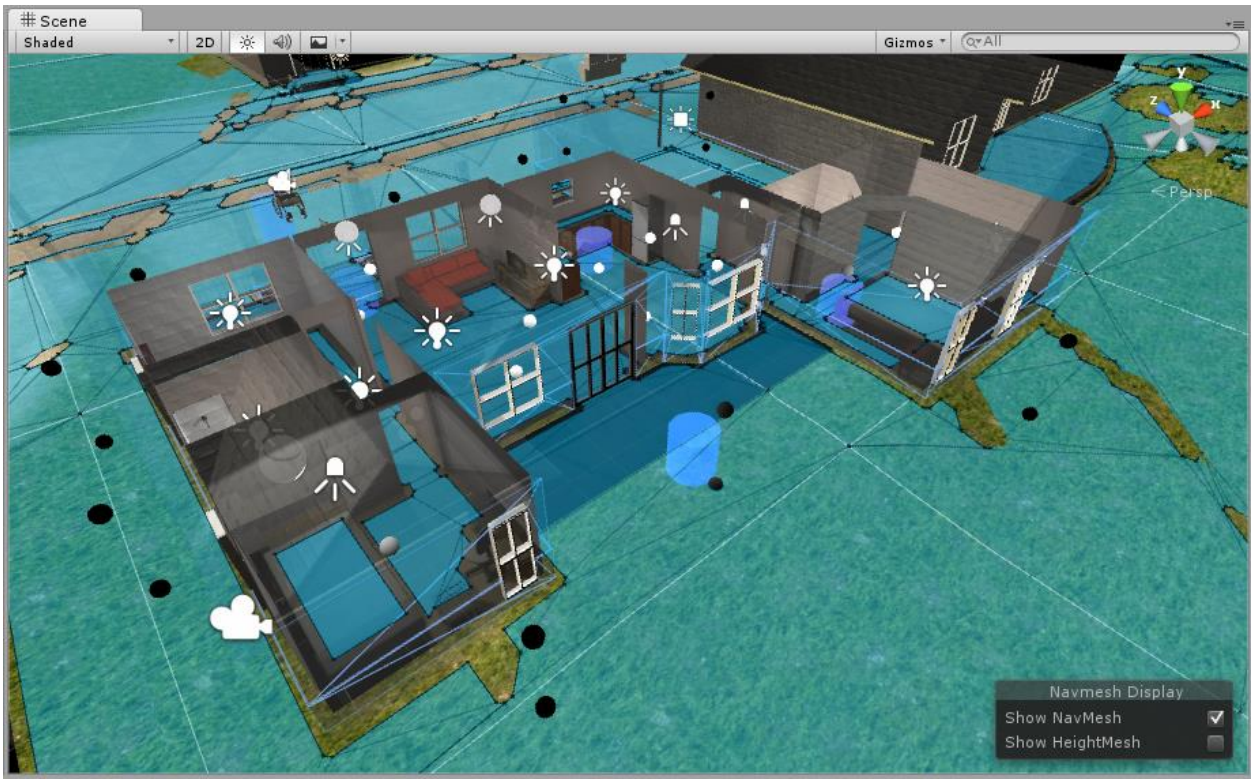


Figure 41: NavMesh in the VRN Home.

The virtual dog is able to freely move about on the blue NavMesh, and can navigate to any position in it. In particular, the dog navigates between the positions of the blue cylinders, which have `GotoTask` components.

In order to add realism to the dog, an `Animator` component is applied that contains animations for walking, lying down, and standing up. The `DogeAnimation` script component

monitors the status of the `NavMeshAgent` component and triggers the different dog animations when appropriate. Since both the `NavMeshAgent` and the `Animator` components are manipulating the dog geometry, it is possible that the manipulations may happen out of synchronization with each other, for example: the `NavMeshAgent` may begin moving the dog before the 'standing up' animation has played to completion. To avoid this, in this architecture, the `VRNLeader` starts and stops the `NavMeshAgent` by means of the `StartMoving()` and `StopMoving()` functions in the `DogeAnimation` component, rather than by directly adjusting the `NavMeshAgent`'s navigation speed.

4.3 Summary

This chapter described the design and implementation of the software used in the VRN Building and VRN Home applications. It began by giving a brief overview of the Unity game engine, and how it simplifies the development of interactive 3D software by a combination of the Unity Editor (for designing and populating scenes) and the script component for adding custom, programmed behaviour. Next, the different components of the VRN Building and VRN Home were discussed in detail, beginning with the software representation of the VRN Chair and VRN Environment packages. These discussions included detailed UML diagrams to illustrate how the various C# classes interact with one another.

Chapter V: Discussion

This thesis describes a novel work in that we are among the first to apply the latest consumer-friendly, immersive HMDs in human navigation studies. Due to the rapid pace at which consumer-friendly VR technology is progressing, it is difficult for researchers to keep their experiments up-to-date. Since we are on the crest of the VR wave, the work presented here is exploratory, but the results promise exciting future developments. Our VR system brings many benefits as compared with other designs, and addresses some limitations inherent to ambulatory VR systems. The revisions to the VRN Building assessment, especially the implementation of an HMD interface, have improved its ability to accurately assess spatial navigation by removing barriers that were present in previous versions of the VRN Building assessment. The case studies described in Chapter III show that VRN exercises might be used as an effective treatment for AD patients and/or individuals with dementia in general, in particular at early stages of the disorder.

5.1 VR Interaction

Our VRN Chair/HMD platform removes the typical experience bias exhibited by elderly people inexperienced with using computer systems, and allows us to study spatial navigation more effectively than researchers without this platform. Our walking (or more accurately, 'shuffling') paradigm allows people to physically move about, and therefore substantially reduces simulator sickness, especially as compared to stationary VR systems [22], [31], [68], [69]. Indeed, the vestibular stimulation that our system provides is a key differentiator amongst other work, which is important as this additional stimulation may be part of path integration and environment encoding at a low level [70]. Immersive VR has been shown to engage patients on a deeper level than existing cognitive therapies such as Lumosity brain games [75], but until the VRN Chair, VR games were outside the reach of people with cognitive impairment who lacked the manual

dexterity to operate traditional game controllers. The VRN Chair breaks down those barriers and makes immersive VR treatments accessible even to people who cannot use standard controllers.

One well-known limitation of ambulatory VR systems is the issue of projecting large virtual environments into smaller real-world environments. At the time of designing our solution, no commercial games existed that attempted to solve this problem, although the academic world has expressed significant interest in a solution. A popular technique is Redirected Walking, where the VR environment translates slightly differently from the user's captured motion. As an example, it can allow a virtual environment to present the illusion that the user is walking forward, when the user is actually walking in a circle. In this way, large virtual environments can be cast into smaller capture spaces [17], [76, Ch. 14]. Unfortunately, these techniques were not really suitable for our application, as we wished to remove possible sources of disorientation and confusion from our participants. Therefore, our 1:1 tracking combined with the decoupled mode discussed in Chapter II is a much better fit for this type of requirement. In the time since we designed the decoupled mode strategy, other commercial game developers have been working on this problem too. In many cases, they simply design virtual environments that are physically smaller than the intended real-world tracking space, although some producers in the gaming industry are instead gravitating to a 'teleportation'- based system where users select points in the virtual world to teleport themselves to [77], [78], rather than having the user walk in real space and then be re-directed when they get too close to a boundary (as in our decoupled system). This 'teleportation' paradigm is space efficient, but comes with a steep learning curve, since it requires mastery of the controllers and is definitely disorienting for inexperienced users. This makes it unsuitable for our target demographic of elderly participants or even participants with cognitive impairments. We briefly explored the impact of our decoupled mode repositioning technique with the ESA participant

during his later trials by varying the number of times that it was used. We did not observe that he made more navigation errors in the VRN Building with more frequent decoupling, but this may be in part due to the fact that he had already encoded the environment and was very familiar with it.

As we have shown in this work, our navigation exercises can have cognitive benefits, provided that the level of challenge is scaled to the participant's ability. In the case of the LSA participant, we learned that this meant providing him with constant verbal reminders of which room he should be going to when trying to reproduce the dog's path. In the case of ESA participant, it involved training him to navigate in a reduced subset of the VRN Building, allowing him to visit each window twice to cement his understanding. Future work should note this, and practical treatments should allow some scaling of difficulty for real patients, whose abilities may vary even within the same AD classification. Possible examples of scaling difficulty in the case of the VRN Home are suggested in Table 10.

Table 10: Examples of how to scale difficulty in the VRN Home

INCREASE DIFFICULTY	REDUCE DIFFICULTY
Removing furniture from rooms	Verbal prompts
Reducing the number of training trials	Show path plotted on map before starting trial
Increasing the number of waypoints	Reducing the number of waypoints

5.2 Case Studies

In our two treatment case studies, we observed a distinct performance difference between the ESA and LSA participants: the ESA participant was able to learn and encode the VRN Building environment, but the LSA participant was unable to encode the VRN Home environment, even with the help of the dog guide. This suggests there might be an effect of diminishing returns on VR training as AD develops; as AD progresses, it becomes more difficult to develop backup neural pathways since the neural cells are being destroyed.

One of the issues with any interactive cognitive therapy is the possibility of confounding factors, such as the placebo effect. Since the treatments described in Chapter III were delivered in one-on-one scenarios, it is possible that the outing and the interactions with the investigators that are inherent to these types of studies may have benefitted and stimulated the participant. These benefits may have been in addition to any effects that the actual treatment might have had, or they may even have overshadowed treatment benefits. This "outing bias" means that any treatment could be beneficial, provided it was delivered in a one-on-one context outside the participant's home. Future experiments can account for this by designing a blind study with a sham treatment.

Alternatively, a method for carrying out an efficacy study would likely be to compare VRN treatment with an alternative cognitive therapy that places subjects in one-on-one contact with a therapist, such as music therapy or physical therapy. This would control for any benefits caused by interacting with the investigator, and allow us to assess the relative merits of our intervention, while still providing treatment for all patients. This comparison strategy was used by Talassi et al. [72]. The authors treated 54 participants (30 MCI, 24 mild dementia) using the Tonetta TNP software (a software package that exercises memory, attention, spatial cognition, etc.), and observed minor improvements in some cognitive functions in the MCI and dementia groups. They also gave a control group occupational/physical therapy, and observed no corresponding improvement in cognitive functions.

5.3 Commercialization

Past experiments have taught us a great deal about human spatial navigation and how it can be stimulated by virtual environments, but the ongoing improvement and miniaturization of VR hardware means that experiments can be conducted that were not possible 20 or even 10 years

ago. fMRI compatible HMDs could hopefully be developed in the future that might allow investigators to capture brain signals from participants in truly immersive environments, instead of relying on conventional displays outside the field of the MRI scanner [79], [80]. This would remove any bias that may be presented by requiring participants to mentally project themselves into the 3D environment shown on the 2D display. Indeed, other brain monitoring techniques such as functional Near-Infrared Spectroscopy (fNIRS) are compatible with current HMDs, and can be used to study brain activity while actively engaged in a virtual environment [81]. While the literature [36] shows that there is a strong correspondence between real-world and virtual-world navigation (even while using a 2D display), this does not necessarily apply to AD patients, which is a large reason that we designed the VRN Chair. In this work we have shown that there is potential benefit in using VRN exercises to rehabilitate spatial navigation in AD patients, but in the future a larger trial ought to be carried out, perhaps in a residence. Depending on the success of that trials, the VRN Chair device could be commercialized and sold to institutions wishing to establish VRN treatment programs.

We believe that our diagnostic tools have commercial applications primarily in care centres, and in hospitals/clinics. We perform our VR experiments in empty banquet halls and common spaces, since the wheelchair works best in a large open area. Since hospitals and day-centres for the elderly usually have these types of spaces, they are appropriate places in which to operate, and VR-based rehabilitation exercises could be integrated into a hospital/day-centre's regular programming. Session data can be used to track patient progress over time, as well as study human spatial navigation.

Chapter VI: Conclusion

The chief contributions of the work presented in this thesis were to update the VRN Building experiment by improving its visual quality and immersion, and making it easier to extend for alternative VR applications. I also designed and tested a preliminary VR-based treatment regime with support from colleagues in our lab. The contributions of this work are summarized in Table 11.

Table 11: Contributions of this work

UPDATE VRN BUILDING TO BE FULLY IMMERSIVE	Switch to Unity 3D engine for better extensibility in the future
	Correct geometry to be correct scale for accurate sense of size
	Add Inertial Measurement Unit sensor to correct drift in VRN Chair
	Enforce seated to posture to reduce simulator sickness
TEST VRN BUILDING FOR EFFICACY AT IMPROVING IMMERSION	Compare pathfinding in healthy young participants in HMD and conventional display cases [62]
	Compare performance of people in first-generation VRN Building with second-generation VRN Building
BUILD VRN TASK ENVIRONMENT	Design VRN Home environment to test task completion framework
	Re-use assets and framework created while updating VRN Building
VRN REHAB CASE STUDY	Case study with ESA/LSA patients
	Pilot test by practicing VRN Building, and VRN Home

Evaluating the changes made to the VRN Building, we found that it improved people's immersion in the virtual environment, and therefore made it a more effective measure of their spatial navigation ability. We also tested the efficacy of the VRN-based treatment regimes in two case studies, and found that VRN-treatment may be beneficial to patients early in the progression of AD.

References

[1] A. A. Rizzo, J. G. Buckwalter, and U. Neumann, "Virtual reality and cognitive rehabilitation: a brief review of the future," *J. Head Trauma Rehabil.*, vol. 12, no. 6, pp. 1–15, 1997.

[2] R. I. García-Betances, M. T. Arredondo Waldmeyer, G. Fico, and M. F. Cabrera-Umpiérrez, "A Succinct Overview of Virtual Reality Technology Use in Alzheimer's Disease," *Front. Aging Neurosci.*, vol. 7, no. May, 2015.

[3] A. Byagowi, D. Mohaddes, and Z. Moussavi, "Design and Application of a Novel Virtual Reality Navigational Technology (VRNChair)," *J. Exp. Neurosci.*, vol. 8, pp. 7–14, 2014.

[4] Oculus VR, "Oculus Best Practices," 2016.

[5] A. Pomés and M. Slater, "Drift and ownership toward a distant virtual body.," *Front. Hum. Neurosci.*, vol. 7, no. December, p. 908, 2013.

[6] L. Heydrich, T. J. Dodds, J. E. Aspell, B. Herbelin, H. H. Bülthoff, B. J. Mohler, and O. Blanke, "Visual capture and the experience of having two bodies - Evidence from two different virtual reality techniques.," *Front. Psychol.*, vol. 4, no. December, p. 946, 2013.

[7] Xsens, "Xsens MVN - Products - Xsens 3D Motion Tracking," 2015. [Online]. Available: https://www.xsens.com/products/xsens-mvn/. [Accessed: 14-Oct-2015].

[8] D. Pape, "File:CAVE Crayoland.jpg," 2006. [Online]. Available: https://en.wikipedia.org/wiki/File:CAVE_Crayoland.jpg.

[9] P. James, "The Oculus Rift DK2, In-Depth Review and DK1 Comparison," *Road to VR*, 2014. [Online]. Available: http://www.roadtovr.com/oculus-rift-dk2-review-dk1-comparison-vr-headset/. [Accessed: 14-Oct-2015].

[10] M. K. Young, G. B. Gaylor, S. M. Andrus, and B. Bodenheimer, "A comparison of two cost-differentiated virtual reality systems for perception and action tasks," in *Proceedings of the ACM Symposium on Applied Perception*, 2014, pp. 83–90.

[11] HTC Corporation, *Vive PRE User Guide*. Valve Corporation, 2016.

[12] O. Kreylos, "Hacking the Oculus Rift DK2, part III," *Doc-Ok.org: A developer's perspective on immersive 3D computer graphics*, 2014. [Online]. Available: http://doc-ok.org/?p=1138. [Accessed: 23-Apr-2016].

[13] D. Sachs, "Sensor Fusion on Android Devices: A Revolution in Motion Processing." Google Tech Talks, 2010.

[14] D. M. (U. S. A. R. I. Johnson, "Introduction to and Review of Simulator Sickness Research," *Soc. Sci.*, no. April, 2005.

[15] R. S. (Essex C. Kennedy, N. E. (Essex C. Lane, K. S. (The U. of I. Berbaum, and M. G. (United S. N. Lilienthal, "Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness," *Int. J. Aviat. Psychol.*, vol. 3, no. 3, pp. 203–220, 1993.

[16] S. E. Kober, G. Wood, D. Hofer, W. Kreuzig, M. Kiefer, and C. Neuper, "Virtual reality in neurologic rehabilitation of spatial disorientation.," *J. Neuroeng. Rehabil.*, vol. 10, no. 1999, p. 17, 2013.

[17] G. Bruder, V. Interrante, L. Phillips, and F. Steinicke, "Redirecting walking and driving for natural

navigation in immersive virtual environments," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 4, pp. 538–545, 2012.

[18] T. I. Kesztyues, M. Mehlitz, E. Schilken, G. Weniger, S. Wolf, U. Piccolo, E. Irle, and O. Rienhoff, "Preclinical Evaluation of a Virtual Reality Neuropsychological Test System: Occurrence of Side Effects.," *CyberPsychology Behav.*, vol. 3, no. 3, pp. 343–349, 2000.

[19] Y. J. Kang, J. Ku, K. Han, S. I. Kim, T. W. Yu, J. H. Lee, and C.-I. Park, "Development and Clinical Trial of Virtual Reality-Based Cognitive Assessment in People with Stroke: Preliminary Study," *CyberPsychology Behav.*, vol. 11, no. 3, pp. 329–339, 2008.

[20] M. J. Cevette, J. Stepanek, D. Cocco, A. M. Galea, G. N. Pradhan, L. S. Wagner, S. R. Oakley, B. E. Smith, D. a. Zapala, and K. H. Brookler, "Oculo-Vestibular Recoupling Using Galvanic Vestibular Stimulation to Mitigate Simulator Sickness," *Aviat. Space. Environ. Med.*, vol. 83, no. 6, pp. 549–555, 2012.

[21] M. J. Cevette, G. N. Pradhan, D. Cocco, M. D. Crowell, A. M. Galea, J. Bartlett, and J. Stepanek, "Electrogastrographic and autonomic responses during oculovestibular recoupling in flight simulation," *Aviat. Sp. Environ. Med.*, vol. 85, no. 1, pp. 15–24, 2014.

[22] K. K. Zakzanis, G. Quintin, S. J. Graham, and R. Mraz, "Age and dementia related differences in spatial navigation within an immersive virtual environment.," *Med. Sci. Monit.*, vol. 15, no. 4, p. CR140-R150, 2009.

[23] A. J. Benson, "Motion Sickness," in *Ernsting's Aviation Medicine*, 4th ed., D. J. Rainford and D. P. Gradwell, Eds. London: Edward Arnold, 2006, pp. 459–475.

[24] P. L. Palatty, R. Haniadka, B. Valder, R. Arora, and M. S. Baligaa, "Ginger in the prevention of nausea and vomiting: a review," *Crit. Rev. Food Sci. Nutr.*, vol. 53, no. 7, pp. 659–69, 2012.

[25] Oculus VR, "Oculus Best Practices," 2016.

[26] R. G. M. Morris, "Spatial localization does not require the presence of local cues," *Learn. Motiv.*, vol. 12, no. 2, pp. 239–260, 1981.

[27] M. K. Rodgers, J. a. Sindone, and S. D. Moffat, "Effects of Age on Navigation Strategy," *Neurobiol. Aging*, vol. 33, no. 1, p. 202.e15-202.e22, 2012.

[28] S. A. Livingstone and R. W. Skelton, "Virtual environment navigation tasks and the assessment of cognitive deficits in individuals with brain injury," *Behav. Brain Res.*, vol. 185, no. 1, pp. 21–31, 2007.

[29] E. Kalová, K. Vlček, E. Jarolímová, and J. Bureš, "Allothetic orientation and sequential ordering of places is impaired in early stages of Alzheimer's disease: Corresponding results in real space tests and computer tests," *Behav. Brain Res.*, vol. 159, no. 2, pp. 175–186, 2005.

[30] T. Fox, "New Virtual Reality Navigation System to Help Diagnose Cognitive Defects," *UC San Diego News Center*, 2014. [Online]. Available: http://ucsdnews.ucsd.edu/pressrelease/new_virtual_reality_navigation_system_to_diagnose_monitor_and_train_patient.

[31] P. Werner, S. Rabinowitz, E. Klinger, A. D. Korczyn, and N. Josman, "Use of the virtual action planning supermarket for the diagnosis of mild cognitive impairment," *Dement. Geriatr. Cogn. Disord.*, vol. 27, no. 4, pp. 301–309, 2009.

[32] J. Kim, K. Kim, D. Y. Kim, W. H. Chang, C.-I. Park, S. H. Ohn, K. Han, J. Ku, S. W. Nam, I. Y.

Kim, and S. I. Kim, “Virtual Environment Training System for Rehabilitation of Stroke Patients with Unilateral Neglect: Crossing the Virtual Street,” *CyberPsychology Behav.*, vol. 10, no. 1, pp. 7–15, 2007.

[33] E. Klinger, I. Chemin, S. Lebreton, and R. M. Marie, “A virtual supermarket to assess cognitive planning.,” *Annu. Rev. CyberTherapy Telemed.*, vol. 2, pp. 49–57, 2004.

[34] N. Josman, R. Kizony, E. Hof, K. Goldenberg, P. L. Weiss, and E. Klinger, “Using the virtual action planning-supermarket for evaluating executive functions in people with stroke,” *J. Stroke Cerebrovasc. Dis.*, vol. 23, no. 5, pp. 879–887, 2014.

[35] University Hospital Bordeaux, “Virtual Action Planning in Stroke: a Control Rehabilitation Study - Full Text View - ClinicalTrials.gov,” *ClinicalTrials.gov, A service of the U.S. National Institutes of Health*, 2014. .

[36] L. A. Cushman, K. Stein, and C. J. Duffy, “Detecting navigational deficits in cognitive aging and Alzheimer disease using virtual reality.,” *Neurology*, vol. 71, no. 12, pp. 888–895, 2008.

[37] A. Alzheimer, “Über eigenartige Krankheitsfälle des späteren Alters.,” *Zbl. ges. Neurol. Psych.*, vol. 4, pp. 356–385, 1911.

[38] D. Purves, G. J. Augustine, and D. Fitzpatrick, *Neuroscience*, 3rd ed. Sunderland, Massachusetts: Sinauer Associates, Inc., 2004.

[39] M. Goedert, M. G. Spillantini, and R. a Crowther, “Tau proteins and neurofibrillary degeneration.,” *Brain Pathol.*, vol. 1, pp. 279–286, 1991.

[40] Garrondo, “Alzheimer’s disease brain comparison,” *Wikimedia Commons*, 2008. [Online]. Available: https://commons.wikimedia.org/wiki/File:Alzheimer%27s_disease_brain_comparison.jpg. [Accessed: 01-Aug-2016].

[41] K. B. D. & A. P. P. B. M. McGleenon, “Acetylcholinesterase inhibitors in Alzheimer’s disease,” *Br. J. Clin. Pharmacol.*, vol. 48, no. 4, pp. 471–480, 1999.

[42] M. F. Folstein, S. E. Folstein, and P. R. McHugh, “‘Mini-mental state’. A practical method for grading the cognitive state of patients for the clinician,” *J. Psychiatr. Res.*, vol. 12, no. 3, pp. 189–198, 1975.

[43] Z. S. Nasreddine, “MoCA Montreal - Cognitive Assessment.” .

[44] Alzheimer’s Association, “Alzheimer’s and Dementia Testing Advances,” *alz.org*, 2016. [Online]. Available: http://www.alz.org/research/science/earlier_alzheimers_diagnosis.asp.

[45] C. P. Hughes, L. Berg, and W. L. Danziger, “A new clinical scale for the staging of dementia,” *Br. J. Psychiatry*, vol. 140, no. 6, pp. 566–572, 1982.

[46] J. C. Morris, “The Clinical Dementia Rating (CDR): Current vision and scoring rules,” *Neurology*, vol. 43, pp. 2412–2414, 1993.

[47] W. J. Burke, J. P. Miller, E. H. Rubin, J. C. Morris, L. a Coben, J. Duchek, I. G. Wittels, and L. Berg, “Reliability of the Washington University Clinical Dementia Rating.,” *Arch. Neurol.*, vol. 45, no. 1, pp. 31–2, 1988.

[48] J. C. Morris, C. Ernesto, K. Schafer, M. Coats, S. Leon, M. Sano, L. J. Thal, and P. Woodbury, “Clinical dementia rating training and reliability in multicenter studies: the Alzheimer’s Disease Cooperative Study experience.,” *Neurology*, vol. 48, no. 6, pp. 1508–1510, 1997.

[49] S. Lessig, D. Nie, R. Xu, and J. Corey-Bloom, "Changes on brief cognitive instruments over time in Parkinson's disease," *Mov. Disord.*, vol. 27, no. 9, pp. 1125–1128, 2012.

[50] R. Katzman, R. Terry, R. DeTeresa, T. Brown, P. Davies, P. Fuld, X. Renbing, and A. Peck, "Clinical, pathological, and neurochemical changes in dementia: a subgroup with preserved mental status and numerous neocortical plaques.," *Ann. Neurol.*, vol. 23, no. 2, pp. 138–144, 1988.

[51] Y. Stern, "Cognitive reserve," *Neuropsychologia*, vol. 47, no. 10, pp. 2015–2028, 2010.

[52] D. A. Snowdon, "Aging and Alzheimer's disease: lessons from the Nun Study.," *Gerontologist*, vol. 37, no. 2, pp. 150–156, 1997.

[53] D. Rothschild, "Pathological changes in senile psychoses and their psychobiologic significance," *Am. J. Psychiatry*, vol. 93, pp. 757–788, 1937.

[54] S. D. Moffat, "Age differences in spatial memory in a virtual environment navigation task," *Neurobiol. Aging*, vol. 22, no. 5, pp. 787–796, 2001.

[55] G. Weniger, M. Ruhleder, C. Lange, S. Wolf, and E. Irle, "Egocentric and allocentric memory as assessed by virtual reality in individuals with amnestic mild cognitive impairment," *Neuropsychologia*, vol. 49, no. 3, pp. 518–527, 2011.

[56] A. Byagowi and Z. Moussavi, "Design of a virtual reality navigational (VRN) experiment for assessment of egocentric spatial cognition," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 4812–4815, 2012.

[57] D. Zen, A. Byagowi, M. Garcia, D. M. Kelly, B. Lithgow, and Z. Moussavi, "The perceived orientation in people with and without Alzheimer's," in *International IEEE/EMBS Conference on Neural Engineering, NER*, 2013, pp. 460–463.

[58] O. R. Pouya, A. Byagowi, D. M. Kelly, and Z. Moussavi, "Introducing a New Age and Cognition-sensitive Measurement for Assessing Spatial Orientation using a Landmark-less Virtual Reality Navigational Task," *Q. J. Exp. Psychol.*, pp. 1–35, 2016.

[59] L. M. Billings, K. N. Green, J. L. McGaugh, and F. M. LaFerla, "Learning Decreases Aß*56 and Tau Pathology and Ameliorates Behavioral Decline in 3xTg-AD Mice," *J. Neurosci.*, vol. 27, no. 4, pp. 751–761, 2007.

[60] R. L. Klatzky, "Allocentric and egocentric spatial representations: Definitions, distinctions, and interconnections," *Spat. Cogn. - An Interdiscip. approach to Represent. Process. Spat. Knowl.*, no. September 1997, pp. 1–17, 1998.

[61] Ahmad Byagowi, S. Singhal, M. Lambeta, C. Aldaba, and and Z. Moussavi, "Design of a Naturalistic Navigational Virtual Reality Using Oculus Rift," *J. Med. Device.*, vol. 8, no. September 2014, pp. 2014–2016, 2014.

[62] P. J. White, A. Byagowi, and Z. Moussavi, "Effect of Viewing Mode on Pathfinding in Immersive Virtual Reality," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, 2015, p. TBA.

[63] Bosch Sensortech, "Intelligent 9-axis absolute orientation sensor," no. 1.4. Robert Bosch GmbH, Reutlingen, 2016.

[64] R. A. Ruddle and S. Lessels, "The benefits of using a walking interface to navigate virtual environments," *ACM Trans. Comput. Interact.*, vol. 16, no. 1, pp. 1–18, 2009.

[65] Z. S. Nasreddine, "MoCA Montreal - Cognitive Assessment." [Online]. Available:

http://www.mocatest.org/. [Accessed: 01-Mar-2016].

[66] T. T. A. Rahman and M. M. El Gaafary, “Montreal cognitive assessment Arabic version: Reliability and validity prevalence of mild cognitive impairment among elderly attending geriatric clubs in Cairo,” *Geriatr. Gerontol. Int.*, vol. 9, no. 1, pp. 54–61, 2009.

[67] M. Tere Garcia-Campuzano, J. Virues-Ortega, S. Smith, and Z. Moussavi, “Effect of cognitive training targeting associative memory in the elderly: A small randomized trial and a longitudinal evaluation,” *J. Am. Geriatr. Soc.*, vol. 61, no. 12, pp. 2252–2254, 2013.

[68] J. O. Brooks, R. R. Goodenough, M. C. Crisler, N. D. Klein, R. L. Alley, B. L. Koon, W. C. Logan, J. H. Ogle, R. a. Tyrrell, and R. F. Wills, “Simulator sickness during driving simulation studies,” *Accid. Anal. Prev.*, vol. 42, no. 3, pp. 788–796, 2010.

[69] J.-Y. Lee, S. Kho, H. Bin Yoo, S. Park, J.-S. Choi, J. S. Kwon, K. R. Cha, and H.-Y. Jung, “Spatial memory impairments in amnestic mild cognitive impairment in a virtual radial arm maze.,” *Neuropsychiatr. Dis. Treat.*, vol. 10, pp. 653–60, Jan. 2014.

[70] R. S. Astur, L. B. Taylor, A. N. Mamelak, L. Philpott, and R. J. Sutherland, “Humans with hippocampus damage display severe spatial memory impairments in a virtual Morris water task.,” *Behav. Brain Res.*, vol. 132, no. 1, pp. 77–84, 2002.

[71] G. Burdea, K. Polistico, A. Krishnamoorthy, G. House, D. Rethage, J. Hundal, F. Damiani, and S. Pollack, “Feasibility study of the BrightBrainer™ integrative cognitive rehabilitation system for elderly with dementia.,” *Disabil. Rehabil. Assist. Technol.*, vol. 10, no. 5, pp. 1–12, 2014.

[72] E. Talassi, M. Guerreschi, M. Feriani, V. Fedi, A. Bianchetti, and M. Trabucchi, “Effectiveness of a Cognitive Rehabilitation Program in Mild Dementia (MD) and Mild Cognitive Impairment (MCI): A Case Control Study,” *Arch. Gerontol. Geriatr. Suppl.*, vol. 1, pp. 391–399, 2007.

[73] Unity Technologies, “Car Tutorial.” Unity Asset Store, 2015.

[74] S. B. Niku, *Introduction to Robotics: Analysis, Control, Applications*, 2nd ed. Hoboken, NJ: John Wiley and Sons, Inc., 2011.

[75] V. J. Shute, M. Ventura, and F. Ke, “The power of play: The effects of Portal 2 and Lumosity on cognitive and noncognitive skills,” *Comput. Educ.*, vol. 80, pp. 58–67, 2015.

[76] F. Steinicke, Y. Visell, J. Campos, and A. Lecuyer, *Human Walking in Virtual Environments*. Springer Science+Busines Media, LLC, 2013.

[77] Valve Corporation, “The Lab.” Valve Corporation, Bellevue, Washington, 2016.

[78] Indimo Labs, “Vanishing Realms.” Indimo Labs LLC, 2016.

[79] C. Mueller, M. Luehrs, S. Baecke, D. Adolf, R. Luetzkendorf, M. Luchtmann, and J. Bernarding, “Building virtual reality fMRI paradigms: A framework for presenting immersive virtual environments,” *J. Neurosci. Methods*, vol. 209, no. 2, pp. 290–298, 2012.

[80] E. M. Migo, O. O’Daly, M. Mitterschiffthaler, E. Antonova, G. R. Dawson, C. T. Dourish, K. J. Craig, a. Simmons, G. K. Wilcock, E. McCulloch, S. H. D. Jackson, M. D. Kopelman, S. C. R. Williams, and R. G. Morris, “Investigating virtual reality navigation in amnestic mild cognitive impairment using fMRI,” *Aging, Neuropsychol. Cogn.*, vol. 5585, no. September, pp. 1–22, 2015.

[81] Y. Rakhshani Fatmehsari and Z. Moussavi, “Designing and Implementing a Portable Near-Infrared Imaging System for Monitoring of Human’s Functional Brain Activity,” *J. Med. Device.*, vol. 9, no. June 2015, pp. 2015–2016, 2015.

[82] J.-F. Baril, V. E. Beynon, S. M. Koop, S. Rohringer, and P. J. White, "Design and Implementation of a Framework for Remote Automated Environmental Monitoring and Control," University of Manitoba, 2014.

[83] P. J. White, A. Byagowi, and Z. Moussavi, "Design of a Wireless Biological Signal Conditioning System," *J. Med. Device.*, vol. 7, no. 2, pp. 2–4, 2013.

[84] P. J. White, B. W. Podaima, and M. R. Friesen, "Algorithms for smartphone and tablet image analysis for healthcare applications," *IEEE Access*, vol. 2, pp. 831–840, 2014.

[85] P. J. White and Z. Moussavi, "Neurocognitive Treatment for a Patient with Alzheimer's Disease Using a Virtual Reality Navigational Environment," *J. Exp. Neurosci.*, vol. 10, p. 129, 2016.

Appendix A

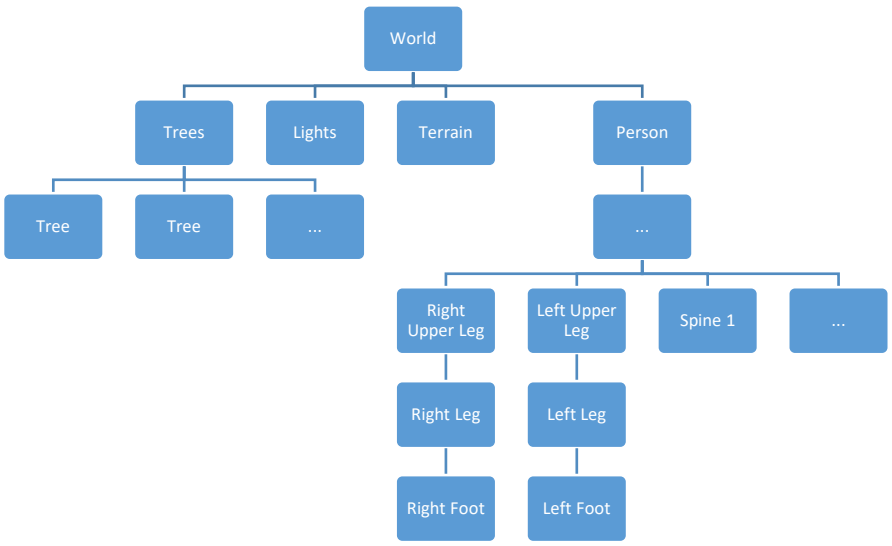
<See attached documentation>

Appendix B: Unity Overview

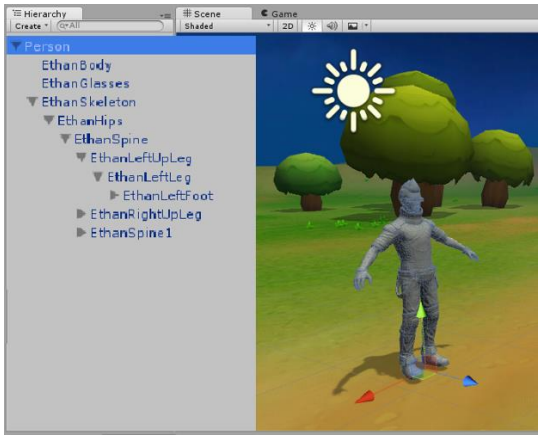
The purpose of this appendix is to provide the reader with a quick introduction to the Unity game engine, and how to use it to program 3D VR environments.

Unity projects are organized into 'scenes', where each scene models a different game level. This organizational strategy lets game developers position entities (i.e. *things* in the scene) in their virtual environments quickly, easily, and intuitively. These entities include the lighting, materials, objects, etc. The editor also lets game developers configure other details that apply to the game in general including the target platform (such as mobile, desktop, or even a game console) and visual quality settings. In addition to positioning and configuring entities in the Unity editor, game developers often need to write software code to define their game's behaviour. This can most easily be done using either the Unity-supplied MonoDevelop or Microsoft Visual Studio.

The basic entity in Unity is modeled by a software construct called a `GameObject`, which allows an entity to be controlled by the game developer's code. The `GameObject` provides each entity with a unique identifier and a transform (which specifies position and rotation information). The `GameObject` class also allows each `GameObject` to get references to other `GameObject`s. In Unity, entities in a given scene exist in a hierarchy. One of the key consequences of modeling entities in a hierarchy like this is that the transforms on each of the entities cascade to the entity's sub-objects, known as 'children'. This means that each level of the hierarchy actually represents a different relative coordinate system, as shown in Figure 42. Applying a transform to a particular entity (e.g. `LeftUpperLeg`) will change the rotation and position of the `LeftUpperLeg` entity and its children, `LeftLeg` and `LeftFoot`, but the `LeftLeg` and `LeftFoot` transforms within *their* respective coordinate spaces will not have changed (i.e. `LeftFoot`'s rotation and position do not change relative to `LeftLeg`).



A



B C

Figure 42: Sample hierarchy of a simple virtual environment.

A) the logical layout of the environment. B) a rendering of this hierarchy in the Unity editor. Since the various joints of the person are all children of the Person `GameObject`, any changes applied to the Person `GameObject`, such as repositioning, rotation and scaling, will propagate downward. In C), the LeftUpLeg and its children have been rescaled and rotated.

Unity's design philosophy is based very strongly on the 'Decorator' design pattern. `GameObjects` may be 'decorated' with one or more 'components' (i.e. components can be added onto each basic `GameObject` to add additional functionality). Figure 43 illustrates some components of a basic `GameObject`, as seen in the Unity editor.

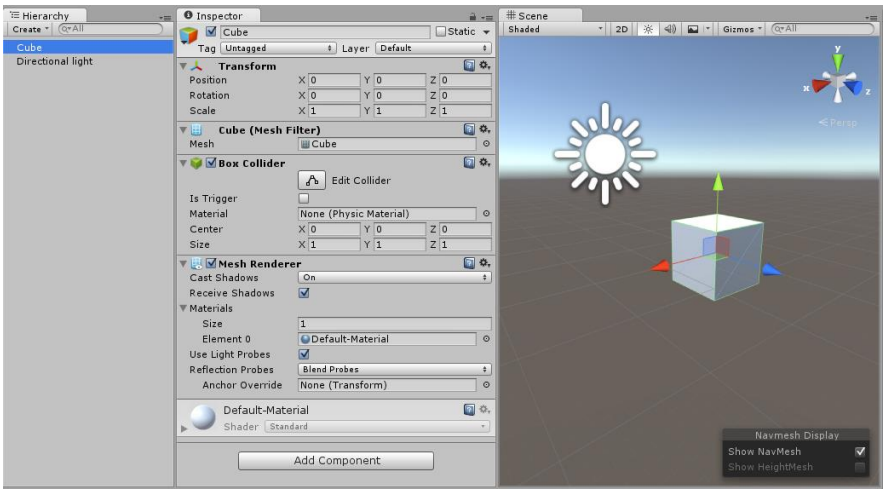


Figure 43: Unity Editor illustrating a simple entity in a virtual environment.

This figure illustrates `Transform`, `MeshFilter`, `Collider`, and `MeshRenderer` components, all 'decorating' a `GameObject` called 'Cube'. This scene also contains a `DirectionalLight GameObject`, which illuminates the scene.

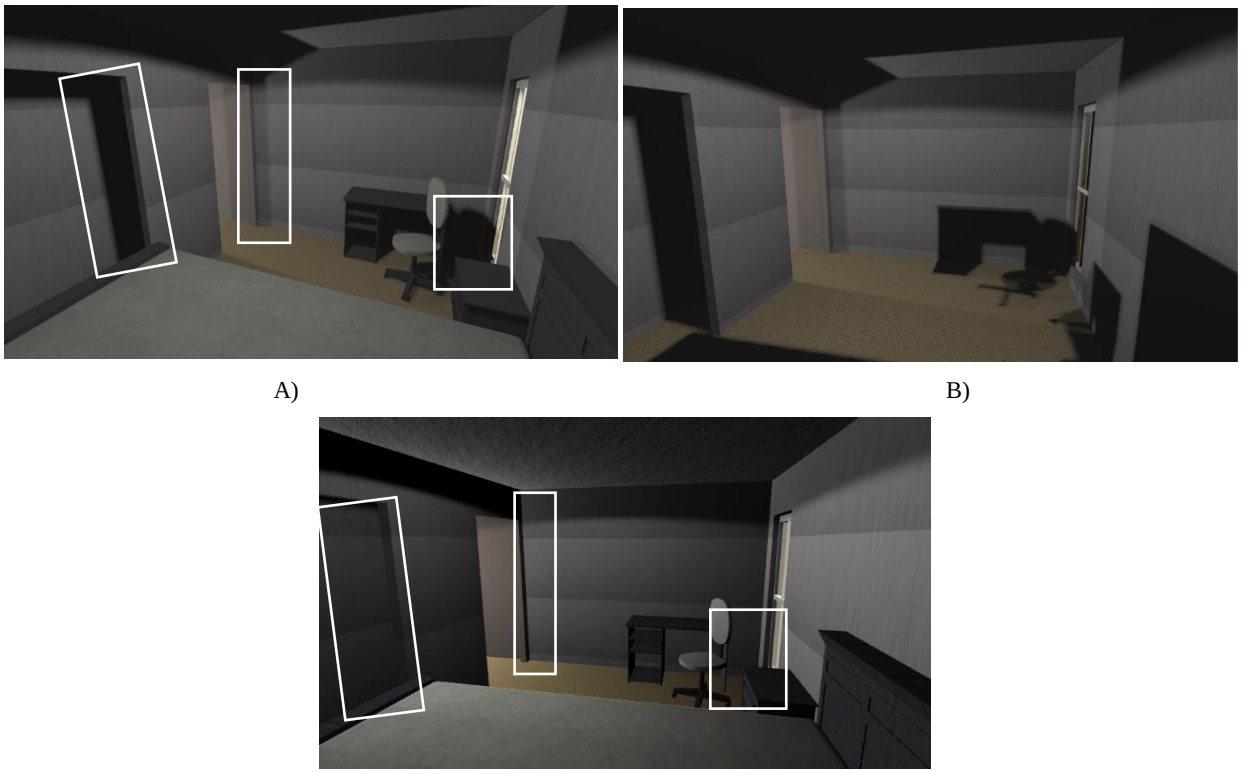
- The `Transform` specifies the position, rotation, and any scaling effects on the cube. These effects propagate to any children in the hierarchy (in this case there are none).
- The `MeshFilter` specifies a 3D mesh to draw for this `GameObject`. These meshes are either produced by a 3D artist, or generated from data (e.g. a point cloud).
- The `MeshRenderer` (with materials) specifies rules for how the `MeshFilter` should be drawn. This includes lighting effects such as reflection and shadows, but also materials (or textures), which cover the geometry like wrapping paper.
- The `Collider` component is used for physics calculations. Two objects with `Colliders` cannot intersect with one another; instead the game engine will adjust their inherent motion parameters based upon the laws of Newtonian physics. Each moving `GameObject` must also include a `RigidBody` component, although static `GameObjects` (such as the walls of a building, or the terrain) do not.

Unity supports other interesting component types, including the `Animator`, `AudioSource`, and several different `LightSource`s, but these are advanced components that are not critical to understanding the fundamentals of how Unity works.

One of the most critical component types is the **script**. Scripts are custom pieces of code that get executed over the course of the game's execution, and allow a game programmer to add custom features that are not provided by any of the other components. Such features may include network connectivity, (for multi-user applications) interaction with input devices, (such as gamepads or joysticks) interacting with other components, (such as triggering animations or adjusting the position/rotation) and all sorts of other interactions and functionality. Script components may be written in either JavaScript and C#; however, C# is widely preferred. The core of a script is the `Update()` function, which is executed each frame as part of the 'run-loop'. By minimizing the time taken to execute an iteration of the run-loop, the framerate (and therefore, performance) can be maximized.

In order for a VR experience to be comfortable for the user, it is absolutely critical that a stable, high framerate is maintained. In the case of the Rift DK2, this framerate is 75 frames/second. If it cannot be maintained, the simulation will appear choppy and unresponsive to the user, and can cause disorientation or even simulator sickness. Framerates can be kept high by using high-performance graphics processors, and by optimizing the software to run quickly. When developing 3D simulations with Unity, in addition to ensuring the code in the `Update()` function executes quickly, developers can optimize their simulations through a combination of selecting simpler visual assets that require fewer computational resources to display, and using Unity's optimization tools. One of the most effective optimizations used in this work is the *static light map*. Static light maps allow shadow information to be calculated in advance, and embedded

directly into the textures of an environment. This improves the efficiency of the virtual environment as the shadow information does not need to be computed for every frame. The effects of static light maps are illustrated in Figure 44.



A) B)

C)

Figure 44: Static light maps in Unity.

A) a rendered room with light maps. Shadows are marked with boxes. B) shows the same room with the furniture removed, but without the light maps recalculated. Note how the light maps are embedded into the textures of the floor and walls. C) The room from A) with all light maps disabled. Note that walls are still illuminated with varying intensities based upon their proximity and rotation relative to the light sources, (in particular in the hallway) but the objects to do not cast shadows.