A Virtual Reality Input API for Medical Applications

Theo Turner[1]

MEng Mathematical Computation

Dr. Dean Mohamedally

Submission date: 27th April 2018

[1]**Disclaimer:** This report is submitted as part requirement for the MEng in Mathematical Computation at UCL. It is substantially the result of my own work except where explicitly indicated in the text. The report may be freely copied and distributed provided the source is explicitly acknowledged.

Abstract

The use of virtual reality, physical activity and game design elements for physiotherapy has repeatedly been shown to improve outcomes for certain rehabilitative treatments. The heterogeneity of past work obstructs large-scale research. This project aims to develop an open, standardised platform for creating VR physiotherapy games, solving this problem.

The project explored the successes and failures of VR game-based approaches to rehabilitation and used these findings to develop an API for this purpose. The platform integrates the emerging FHIR medical data standard, allowing future studies to be readily graded and compared. A demo game serving a useful physiotherapy function has also been built.

The project successfully achieved its intended goal, producing a full-featured platform for developing VR physiotherapy games.

Contents

Chapter 1

Introduction

This chapter outlines the motivations behind undertaking this project, details key objectives and provides an overview of the structure of this report.

1.1 Motivation

Over the course of the past few decades, many aspects of medicine have become increasingly reliant on technology. Rapid advancements in computing power and device applications have driven the integration of technology with healthcare, aided by falling hardware costs and increasing development of computer-aided medical tools. Surgery in particular has been revolutionised by the introduction of optical systems, opening the field of minimally invasive surgery.

A more recent development at the intersection of medicine and technology is the introduction of Virtual Reality (VR) systems to the tool set of medical professionals. VR allows for intuitive interaction with computer-generated environments, which can be used in applications such as simulation-based medical training and pre-surgery rehearsal [1]. The utility of VR systems is increasingly appreciated by the medical community despite a slow rate of adoption.

A still-nascent aspect of this field is the use of VR in enhancing patient experiences. One of the most widespread preventable problems in healthcare is low adherence to physiotherapy treatments following surgery or injury, the primary causes being lack of motivation, boredom and fatigue [2, 3, 4]. VR systems, when used as a medium for physiotherapy, have been shown to improve user enjoyment, self-efficacy and adherence to regimented treatment programs, especially when combined with physical activity [5, 6, 7].

In addition to its potential to motivate, VR has been demonstrated to improve treatment outcomes. The use of VR for occupational therapy has a significant positive effect on rehabilitative treatment results [8, 9]. When used in conjunction with regimented exercise programs, VR may further improve patient outcomes [10, 11]. Most notably, a Cochrane

review of 19 studies in the field featuring more than 500 participants found physically active VR approaches to be significantly more effective than conventional therapy for some rehabilitative treatments [12]. In the context of physiotherapy, the introduction of game design elements may further enhance treatment results [13, 14, 15, 16, 17, 18, 19].

Despite the possibilities presented, little more than trials have been performed for the use of VR, exercise and game design in patient rehabilitation. The limiting factor obstructing large-scale studies of this type is the heterogeneity, or lack of consistency, in the treatments used [15]. Evident from the literature, however, is the improvement in patient outcomes following the use of these three elements in treatments.

Clearly, there is an untapped opportunity to introduce a standardised, open platform for physically active game-based VR physiotherapy in this field. In addition to facilitating further research, such a platform has the potential to ease the transition of innovative, useful technology into healthcare, and may have a profound impact on the effectiveness of a range of rehabilitation types. The project challenge is the construction of this platform.

1.2 Demands and Objectives

As has been made evident by the diversity of the trials performed in this field thus far, there is a wide range of different physiotherapy techniques involving VR and physical activity. A medically useful platform has to be able accommodate a large number of medical functions, many of which may be specific to small groups of people or even individuals [12, 20]. Clearly, the physiotherapy tasks themselves cannot be hard-coded. Instead, the platform should facilitate the creation of these tasks.

The problem of heterogeneity between past studies should also be prevented. Patient metrics should be standardised so that they can be readily documented, graded and compared. In doing so, future studies of VR-based physiotherapy become normalised and consistent medical records are generated for patients.

This projects aims to meets these two key demands in the form of an Application Programming Interface (API). Built for VR hardware and providing real-time information about patients, the API will grant easy access to physiological metrics, standardising and outputting them in a medically useful format. Ultimately, it will facilitate the creation of physiotherapy games by developers.

1.2.1 Research Objectives

- To identify the key characteristics of useful VR physiotherapy software.
- To review relevant VR and exercise technologies, as well as survey existing systems employing the two.

- To identify the key challenges involved in developing these systems and to formulate a set of considerations for their use.
- To catalogue the requirements of a tool for creating VR physiotherapy software.
- To identify considerations for software built using the API.
- To identify the most appropriate programming language(s) and software tools for the development of the API and example software, and to become proficient in their use.

1.2.2 Technical Objectives

- To develop an interface for the creation of VR physiotherapy games, giving access to necessary patient metrics and standardised outputs in a medically useful format.
- To write documentation detailing full use of the API.
- To test use of the API among developers, collect feedback and iteratively update the software according to this data.
- To develop an example game built using the API that serves a useful physiotherapy function involving both VR and physical activity.
- To test use of the game among potential end-users, collect feedback and iteratively update it according to this data.

1.2.3 Development Methodology

Given the objectives of the project outlined above, the prototyping software development model was used for the completion of this project. The prototyping model has four primary phases: identifying requirements, developing a prototype, reviewing through testing and revising the software. Prototyping is especially relevant for this undertaking due to the project's two-part nature (API and game). The methodology provides clarity regarding the functional process of the software being developed, while reducing the risk of failure of specific functionality. It is suitable for the scale of the project and does not demand a team of multiple developers.

1.3 Report Structure

The report consists of seven chapters. Chapter 2 provides context for the project, including a review of literature, related work and relevant technologies. Chapter 3 details project requirements, choice of tools and sources of information. Chapters 4 and 5 catalogue the design and implementation of the API and a game built using it, respectively. Chapter 6

covers testing and iterative updates. Finally, chapter 7 presents a critical evaluation of the project and discusses potential future work.

Chapter 2

Background

This chapter provides context for the project in the form of a review of literature. First, an overview of virtual reality and exercise technologies is given, and their potential for interplay examined. Second, physiotherapy software and previous work involving VR, physical activity and game design for rehabilitation are discussed. Finally, a survey of commercial hardware that may be used for this type of physiotherapy is presented.

2.1 Virtual Reality and Exercise Technologies

Virtual reality is an immersive technology in which interaction with a computer-generated environment is made to seem real. VR simulates experience through the use of purpose-built hardware, the most well-known device being the Head-Mounted Display (HMD). Exercise technology refers to electronics that facilitate or motivate physical activity, such as wristband activity trackers. As outlined in Chapter 1, both of these technologies offer significant potential in improving patient adherence to physiotherapy programs and improving treatment outcomes, especially when used in conjunction [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19].

2.1.1 Virtual Reality Systems

Virtual reality, though most well-known for its applications in computer entertainment, is used across broad range of industries [21]. In VR, interaction with a virtual environment is more immersive than with conventional computer-generated worlds, such as those used in video games. A high level of immersion is achieved through the use of purpose-built hardware to simulate sensory perception.

Early research into the use of VR employed large-screen displays to fill the user's field of view. One evolution of this type of VR was the use of multi-projection devices, such as the Cave Automatic Vitual Environment (CAVE), which superimpose virtual environments onto the walls of a room [22]. HMDs, devices worn on the head with screens directly in front of

the eyes, are another evolution commonly used today. Generally, one screen is used for each eye to achieve stereoscopic vision. The use of an HMD is intended to simulate direct viewing of a virtual environment without the perception of observing it through a display.

Some VR systems integrate technologies for simulation of sensory input other than sight, such as haptics and three-dimensional audio. However, research into their use for VR is less established than with the display [21]. Aside from engineering the display itself, there are two key technical challenges in the development of VR systems.

1. **Virtual reality sickness.** Incorrectly configured virtual environments may cause a range of undesirable physiological symptoms, such as disorientation, nausea, headaches and pallor [23]. Virtual reality sickness is distinct from motion sickness in that it can be induced by visual perception of self-motion without any actual physical movement. Symptoms can be reduced by correctly configuring the rendered field of view, adjusting perspective to better imitate real motion parallax (the perception of more distant objects moving slower than those which are closer) and preventing motion mismatch by translating real, physical user movement to the virtual environment.

2. **Motion tracking.** In order to display a computer-generated world to the user as though it were being viewed directly, the real-world perspective must be translated to the rendered environment. Tracked motion has a total of six degrees of freedom. A user may move in three perpendicular axes: forwards-backwards, up-down, and left-right. In addition, they may rotate their view about each of these axes through head movement, demanding the tracking of pitch, yaw, and roll. User tracking may be achieved through the use of optical sensors, such as infrared cameras, or wearable tracking devices. Tracking outputs must also be accurately mapped to movement in the virtual environment.

2.1.2 Exercise Technologies and Gamification

Formal literature on exercise technologies is limited. For the purposes of this project, the term refers to intelligent hardware devices enabling or motivating exercise. These devices have their own computing capability, most commonly an ability to measure and display physiological data such as the user's heart rate.

The two most widely available exercise technologies are wearable activity trackers, such as the Fitbit, and stationary exercise equipment. Wireless data transfer, data standardisation, safety, hygiene and user fatigue are the primary challenges in the development of these systems. Given the motivating factors described in chapter 1, however, the point of discussion most relevant to this project is how these technologies motivate adherence to regimented exercise programs.

As evidenced by the spectrum of sport popular today, physical activity lends itself to

game-based approaches. In recent years, exercise equipment such as stationary rowing and running machines have seen widespread introduction of game-based motivational systems [24, 25]. Interestingly, many of these systems draw not only on sporting game design, but are also heavily influenced by aspects of video games, keeping track of user profiles, achievements and high-scores. This is an example of *gamification*, a phenomenon defined in its broadest terms as the use of game design in contexts other than games. In its application to exercise, gamification reinforces the already game-like nature of physical activity.

Several dozen studies over the past few decades have shown the use of game design to improve physical activity levels. In the context of exercise, there is a strong correlation between gamification and increased energy expenditure [26, 27]. With its increase in popularity, the application of games to exercise is now often referred to as *exergaming*.

2.1.3 Active Virtual Reality (AVR)

Virtual reality is a useful tool in motivating exercise, with applications designed for the use of both VR and physical activity showing strong synergy between the two [5, 6, 7, 28]. In the context of physiotherapy, their combined use has strong potential in improving patient outcomes [10, 11]. Gamification of the tasks involved may further enhance these benefits [13, 14, 15, 16, 12, 17, 18, 19]. The use of gamified exercise in VR, as employed in a significant portion of the literature presented to date, may be referred to as Active Virtual Reality (AVR).

The primary challenges in the development of AVR systems carry over from those of standard VR and exercise technologies, albeit with two notable differences. First, some hygiene and safety issues associated with exercise technologies become more apparent with the use of an HMD. As a result of the perspiration resulting from physical activity, HMD padding may need to be adapted to address hygiene concerns. Furthermore, as HMDs obscure the user's vision, additional safety concerns arise, principally trip hazards.

Second, motion tracking becomes a more complex task. To reiterate, tracked motion has a total of six degrees of freedom: movement along the three spatial axes as well as rotation about each of them. The additional challenge introduced with AVR, however, is the introduction of exercise equipment, which may hinder tracking mechanisms by obscuring the user and introducing a greater number of moving parts to the system. While there are obvious solutions to this problem, such as additional optical sensors or the combined use of optical systems and wearable tracking devices, these are high-cost approaches.

Huang et al. (2017) were able to mitigate the AVR tracking problem at low cost using Microsoft's Kinect sensor [29]. The Kinect makes use of an RGB camera and a depth sensor to build a world model. User skeleton tracking is achieved in two phases: the device builds a depth map of the scene being observed and subsequently infers body pose using machine learning techniques [30]. The depth map is constructed using structured light depth acquisition, which involves projecting a known pattern of infrared light onto the scene and computing

depth from the deformation of the pattern. Inferring body pose is done by means of a neural network trained on a dataset of known body poses. In Huang et al.'s AVR experiments, the Kinect sensor achieved accurate full-body motion tracking even with the user partly obscured by an exercise bicycle.

2.2 Physiotherapy Software and AVR

Although use of software for physiotherapy is described in a wide range of studies and is gaining momentum in the commercial healthcare sector, no single solution or standard dominates the field. Commercial physiotherapy software, such as SimpleSet, Physiotools and Rehab Guru, is almost exclusively oriented on exercise prescription. These solutions rely on static exercise databases, resulting in a more generalised approach to physiotherapy. In contrast, software used in rehabilitation studies is often purpose-built for recovery from specific injuries or diseases. The consequence of the disconnect between commercial and research-oriented physiotherapy software is that no single platform exists that facilitates sufficiently specific treatments to replace conventional therapy *and* that can service a range rehabilitation functions broad enough to be usable on a large scale. This reinforces the impracticality of hard-coding physiotherapy exercises identified in Chapter 1.

The primary challenge in the development of physiotherapy software that is sufficiently specific *and* medically useful for a wide range of rehabilitation functions is that there is no single catch-all set of guidelines for physiotherapy. This can be attributed to the fact that physiotherapy treatments evolve over time, are not consistent across borders, and guidelines are not always followed [31, 32]. Guidelines, if possible to formulate, would be helpful in defining physiotherapy software specifications.

While the introduction of AVR to physiotherapy significantly reduces the required scope of guidelines, balancing the generalisability of software and the specificity of its rehabilitation functions is difficult. At the time of writing, there are no guidelines for AVR physiotherapy, however, Chapter 3 discusses considerations and requirements for the development of systems for this purpose.

One potential approach to mitigating the inconsistency of physiotherapy software is the adoption of an Electronic Health Record (EHR) data standard. A number of physiotherapy-specific medical record specifications exist, such as WebPT and ECLIPSE Electronic Health Records, however their primary functions are often scheduling and medical billing [33]. More applicable to AVR is MIRA Rehab, which focuses on rehabilitation through exergaming. MIRA Rehab has been used for AVR physiotherapy in two formal studies with limited success [34, 35]. However, currently available physiotherapy-specific standards are commercial, often using proprietary file formats and having limited interoperability between points of care.

Better interoperability may be found in more general medical data standards. As of

2016, Health Level 7 (HL7) International maintains the most widely-used and interoperable standards [36]. The most up-to-date HL7 standards, currently at a trial release stage, are the Fast Healthcare Interoperability Resources (FHIR). In contrast to traditional healthcare data standards, FHIR adopts an up-to-date Representational State Transfer (REST) architecture and agile programming principles [37]. The specification is written for web protocols, supports modern standards like HTML, XML and JSON for user interfacing as well as includes an HTTP-based API [38].

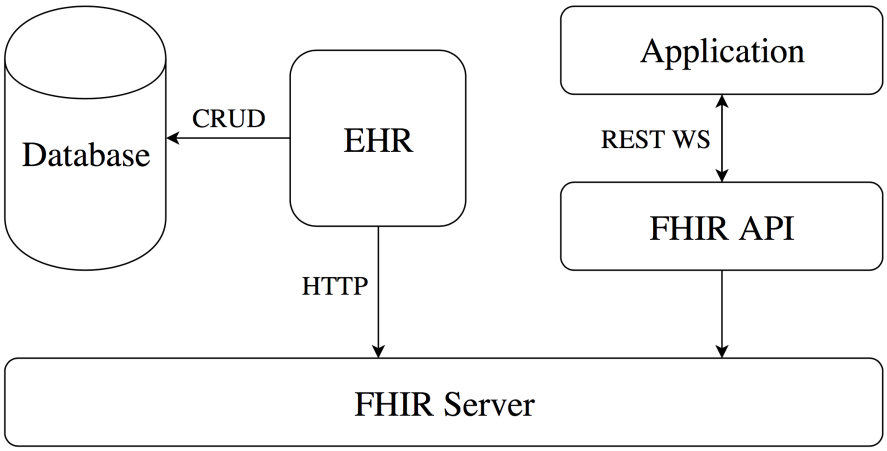

Figure 2.1: Architectural diagram of the FHIR specification.

Thanks to its use of up-to-date architecture, the development of FHIR-based applications is considerably less difficult than using its traditional counterparts, such as ASC X12 and previous HL7 standards. Most notably, the Substitutable Medical Applications and Reusable Technologies (SMART) platform was adapted for use with FHIR in 2016 [39]. A small library of SMART applications using FHIR are currently available for download, though none are written specifically for physiotherapy.

The data transfer framework used for FHIR is most reflective of industry best practices for complex systems development among similar specifications. Though the most appropriate choice for medical applications of virtual reality, FHIR has yet to be used with VR.

2.2.1 AVR Physiotherapy and Existing Software

To reiterate, AVR physiotherapy refers to VR approaches to rehabilitation that involve both physical activity and gamification of physiotherapy tasks. The limited number of studies and two commercial systems in this sector largely focus on upper extremity rehabilitation. Many solutions also do not make use of HMDs, but rely on outdated large-screen VR solutions.

The most remarkable, and perhaps earliest use of AVR physiotherapy is a study by Holden and Dyar, which employed techniques such as tracked finger movement, 3D glasses and head-mounted displays for motor control rehabilitation as early as 2002 [16]. Though physical

activity played a smaller role than in more recent studies, Holden and Dyar's system was highly effective in rehabilitating upper extremity motor control and strength.

Fung et al. (2004) developed an AVR system using a treadmill mounted on a six degrees of freedom motion platform, simulating terrain incline underfoot [40]. Tracking involved the use of electromagnetic sensors worn by the user. The high-cost system was intended for locomotor therapy, with later studies using the device seeing some success in gait rehabilitation [17, 18].

Kizony et al. (2006) proposed the first low-cost system for home-based AVR physiotherapy [41]. The software, dubbed TheraGame, involved interaction with a virtual environment using tracked upper body movement. VR was achieved through use of a large-screen display, while user tracking was achieved using a standard webcam. This approach produced modest results.

Cameirão et al. (2007) used a similar approach in developing the Rehabilitation Gaming System (RGS), though tracking required the user to wear coloured patches [42]. A later study using the RGS found it to be significantly more effective than time-matched occupational therapy in upper extremity rehabilitation for acute phase stroke patients [13].

At the time of writing, there are two commercial platforms for AVR physiotherapy, VRHealth and Immersive Rehab [43, 44]. VRHealth has three physiotherapy applications available for purchase: two for cervical spine range of motion rehabilitation and one for upper body extremity rehabilitation. These are available at $99 per month and run on the Oculus Rift. Immersive Rehab has two physiotherapy applications, both for upper limb rehabilitation. Immersive Rehab applications are compatible with the Oculus Rift and HTC Vive. Both AVR physiotherapy platforms have considerable hardware requirements.

As stated in Chapter 1, the project challenge is the development of a standardised, open platform for physically active game-based VR physiotherapy. As evidenced by the literature reviewed thus far, a useful contribution to the field must enable physiotherapy tasks to be created and should use an established medical data specification for standardisation. None of the aforementioned systems meet these criteria.

2.3 Commercial AVR Hardware

Commercial AVR hardware, though not explicitly designed for physiotherapy, could be used to construct a system for an open, standardised rehabilitation platform. Reviewing these systems creates a background of available technologies that could be applied to the project.

At the time of writing, there are two principal methods that enable virtual reality to be employed with exercise equipment:

1. A system of an HMD and mechanical exercise equipment
2. an HMD only, directing the use of equipment external to the system

2.3.1 Systems featuring HMDs and Mechanical Exercise Equipment

The most common type of AVR systems feature equipment with moving parts networked to an HMD. There are two distinct equipment categories: hand-operated equipment and omnidirectional treadmills.

Systems featuring hand-operated exercise equipment are designed with exercise as the primary function. One example is VReeMotion, a motion, inertia and wind-simulating recumbent stationary bicycle. Another is ICAROS, a forward-leaning gyroscopic chair. A third is the VirZOOM, an upright stationary bicycle with a public software development kit and wide range of available games.

Figure 2.2: ICAROS Pro, a system of an HMD and mechanical exercise equipment [45].

In contrast, for the vast majority of systems featuring omnidirectional treadmills, exercise is not the primary function. As the name suggests, these treadmills allow for locomotion in any direction, facilitating a 360-degree range of movement. Devices such as the Virtuix Omni and WizDish ROVR2 employ convex, circular bases on which the user slides their feet in order to move. Other products, such as the Infinadeck, feature a traditional treadmill band able to move in any direction underfoot.

Figure 2.3: Virtuix Omni, a virtual reality product featuring an omnidirectional treadmill [46].

Due to their versatility when compared to HMD-only solutions, AVR systems featuring mechanical exercise equipment offer the greatest level of immersion and have a novelty factor. Their primary disadvantage, however, is high cost. For example, excluding an HMD, the current retail value of the ICAROS Pro is $9200, the WizDish ROVR2 is priced at $699 and the VirZOOM at $399.

2.3.2 HMD-only systems

Several HMD-only AVR devices are currently available for purchase. For these systems, the software experience consists of instruction for the use equipment not networked to the HMD, or for calisthenic exercises only. The vast majority of HMD-only virtual reality fitness experiences run on the HTC Vive, including the Holodia Holofit, iDEACLOUDs VR Fitness, and the still-in-development Black Box VR. These products vary greatly in terms of game design, with a number opting for a fitness-oriented virtual environment, such as a gym, and some for a more fantastical setting.

There are two obvious advantages of HMD-only AVR devices over systems of HMDs and mechanical exercise equipment: significantly lower cost and inherent portability. These advantages, inevitably, make them more user-friendly. One caveat of this type of system, however, is that exercise metrics, such as heart rate, power generation and output speed are rarely available.

Figure 2.4: Holodia Holofit, an HMD-only system, used with indoor rowing machines [47].

2.4 Summary

The intersection of virtual reality, exercise and gamification has strong potential in physiotherapy applications [13, 14, 15, 16, 12, 17, 18, 19]. The primary challenges in constructing AVR physiotherapy platforms are the absence of catch-all guidelines and technical considerations such as motion tracking. Any useful contribution to the field must allow physiotherapy tasks to be created and should use an established medical data specification for standardisation. No solution meeting these criteria has been developed to date. A selection of hardware is available for constructing this platform.

Chapter 3

Requirements and Considerations

This chapter reviews considerations for the development of AVR systems, identifies use cases and establishes a structured list of requirements for the project task. Choice of tools and sources of information are subsequently identified.

3.1 Problem Statement

The aim of this project is to construct an open, standardised platform for physiotherapy that makes use of:

1. virtual reality;
2. physical activity;
3. gamification.

Through a review of literature, it has been established that the platform should not hard-code physiotherapy tasks, but allow them to be created. Standardisation may be achieved through the use of a medical data specification.

An Application Programming Interface (API) is best suited to these demands. If developed using hardware that provides real-time information about patients, an API grants developers easy access to physiological metrics that may be stored in accordance with a medical data standard. These outputs can be used to create physiotherapy games.

3.2 Considerations for the Development of AVR Systems

As physiotherapy tasks will not be hard-coded, most development considerations stem from the use of AVR. Though more relevant to games built on top of the API, these considerations are useful in formulating structured requirements for the project task. Physiotherapy game design considerations are discussed in more depth in Chapter 6.

A number of contemporary publications have documented the challenges in design and implementation of AVR systems.

3.2.1 Design and Evaluation Models

Successful implementation of AVR is contingent on a well-received user experience, highlighting the importance of the use of design and evaluation models [28]. Few publications presenting such models for AVR systems discuss aspects outside of the game design itself. Of those that do, however, one of the most succinct is in Alexander Whaley's dissertation *Evercycle - A Virtual Reality Exercise Game to Accelerate Exercise Performance Improvement using the Feedforward Effect* [48]. Whaley identifies three major points of consideration for the successful implementation of AVR:

Consideration	Related Challenges
Hardware	• Defining hardware requirements • Ensuring hardware interfaces correctly with software
Design	• Achieving a suitable, immersive game aesthetic • Developing enjoyable gameplay • Implementing a good user interface • Designing an intuitive control scheme • Ensuring accessibility to a variety of users • Setting appropriate in-game goals for user motivation
Safety	• Ensuring system wires cannot get caught during use • Preventing loss of balance for the user • Addressing hygiene concerns

Table 3.1: Points of consideration for AVR systems design.

The above considerations are echoed in a number of publications discussing AVR. One such paper, *Challenges in Virtual Reality Exergame Design*, also presents a number of problems encountered during software development for systems featuring exercise bicycles, including

potential solutions to these issues [49]:

1. **Virtual reality sickness.** As discussed in Chapter 2, use of virtual reality can cause undesirable physiological symptoms in some cases. Virtual reality sickness may be alleviated by adjusting field of view, rendering realistic motion parallax and preventing motion mismatch.

2. **Motion tracking.** Another consideration discussed in Chapter 2, motion tracking demands modelling the user in six degrees of freedom, including when they are partly obscured by equipment. Certain optical sensors and wearable tracking devices can be configured for this purpose.

3. **Feedback latency.** In order to avoid dissociation between game events and system output, sensory feedback should have no perceptible delay. Game design should be adjusted to avoid scenarios where feedback delay is unavoidable, such as large changes in the resistance of an exercise bicycle.

4. **Suitability of hardware.** The level of immersion achieved by AVR devices may be inhibited by insufficient display clarity or screen clouding as a result of increased body temperature during physical activity. In order to mitigate these problems, a high resolution display and appropriate HMD housing should be used.

Another paper echoing Whaley's considerations, published in a recent issue of the European Scientific Journal, aims to construct a complete model for the evaluation of AVR systems featuring stationary bicycles [50]. This model, presented from the perspective of the end user, is divided into four distinct categories: usability, affect, user value and presence feeling. Though it lists many of the same considerations as those detailed by Whaley, it also includes social aspects and user attachment as points of evaluation. A similar model can be found in *Mixed Reality Game Prototypes for Upper Body Exercise and Rehabilitation* [51].

3.2.2 Additional Factors

Various additional factors affecting the user experience of AVR systems have been outlined in related literature. These include use of a simple game design, development of sufficient software content to keep the user motivated for the long term and varied reciprocative features [28, 52, 53, 54]. The use of a personalised or individually adopting game experience is also encouraged, and in the context of medical applications, exercise compliance–the quality of in-game exercises versus their traditional counterparts–may serve as a useful measurement of AVR efficacy [5, 55, 56].

3.2.3 Consolidating Considerations

By consolidating the content of literature on AVR in the context of currently available systems, a more comprehensive set of considerations than those previously written can be tabulated.

Category	Challenge	Considerations
Hardware	Design	• Minimising hardware cost • Motion tracking accuracy • Promoting user attachment
	Implementation	• Correct hardware-software interfacing • Minimising feedback delay
	Safety	• Using an appropriate HMD for exercise • Managing wires and trip hazards
Software	Design	• Aesthetics, user interface and graphical fidelity • Developing high-quality, varied gameplay • Appropriate game complexity and in-game goals • Developing a user-friendly control scheme • Individually adopting challenge
	Implementation	• Including varied reciprocative features • Minimising scenarios with feedback delay
	Safety	• Countering loss of balance and VR sickness • Achieving good exercise compliance

Table 3.2: Considerations for AVR systems development.

3.3 Use Cases

In an AVR physiotherapy session, a patient completes a gamified, physically active rehabilitation task in VR. Physiological metrics, such as heart rate, are read from the patient for medical assessment. Traditionally, a medical professional takes these measurements and adjusts treatments as necessary.

While the core benefit of AVR in physiotherapy is improving treatment outcomes, the technology may also be used for automation of medical assessment. Though physiological metrics may be taken manually by a medical professional, digitising them through hardware readings has two key benefits.

First, automation saves time and reduces equipment required for treatments–if the hardware required for physical activity and VR is able to take measurements, it should do so. Needless to say, minimising overhead is good practice in systems design. Much of the previous work outlined in Chapter 2 failed to take full advantage of hardware for patient assessment.

Second, digitising patient records allows for non-concurrent, remote treatments. If physiological data is sent to a server, it may be assessed remotely by a medical professional. Furthermore, if stored in an appropriate format, the actors' actions need not be concurrent–the medical professional may view treatment records at any time after a session has taken place. Treatments not demanding actor co-presence or concurrent actions may be referred to as *patient-led*. Digitising patient outputs provides the *option* for patient-led rehabilitation, but does not limit physiotherapy to non-concurrent, remote treatment.

An additional useful entity relationship in AVR systems is the flow of information back to the patient. Real-time feedback may promote a patient's ability lead treatment sessions, as well as aid motivation, for example through the use of metric-based in-game goals. One potential hazard associated with this, however, is that it may encourage excessive exertion. Patient safety should be addressed in the development of both the API and games built on top of it.

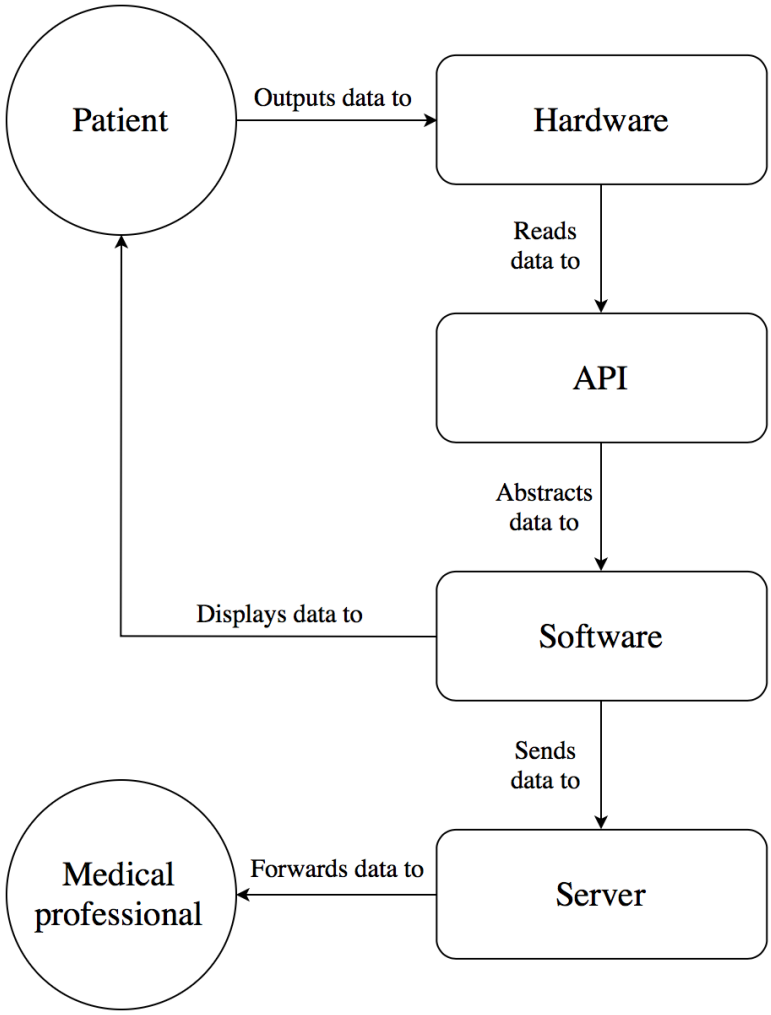

Figure 3.1: Data flow of an AVR system with support for patient-led treatments.

3.4 Structured List of Requirements

System functionality requirements are detailed below, prioritised according to the MoSCoW method [57].

3.4.1 Must-haves

Must-have requirements are necessary for successful project delivery.

- **Appropriate hardware.** The hardware configuration must be appropriate for physiotherapy.
- **Read patient metrics.** The API must read outputs from hardware taking physiological measurements.

- **Abstract data.** The API must perform any data transforms required for human readability, such as specifying units or averaging a metric over a physiotherapy session.
- **Standardise data.** The API must save all records in accordance with a medical data standard.
- **Motion tracking.** User motion must be fully tracked, with movement accurately translated to the virtual environment.
- **Create physiotherapy games.** The API must be usable in a game development environment.

3.4.2 Should-haves

Should-have requirements are important but not necessary for successful project delivery. Should-have requirements are prioritised over could-have requirements.

- **Export data.** The API should allow data to be transferred to a server for patient-led treatment options.
- **Display data.** The API should allow metrics to be displayed to the user for real-time feedback and in-game goals.
- **Documentation.** The API should be packaged with a user guide for developers.
- **Speed.** The API should be efficient and not cause perceptible lag in a game.
- **Safety.** The API should be configured to minimise virtual reality sickness and should not encourage dangerous use of virtual reality equipment or excessive exertion.
- **An example game.** The API should be packaged with an example game that illustrates use of the API and serves a useful physiotherapy function.

3.4.3 Could-haves

Could-have requirements are desirable but not necessary. Could-have requirements are prioritised below should-have requirements.

- **Low hardware requirements.** The API could be undemanding, allowing compatibility with a greater range of systems.
- **Extensibility.** The API could be made extensible so that development may continue after the conclusion of the project.

3.4.4 Won't-haves

Won't-have requirements are not appropriate at this time, but are useful if implemented at a later date.

- **Security.** The API won't be assessed for security vulnerabilities as this is outside of the scope of the problem statement. It is, however, relevant to physiotherapy as medical records should remain confidential. Assuming extensibility of the API is maintained, security features may be added after the conclusion of the project.

3.5 Choice of Tools and Sources of Information

In addition to the tools and sources of information listed below, the project was guided by the challenges, successes and failures of AVR systems detailed in the literature cited.

3.5.1 Software

In terms of development tools, C# and Unity were chosen for their ease of integration with VR tools, widespread use in game development and support for a variety of architectures, including mobile devices. Git was utilised for repository management. For the choice of medical data specification, FHIR was identified as the most up-to-date and appropriate standard for use with complex systems. This project is the first to integrate FHIR with virtual reality.

The sources of information used for software tools are:

1. The C# language specification [58].
2. The official Unity documentation [59].
3. The FHIR Standard for Trial Use (STU) documentation [38].

3.5.2 Hardware

The VirZOOM was identified as the most appropriate AVR hardware device for physiotherapy. The VirZOOM is able to output a more comprehensive set of patient metrics than competing AVR devices at low cost and has an SDK that is compatible with Unity. All HMDs supported by the VirZOOM are intended to function with the API (Samsung Gear VR, Google Daydream, HTC Vive, Oculus Rift and PlayStation VR). The Samsung Gear VR was chosen as the primary development platform to keep API hardware requirements minimal. The Gear VR also has lightweight housing, making it appropriate for physical activity. Note that motion tracking is specific to each headset used, though all solutions allow for six degrees of freedom tracking with a partially obscured user.

The sources of information used for hardware tools are:

1. The Oculus developer documentation (Oculus Rift and Samsung Gear VR) [60].
2. The VirZOOM support documentation [61].

3.6 Summary

A formal problem statement, considerations for AVR systems development and API use cases have been used to formulate a structured list of project requirements. Based on these requirements, C#, Unity, Git, FHIR, the VirZOOM and compatible HMDs have been identified as core development tools.

Chapter 4

FHIRWEAVR

This chapter covers the design and implementation of the API. It presents an overview of the system, followed by a detailed description of each key functionality. Iterative updates in the development process and API documentation are also discussed.

4.1 System Overview

The key challenge in developing a technology platform for medical purposes is balancing ease of use with medical usability. It cannot be assumed that medical professionals have in-depth technical knowledge, or that developers possess in-depth medical knowledge. Patients cannot be assumed to possess either. Accordingly, the full AVR physiotherapy system was designed for ease of understanding by patients, developers and medical professionals alike.

As the first resource to integrate FHIR with virtual reality, the API has been named consistently with other FHIR projects. The title given is *FHIRWEAVR*, an acronym for *FHIR Workbox for Exercise and Active Virtual Reality.*

4.1.1 System Components

The primary challenge in realising the use cases described in Chapter 3 is facilitating the required data flow. This involves three key steps: collecting physiological data through the system hardware, posting that data to a server and displaying it to the patient.

The VirZOOM bicycle alone is not capable of capturing the full range of physiological information that the AVR system requires. While the VirZOOM measures *activity data*, such as heart rate and pedal cadence, is does not track patient motion. *Tracking data* is recorded by the head-mounted display. VirZOOM outputs are transmitted to the system running the API and physiotherapy game via a Bluetooth Low Energy (BLE) connection. HMD outputs are transmitted either via BLE or wired connection depending on the device used–in either case, the flow of data does not change.

The VirZOOM and HMD outputs comprise the two branches of data flow which are merged and abstracted by FHIRWEAVR. The complete set of physiological data for physiotherapy games (abbreviated to *physio data*) is made available to the developer through the API. Posting information to a server is direct in terms of data flow. Displaying physio data to the user, however, involves going back through the HMD. Therefore, the HMD plays two key roles in the system: motion tracking and feedback to the patient.

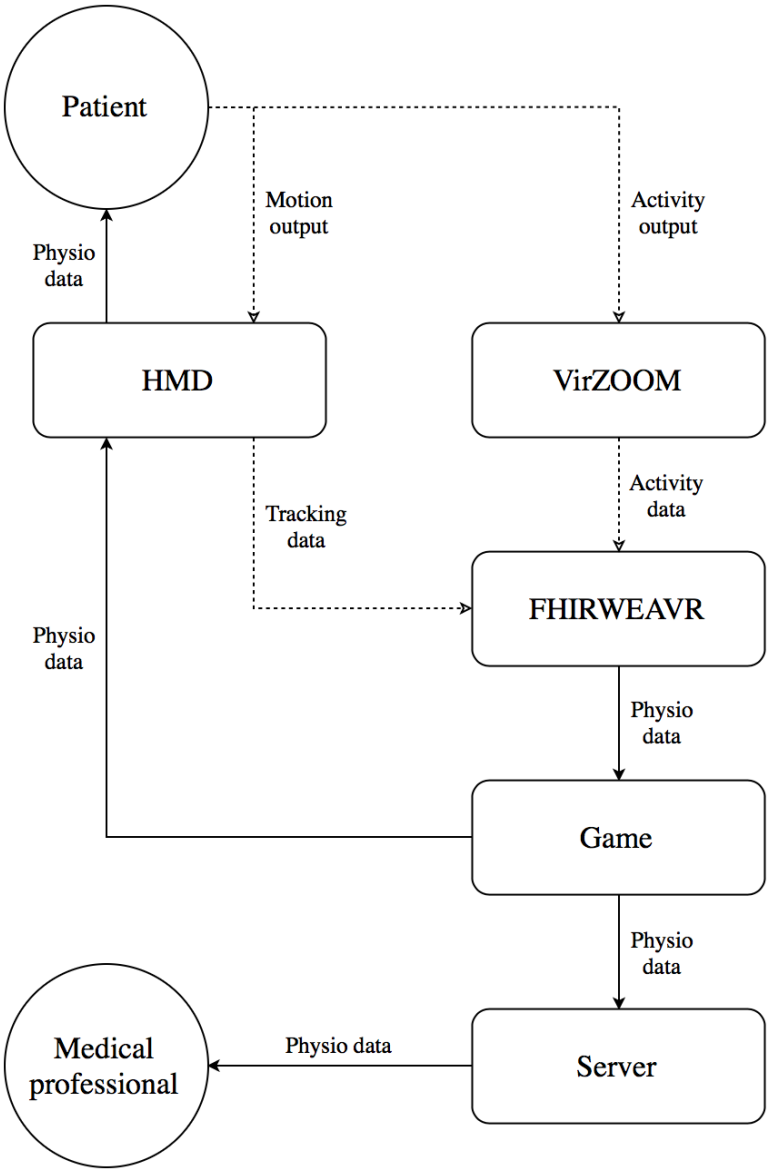

Figure 4.1: Data flow of an AVR system using FHIRWEAVR, cf. Figure 3.1. Dotted lines denote incomplete sets of patient data that are not sufficient for developing physiotherapy games alone.

The API and Game components run on the same physical device. When FHIRWEAVR

is used with mobile VR devices, i.e. the Samsung Gear VR or Google Daydream, the HMD *is* this device. In the case of the Oculus Rift, HTC Vive and PlayStation VR, the HMD is a separate piece of equipment connected to a computer with a wired connection. The primary safety hazard–wires getting caught on the patient or the VirZOOM pedals–is resolved by fasteners on the bicycle.

4.1.2 Software Structure

The complete set of assets for this project is divided into three folders:

1. **FHIRWEAVR.** The API scripts.
2. **DemoGame.** The scripts and 3D assets for an example game built on top of FHIRWEAVR (discussed in Chapter 5).
3. **VirZOOMCore.** A slightly modified version of the VirZOOM SDK. The SDK has been adapted so that a developer may easily move from building on top of the demo game (default FHIRWEAVR project) to a blank scene. FHIRWEAVR UI elements are stored here.

Full documentation is included in the FHIRWEAVR download. The documentation serves as a developer guide for use of the API. The files in the FHIRWEAVR folder that are relevant in this chapter are:

- `DataHandler.cs`: functions for reading and abstracting hardware outputs, as well as displaying real-time feedback to the patient.
- `GenFHIR.cs`: functions for saving data in FHIR-compliant format and for creating FHIR device profiles.
- `PushData.cs`: functions for interacting with and posting data to a server.
- `DateTimeFormats.cs`: functions for handling FHIR's *dateTime* data type.
- `ExampleUses.cs`: example uses of FHIRWEAVR in a class derived from `MonoBehaviour` (the base class of Unity scripts).

4.2 Core Functionality

One easily overlooked aspect of API development is that the architecture alone should not be limited to good software design principles, but **should enable developers building on top of the API to use them as well.** The code base is structured to balance four core properties between the API *and* games built using it: readability, modularity, maintainability and extensibility.

Games favour the use of a component-based architecture because game entity capabilities only become clear at runtime. This makes the principle of *composition over inheritance* a staple of modern game design. Consequently, the core functionality of FHIRWEAVR is available through object instances.

4.2.1 Reading and Abstracting Hardware Outputs

Developers are given access to two types of data defined by FHIRWEAVR: `current` data, real-time readings, and `session` data, measurements taken over a physiotherapy task. For each of these, developers are also given the choice of returning a specific metric or all metrics.

In order to enable *composition over inheritance* in the physiotherapy games to be built on top of FHIRWEAVR, access to these metrics should be from an instance. This presents a problem for a component that reads hardware outputs–with conventional instancing, each instance polls the hardware separately. The outcome is several slightly different accounts of the same information across instances, when in reality, each metric only has a single value at any one time.

The solution to this problem lies in a design pattern known as the *Singleton*. Use of a Singleton guarantees that only one instance of a particular class exists at any one time. While this design pattern can introduce unnecessary restrictions in traditional inheritance-based architectures, it is ideal for classes with complex responsibilities in game code.

The Singleton implementation in FHIRWEAVR's `DataHandler` class is adapted from Gamma's *Design Patterns: Elements of Reusable Object-oriented Software*. The approach is particularly useful for game development because performance optimisation is important in demanding interactive software, and Gamma's Singleton allows for *lazy instantiation*. This means that instantiation occurs only on instance requests.

```
public class DataHandler : MonoBehaviour
{
    private static DataHandler local = null;
    public static DataHandler Instance
    {
        get
        {
            if (local == null)
            {
                local = (DataHandler)FindObjectOfType(typeof(DataHandler));
            }
            return local;
        }
    }
    void Awake()
    {
        local = this;
    }

    ...

}
```

Listing 1: Adaptation of Gamma's Singleton for FHIRWEAVR. `DataHandler` is attached to the patient's game object.

It is important to note that Gamma's Singleton is not thread safe. If multiple threads concurrently enter the instance property method, more than one instance may be created. The Unity API is not available outside the main thread, however, and `DataHandler` derives from `MonoBehaviour`. Hence developers do not need to consider concurrency control when requesting physio data. Thanks to *separation of concerns* in the API, Singleton-related thread safety issues do not affect FHIRWEAVR classes demanding use of multi-threading (see Data Transfer).

FHIRWEAVR provides simple functions for starting and ending a physiotherapy session. During a session, VirZOOM activity information and HMD motion outputs are polled at each rendered frame. This is sufficient for displaying feedback to the patient that appears 'smooth' since the displayed physio metrics update at the same rate and at the same time as the virtual environment. The frame rate also varies between HMDs, so by not fixing the poll rate, the best possible user experience is rendered on each device.

To reiterate, FHIRWEAVR defines two types of data: `current` and `session`. Current data consists of readings at the time of the function call. Session data is computed by totalling or averaging metrics (as appropriate for the metric) from the start of the session to the time of the function call. A specific metric can be returned using the function `GetMetric(metric,`

`type)` and all available metrics can be returned using the function `GetAllData(type)`. The latter function is composed using the former in accordance with the *DRY principle* (Don't Repeat Yourself). Use of the DRY principle eases code maintenance and improves readability.

4.2.2 FHIR Integration and Saving Data

As identified in Chapter 2, the FHIR standard includes a RESTful API. While the original intention was to use this for the project, it was discovered that the VirZOOM and FHIR API have incompatible .NET version requirements. These requirements are not documented, and therefore were not discovered during the research stage of the project. Consequently, a custom implementation bridging AVR and FHIR has been written for the project.

As a result of the 'uncharted' nature of writing a new bridge between the two technologies, a substantial number of options arose regarding data representation in the FHIR specification. Making the most appropriate choices was one of the major challenges of the project. The *KISS principle* ('Keep It Simple, Stupid') was employed to maintain the four core properties of the code base.

The FHIR `Documents` framework was chosen to represent data as it is the only mechanism by which physio data (as defined earlier) is not reliant on an exchange protocol, i.e. information can be saved in FHIR-compliant format without the need for communication with a server. FHIR `Documents` are also the most appropriate FHIR-based data representation for the non-concurrent actions aspect of patient-led treatments. Of the file formats supported by FHIR, XML was chosen for its widespread use in healthcare and inherent extensibility [62].

FHIRWEAVR saves a set of device metrics. For the information to be interpretable, the FHIR `Document` must describe the device, the metrics and the time of their recording. While there is a `DeviceMetric` resource in the FHIR specification, use of this resource explicitly requires the device to implement or derive from the ISO/IEEE 11073 standard, which the VirZOOM does not [63].

In accordance with KISS, a FHIR `Bundle` containing a `Basic` resource was chosen for the physio data. The `Basic` resource handles concepts not yet defined in FHIR and allows a data structure of AVR physio metrics to be represented in the specification. Information about the date and time of the data capture is expressed using FHIR's surprisingly complex `dateTime` data type, which is also used in the file name for quick identification of the correct `Document`. Each `Document` containing a set of physio data references a separate FHIR `Document` containing a `Device` resource which describes the hardware used to record the data.

FHIR `Documents` are generated using C#'s `XmlWriter` class. When calling the function for starting a physiotherapy session, the `Document` containing the `Device` resource is created. Developers can save physio outputs by calling `GenFHIR.Device(type, overwriteName)`, where the `type` parameter functions as with `GetMetric()`/`GetAllData()` and `overwriteName` is an optional parameter for specifying a non-default file name. The `overwriteName` parameter

is useful for overwriting an existing document with new physio data before sending it to a server. Note, however, that FHIR does not allow modification of existing resources with a unique identifier, so the file is given new identifying information each time it is changed.

4.2.3 Data Transfer

As discussed in Chapter 3, patient-led treatments should be optional. Where AVR physiotherapy does not demand patient-led treatment capabilities, game architecture should not be complicated by the module providing this functionality. This allows game developers to follow the *YAGNI principle* ('You Aren't Gonna Need It') and is a good example of why *composition over inheritance* was used in the design of FHIRWEAVR [64]. The YAGNI principle is particularly useful where software requirements change regularly, making it highly appropriate to physiotherapy, which often involves progressive treatment adjustment. Accordingly, the `PushData` class can be instanced only if it is needed.

Unlike `DataHandler`, `PushData` is a standard class not derived from `MonoBehaviour`. It does not follow the Singleton design pattern for two reasons:

1. There is no need for it (KISS principle).
2. For a smooth game experience, data upload demands multi-threading and Gamma's Singleton is not thread safe.

`PushData` uses HTTP for data transfer. HTTP was chosen because the FHIR RESTful API (which was incompatible with the VirZOOM) uses this protocol for communication with FHIR servers. `PushData` uploads FHIR `Documents` to a server through an HTTP POST. A HEAD request checks if the host is available. HTTP GET is used to check if a copy of the `Device` profile is present on the server–if not, this FHIR `Document` is uploaded in addition to the physio data `Document` specified in the function call.

Use of the prototyping model, as discussed in Chapter 1, revealed a performance issue related to use of HTTP protocols early in the development process. When run on the main thread, data transfer caused significant stuttering in the 3D rendering. Accordingly, functions for uploading data to a server are given their own thread when called and do not cause stuttering.

4.2.4 Displaying Data

Displaying physio data to the patient provides real-time feedback for both improved motivation and more effective patient-led physiotherapy. In FHIRWEAVR, this is implemented through a customisable User Interface (UI).

One novel aspect of VR development is UI design. On conventional computing devices, game UIs are rendered as though 'painted' on top of the screen. In the case of an HMD, a

'painted-on' UI is too close to the eyes for the user to focus on it. There are two ways of displaying two-dimensional information to the user in VR:

1. **Spatial UI.** The UI is rendered as an object in the three-dimensional virtual environment space, rather than on the screen.

2. **Diegetic UI.** The game objects themselves display information to the user.

By default, the FHIRWEAVR UI object is rendered as a spatial UI. However, it may be made diegetic by a developer. The UI is customisable, including position, font and colour–it is intended as a skeleton for constructing more complex interfaces. It is also only displayed if the developer calls the `DisplayMetric()` or *DisplayAllData()* function, not obligatory.

The aforementioned functions have optional parameters for specifying the length of time the information is displayed (indefinite by default) as well as for adding custom information to the UI. The metric and type (`current` or `session`) are specified in the same way as with the `GetMetric()` and `GetAllData()` functions. `DisplayMetric()` and `DisplayAllData()` are composed using `GetMetric()` in accordance with the DRY principle.

Functions for displaying data are part of the `DataHandler` class and called from the same instance as those for getting data. Originally, these functions were defined in a separate class in accordance with the *single responsibility principle*, which states that each class should implement a single functionality, preventing them from having more than one reason to change during code maintenance. However, user testing revealed that developers found instantiating several FHIRWEAVR modules inelegant, so the two were merged (see Chapter 6).

4.3 Summary

FHIRWEAVR has four core functions: it reads and abstracts hardware outputs, integrates the FHIR specification with VR, uploads physio data to a server and displays real-time feedback to a patient. The API is designed so that both its own architecture *and* games built on top of it follow good software development principles.

Chapter 5

AVR Demo Game

This chapter covers the design and implementation of an active virtual reality demo game built using FHIRWEAVR. An overview of the game, including how it serves a useful physiotherapeutic function, is followed by a description of core functionality.

5.1 Game Overview

The demo game is an example use of FHIRWEAVR. Developers are encouraged to extend the demo game (non-commercially) in the API documentation.

5.1.1 Serving a Useful Physiotherapy Function

To reiterate, AVR physiotherapy involves three elements:

1. virtual reality;
2. physical activity;
3. gamification.

A number of studies demonstrate that the components of AVR physiotherapy are effective in the rehabilitation of balance [65]. In regard to stationary bicycles, the combined use of VR and exercise has been shown to be beneficial for balance therapy [10, 66]. Aspects of gamification have yet to be applied in this context.

Inness and Howe (2002) as well as Bisson et al. (2004) identified VR tasks where the upper body leans and extends from a fixed lower-body position as an effective form of physiotherapy for balance [11, 9]. Intuitively, an exercise bicycle such as the VirZOOM would be particularly appropriate for this approach given its stationary base.

A widely-cited review of motor rehabilitation using virtual reality found stationary bicycles to be safe and effective tools for VR balance physiotherapy [65]. The review remarked that

previous work in this field did not make sufficient use of physical activity or auditory cues. Consequently, the demo game presented here aims to do precisely that.

5.1.2 AVR Game Design

The demo game is intended for balance rehabilitation. Following the accomplishments of past work in VR balance therapy, it aims to improve motor control and help treat weaknesses in muscles used for postural stability. Drawing on the successes and failures of the aforementioned studies, the demo game encourages a patient to steadily shift their centre of gravity while keeping their point of contact with the ground constant.

The AVR software considerations identified in Chapter 3 influenced the design of the demo game. These considerations can be grouped into three principal game design elements:

1. Aesthetics;
2. Controls;
3. Gameplay.

Gameplay is perhaps the most complex aspect, involving the incorporation of reciprocative features and an individually adopting challenge. Before developing gameplay, however, a means of encouraging a patient to steadily shift their centre of gravity was identified.

Controls, being the sole method of interaction with a game, are inevitably and continually used during AVR physiotherapy. Therefore, in the demo game, steady shifts in a patient's centre of gravity *are* the controls. In-game movement is achieved by leaning in the desired direction of travel. Given FHIRWEAVR's inclusion of tracking information in its physio data, implementing this control scheme was straightforward. Leaning-based controls have the added benefit of being intuitive when using the VirZOOM, since leaning is the natural way to steer a bicycle.

Given the current popularity of the *endless runner* genre, this gameplay style was chosen for the demo game. Endless runners involve never-ending 'courses' or 'tracks' that the player travels along, avoiding obstacles and collecting items to increase their score. The primary goal is to achieve as high a score as possible. Three-dimensional endless runners lend themselves particularly well to the leaning-based controls described above for two reasons:

1. Left/right movement is the sole mechanism for avoiding obstacles and picking up items.
2. Understanding player collisions with objects is highly intuitive when the player's perspective is in first person, as is the case with VR.

Conventionally, endless runners display a 'game over' dialog on the first collision with an obstacle. In the case of AVR physiotherapy, however, rehabilitation takes priority over entertainment. Needless to say, treatment requirements should interrupt the game, not the

converse. Accordingly, the demo game differs from traditional endless runners in that there is no 'player death' mechanic–instead, collisions with obstacles decrease the player's score.

In terms of aesthetics, a non-threatening, low polygon count graphics style was chosen. The use of a minimalist virtual environment makes the game easy to understand, appeals to a wide audience and is undemanding in terms of hardware. This choice is minimally restrictive in terms of target users and budget, thereby making it the aesthetic that is least obstructive to the physiotherapy itself.

Figure 5.1: An in-game screenshot. The player must avoid obstacles and pick up coins by leaning left and right. Forward movement is controlled by pedalling speed (see Core Functionality below).

5.2 Core Functionality

The software development principles and core properties used to develop FHIRWEAVR were also used for programming the demo game.

5.2.1 Gameplay

Gameplay involves collecting coins to increase the player's score. Collisions with obstacles decrease the player's score. In order to enable setting individual challenges, forward momentum is controlled by pedalling speed. Patients struggling with reaction times can pedal more slowly, while those seeking a challenge can–intuitively–speed up as they would on a regular bicycle. All game controls are mapped directly from FHIRWEAVR physio data.

The demo game also monitors heart rate and displays feedback to the patient, discouraging excessive exertion if the reading approaches a dangerous level. Similarly, if heart rate nears a resting value, the patient is encouraged to speed up (on the recommendation of the aforementioned review of VR-based motor rehabilitation). Heart rate readings are taken from

FHIRWEAVR physio data. Feedback is shown using the API functions for displaying data and includes use of the optional parameter for custom information.

The demo game includes a number of reciprocative features. First, different sounds are played when the patient successfully picks up a coin or collides with an obstacle [67]. Second, a 'camera shake' effect is displayed on collisions with obstacles. Third, a 'rumble strip' effect is rendered if the player continues to lean against the fence on either side of the track. Each of these features allow the player to identify and understand in-game events quickly.

5.2.2 Procedural Generation

From a game development perspective, the most challenging aspect of endless runners is conspicuously in the name: they are *endless*. Clearly, content must be generated as the player progresses. However, naive approaches to content generation such as repeatedly copying in pre-made sections of track quickly lead to predictable and monotonous gameplay.

A solution to this problem can be found in *procedural generation*, which involves computing game layouts algorithmically. Procedural generation does not repeat object arrangements, thereby ensuring the environment always appears new and interesting. A simple procedural generation algorithm has been developed for FHIRWEAVR's demo game.

The demo game has a finite set of game object types, each of which has a finite set of three-dimensional models. For example, `Obstacles` (object type) can have four different appearances (models). For each object type, an acceptable range of values for the position, orientation and scale is defined. The acceptable range ensures objects don't render unnaturally, for instance floating in mid-air, or in a way that is unfair to the player, such as two obstacles placed next to each other making a collision inevitable.

Game objects render cyclically about the player so that onward progress can continue indefinitely: once an object goes out of sight behind the player, it 'respawns' just out of sight in front of the player. When an object respawns, it does so with a random position, orientation and scale in the acceptable range for that object type. The same objects are re-used, avoiding memory leaks, but the environment generated is always unique. The model set is static, but the applied transforms are dynamically generated.

As game objects have common moving behaviour, a central class is used for respawning objects using the generation algorithm. This class, `EnvironmentTranslator`, has customisable properties set by each object type to define their acceptable ranges. It also handles non-specification, so if a property is not relevant to a certain object type, it doesn't need to be assigned a value. This is an example of the *state* and *facade patterns*, which make a relatively complex algorithm modular, extensible and completely human-readable.

```
using UnityEngine;
public class Palm : MonoBehaviour
{
    EnvironmentTranslator objectMover;
    void Start()
    {
        objectMover = gameObject.AddComponent<EnvironmentTranslator>();
        objectMover.Limit = 72.39;
        objectMover.DoesScale = true;
        objectMover.IsScenery = true;
        objectMover.XMin = 31;
        objectMover.XMax = 32;
        objectMover.ScaleMin = 8;
        objectMover.ScaleMax = 15;
        objectMover.ZSpread = 32;
    }
}
```

Listing 2: Full class for the `Palm` object. Palms can spawn either side of track but not on it, in a large range for all three spatial axes, in a variety of sizes and in any orientation. Despite having one of the most sophisticated spawning behaviours, its class is remarkably simple.

5.2.3 Graphics Rendering

A number of non-essential animations were added to the demo game to improve the user experience, aiding in immersion and enhancing the motivational benefits of AVR physiotherapy. For example, objects regularly in peripheral vision are rendered with realistic motion parallax to mitigate virtual reality sickness and coins spin in place for visual appeal.

The game's three-dimensional models are partly new, partly from the VirZOOM SDK and party from a defunct free Unity asset [61, 68]. Occlusion culling is used to improve performance when rendering these models. To prevent 'pop-in' of objects as they are spawned ahead of the player, volumetric fog is used to fade them in.

Figure 5.2: Distant objects fade in through volumetric fog.

5.3 Summary

A demo game has been built on top of FHIRWEAVR, showcasing the API and acting as a springboard for future AVR physiotherapy games. The game rehabilitates balance by encouraging patients to steadily shift their centre of gravity, an approach supported by the surrounding literature. The game is in the endless runner style and uses procedural generation.

Chapter 6

Testing and Iterative Updates

This chapter outlines the testing process used for the project, including the technical, developer and end-user stages.

6.1 Quality Assurance

Software testing is a form of quality assurance that can be used to ensure a project both meets requirements and has a sufficiently stable, usable code base. *Unit*, *integration* and *system testing* assess the technical aspects of software, whereas *acceptance testing* ensures projects meet end-user demands. These assessments reveal weaknesses in software that can be patched through iterative updates following each round of testing.

6.2 Unit and Integration Testing

Unit and integration tests were performed at each cycle in the prototyping development model, both for FHIRWEAVR and the demo game. Testing used Unity's *Test Runner*, an integration of the *NUnit* library in the Unity editor [69]. Testing code that derives from `MonoBehaviour` is notoriously difficult and the Test Runner is the only comprehensive test suite for Unity code. Unfortunately, it does not provide any complex test metrics such as code coverage. Unit tests can be performed using *EditMode* in the Runner and integration tests can be performed using *PlayMode*.

Given the use of the prototyping software development model, the entire system underwent testing at each cycle of the prototyping process. Work on the next prototype did not begin until all tests were passed. This approach to testing allowed for rapid development of each major piece of functionality and ensured that all components were reliable before work on the next prototype began. This proved effective in systems integration, however the approach did not handle unexpected events–where new information revealed that previously tested code required updating, additional tests had to be written.

Where possible, mocking was used to counter difficulties in testing with Singleton instances and data upload functions were tested using the Post Test Server V2 [70]. Unit and integration testing across different system configurations was filtered by the `UnityPlatformAttribute`.

```
using UnityEngine;
using UnityEngine.TestTools;
using NUnit.Framework;
using System.Collections;

public class CoinTest {

    [UnityTest]
    public IEnumerator CoinSetsEnvironmentTranslatorProperties()
    {
        var coin = new GameObject();
        coin.AddComponent<Coin>();
        var start = coin.transform.position.z;

        yield return new WaitForSeconds(1);
        Assert.Greater(start, coin.transform.position.z);
    }

    ...

}
```

Listing 3: An example test checking if the `Coin` script successfully sets properties of `EnvironmentTranslator` and the object begins moving in the correct direction.

FHIRWEAVR has been tested with the Samsung Gear VR, Oculus Rift and and HTC Vive, though the API is also compatible with the PlayStation VR and Google Daydream.

6.3 System Testing

Three novice developers and one experienced developer were given access to FHIRWEAVR and the demo game. The developers were referred to the API documentation and the Unity user manual, then challenged to add a new game mechanic in the space of an hour. At the end of the session, they were asked to fill out a feedback form detailing their impressions of the API.

The experienced developer introduced a new game object type, a `Gem`, which increased the player's score by ten instead of the standard one from coins. The developer wrote a class for gems that specified properties of `EnvironmentTranslator`, successfully adding it as an object in the procedural generation algorithm (see Appendix A). Though the novice developers

implemented less impactful features, all participants successfully added new mechanics using FHIRWEAVR functions.

In terms of feedback, three of the four developers remarked that they disliked instancing three different modules to make full use of the API. This feedback was incorporated by merging the two most commonly used features, getting and displaying data, into the single instantiable class `DataHandler`. When polled at a later date, the developers agreed that this was an acceptable solution. Other than the objection to multiple instancing, feedback was altogether positive–two of the four developers wrote that they particularly liked the ability to display UI elements for a specified period of time.

6.4 Acceptance Testing

Acceptance testing involved five individuals playing a short round of the demo game. Two rounds of testing were performed with the same subjects. Feedback from the first round was used to update the API and demo game before the second round.

Of the five individuals who took part in the acceptance tests, four had no disability. The final participant, referred to here as E, has impaired motor control and visual perception following a car accident. This individual has regularly engaged with physiotherapy for almost two years.

Participants were asked to fill out a simulator sickness questionnaire before and after each round of testing to gauge changes in physiological symptoms, if any. A feedback form was also completed following each round of testing. These forms, as well participant information sheets, were adapted from those used for recent virtual reality interaction studies at University College London [71].

Figure 6.1: The system arrangement used in acceptance testing (picture consent obtained).

In the first round of testing, three participants reported increased feelings of nausea after the test. No participant was able to identify the cause. At the time of the test, realistic motion parallax rendering had not yet been added (now the `DoesFishEye` property in `EnvironmentTranslator`). Consequently, between the rounds of testing, objects in the peripheral vision were made to appear as though moving faster and the camera viewpoint was adjusted downward. After the second round of testing, only one participant reported increased feelings of nausea.

All participants expressed frustration with the sensitivity of controls in the first round of testing. Before the update between rounds, leaning any amount in either direction caused in-game movement. Participants requested that moving in a straight line be made easier. Accordingly, controls were adjusted so that lateral movement only occurred above a certain degree of leaning. In the second round of testing, more than one participant noted that this gave the virtual bicycle a realistic feeling of weight.

An interesting finding of acceptance testing was that the perceived location of the virtual bicycle and its actual location in-game did not coincide. Despite the demo game's inclusion

of a full three-dimensional rendering, three participants found collision detection unfair in the first round of testing, all of them stating that they expected the bicycle to be directly below them. Pursuant to this feedback, the player hit-box was adjusted to extend below the player camera rather than around the bicycle. In the second round of testing, no player reported that gameplay seemed unfair.

General feedback for FHIRWEAVR was encouraging. Participants reported high levels of enjoyment during testing, with four out of five expressing that they would be happy to continue the task for an extended period of time (more than five minutes). Participant E stated that the experience was significantly more enjoyable than physiotherapy treatments they had completed in the past.

6.5 Summary

Unit, integration, system and acceptance testing were conducted in turn. Testing revealed weaknesses in the project software that have been patched through iterative updates. Feedback from developers and end users is highly encouraging and bodes well for FHIRWEAVR's potential as an AVR physiotherapy games platform.

Chapter 7

Conclusions and Evaluation

This chapter outlines project achievements and critically evaluates FHIRWEAVR in the context of previous work. Potential future extensions to the API are also discussed, followed by closing remarks.

7.1 Project Achievements

All of the project objectives identified in Chapter 1 have been achieved:

7.1.1 Research Objectives

1. **To identify the key characteristics of useful VR physiotherapy software.** Chapter 1 identified physical activity and gamification as the essential components of effective rehabilitation using VR. Chapters 2 and 3 provided a comprehensive discussion of these elements and introduced additional favourable characteristics of AVR physiotherapy systems. Chapter 5 expanded on these characteristics for a specific class of physiotherapy.

2. **To review relevant VR and exercise technologies, as well as survey existing systems employing the two.** VR, exercise and AVR technologies have been discussed at length. A review of devices used for previous work in this field, both research-oriented and commercial, has been presented. Finally, AVR technologies that could be applied to the project were comprehensively surveyed.

3. **To identify the key challenges involved in developing these systems and to formulate a set of considerations for their use.** Chapter 2 detailed the development challenges associated with VR and exercise technologies, as well as additional problems arising from their combined use. An extensive set of considerations for use of the two was formulated in Chapter 3.

4. **To catalogue the requirements of a tool for creating VR physiotherapy software.** Drawing on the project research and use cases, structured requirements were established in Chapter 3. Requirement prioritisation was also specified using the MoSCoW model.

5. **To identify considerations for software built using the API.** AVR software development considerations were established in Chapter 3 (see Table 3.2). These considerations were refined for development of the demo game in Chapter 5. Further physiotherapeutic, type-specific requirements were also introduced in this chapter.

6. **To identify the most appropriate programming language(s) and software tools for the development of the API and example software, and to become proficient in their use.** C#, Unity, FHIR and Git were identified as the most appropriate tools for developing FHIRWEAVR following the project research. The successful completion of the project by these means would not have been possible without acquiring proficiency in their use.

7.1.2 Technical Objectives

1. **To develop an interface for the creation of VR physiotherapy games, giving access to necessary patient metrics and standardised outputs in a medically useful format.** FHIRWEAVR's ability to meet these demands has been discussed extensively and is evidenced by the successful development of a demo game that serves a useful physiotherapy function. System and acceptance testing have further demonstrated the API's real-world practicality at developer and end-user levels.

2. **To write documentation detailing full use of the API.** Comprehensive documentation has been written for FHIRWEAVR. System testing evidenced its usability for experienced and novice programmers.

3. **To test use of the API among developers, collect feedback and iteratively update the software according to this data.** System testing with external developers engendered encouraging feedback and revealed an inelegance in the API design. This was patched and accepted as a solution by the testers.

4. **To develop an example game built using the API that serves a useful physiotherapy function involving both VR and physical activity.** A demo game for balance rehabilitation has been developed drawing on relevant medical literature. The gameplay demands extensive use of both VR and exercise.

5. **To test use of the game among potential end-users, collect feedback and iteratively update it according to this data.** Multiple rounds of acceptance testing,

including a motor-impaired participant, have been completed. Several updates pursuant to the feedback collected have been applied, most notably improving safety through VR sickness mitigation.

In addition to meeting project objectives, FHIRWEAVR achieves complete coverage of the must-have, could-have and should-have components of the MoSCoW structured requirements. The above examination of project objectives delineates the vast majority of these, however *speed* and *low hardware requirements* may need additional clarification. The API's efficiency is evidenced by its ability to run on mobile devices. Testing, which included use of the two-year-old Samsung Galaxy S7, gave no indication of poor performance. As discussed in Chapter 4, API efficiency was significantly improved by the implementation of multi-threading.

7.2 Comparison to Previous Work

To date, there have been five different attempts at AVR physiotherapy platforms:

1. TheraGame (2006) [41].
2. The Rehabilitation Gaming System (RGS) (2007) [42].
3. VRHealth (2016) [43].
4. Immersive Rehab (2016) [44].
5. MIRA Rehab for VR (2017) [34, 35].

Favourable characteristics of AVR physiotherapy platforms and their exhibition in the above, as well as in FHIRWEAVR, are tabulated below. These properties have been identified from the surrounding literature and are discussed throughout this report. Given their similarity in implementation, VRHealth and Immersive Rehab (IR) have been grouped.

Characteristic	TheraGame	RGS	VRHealth/IR	MIRA	FHIRWEAVR
Fully immersive (HMD)	✗	✗	✓	✗	✓
Standardised medical data	✗	✗	✓	✓	✓
Treatments not hard-coded	✗	✗	✗	✗	✓
Patient-led treatments	✓	✗	✓	✓	✓
Open-source software	✗	✗	✗	✗	✓*

Table 7.1: Comparison of AVR physiotherapy approaches. *FHIRWEAVR will be made open-source in the next year.

Compared to the aforementioned approaches, FHIRWEAVR's principal shortcoming is that it has yet to be trialled in a formal medical study. It is difficult to make claims to medical utility before trials have taken place.

Another minor difficulty with the API is that FHIR is currently a standard for trial use. Changes to the specification before the full release may require FHIRWEAVR to be revised. A significant update to FHIR took place during the project which required the API to be reworked.

7.3 Future Work

As FHIRWEAVR is an API, all future work should follow the *open/closed principle*: the software is open to extension but closed to modification. This ensures that projects built on top of FHIRWEAVR do not become non-functional following future updates.

There are three principal avenues of future work. First, the won't-have MoSCoW requirement of security should be implemented. Confidentiality of patient records is a legal requirement in many countries, and FHIRWEAVR's large-scale serviceability is limited until privacy features are put into effect. Second, additional storage formats could be added. JSON and Turtle are supported by FHIR in addition to XML, and their inclusion would improve the versatility of FHIRWEAVR. Finally, the API could be extended to a greater number of

AVR devices. A set of scripts between the API and hardware levels could be introduced to differentiate outputs from different devices and map them to functions in `DataHandler`. This approach ensures compliance with the open/closed principle.

7.4 Conclusion

Active virtual reality is a novel approach to physiotherapy, offering considerable potential to improve both treatment outcomes and patient experiences. This project drew on the successes, and learned from the failures, of previous work in the field to create an open, standardised virtual reality input API for medical applications. A demo game built using the API attests to its serviceability, reinforced by encouraging results from testing by developers and end users.

FHIRWEAVR is the first project to integrate the emerging FHIR specification with virtual reality. The API eases the transition of an innovative, useful technology into healthcare, and can improve the homogeneity of future studies in AVR rehabilitation. With continued work, FHIRWEAVR could play an important role in improving and developing the practice of physiotherapy.

Bibliography

[1] W. Müller-Wittig, "Virtual reality in medicine," in *Springer Handbook of Medical Technology*. Springer, 2011, pp. 1167–1186.

[2] F. Rose, E. Attree, B. Brooks, and D. Johnson, "Virtual environments in brain damage rehabilitation: A rationale from basic neuroscience." 1998.

[3] D. Tinson, "How stroke patients spend their days: an observational study of the treatment regime offered to patients in hospital with movement disorders following stroke," *International Disability Studies*, vol. 11, no. 1, pp. 45–49, 1989.

[4] D. Johnson, E. Rose, S. Rushton, B. Pentland, and E. Attree, "Virtual reality: a new prosthesis for brain injury rehabilitation," *Scottish Medical Journal*, vol. 43, no. 3, pp. 81–83, 1998.

[5] C. Bryanton, J. Bosse, M. Brien, J. Mclean, A. McCormick, and H. Sveistrup, "Feasibility, motivation, and selective motor control: virtual reality compared to conventional home exercise in children with cerebral palsy," *Cyberpsychology & behavior*, vol. 9, no. 2, pp. 123–128, 2006.

[6] N. Zeng, Z. Pope, and Z. Gao, "Acute effect of virtual reality exercise bike games on college students' physiological and psychological outcomes," *Cyberpsychology, Behavior, and Social Networking*, vol. 20, no. 7, pp. 453–457, 2017.

[7] J. J. Annesi and J. Mazas, "Effects of virtual reality-enhanced exercise equipment on adherence and exercise-induced feeling states," *Perceptual and motor skills*, vol. 85, no. 3, pp. 835–844, 1997.

[8] D. Cunningham and M. Krishack, "Virtual reality: a wholistic approach to rehabilitation." *Studies in health technology and informatics*, vol. 62, pp. 90–93, 1999.

[9] E. Bisson, B. Contant, H. Sveistrup, and Y. Lajoie, "Balance training for elderly: comparison between virtual reality and visual biofeedback," in *JOURNAL OF AGING AND PHYSICAL ACTIVITY*, vol. 12, no. 3. HUMAN KINETICS PUBL INC 1607 N MARKET ST, CHAMPAIGN, IL 61820-2200 USA, 2004, pp. 337–337.

[10] N. G. Kim, C. K. Yoo, and J. J. Im, "A new rehabilitation training system for postural balance control using virtual reality technology," *IEEE Transactions on Rehabilitation Engineering*, vol. 7, no. 4, pp. 482–485, 1999.

[11] L. Inness and J. Howe, "The community balance and mobility scale (cb&m)–an overview of its development and measurement properties," *Synapse*, vol. 22, pp. 2–6, 2002.

[12] K. Laver, S. George, S. Thomas, J. Deutsch, and M. Crotty, "Cochrane review: virtual reality for stroke rehabilitation." *European journal of physical and rehabilitation medicine*, vol. 48, no. 3, pp. 523–530, 2012.

[13] M. da Silva Cameirão, S. Bermúdez i Badia, E. Duarte, and P. F. Verschure, "Virtual reality based rehabilitation speeds up functional recovery of the upper extremities after stroke: a randomized controlled pilot study in the acute phase of stroke using the rehabilitation gaming system," *Restorative neurology and neuroscience*, vol. 29, no. 5, pp. 287–298, 2011.

[14] B. Najafi, "Gamification of exercise and its application for fall prevention among patients with diabetes and peripheral neuropathy," in *Qatar Foundation Annual Research Conference*, no. 2013, 2013, pp. BIOP–0109.

[15] M. Yates, A. Kelemen, and C. Sik Lanyi, "Virtual reality gaming in the rehabilitation of the upper extremities post-stroke," *Brain injury*, vol. 30, no. 7, pp. 855–863, 2016.

[16] M. K. Holden and T. Dyar, "Virtual environment training-a new tool for neurorehabilitation?" *Neurology Report*, vol. 26, no. 2, pp. 62–71, 2002.

[17] J. Fung, C. L. Richards, F. Malouin, B. J. McFadyen, and A. Lamontagne, "A treadmill and motion coupled virtual reality system for gait training post-stroke," *CyberPsychology & behavior*, vol. 9, no. 2, pp. 157–162, 2006.

[18] J. Fung and C. F. Perez, "Sensorimotor enhancement with a mixed reality system for balance and mobility rehabilitation," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE.* IEEE, 2011, pp. 6753–6757.

[19] R. Kizony, N. Katz *et al.*, "Adapting an immersive virtual reality system for rehabilitation," *Computer Animation and Virtual Worlds*, vol. 14, no. 5, pp. 261–268, 2003.

[20] E. Steele, K. Grimmer, B. Thomas, B. Mulley, I. Fulton, and H. Hoffman, "Virtual reality as a pediatric pain modulation technique: a case study," *Cyberpsychology & Behavior*, vol. 6, no. 6, pp. 633–638, 2003.

[21] Q. Zhao, "A survey on virtual reality," *Science in China Series F: Information Sciences*, vol. 52, no. 3, pp. 348–400, 2009.

[22] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, and J. C. Hart, "The cave: audio visual experience automatic virtual environment," *Communications of the ACM*, vol. 35, no. 6, pp. 64–73, 1992.

[23] J. J. LaViola Jr, "A discussion of cybersickness in virtual environments," *ACM SIGCHI Bulletin*, vol. 32, no. 1, pp. 47–56, 2000.

[24] J. Hamari and J. Koivisto, "Why do people use gamification services?" *International Journal of Information Management*, vol. 35, no. 4, pp. 419–431, 2015.

[25] T. Kari, J. Piippo, L. Frank, M. Makkonen, and P. Moilanen, "To gamify or not to gamify?: Gamification in exercise applications and its role in impacting exercise motivation," *BLED 2016: Proceedings of the 29th Bled eConference" Digital Economy", ISBN 978-961-232-287-8*, 2016.

[26] J. Sween, S. F. Wallington, V. Sheppard, T. Taylor, A. A. Llanos, and L. L. Adams-Campbell, "The role of exergaming in improving physical activity: a review," *Journal of Physical Activity and Health*, vol. 11, no. 4, pp. 864–870, 2014.

[27] S. R. Siegel, B. L. Haddock, A. M. Dubois, and L. D. Wilkin, "Active video/arcade games (exergaming) and energy expenditure in college students," *International journal of exercise science*, vol. 2, no. 3, p. 165, 2009.

[28] A. Amresh and R. Salla, "Towards a home-based virtual reality game system to promote exercise," in *Proceedings of the 50th Hawaii International Conference on System Sciences*, 2017.

[29] S.-Y. Huang, J.-P. Yu, Y.-K. Wang, and J.-W. Liu, "Designing an exergaming system for exercise bikes using kinect sensors and google earth," *Multimedia Tools and Applications*, vol. 76, no. 10, pp. 12 281–12 314, 2017.

[30] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on.* Ieee, 2011, pp. 1297–1304.

[31] J. E. Cromie, V. J. Robertson, and M. O. Best, "Occupational health and safety in physiotherapy: guidelines for practice," *Australian Journal of Physiotherapy*, vol. 47, no. 1, pp. 43–51, 2001.

[32] I. C. Swinkels, C. H. van den Ende, W. van den Bosch, J. Dekker, and R. H. Wimmers, "Physiotherapy management of low back pain: does practice match the dutch guidelines?" *Australian Journal of Physiotherapy*, vol. 51, no. 1, pp. 35–41, 2005.

[33] G. Postolache, P. S. Girao, and O. Postolache, "Applying smartphone apps to drive greater patient engagement in personalized physiotherapy," in *Medical Measurements and Applications (MeMeA), 2014 IEEE International Symposium on.* IEEE, 2014, pp. 1–6.

[34] W. J. Farr, M. Poole, C. Thursfield, and I. Male, "Acceptance of a neuropaediatric exergame rehabilitation system with severe cerebral palsy," in *Extended Abstracts Publication of the Annual Symposium on Computer-Human Interaction in Play.* ACM, 2017, pp. 255–260.

[35] I. Moldovan, L. Tric, R. Ursu, A. Podar, A. Călin, A. Cantea, L. Dascălu, and C. Mihaiu, "Virtual rehabilitation programme using the mira platform, kinect and leap motion sensors in an 81 years old patient with ischemic stroke," in *E-Health and Bioengineering Conference (EHB), 2017.* IEEE, 2017, pp. 325–328.

[36] F. Oemig, B. Blobel, and H. Germany, "Hl7 v2+: The future of hl7 version 2?" *EJBI*, vol. 12, no. 1, 2016.

[37] D. Bender and K. Sartipi, "Hl7 fhir: An agile and restful approach to healthcare information exchange," in *Computer-Based Medical Systems (CBMS), 2013 IEEE 26th International Symposium on.* IEEE, 2013, pp. 326–331.

[38] "HL7 FHIR documentation," accessed April 19, 2018. [Online]. Available: https://www.hl7.org/fhir/

[39] J. C. Mandel, D. A. Kreda, K. D. Mandl, I. S. Kohane, and R. B. Ramoni, "Smart on fhir: a standards-based, interoperable apps platform for electronic health records," *Journal of the American Medical Informatics Association*, vol. 23, no. 5, pp. 899–908, 2016.

[40] J. Fung, F. Malouin, B. McFadyen, F. Comeau, A. Lamontagne, S. Chapdelaine, C. Beaudoin, D. Laurendeau, L. Hughey, and C. Richards, "Locomotor rehabilitation in a complex virtual environment," in *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, vol. 2. IEEE, 2004, pp. 4859–4861.

[41] R. Kizony, P. L. T. Weiss, M. Shahar, and D. Rand, "Theragame: A home based virtual reality rehabilitation system," *International Journal on Disability and Human Development*, vol. 5, no. 3, pp. 265–270, 2006.

[42] M. S. Cameirão, S. B. i Badia, L. Zimmerli, E. D. Oller, and P. F. Verschure, "The rehabilitation gaming system: a virtual reality based system for the evaluation and

rehabilitation of motor deficits," in *Virtual Rehabilitation, 2007*. IEEE, 2007, pp. 29–33.

[43] "VR Health Group," accessed April 22, 2018. [Online]. Available: https://www.vrhealthgroup.com

[44] "Immersive Rehab Ltd." accessed April 22, 2018. [Online]. Available: https://immersiverehab.com

[45] ICAROS, "Image from article: Icaros pro promotional material," 2017, accessed September 27, 2017. [Online]. Available: https://www.icaros.com

[46] W. Wei, "Image from article: Forget the playstation 4 and xbox one, this device actually lets you run around inside video games," 2013, accessed September 28, 2017. [Online]. Available: http://www.businessinsider.com/virtuix-omni-product-test-2013-11

[47] T. Marriott, "Image from article: Virtual reality work-outs have arrived, and they're awesome," 2016, accessed September 27, 2017. [Online]. Available: https://digitalsport.co/virtual-reality-work-outs-have-arrived-and-theyre-awesome

[48] A. Whalley, "Evercycle-a virtual reality exercise game to accelerate exercise performance improvement using the feedforward effect," 2017.

[49] L. A. Shaw, B. C. Wünsche, C. Lutteroth, S. Marks, and R. Callies, "Challenges in virtual reality exergame design," 2015.

[50] S.-U. Kim, K. Lee, J.-H. Cho, K.-C. Koo, and S. B. Kim, "Toward an evaluation model of user experiences on virtual reality indoor bikes," *European Scientific Journal, ESJ*, vol. 13, no. 15, 2017.

[51] M. Gotsis, A. Tasse, M. Swider, V. Lympouridis, I. C. Poulos, A. G. Thin, D. Turpin, D. Tucker, and M. Jordan-Marsh, "Mixed reality game prototypes for upper body exercise and rehabilitation," in *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE*. IEEE, 2012, pp. 181–182.

[52] S. Finkelstein, A. Nickel, Z. Lipps, T. Barnes, Z. Wartell, and E. A. Suma, "Astrojumper: Motivating exercise with an immersive virtual reality exergame," *Presence: Teleoperators and Virtual Environments*, vol. 20, no. 1, pp. 78–92, 2011.

[53] S. Ganesan and L. Anthony, "Using the kinect to encourage older adults to exercise: a prototype," in *CHI'12 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2012, pp. 2297–2302.

[54] J. Bolton, M. Lambert, D. Lirette, and B. Unsworth, "Paperdude: a virtual reality cycling exergame," in *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2014, pp. 475–478.

[55] A. Leibetseder and M. Lux, "Gamifying fitness or fitnessifying games: a comparative study." in *GamifIR@ SIGIR*, 2016, pp. 37–44.

[56] C. S. González González and V. Navarro Adelantado, "Gamification and active games for physical exercise: A review of literature," 2017.

[57] D. Clegg and R. Barker, *Case method fast-track: a RAD approach*. Addison-Wesley Longman Publishing Co., Inc., 1994.

[58] "C# language specification," digital download, accessed April 19, 2018. [Online]. Available: https://www.microsoft.com/en-us/download/details.aspx?id=7029

[59] "Unity documentation," accessed April 19, 2018. [Online]. Available: https://docs.unity3d.com/Manual/index.html

[60] "Oculus developer documentation," accessed April 19, 2018. [Online]. Available: https://developer.oculus.com/documentation/

[61] "VirZOOM support documentation," accessed April 19, 2018. [Online]. Available: https://virzoom.atlassian.net/wiki/spaces/help/overview

[62] J. Brewton, X. Yuan, and F. Akowuah, "Xml in health information systems," in *Proceedings of the International Conference on Bioinformatics & Computational Biology (BIOCOMP)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2012, p. 387.

[63] "HL7 FHIR documentation: DeviceMetric resource," accessed April 19, 2018. [Online]. Available: https://www.hl7.org/fhir/devicemetric.html

[64] S. Schach, "Object-oriented classical software engineering, book," 2007.

[65] H. Sveistrup, "Motor rehabilitation using virtual reality," *Journal of neuroengineering and rehabilitation*, vol. 1, no. 1, p. 10, 2004.

[66] J. R. Bruun-Pedersen, K. S. Pedersen, S. Serafin, and L. B. Kofoed, "Augmented exercise biking with virtual environments for elderly users: a preliminary study for retirement home physical therapy," in *Virtual and Augmented Assistive Technology (VAAT), 2014 2nd Workshop on*. IEEE, 2014, pp. 23–27.

[67] "8-Bit Sounds Free Package," accessed December 2, 2017. [Online]. Available: http://electrodynamicsproductions.blogspot.co.uk/2012/07/added-free-pack-for-8bit-style-sound.html

[68] "Grass Road Race 3D models Unity asset," accessed December 2, 2017. [Online]. Available: http://simpleassets.blogspot.co.uk/2015/10/grass-road-race-asset-youtube-httpswww.html

[69] "Unity Test Runner," accessed April 20, 2018. [Online]. Available: https://docs.unity3d.com/Manual/testing-editortestsrunner.html

[70] "Post Test Server V2," accessed April 22, 2018. [Online]. Available: http://ptsv2.com

[71] A. Steed, "Participant information sheet and feedback forms for COMPGV07 virtual reality interaction tests," 2017.

Appendix A

Gem Class

During system testing, an external developer created a new object type, a `Gem`. The developer wrote a class for gems that specified properties of `EnvironmentTranslator`, successfully adding it as an object in the procedural generation algorithm. That class is presented below.

```
1  using UnityEngine;
2
3  public class Gem : MonoBehaviour
4  {
5      EnvironmentTranslator et;
6      bool gemTrigger;
7
8      void Start()
9      {
10         et = gameObject.AddComponent<EnvironmentTranslator>();
11         et.Limit = 76.2;
12         et.ZSpread = 5;
13         et.XMin = -10;
14         et.XMax = 10;
15     }
16
17     void Update()
18     {
19         if (transform.position.z <= -3)
20         {
21             gemTrigger = false;
22         }
23     }
24
25     void OnCollisionEnter(Collision collision)
26     {
27         if (collision.gameObject.name == "Camera" && gemTrigger == false)
28         {
29             gemTrigger = true;
```

```
30              Player.score += 10;
31          }
32      }
33  }
```