Informatica — Universiteit van Amsterdam

Bachelor Informatica

University of Amsterdam

Virtual reality exposure therapy for treatment of selective mutism in children

Younes Hamdoud

June 28, 2021

Supervisor(s): Dr. Robert Belleman

Signed:

Abstract

Selective mutism is a rare anxiety disorder where a person does not speak in certain situations. Exposure therapy has proven to be an effective psychological treatment, but generally involves intensive *in vivo* exposure supervised by a therapist. This research investigates whether virtual reality exposure therapy can make treatment of selective mutism in children more accessible. Studies on the adoption of virtual reality treatment in related disorders exists, but our intended use is novel. Findings are based on the development of an experimental application, qualitatively evaluated by domain experts. The application is based on an existing protocol, which revolves around the school environment. We present a robust approach for systematically implementing this protocol using reusable components, driven by intuitive interaction methods, and guided by a virtual teacher. We aim to empower therapists by supporting the ability to practice freely in a tailored virtual environment, while relaying its state through a dashboard for monitoring. In conclusion, the application was positively received by the interviewed domain experts, who regarded it as a sufficient foundation for further research and development.

Acknowledgements

I would like to sincerely thank my supervisor Robert Belleman for his guidance. I would also like to thank Els Wippo for her work on the selective mutism treatment protocol on which this thesis relies, and for her excellent feedback. In addition, I would like to thank her colleagues Lynn Meester and Emma Verwey at Levvel for their involvement in the evaluation of the VRET application. Many thanks to Jeroen Legerstee from Erasmus MC and Lisbeth Uthens from the University of Amsterdam for their assessments and insights. Finally, thanks should go to Emre Oktay for his contribution to the design of the application, and his work in carrying out the evaluation.

Contents

CHAPTER 1
Introduction

Selective mutism (SM) is an anxiety disorder where a child does not speak in certain situations [4]. SM has low prevalence, but can lead to severe problems in those affected when left untreated.

Extensive research supports cognitive behaviour therapy (CBT) as a psychological treatment for SM. Starting such treatment as early as possible is important, as it can increase effectiveness. The primary incentive of this project is therefore to assess whether virtual reality (VR) can make treatment more accessible. In addition, the expectation is that greater accessibility incentivises more practice, in turn reinforcing treatment.

Psychologists from Levvel, and researchers at Erasmus MC and the University of Amsterdam are currently evaluating the effectiveness of SM treatment protocols based on exposure therapy, a specific form of CBT. Besides research on *in vivo* treatment, there is also interest in emerging technology such as VR. The adoption of VR in healthcare is already widespread; clinical applications similar to the efforts advocated by the aforementioned parties have been studied before [13]. However, the application of virtual reality exposure therapy (VRET) for treating SM in particular is novel.

Several prominent qualities make VRET a subject of interest:

- Treatment is not constrained by location,
- Scheduling can be flexible for unguided treatment,
- Tailoring to patients requires less resources,
- Monitoring and analysis are easier in a controlled environment.

The freedom of location that VR grants is of particular importance in the case of SM, as schools are often the environment where patients are severely affected [4]. Supporting the ability to practice freely in this particular environment under controlled circumstances could greatly empower therapists in their work.

Evidently, VR as a platform has added benefit without modification to the treatment itself. However, the shift to an immersive virtual environment allows for approaches that are infeasible when treating patients *in vivo*. To clarify, measures can be taken in VR that would otherwise be dangerous, impossible, counterproductive or expensive [2]. For example, a therapist can have fine-grained control over the virtual environment and can focus on the interactions of the child, while the application collects metrics and analyses these in real-time.

In this thesis we perform preliminary research to evaluate the applicability of VRET for the treatment of SM. Findings will be based on the development and evaluation of an exploratory VRET application. This application will be based on the first steps of the existing protocol 'Praten op school, een kwestie van doen' (Talking at school, a matter of doing) published by Wippo and Güldner [20]. We refer to this protocol as the '*in vivo* protocol' in the remainder of this document, the abridged VR adaptation will be referred to as the 'SM VRET protocol'.

1.1 Research questions

The main research question of this thesis is as follows:

> RQ1. Can an existing exposure therapy treatment for children with SM be adopted to function in VR?

To answer this question the following has to be determined first:

> RQ1.1. How must the *in vivo* protocol be implemented to function under the constraints of VR?
>
> RQ1.2. How could the SM VRET protocol leverage VR to support the *in vivo* protocol?

1.2 Structure

In chapter 2, we commence with a detailed description of SM and the usage of VRET for similar disorders. This chapter also includes a review on literature related to the concept of 'immersion'. Next, chapter 3 describes the design of the SM VRET protocol. Afterwards, chapter 4 details the choice of platform and the actual implementation of the SM VRET application. Chapter 5 presents the resulting application and its evaluation by domain experts. Lastly, we discuss earlier results and conclude in chapter 6.

CHAPTER 2

Background

2.1 Selective mutism

Selective mutism (SM) is an anxiety disorder in which a young child cannot speak in specific situations [4]. Children suffering from SM might talk in scenarios where they feel at ease, e.g., at home or when talking to parents. While SM can occur in adults and older children, it is an uncommon occurrence.

The DSM-5, an authoritative volume on psychiatric diagnoses, lists the following diagnostic criteria [1, p. 195]:

(A.) Consistent failure to speak in specific social situations in which there is an expectation for speaking (e.g., at school) despite speaking in other situations.

(B.) The disturbance interferes with educational or occupational achievement or with social communication.

(C.) The duration of the disturbance is at least 1 month (not limited to the first month of school).

(D.) The failure to speak is not attributable to a lack of knowledge of, or comfort with, the spoken language required in the social situation.

(E.) The disturbance is not better explained by a communication disorder (e.g., childhood-onset fluency disorder) and does not occur exclusively during the course of autism spectrum disorder, schizophrenia, or another psychotic disorder.

Research has shown that SM cannot be linked to either specific traumatic events, psychological conflicts or a dysfunctional family in the general case [4]. A specific cause is yet to be determined, but the current consensus regards SM as a result of untreated social anxiety.

Younger children are more receptive to treatment and have a better overall prognosis [4]. While some studies have shown that symptoms can become less severe as children get older, they warn that other anxiety disorders can become more prominent in these cases. These findings further solidify our goal of making treatment more accessible.

2.1.1 Treatment

SM mutism is classified as an anxiety disorder. This leads researchers to consider treatment techniques that have been proven to work for related disorders. Indeed, a literature review of 23 studies exploring treatment of SM concludes that existing research provides supports for cognitive behaviour therapy (CBT) [6]. In addition, the review determined the following treatment techniques to be especially effective: shaping, fading, contingency management, socials skills training, and self-modelling.

The earlier discussed *in vivo* protocol (see chapter 1) can be categorised as exposure-based, a specific form of CBT [20]. The rationale behind this type of therapy is to apply exposure

Table 2.1: A breakdown of therapeutical techniques commonly applied in CBT. This subset forms the core behind the *in vivo* protocol.

Technique	Description
Shaping	Desired behaviour that is demonstrated is reinforced, while undesired behaviour is ignored.
Fading	The environment where desired behaviour is demonstrated is gradually changed without loss of the desired behaviour.
Modelling	Desired behaviour is demonstrated by a therapist; the child learns the behaviour by observation.

to lessen avoidance behaviour in anxious patients. Gradual exposure therapy helps individuals overcome their fears by progressively exposing them to problematic stimuli. The *in vivo* protocol integrates techniques that are believed to be effective for treating those suffering from SM. This includes shaping, fading and modelling. Table 2.1 elaborates on these techniques.

The *in vivo* protocol is designed for children between the ages of 3 and 12. The concrete objective of the treatment is to get children talking in their school environment. Treatment is considered successful when a child is able to speak one-to-one with peers and teachers, and able to participate in group discussions.

The protocol is divided in several steps, often taking the form of playful games. Gamification lowers the barrier of entry, which is important considering the age and condition of the patients. For a complete description we refer to the original publication [20]. However, for our purposes the general progression is of interest. The first steps of the treatment encourage children to make audible noises. These steps can broadly be described as blowing games. From there on, patients slowly progress into making sounds. Eventually, the child will start uttering words and phrases.

2.2 Virtual reality exposure therapy

Powers and Emmelkamp [13] present a meta-analysis on the viability of VRET for anxiety disorders. They conclude that VRET is slightly but significantly more effective than *in vivo* therapy and call for broader application in clinical practice. Furthermore, they underline the granularity that VRET grants when dynamically generating scenarios during exposure therapy.

Boon and Fung [3] briefly describe a VRET trial for children with SM. Their exposure-based protocol required children to interact in classroom scenarios via positively reinforced gestures or speech. Their evaluation of this protocol shows promising results. Furthermore, they list several challenges related to general development of serious games. Foremost is balancing the requirements of the child and the serious objective of the game. In their case the requirements of the patient were amusement, curiosity, challenge, and socialisation.

Lindner et al. [9] include important technical considerations when designing VRETs. First off is the target platform They group VR hardware into three categories:

- *Stationary* headsets, like the Oculus Rift, are tethered to a computer and rely on the computer to render imagery,
- *Mobile* headsets, like the Samsung Gear VR, require a smartphone to function.
- *Standalone* headsets, like the Oculus Quest series, are capable of generating their own imagery.

These platforms can be ranked using several criteria, like the graphical fidelity that can be supported by their hardware. Generally, stationary headsets perform best, while mobile headsets perform the worst [9].

Aside from the visual fidelity that these platforms provide they also differ in input capabilities. Mobile headsets are often limited to three degrees of rotational movement while stationary and standalone headsets can track an additional three degrees of translation movement. Extra tracking capabilities allows for a better freedom of movement in the VRET application.

Table 2.2: An overview of design techniques that can used in the creation of VRET applications.

Technique	Description
Hierarchical progression	An initial fear hierarchy is created to guide a gradual progression of stimuli.
Monitoring	Patients are monitored and safety behaviours (avoiding stimuli) are corrected by therapists.
Psychoeducation	Information and support is provided to a patient to better understand and cope with illness.
Gamification	Addition of game elements such as level-style progression, points, rewards, goals, and feedback.

Lindner et al. [9] also list several design techniques that can be used in a VRET application. These techniques can be exploited in VR and are listed in table 2.2. For example, psychoeducational content can be enhanced by using engrossing VR material like diagrams, interactive media and a virtual therapist. Monitoring can be aided by tracking the patient directly in VR. Finally, hierarchical progression can conveniently be tailored, since we have fine-grained control over the virtual environment.

Gamification can play a particularly important role. The primary goal of this technique is to keep patients motivated by making the experience engaging and enjoying [9]. Open-ended scenarios that can be played regularly could motivate patients to practice more often. In addition, gamification can be used to make VRET less distressing. In spite of these benefits, the adoption of gamification should never distract from the treatment itself.

2.3 Immersion

In this section we discuss a subset of important VR concepts introduced by scientific literature over the past decades. These concepts are useful for exploring and ratifying the design of the SM VRET application.

VRET has to trigger anxiety in the patient to be effective. To that extent, the patient has to suspend the belief that they are in a simulated situation. Reviews on the topic of VRET indeed underline the importance of creating an ‘immersive’ experience. However, immersion and related terms like presence and embodiment have a precise meaning in the context of VR research. Moreover, researchers might disagree on their exact definition.

Slater [15] stresses the contrast between immersion and presence in his work. He defines immersion as the objective level of sensory fidelity provided by a system. Presence on the other hand, is a subjective response of a user to a given level of immersion. In the context of VR, there are typically only two competing systems: the real world (our ground truth) and the artificial VR system. Often, presence is implicitly defined as the response to the artificial system. For example, an artificial system evoking the feeling of ‘being there’ is considered a sign of presence.

Slater [16] further extends the presence formalism using the terms 'place illusion' and 'plausibility illusion'. Place illusion is the sensation of being in a real place, while plausibility illusion is the belief that a situation is actually occurring. For example, seeing an arm in the expected position when using VR induces place illusion [14]. Observing movements of the virtual arm corresponding to your real arm induces plausibility illusion.

2.3.1 Uncanny valley

In 1970, Masahiro Mori hypothesized that robots will appear more familiar when they seem more human, until a point of close resemblance is reached where subtle deviations evoke an uneasy response [11]. Mori identified two peaks of familiarity, as shown in figure 2.1, but cautioned that designers should not make the second peak their goal. Rather, they should only aspire for the level of human likeness where the first peak occurs to avoid the 'uncanny valley' between the peaks.

The uncanny valley does not only affect the design of robots, the phenomenon can also be applied to computer-generated imagery (CGI). In the case of VR, the amount of scrutiny that virtual humans receive has increased. While before CGI humans were displayed on a screen in front of us, now they are met face-to-face.

The existence of the uncanny valley has been questioned before. Notably for our case is the rebuttal that it is learned behaviour. A recent study has shown that younger children do not judge human-like robots as more unsettling than machine-like robots [5]. Furthermore, development of the behaviour was linked to the children's perceptions of the mental capacities of the robot. Younger children preferred a robot when they believed it had a human-like mind, while older children showed the opposite reaction.

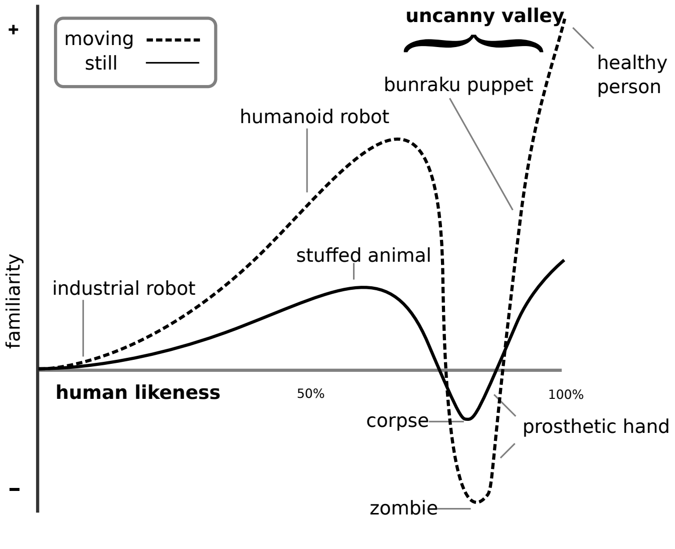


Figure 2.1: Familiarity increases with human likeness until a point is reached at which deviation from human likeness create an unnerving effect: the uncanny valley [11]. The familiarity of moving characters is also plotted.

CHAPTER 3

Design

In this chapter, we cover the design of the SM VRET protocol, starting with a brief outline. The protocol consists of an onboarding process and three treatment steps: blowing, vocalisation, and verbalisation. Each treatment step is characterised by the mode of communication that is tested, and is modelled after one or more steps from the *in vivo* protocol.

The first part of the SM VRET protocol invites the child to blow air from his or her mouth (section 3.3). The proficiency of the child is tested with a game, where a ball must be blown into a goal. The second part requires the child to use vocal communication (section 3.4). Likewise, the progress of the child is tested with a game, in this case the child must mimic animals shown on pictures. Finally, in the third part, the child is required to name several subjects depicted on pictures (section 3.5). The format of the game is similar to the previous part, but the child will now be asked to utter actual words. Figure 3.1 shows a schematic overview of the protocol.

3.1 Environment

In chapter 1, we put forward that VRET is not constrained by location. This section describes how we leverage location freedom as a therapeutical instrument in the SM VRET protocol.

The *in vivo* protocol revolves around the school environment, but one-on-one sessions with a therapist do not take place *in situ* [4]. The child must travel to the therapist to receive treatment instead. When implementing the protocol in VR however, we can transport the child anywhere without being constrained by logistics.

In the SM VRET protocol, the child encounters three different school environments as treatment progresses. First, the child is situated in a playground before entering the school. Afterwards, he or she traverses the corridors before finally reaching the classroom. We elaborate on these different environments in the sections of their corresponding treatment steps.

Changing the environment as treatment advances has two incentives. The creation of narrative progression being the first. To clarify, the ordering of the three environments reflects the schedule of an ordinary school day. Second, the change brings variation in the level of exposure, as we aim to increase the severity of stimuli between consecutive environments.

Aside from applying ‘fading’ between different environments as discussed before, it is also applied in the scope of individual environments. In practice, this involves changing the amount of non-player characters (NPC) and the volume of ambient noise. These dynamics changes can be controlled by the application, but a manual override is also accessible to supervisors.

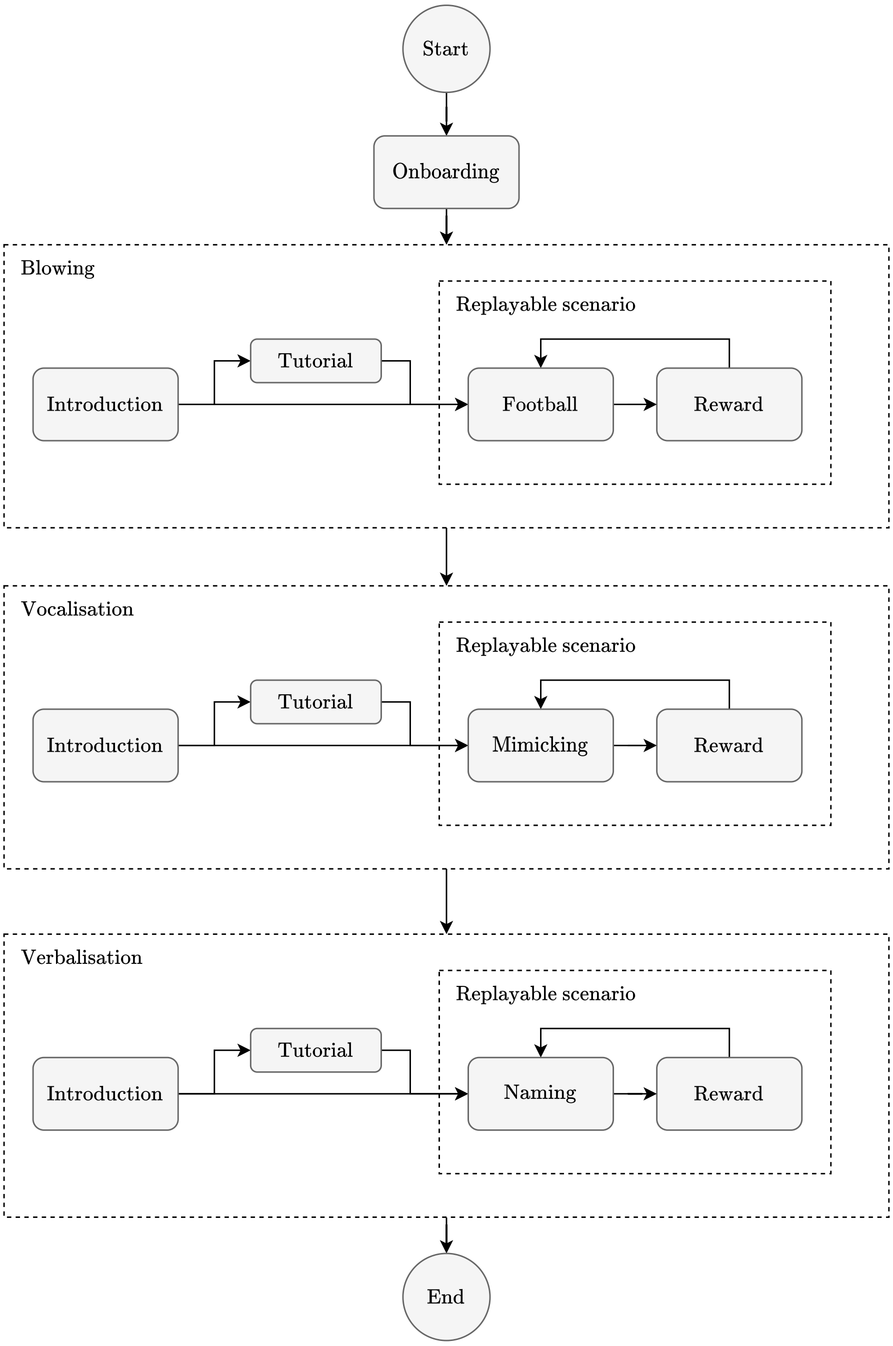


Figure 3.1: A schematic overview of the SM VRET protocol. Similar to the *in vivo* protocol, the SM VRET protocol is subdivided by mode of communication. Note the optional steps and the replayable scenarios in each section.

3.2 Onboarding

The goal of the onboarding section of the SM VRET protocol is to seamlessly prepare the child in a therapeutical and practical sense. When the child has finished, he or she should be comfortable in the virtual environment and have no problem completing the tasks required in the subsequent sections of the protocol.

The onboarding section starts with a psychoeducational introduction by the virtual teacher, as expected in the *in vivo* protocol. The teacher will guide the child throughout the rest of the protocol. The child will progress treatment through his or her interactions with the teacher. It is thus of great importance to keep the child comfortable in the teacher's presence.

After the introduction, the child is instructed to complete two objectives, which require looking and pointing at balloons in the environment. The virtual teacher models the desired behaviour in front of the child.

3.3 Blowing

The goal of the first step of the *in vivo* protocol is to make the child move his mouth, by enticing him or her to blow [4]. Blowing is at the bottom of the fear hierarchy and is thus an apt first step towards the ultimate goal of speech (section 2.1). The SM VRET protocol mirrors this initiation through the implementation of blow football.

The first part of the SM VRET protocol is situated in a playground, as shown in figure 3.2. Several other children walk around the playground, they look at each other and the child while moving. The child can locate several outdoor playsets like a swing and a slide by sight and sound. A blow football table is situated in front of the child.

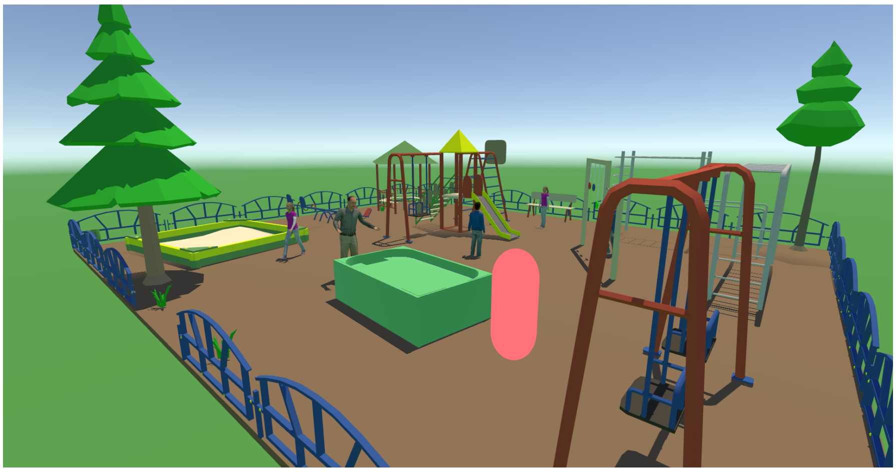


Figure 3.2: The playground environment used in the blowing step of the SM VRET protocol. The red capsule denotes the child's approximate position in the virtual environment.

3.3.1 Football

The goal of blow football is to blow a miniature ball into the goal of the virtual teacher. To accomplish this, the child must move his head and make audible noises. The orientation of the child's head and the volume of the produced noise determine the trajectory of the ball.

Blow football is faithfully adapted from the *in vivo* protocol. The only surface level change is the appearance of the ball; *in vivo* the child blows a cotton ball, to lower the required blowing threshold. In VR, we can use a real ball as the appearance of an object is not necessarily linked to its physical properties.

Before the game commences, the virtual teacher verbally explains the rules. The virtual teacher invites the child to the game and awaits affirmation. When the child accepts the invitation, the teacher will demonstrate how the ball can be blown into a goal ('modelling', see section 2.1). After the demonstration, the child will be asked to score a goal. On completion, the teacher celebrates the achievement using gestures and speech ('shaping'). These two subsections can be identified as the introduction and the optional tutorial respectively in the blowing section of figure 3.1.

After the child is introduced to blow football, the virtual teacher invites him or her to a competitive match where the first to score three goals wins. Throughout the match, the teacher continues to guide the child by performing the required actions. The teacher congratulates the child after completion, and rewards them with a sticker (section 3.6). When the child has completed the game the teacher asks if he or she want to play again. This created a gamified scenario were the child can continue practising until satisfied.

Several design choices serve to create an immersive experience. For example when the child blows, multiple particles are rendered in relation to the strength of his or her puff. Aside from immersion, this allows the child to gauge the blowing intensity. To make this explicit during blow football a short trail is rendered, whose colour gets more vibrant as the child blows harder ('shaping'). Lastly, the child can stun the teacher by blowing in his or her direction to make scoring easier, further reinforcing 'plausibility illusion'.

3.4 Vocalisation

After blowing games, the *in vivo* protocol progresses by compelling the child to make noises while blowing [4]. Keeping the experimental scope of the application in mind, we do not treat this as a separate mode of communication in our implementation. Instead, we immediately progress to the vocalisation step, which succeeds 'blowing with noises' in the *in vivo* protocol.

We define vocalisation as communicating through utterances that are not words. For this treatment step, we retain the overall structure (introduction → (tutorial →) game) of the previous step as exemplified by figure 3.1.

This part of the SM VRET protocol is situated in the school corridor shown in figure 3.3. The child can observe several other children walking around in a classroom through a window in front of them. The virtual teacher is situated in front of the window. The child can faintly here hallway and classroom noises in the corridor.

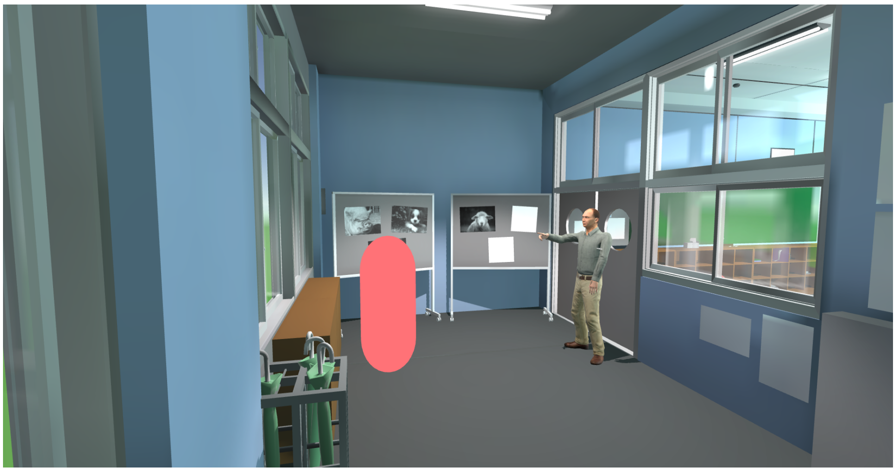


Figure 3.3: The corridor environment used in the vocalisation step of the SM VRET protocol. The red capsule denotes the child's approximate position in the virtual environment.

3.4.1 Mimicking game

The child must identify the animal on a picture and mimic the sound that the animal makes. Before the game starts, the child sees several black and white pictures of animals hanging on the wall. Like before, the virtual teacher invites the child to play the game. When the game starts, one of the pictures fades from black and white to colour. The teacher points to the picture and names the animal. Hereafter, the teacher invites the child to make a noise that sounds like the animal. If the child makes the correct sound the teacher encourages them ('shaping'). The game continues like described until every picture has regained its colour.

3.5 Verbalisation

The final part of the protocol is based around verbal communication. This part takes place in a classroom environment shown in figure 3.4. The child is located in the back of the classroom facing a window. To the right are several empty seats, to the left is a chalkboard. The child can now clearly hear the classroom ambience that was faintly audible on the corridor in the second part. Like before, the classroom is populated by several children. They enter the classroom from a door located at the opposite end and walk to vacant desks.

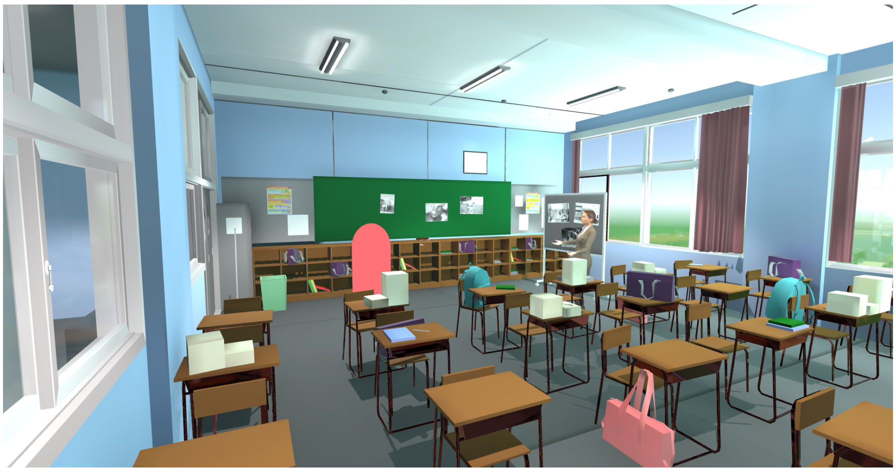


Figure 3.4: The classroom environment used in the verbalisation step of the VRET protocol. The red capsule denotes the child's approximate position in the virtual environment.

3.5.1 Naming game

The naming game is similar to the mimicking game from the previous part. The key difference is that the child does not mimic the animals shown on the picture. Rather, the teacher instructs the child to name the pictured subjects. Like before we make use of 'modelling' to teach the game during the tutorial and to motivate the child during the game.

3.6 Reward system

Providing positive reinforcement through rewards can be a strong therapeutic instrument, as we discussed earlier in section 2.1. Accordingly, the child obtains a reward when he or she complete a section of the therapy as shown in figure 3.1. When a child is eligible he or she gets to choose a single sticker from a set shown to them. The choice of sticker is relayed to a supervisor (section 4.7). The supervisor can use this information to prepare a real sticker, identical to the one chosen in the SM VRET application. In this manner, we strive to achieve a greater sense of immersion by bridging the gap between the real and virtual world.

CHAPTER 4

Implementation

This chapter discusses the implementation of the SM VRET application. Section 4.1 discusses the hardware and software platforms upon which the application is build, explaining the rationale behind our choices, and how these affect our implementation. The remainder of this chapter is dedicated to the analysis of the actual implementation, which consists of five major components:

1. Event system (section 4.2),
2. Player interaction (section 4.3),
3. Environment (section 4.4),
4. Virtual teacher (section 4.5),
5. Non-player characters (section 4.6).

When these components are consolidated we should be left with a system that is sufficiently expressive to support the SM VRET protocol, as laid out in chapter 3. The inner workings of each component are addressed in their respective section. Nonetheless, we continue with a brief outline of each component and their relation to one another.

The SM VRET protocol has a coherent structure, as is evident by its depiction in figure 3.1. The event system seeks to capture this structure in a programmatic manner. It drives the progression of treatment, relying on the other components for its perception of the virtual environment.

The systems that make up the player interaction component translate actions of the child from the real to the virtual world. For example, detecting blowing, and head gestures. The respective systems for each are responsible for both visualising the action in the virtual world and informing the other components when they occur. Visualisation is especially important for achieving ‘plausibility illusion’.

As discussed in section 4.4, the protocol features three different school environments. The implementation of these environment consists of their visual and auditory realisation. Besides this, there are occurrences where changes in the environment are triggered by the other components, which need to be actualised.

Several NPCs are involved in the SM VRET protocol. Foremost is the virtual teacher, who plays an active role. Creating a believable representation of the teacher is of paramount importance, as the child must interact with this representation to progress. Likewise, we must create proper representations for the others NPCs.

4.1 Platform

We build the SM VRET application using an existing game engine. There are several benefits to using commercially available engines as a starting point [9]:

- Control over all aspects of the design while remaining user-friendly,
- Less programming effort compared to starting from scratch,
- Ability to target multiple platforms.

Many low level problems are abstracted away when using a sufficiently advanced game engine, bringing design and implementation of the application to the forefront. Nonetheless, optimisation and interaction between the subsystems of such an engine still remain as formidable challenges.

We used the Unity game engine as it fits the criteria we listed earlier. First, due to its widespread adoption many resources and assets are available. Second in regard to complexity, Unity's `C#` scripting API combined with its GameObject and Component workflow allows for quick development. Finally, Unity provides excellent support for all major VR platforms.

Unity supports VR and AR through the adoption of the Khronos OpenXR standard or by direct support of vendors, like the Oculus SDK (figure 4.1). This support includes Windows as well as Android, which is of importance as many mobile and standalone headsets run on the latter platform.

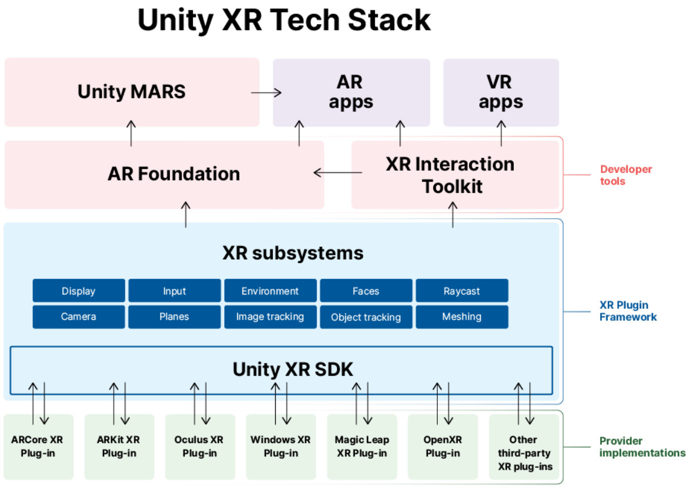


Figure 4.1: Unity's (2020 LTS) XR technology stack.[1]

Unity's XR abstraction allows developers to target every supported platform without writing platform-specific code. Furthermore, it allows interfacing with VR hardware on different abstraction levels. Raw sensor data can be retrieved directly through the XR SDK. While the XR Interaction Toolkit (XRIT), built on top of this SDK, exposes a higher level API.[2] The XRIT workflow revolves around interactor and interactable components, attached to GameObjects. The interaction manager ties these components together delivering a robust system for player interaction out of the box.

For our testing we target the Oculus Quest 2, a standalone headset (section 2.2) running a modified version of Android. As we use Unity, we can easily distribute the SM VRET application to other hardware platforms. Nevertheless, targeting standalone headsets does limit us in other ways. Principally, the processing power and graphical fidelity of our application are limited by the performance of current generation system on a chip hardware, utilised by the majority of standalone headsets.

[1] `https://docs.unity3d.com/Manual/XRPluginArchitecture.html`
[2] `https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@1.0`

4.2 Event system

Treatment can be separated in several sections depending on the context. Figure 3.1 exemplifies how the protocol can be divided by mode of communication, and how these modes can further be subdivided into smaller treatment steps. When we regard the protocol from an implementation perspective, it helps to go one step further. In figure 4.2, we illustrate how one could generally subdivide the introduction step. In this example, the child must look at the virtual teacher to trigger the introduction. When the child successfully completes the task, the teacher will perform an introduction, followed by a question. When the child answers, the introduction is over.

Ideally, we want to split the SM VRET protocol into generic and reusable components like shown in figure 4.2. These components or *events* must serve as the smallest unit for building out the protocol and should encompass a single action that triggers completion. The benefits of an event-driven approach for implementing the SM VRET protocol are the following:

- Reusable units can decrease the overall implementation complexity, decreasing the chance of introducing implementation bugs,
- Events can be managed with a GUI, allowing designers to build out treatment without knowledge of the underlying implementation,
- Generic building blocks allows for fast iteration, a sequence of events can easily be reordered or modified in other ways since they are decoupled from one another.

Our implementation of this system refers to events as 'GameEvents'. Since we only discuss this implementation, we use these two terms interchangeably in this document.

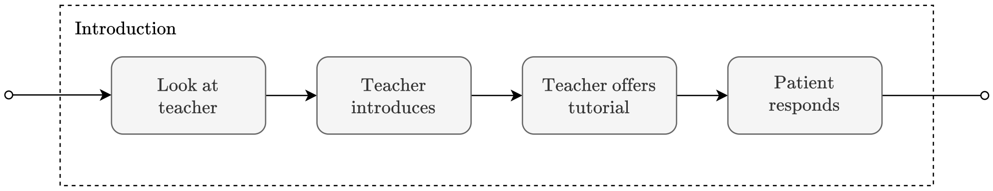


Figure 4.2: A possible interpretation of the introductory section of a treatment step.

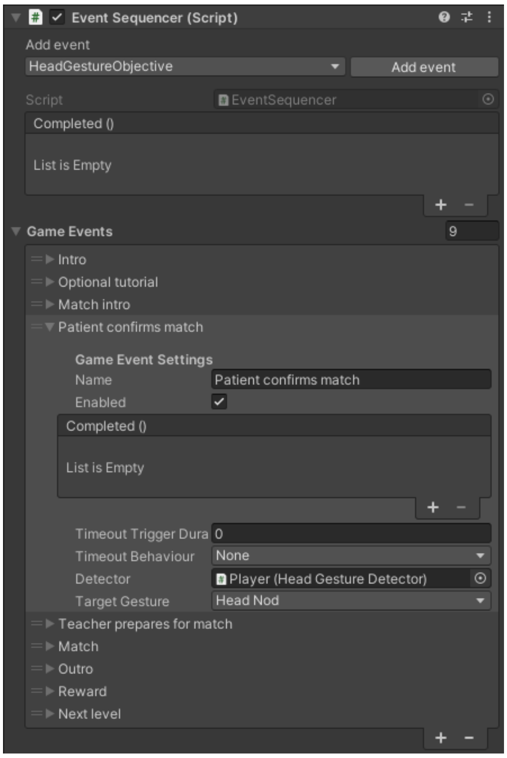
4.2.1 Sequencing

We need a way to combine events to create coherent progression in the SM VRET application. For this purpose we use a linear event sequencer. When an event is completed in such a system, the next event is started and so forth. Linear progression is enough for our purposes as this is the fashion in which the treatment protocol generally progresses. However, there are sections of treatment that do not fit this generalisation. For example, when playing blow football, the child might interrupt the teacher by blowing in his face. Such non-linear sections can occur in the scope of a single event.

The implementation of the sequencer builds upon UnityEvents, Unity's lower level event system. A UnityEvent can decouple components by setting up a persistent callback. To be concrete, a GameEvent can hook into the subsystems of the child or teacher by listening to the UnityEvents invoked by these systems. A GameEvent processes these callbacks and can invoke their own UnityEvent upon completion. The event sequencer can ultimately listen for these completion events and update its state accordingly.

Aside from a way of tracking treatment through events, there are other requirements for the sequencer. Like we mentioned earlier, it should be possible to sequence events through an intuitive graphical interface. Unity's editor can be extended through so-called *Custom Editors*.[3] We use this functionality to create a simple editor that allows users to add different event types, visualise them on a timeline, and reorder them. The sequencer GUI is shown in figure 4.3.

[3] https://docs.unity3d.com/Manual/editor-CustomEditors.html



Event Sequencer (Script)

Add event

HeadGestureObjective

Script: EventSequencer

Completed ()

List is Empty

Game Events: 9

- Intro
- Optional tutorial
- Match intro
- Patient confirms match

Game Event Settings

Name: Patient confirms match

Enabled: ✓

Completed ()

List is Empty

Timeout Trigger Dura: 0

Timeout Behaviour: None

Detector: Player (Head Gesture Detector)

Target Gesture: Head Nod

- Teacher prepares for match
- Match
- Outro
- Reward
- Next level

Figure 4.3: The editor GUI of the event sequencer.

4.2.2 Events and objectives

Our GameEvent implementation is driven by composition and inheritance. We define an abstract `GameEvent` class providing basic functionality and derive specific events from this class. Figure 4.4 shows a simplified class diagram of events. In this section we give a short overview of the shown events, referring to later sections where necessary.

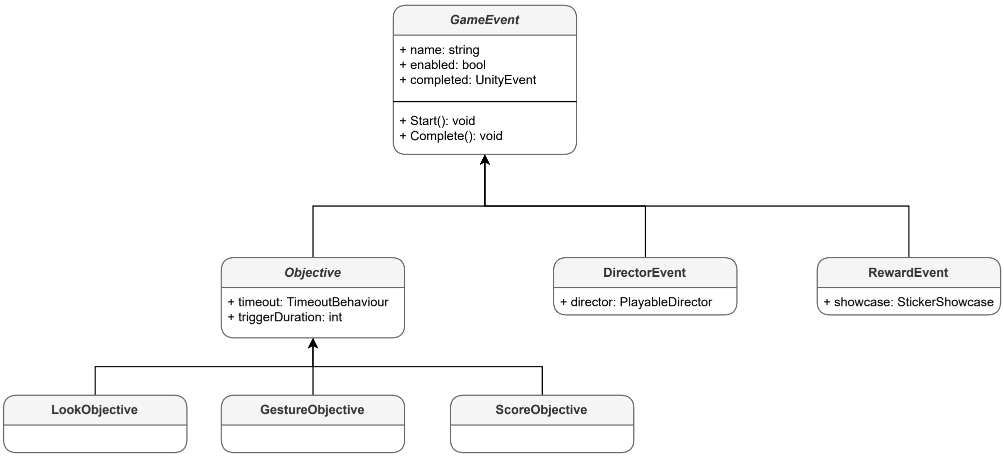


Figure 4.4: A simplified overview of the event class diagram, displayed using a subset of the UML standard.

The abstract `GameEvent` class forms the basis of the event system. It contains a reference to a `UnityEvent` object for notifying listeners when an event is completed.

The abstract `Objective` class inherits from `GameEvent` and provides some basic timeout functionality. For our purposes, objective are most often used to detect child interaction.

The `DirectorEvent` plays the active Timeline on the selected Director in the current scene (section 4.2.3 on the Timeline). The event is completed when the Timeline finishes or when it is pre-empted.

The `LookObjective` hooks into the gaze tracking subsystem (section 4.3.3). The objective can be configured by selecting a target object in the active scene, a minimum look duration, and a maximum look distance. A possible use is making the child look at the teacher, like the first step of figure 4.2.

When the `RewardEvent` is started, the child can choose between several rewards visible on the screen (see chapter 5). The event is completed when a reward is selected.

The `GestureObjective` hooks into the gesture detection system (section 4.3.3). The objective can be configured by selecting a target gesture, like nodding or shaking the head. The objective is completed when the child performs the correct gesture.

The `ScoreObjective` interfaces with the blow football game used during the blowing section of the treatment. The objective can be configured by selecting a goal scorer, either the child or the virtual teacher, and a target score count. The objective is completed when the target score is reached.

4.2.3 Timeline

The `DirectorEvent` is used for playing a sequence of scripted events. This type of event is backed by Unity's built-in Timeline feature, a linear editing tool for sequencing animation, music, and sound or particle effects among others.[4] For example, figure 4.6 shows the Timeline view of the blowing football tutorial section.

Timeline is implemented on top of the Playables API.[5] This API allows for processing and mixing of multiple data sources like animation and audio. When a Timeline starts playing, Playable nodes are created and organised in a tree-like structure. We can extend Timeline by creating custom Playables, allowing us to integrate our own systems. For example, in section 4.5.3 we explain how we integrated the inverse kinematics subsystem of the virtual teacher into the Timeline.

Playables are useful for representing processes on the timeline that happen over a longer timespan. Conversely, there are cases where we want to represent a process that happens instantaneously. In such a situation we can create a custom signal. We can emit a signal from a Timeline track and any GameObjects listening to this signal will be informed. This process is shown in figure 4.5.

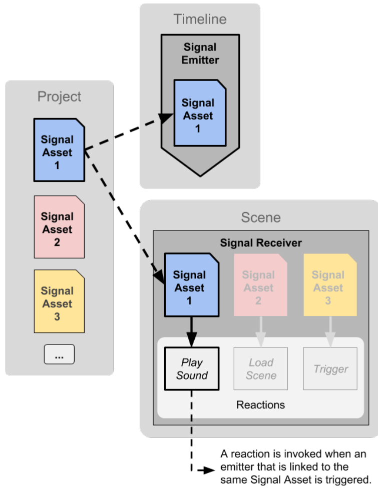


Figure 4.5: A schematic overview on how Timeline signals work.[6] The dotted lines represent signal references. We can create a signal asset and refer to it from the timeline. GameObjects can listen to a signal emission using a signal receiver component and a signal reference.

[4] `https://blog.unity.com/technology/extending-timeline-a-practical-guide`

[5] `https://docs.unity3d.com/Manual/Playables.html`

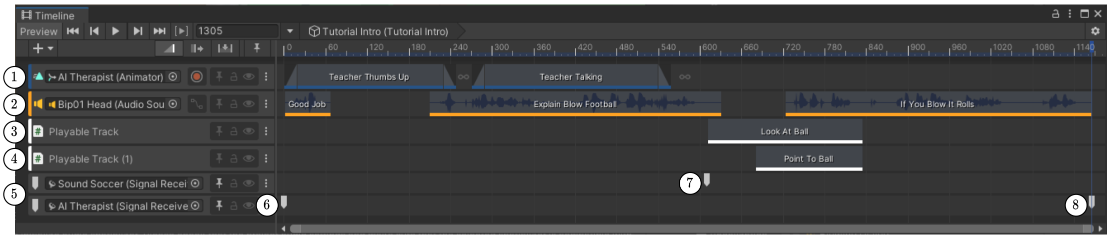


Figure 4.6: The Timeline view of blow football tutorial section. Annotations:

1. Animation track
2. Audio track.
3. Track for custom Playables. Used for IK Playables in this case (section 4.5.3).
4. Second track for custom Playables, used to layer additional clips.
5. Signal tracks.
6. Signals teacher to look at child.
7. Signal to spawn a ball.
8. Signals teacher to start playing blow football in preparation of next event.

4.3 Player interaction

4.3.1 Hand and arm presence

The game should be playable for young children as we noted earlier (section 2.1). As a consequence, we keep the input methods as straightforward as possible. Tracking hand movements of the child using controllers is one of the intuitive methods we opt for.

Like the introduction of this chapter mentioned, visualisation of input is important. We can use inverse kinematics (IK) for this visualisation. IK is the process of calculating the joint parameters needed to place the end of a kinematic chain in a given position and rotation relative to its start [10]. For example, it allows us to determine the orientation of the bones of the arm when we know the position of the hand, the shoulder, and the direction that the elbow should point.

Generally, we only have access to tracking data for the controllers and the HMD. We could use a simple two bone IK system, but this does not include the subtleties of the human arm, like rotation of the upper body when reaching out and twisting of the forearm. To overcome this, we look to specialised IK solvers, specifically the open-source VRArmIK library discussed in a publication by Parger et al. [12]. This solver contains heuristics for elbow positioning based on the shoulder-to-hand distance and avoids unnatural joint limits. For the SM VRET application, we integrated the VRArmIK library with the XRIT.

4.3.2 Blowing detection

When the child plays blow football (section 3.3) we need to determine at what intensity he or she blows, if at all. Specialised hardware exists for this purpose, but it can also be achieved with reasonable accuracy using a conventional microphone [18]. The latter is preferable, as the target hardware platform (section 4.1) features a built-in microphone.

There are downsides to detecting blowing using a conventional microphone. A microphone works by converting air pressure variations of sound waves to electrical signals. When one blows

[6] `https://forum.unity.com/threads/new-in-2019-1-timeline-signals.594142/`

into a microphone the air pressure will drastically fluctuate; much more than if one would speak. These severe fluctuations result in a very distorted signal. As a result, detecting blowing using a microphone effectively comes down to identifying a noisy signal.

Our approach to detecting blowing can be split into two steps. First, we estimate the loudness of the signal. Determining the loudness of the signal helps to discern quiet background noise from the relatively loud blowing noise. We start by sampling a short fragment of audio, we refer to this series as x of length n. We calculate the root-mean-square of the fragment as follows

$$\mathrm{RMS} = \sqrt{\frac{1}{n}\left(x_1^2 + x_2^2 + \ldots + x_n^2\right)}. \tag{4.1}$$

We can also use equation 4.1 to calculate the reference noise level RMS_0. This reference noise level serves for calibration, since the internals of a microphone can introduce a small amount of noise into an otherwise silent recording. Additionally, the acoustics of the room and background noise affect this value. Ideally, one would capture this fragment on site; recording it in a relatively quiet room sufficed in our case.

We can calculate how loud the current input fragment is compared to our reference recording:

$$L = 10 \log_{10}\left(\frac{\mathrm{RMS}^2}{\mathrm{RMS}^2}\right) = 20 \log_{10}\left(\frac{\mathrm{RMS}}{\mathrm{RMS}}\right) \text{ dB}. \tag{4.2}$$

Before using this value for classification, we use a minimum dB level to filter out background noise and set a maximum level according to the desired blowing strength.

We would still like to discern blowing from speaking into the microphone, the second step in our approach deals with this. A relatively fast and simple way of detecting a noisy signal is to determine its flatness in the frequency spectrum. We can calculate the spectral flatness as follows [17]:

$$F = \frac{\exp\left(\frac{1}{n}(\ln x_1 + \ln x_2 + \cdots + \ln x_i)\right)}{\frac{1}{n}(x_1 + x_2 + \cdots + x_i)}. \tag{4.3}$$

Earlier, we characterised blowing as a loud and noisy signal. However, our current implementation is sensitive to other similar noises. To deal with this we use a spring-damper like smoothing function. This function gradually changes the output intensity over time towards the detected blowing intensity. In essence, sustained blowing is detected while other, loud but short-lived, noise is ignored.

4.3.3 Gaze tracking and pointing

To track the child's gaze and to detect what he or she is pointing at, we use 'raycasting'. A raycast sends an imaginary beam along the ray from its origin until it hits a collider in the scene [10]. In Unity, information is returned about the object and the point that was hit in a `RaycastHit` object.[7] Gaze tracking can be implemented by casting a ray from the point between the left and right eyes in the direction that the HMD is facing.

At several points in the treatment the child has to perform a pointing gesture. For example, when selecting a reward the child must point to a sticker. Such a gesture can trivially be implemented using a raycast.

4.3.4 Head gesture recognition

In the SM VRET protocol, the child is required to accept or reject propositions. Since the child might not yet be comfortable with verbal communication, an alternative is important. We opted for head gestures, shaking and nodding in particular.

Head gestures are spatio-temporal patterns. The HMD tracks spatial data over a certain interval. We effectively end up with a time series of points containing data like the position and velocity of the head. Several existing approaches are known for robustly matching spatio-temporal patterns. Most of these revolve around the use of hidden Markov models, as they excel

[7] `https://docs.unity3d.com/ScriptReference/Physics.Raycast.html`

at pattern matching [21]. We settle for a more naive approach, as the window of time in which gestures take place is known, making false positives less of a problem.

Our implementation makes the naive assumption that the angular velocity of the child's head in the plane of motion of a gesture will be of the same magnitude when shaking back and forth. For example, when shaking 'no', the head of the child will turn left as quickly as it turns right. When we track the angular velocity of the head over the duration of this gesture, two things must be true based on our assumption: the minimum yaw rate (angular velocity component around the yaw axis) and the maximum yaw rate are equal and the average yaw rate should be zero. The same is true for the pitch rate when nodding the head.

Listing 1 shows a simplified code example of the detection system.

Listing 1 Simplified code example of head gesture detection.

```
bool DetectShake(IEnumerable<float> speeds) =>
    (
        speeds.Max() >= triggerSpeed &&
        speeds.Min() <= -triggerSpeed &&
        Mathf.Abs(speeds.Average()) < maxAvgSpeed
    );

if (DetectShake(angularVels.Select(v => v.y)))
    // Head shake
else if (DetectShake(angularVels.Select(v => v.x)))
    // Head nod
```

4.4 Environment

In this section, we discuss some technical aspects of creating the environments for the SM VRET application. As we noted in section 4.1, the target hardware platform has limited processing power. Creation of the environments is thus a balancing act between good performance and the fidelity expected of an immersive experience.

4.4.1 Lighting

An important part of immersion is visual fidelity. Creating assets for a virtual world with a consistent art style is mostly in the domain of 3D artists. However, there are also technical aspects to creating a believable environment that do not necessarily require the creation of better assets.

In computer graphics there is a clear distinction between baked and dynamic lighting. A dynamic light casts shadows in real-time, but comes with a high performance cost depending on the amount of moving objects in the scene and the amount of lights.

Baked lighting forgoes the performance costs of dynamic lighting by calculating bounces beforehand and storing these in a lightmap texture. This saves run-time performance, as lighting data can be retrieved from the lightmap. Moreover, baking allows us to perform calculations that are too expensive to perform at run-time, e.g, multiple light bounces and reflections. However, baked lighting only takes static objects in account.

We can use baked lighting to create visually appealing environments. Leaving dynamic lights for certain sections, to ground the child in the virtual environment. For example when the child moves his arms or blows at the ball, shadows cast by the sun or lamps update accordingly.

4.4.2 Spatial audio

By spatialising an audio source, emitted sound can be transformed in such a way that it appears to be positioned at a specific location in the virtual environment. Spatial audio is an important part of creating immersion. VR headsets provide the ability to track a listener's head orientation and position. We can use this information when spatialising.

In the SM VRET application several audio clips are spatialised. Foremost is the dialogue uttered by the virtual teacher. Aside from the aforementioned reasons there is also a therapeutical benefit. The child is encouraged to look in the direction of the teacher, potentially bringing them closer to making eye contact.

Spatialised audio can be drowned out by background noises or sources that are closer to the listener. Aside from this, important sounds can become inaudible when the listener is facing the wrong way. For example, the child might not hear instructions given by the virtual teacher because he or she is afraid to look at them. The SM VRET application solves these issues in two ways. First off, indoor environments model sound reverberation, which makes sounds audible even when the listener is not directly facing the source. Secondly, dialogue is isolated in a separate track to boost its volume. Finally, a linear audio roll-off, which is not true nature, is utilised to make certain sounds more audible at a distance.

4.5 Virtual teacher

The virtual teacher plays a central role in the treatment. The child is exposed to his or her fears through interaction with the teacher. Consequently, we regard the creation of a virtual representation of the teacher that is approachable and lively as essential. In subsection 4.5.1 we discuss the appearance of the teacher, in the remainder we explain how we animate the teacher.

4.5.1 Appearance

The representation of avatars like the teacher are a key contributor to the 'plausibility illusion' of a VR experience [16]. There are a range of alternatives to chose from when representing an avatar, from abstract, cartoon-like to realistic. The choice between these alternatives is not straightforward as we seek to avoid the uncanny valley (section 2.3.1). On the other hand, straying too far from realism might be detrimental to immersion.

One aspect of creating an avatar is choosing its visual appearance. In the SM VRET application we experimented with three different looks, as shown in figure 4.7. Our initial avatar had the appearance of a humanoid robot. The head of the robot contained a screen that played back a recording of a human face. After preliminary feedback, we modified the look of the robot to make it less intimidating, bringing its stature in line with that of the child. Eventually, we settled for human avatars based on supplemental feedback.

The Rocketbox library contains a variety of rigged avatars representing humans of different genders, races, and occupations [8]. The library is freely available for academic purposes and has seen use in research on crowd simulation, real-time avatar embodiment and VR. Aside from the visual fidelity of the library, several other things are of interest:

- The skeletal rigs are compatible with Unity's humanoid animation system and contain bones for facial animation,
- The availability of meshes with multiple details levels allows for seamless blending on low performance hardware,
- The existence of both adult and child avatars,

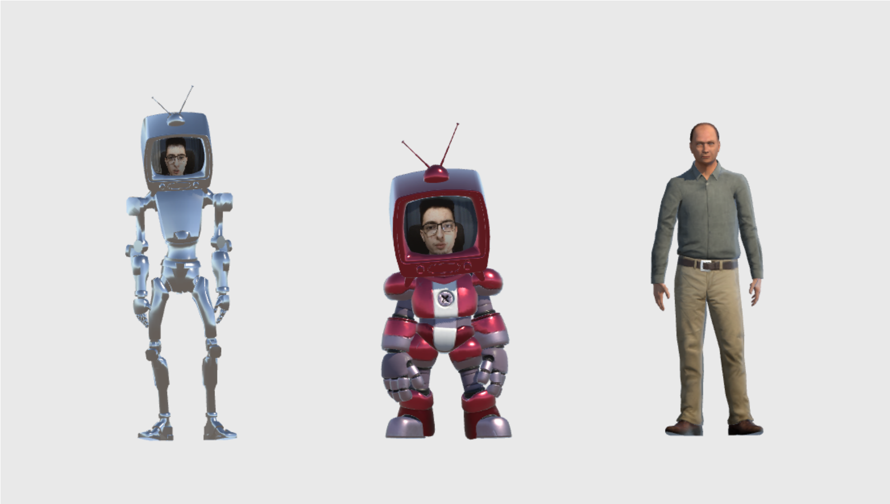


Figure 4.7: The different virtual teacher avatars that we experimented with. From left to right: adult robot with screen head[8], child robot with screen head[9], and human. The robot avatars are modified assets sourced from the Unity Asset Store and the human from.

Screen head

To facilitate the playback of recorded videos on the screen head of the robot we use Unity's `VideoPlayer` component, abstracting its functionality with a custom behaviour script. Usage consists of creating a custom `ScreenClip` asset in the editor. Such an asset contains the actual video data, an identifiable name, and a flag to enable or disable audio. The interface of our behaviour script allows the playback of these clips. To make the system work with our sequencer, we integrate it with the Timeline using a custom Playable behaviour.

The built-in `VideoPlayer` can stutter when switching the video. This is problematic for our purposes as this can break immersion when the child is looking at the face of the teacher. We added double buffering functionality by using two `VideoPlayers`. When we switch to a different video we load it on a separate player. We listen for the event that emits when the engine has loaded the video. When this event is invoked we turn off the playing video player. This introduces a slight delay, but it is less jarring than having the screen flicker when switching clips.

4.5.2 Motion capture

Animating a character by hand takes time and expertise. An alternative to keyframe animation is motion capture. Mixamo provides high quality body motion capture data for free.[10]

The actor performing motion capture does not always match the rig of the animated character. Unity can retarget animation from one humanoid rig to another with different proportions. This works good for real humans, but when using bipedal cartoon characters we need to manually tweak the muscle limits to avoid clipping of extremities.

4.5.3 Inverse kinematics

We can use inverse kinematics to layer secondary movement on top of generic motion capture animation. For example, if we have an animation clip containing a waving gesture we can use IK to turn the upper body to face the child. We extend Unity's IK system to the Timeline by implementing a custom Playable. We can configure this Playable with an IK target, and a target skeleton, often attached to an NPC. Furthermore, we use linear interpolation of the IK target to smooth the movement. For example, the teacher does not snap his head when a different target is selected but slowly rotates it instead.

[8] `https://assetstore.unity.com/packages/3d/characters/robots/space-robot-kyle-4696`

[9] `https://assetstore.unity.com/packages/3d/characters/jammo-character-mix-and-jam-158456`

[10] `https://www.mixamo.com`

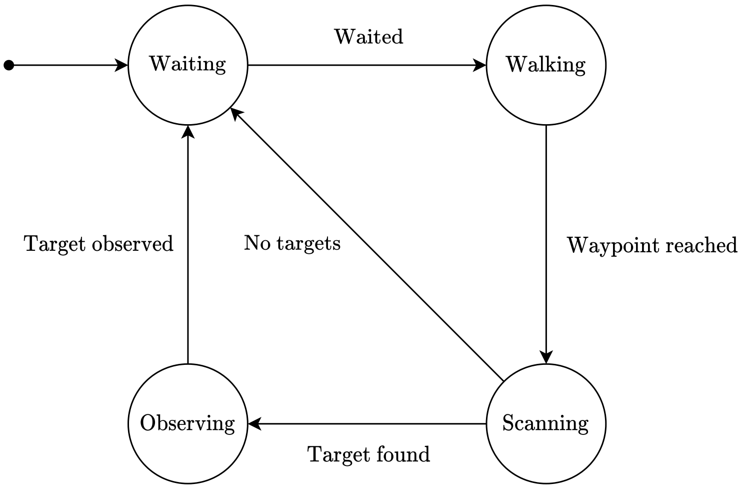


Figure 4.8: A finite state machine that represents the emergent behaviour of the child agents in the SM VRET application.

4.5.4 Procedural facial animation

Procedural animation can be used for realistic facial animations without requiring intensive motion capture. The fidelity of procedural animation depends on the complexity of the techniques that are applied, and the data that is available.

MoveBox is an open source toolbox for targeting motion captured movements onto Rocketbox avatars [7]. The toolbox can make use of consumer hardware that contains depth sensor cameras, like the Microsoft Kinect. It serves as a low-cost alternative to studio motion capture, but also includes basic audio lip sync and blinking functionality.

MoveBox is primarily concerned with authoring animation, this is reflected in its procedural facial animation implementation as well. However, the systems MoveBox provides are optimised for the facial rigs of Rocketbox avatars. We adopted a modified version of the facial animation systems for the SM VRET application. These modifications include making the system useful for playback of animations on Timeline tracks.

4.6 Non-player characters

Alongside the virtual teacher other children appear as non-player characters (NPC) in the SM VRET application. Unlike the teacher, the children do not play an active role in the games. Rather, they are used as a therapeutical tool for 'fading' (section 2.1). The behaviour of NPCs is controlled by a basic controller script. This is in contrast to the virtual teacher whose behaviour is orchestrated through the Timeline, except during games like blow football.

When we discuss the emergent behaviour of the NPC we refer to it as an (intelligent) agent. The behaviour that arises from the agent's scripting can be intuitively represented as a finite state machine, as shown in figure 4.8. When an agent enters the scene he spends an arbitrary amount of time waiting (*Waiting* state in the figure). The upper and lower bound of this waiting time can be controlled from the Unity editor. After the agent has waited, a random location in the scene is chosen from a set of predetermined way-points. The agent proceeds by walking to the chosen location (*Walking* state). When the destination is ultimately reached, the agent scans the surrounding area for points of interests (*Scanning* state). The agent returns to the initial *Waiting* state if no points of interest are found. On the contrary, if scanning was successful the agent will first rotate its body to examine an arbitrary point of interest (*Observing* state) before eventually returning to the *Waiting* state.

Agents perform alternative behaviour in the classroom environment. This behaviour is shown in figure 5.2.

Scanning points of interest

The Unity engine utilises a layer-based collision detection system. We create a distinct layer and attach a collider on this layer to our points of interest. By performing a `SphereCast`, a spherical substitute to a raycast, on this layer we find the position of all points of interest in a given radius.[11]

Navigation meshes

The agent must traverse an environment to reach way-points, while avoiding obstacles like inaccessible terrain, and other agents. We create a navigation mesh to aid the agent in path-finding. A navigation mesh is constructed from level geometry and defines which sections can be traversed. The Unity engine has built-in support for generating navigation meshes.

4.7 Web dashboard

VR applications can easily collect data due to the highly controlled nature of virtual environments, as was mentioned in chapter 1. However, to take advantage of this we need to relay what is happening in the virtual environment to the supervising therapist. For this purpose we implemented the ability to communicate over the internet with a remote server in the SM VRET application. Consequently, the remote server can be used to distribute the information to therapists through an accessible web portal.

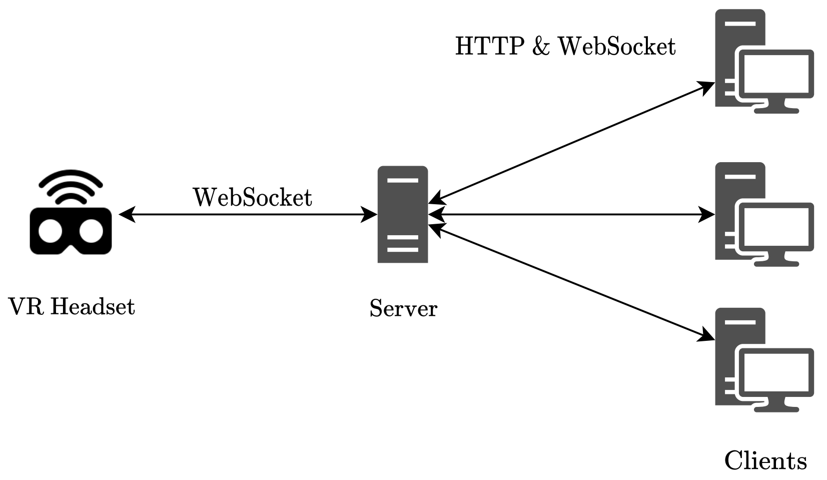


Figure 4.9: Network diagram showing a possible network structure and the protocols that are used to communicate between each node.

Figure 4.9 shows a network diagram of a possible usage scenario. The SM VRET application connects to the server using a WebSocket. We use the WebSocket protocol since it allows for full duplex communication. For a real-time application like ours this is superior to an API or HTTP long polling. Supervisors can access the server through a web dashboard. After establishing an HTTP connection, a WebSocket is opened. The server forwards the messages of the SM VRET application through the WebSocket to the dashboard, and the other way around.

The games played in the vocalisation and verbalisation section require human intervention. These games rely on vocal and speech input from the child. Analysing this input on a standalone or mobile platform can be problematic due to performance constraints. Furthermore, imperfect speech recognition can lead to a frustrating treatment experience. Hindering child progression in this way has detrimental effects on treatment. Future work regarding this issue is discussed in section 6.2.

[11] `https://docs.unity3d.com/ScriptReference/Physics.SphereCast.html`

4.7.1 Protocol

The SM VRET application and the web dashboard clients communicate using a rudimentary JSON protocol. Messages consist of a string `type` field and an optional `content` field. The data type of the `content` field is determined by the message `type`. The following message types can be received by the dashboard (format `(type, content)`):

- `(NextCard, CardContent)`, sent when a new card appears when playing the vocalisation and verbalisation games, `CardContent` lists the response that the supervisor should evaluate;
- `(EventStarted, EventName)`, sent when a new event is started by the child;
- `(EventCompleted, EventName)`, sent when the current event is completed by the child;
- `(StickerClaimed, StickerName)`, sent when the child chooses a sticker.

The following messages can be sent from the dashboard to the SM VRET application:

- `(NextCard, _)`, sent when the supervisor determines that the child made the correct response during the vocalisation and verbalisation game;
- `(AddNPC, _)`, sent when the supervisor wants to spawn a NPC child (section 4.6);
- `(SkipEvent, _)`, sent when the supervisor wants to skip the current event.

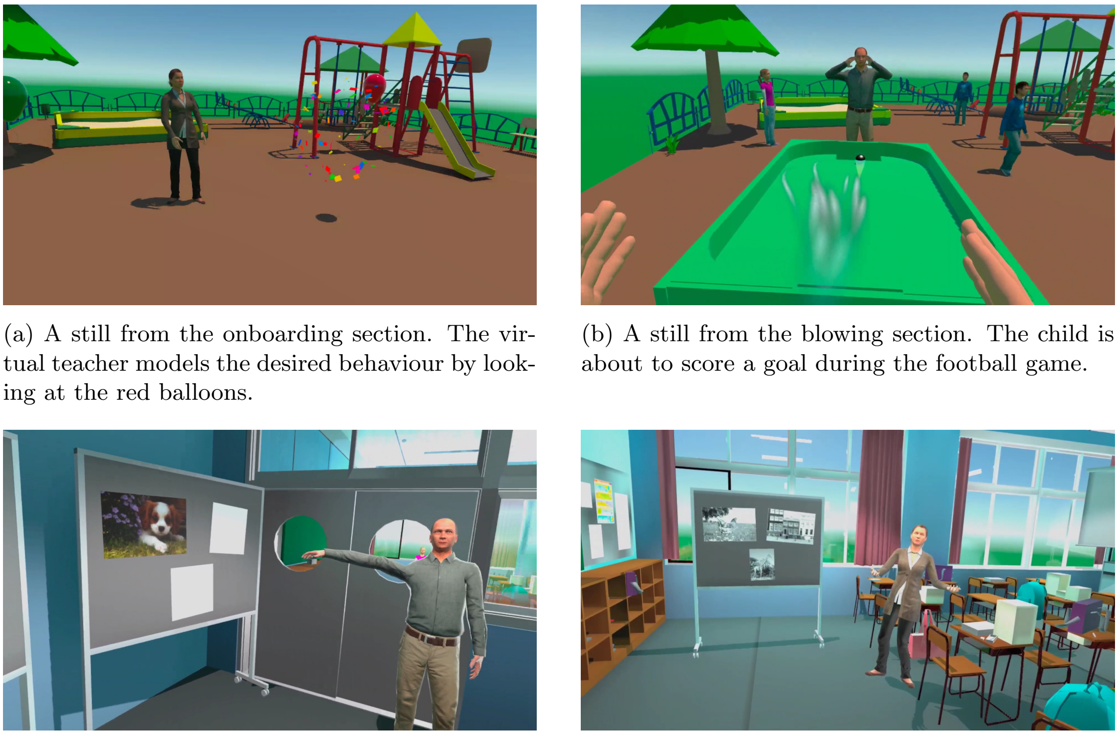
CHAPTER 5

Results

This chapter is dedicated to exhibiting the SM VRET application, and discussing the results of its evaluation.

5.1 Application

Figure 5.1 shows the SM VRET application from the point of view of the child. The figure also links to recording of the application, to give a more complete overview. In addition, figure 5.2 shows the behaviour of an NPC.



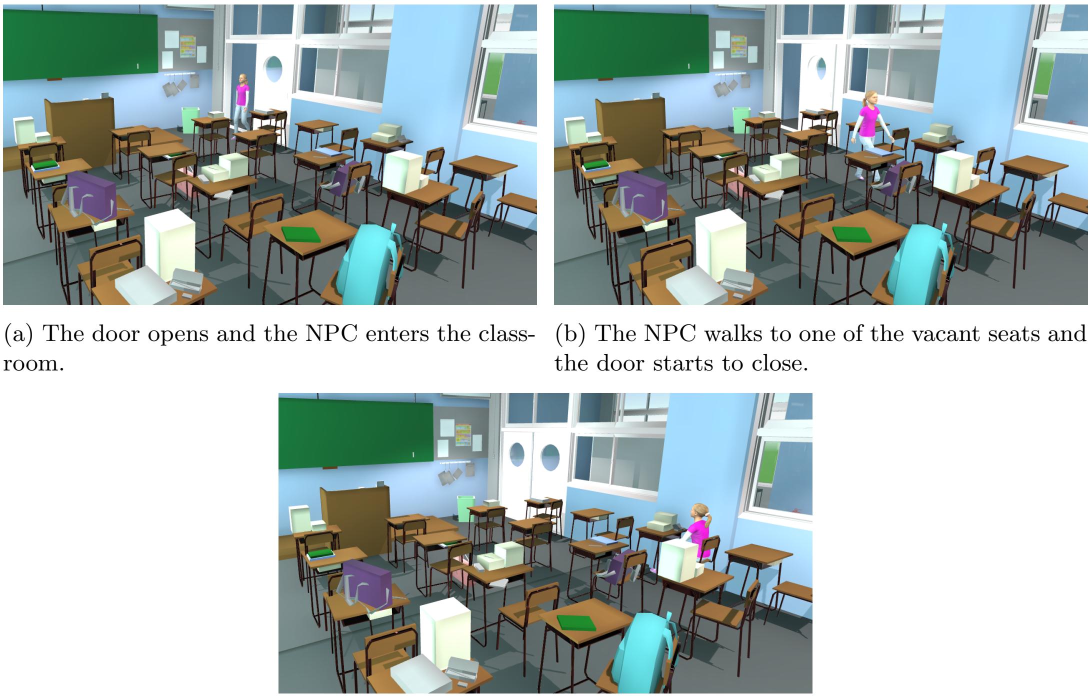
(a) A still from the onboarding section. The virtual teacher models the desired behaviour by looking at the red balloons.

(b) A still from the blowing section. The child is about to score a goal during the football game.

(c) A still from the vocalisation section. The virtual teacher asks the child to mimic a dog.

(d) A still from the verbalisation section. The virtual teachers is in the process of explaining the rules of the game.

Figure 5.1: The four sections of the SM VRET application, as seen from the point of view of a child. In addition, the stills link to a recording of their corresponding section in the digital document. Alternatively, a playlist with all recordings can be accessed using the following link: `youtu.be/playlist?list=PLBA-P-77ZVGabjDEld8zAT9h8Ig-RJNGs`.



(a) The door opens and the NPC enters the classroom.

(b) The NPC walks to one of the vacant seats and the door starts to close.

(c) The NPC is seated and the door is now completely closed.

Figure 5.2: A sequence of images showing the behaviour of NPC children in the classroom environment.

5.2 Evaluation

Besides preliminary feedback obtained during development, the finished VRET application was also evaluated by domain experts. Both researchers and psychologists were invited to a demonstration session, where they tried the application themselves in VR. After completion, they were asked to complete a questionnaire followed by a short interview.

This section is dedicated to discussing a subset of the interviews. We are interested in gaining qualitative insights into the technical implementation of the VRET application. To this extent, we examine the portions of the interview that can be related to the accomplishments and shortfalls of the implementation. Further analysis of the interview and the questionnaire are left to future work. Nevertheless, we start off with some excerpts that contain broader feedback.

The response of the interviewees on whether the SM VRET application could see future development and use:

> *Interviewee 1*: Definitely. I really do hope so. But I think a lot of psychologists and psychiatrists are a bit scared. ... We do have virtual reality tools. ... Nobody's using them. This serves a purpose. Okay, this serves a purpose because you can exercise in the classroom, and we don't have a classroom at home.
>
> *Interviewee 2*: Yes, definitely. It would be nice if there could be a smartphone version with Google Cardboard. Of course with that, some of the interactions with the environment would be impossible. But I think that with this, they can also train at home. But that is for further development.

The interviewed researchers found their expectations met, and in certain aspects exceeded. Considering the exploratory nature of the SM VRET application, they regarded it as a good foundation for further research.

The second interviewee does touch on a particularly important point. We chose to target standalone VR platforms (section 4.1), instead of more accessible mobile platforms, like Google

Cardboard.[1] This choice was based on the better input capabilities and processing power that standalone hardware generally provides. However, future research could look into the possibility of porting parts of the VRET application to less powerful platforms.

The positive aspects of the VRET application according to the interviewees:

> *Interviewee 1*: I appreciate the classroom, et cetera, the surroundings are really real life. Yes, the teachers are a little stiff and automatic, but I can't expect that on this budget to be better. I think the fastness with which you produce this product is higher than my expectations. I appreciate what you did in this short time.
>
> *Interviewee 2*: I think what was done well, was the environment. I think the colours were vibrant and nice for children. Maybe some more detail would be nice. ... It was nice that there was a feedback voice recorded, responding well with what you saw in virtual reality, like the blowing game or like saying the name of the pictures. I think that was very well done. The interaction was quite smooth. There were no time delays.

The interviewees praised the realisation of the virtual environments. Underlining how the availability of convincing school environments in VR can support the *in vivo* protocol. However, they did note that they should be more child-friendly.

The shortcomings of the SM VRET application according to the interviewees:

> *Interviewee 1*: Perhaps if there could be some fine-tuning, it will be better regarding the sound. As a delay between the nodding or the directing with your finger occurs, before the reaction. ... It would be nice if the vision of the pictures on the wall became clearer, or if the teacher is more friendly.
>
> *Interviewee 2*: The introduction of the female character was quite long, and I got distracted. Also, the people were very uncanny valley, and the lip sync, for me, didn't work very well. I couldn't see their mouths moving. ... I was also thinking that the children probably do not have experience with virtual reality, or that the social interactions are difficult for them. So maybe they can play in the playground for a while, alone not with a character, to get accustomed to virtual reality.

The initial response to the virtual teacher was mixed. The second interviewee felt that the virtual teacher was uncanny (section 2.3.1). One of the apparent shortcomings is the mouth animation, which was barely noticeable during certain sections. We argue that this can be partially attributed to the limited resolution of the displays of the VR headset, which make the subtle movements of the mouth difficult to see at a distance. A possible solution is to place the virtual teacher closer to the child. Alternatively, the procedural mouth animation could be exaggerated. However, this might expose the flaws of our current rudimentary approach to animation. In section 6.2, we discuss a more faithful approach that might overcome the current shortcomings.

The interview revealed how the body animations of the virtual teacher seemed awkward at times. This touches upon the downsides of our choice of avatar, as unnatural motion is especially evident when using a human avatar. This issue could be rectified by curating more sources of motion capture, eliminating our reliance on IK techniques. Alternatively, our current IK techniques might be improved by reusing the existing advanced upper body system, as explained in section 6.2.

The first interviewee had issues with using head gestures. Similar instances are visible in the recordings (figure 5.1d at `0:40`). This could be caused by an inability to pre-empt the current event. The event system has rudimentary support for pre-empting `DirectorEvents`. Perhaps a visual indication can be implemented to inform the child exactly when pre-empting is possible. Alternatively, the head gesture detector might fail to detect the gesture. A more robust approach to head gesture detection is described in section 6.2.

[1] `https://arvr.google.com/cardboard/`

Lastly, one of the interviewees regarded the pictures on the board in the classroom as unclear (figure 3.4). These unclear images could be fixed by using a more suited texture filtering technique, like anisotropic filtering, which retains the sharpness of a texture at extreme angles [10]. Alternatively, a simpler art style could be considered for the pictures.

CHAPTER 6

Discussion

In chapter 5, images and footage of the completed SM VRET application were shown. The application is based on the *in vivo* protocol, which revolves around the school environment. Up to this point, we presented a systematic approach to implementing this protocol using an event system and sequencer, driven by intuitive interaction methods and guided by the virtual teacher. This chapter is dedicated to putting the application and its evaluation in a broader perspective.

In general, the interviewed domain experts where impressed with the SM VRET application, considering its short turnaround time and the available resources. They praised the environment and interactions that were showcased. In short, their findings consisted of the following:

- Added benefit through an accurate realisation of the school environment,
- Good interactions with responsive feedback by the virtual teacher and environment,
- Detection of input needs to be more robust,
- Animation of the virtual teacher needs refining.

Overall, the application was regarded as a sufficient foundation for further research and development.

6.1 Conclusion

In conclusion, exposure therapy treatment for children with SM can be adopted to function in VR. We base this on the results of the evaluation of the presented SM VRET application, and the subsequent resolution of the research questions established in section 1.1.

The *in vivo* protocol was implemented to function under the constraints of VR by dividing it into generic and reusable events. These events are sequenced to mirror the progression of the protocol.

Exposing the child to interaction with others forms the foundation of the *in vivo* protocol. For this purpose, we implement a virtual teacher integrated with the event system. The teacher and other NPCs have realistic human avatars, sourced from an existing library. These avatars are animated using motion capture data refined with an IK pass. Additionally, procedural facial animation is used for blinking and speech.

To show how the SM VRET protocol leverages VR, we explain how we exploited four qualities of this platform listed in chapter 1: flexible scheduling, location freedom, tailoring, and monitoring. Scheduling is eased through unguided and gamified sections that can be replayed by the child, like blow football. We leverage location freedom as a clinical tool by creating narrative progression and by varying exposure levels through three different environments. Supervisors can control the amount of NPCs in the environment using an accessible web dashboard, to tailor the experience of the child. In addition, the dashboard allows for monitoring, by relaying the state of the virtual environment. Allowing supervisors to prepare real rewards equivalent to those received in VR, for example.

6.2 Future work

Our current approach to mouth animation proved to be insufficient based on the evaluation. Currently, we map the amplitude of the dialogue audio to movement of the mouth. A logical next step is to extend this approach with phoneme analysis. This entails estimating the viseme based on the uttered phoneme in real-time [7]. The Rocketbox avatars already come with the required viseme poses [8]. A possible approach to implementing this system in our current application is through integration of the Oculus Lipsync plugin for Unity.[1]

Alongside facial animation, body motion also appeared unnatural at times. A possible solution is to replace Unity's humanoid IK system with the upper body IK solver used for visualising the arms of the child. Moreover, the upper body IK system could be used to record and author custom animations, forgoing the need for IK to modify generic animations altogether.

Steps must be undertaken to improve the player interaction systems. For example, the assumptions we make in our head gesture detector are too simple, and the parameters we use for detection do not generalise to every potential user. A more robust system utilising hidden Markov models, like mentioned in section 4.3.4, could give better results. Likewise, blow detection does not always work as expected. A possible solution is to use specialised hardware like we mentioned in section 3.3. Alternatively more sophisticated audio classification techniques could be used, potentially employing deep learning methods like convolutional neural networks.

Finally, the possibility of using speech recognition to avoid the need for human intervention during the vocalisation and verbalisation sections could be evaluated. A possible candidate for lightweight offline speech recognition is PocketSphinx.[2] On the contrary, one might look to more powerful hardware through the use of cloud speech-to-text providers.

6.3 Ethical considerations

Facebook acquired Oculus, the creator of our target platform, in 2014.[3] Over the past years, Facebook has tightly integrated the Oculus platforms with their software ecosystem. Going as far as requiring a Facebook account to use the Quest 2 headset, rather than a separate Oculus account as before. Widespread concern over this change has even forced Facebook to halt sales in Germany [19].

Mandating the use of a Facebook account can be problematic when sharing VR headsets, in both research and therapeutical contexts. As a potential solution, Oculus provides a special model of the Quest 2 for business use, which can be used without a Facebook account.[4] We note that this model is drastically more expensive and is attached to a paid subscription.

Despite all of this, our research still targets the Quest 2 as it is currently the most affordable and best performing according to our criteria (section 4.1). By doing this we might incentivise potential users to sign up for an expensive service with recurring costs. Or worse, urge a breach of the terms of service.

The outcome of the German antitrust investigation might change the current state of affairs. But in the meantime, we stress the importance of surveying other promising hardware platforms.

[1] `https://developer.oculus.com/documentation/unity/audio-ovrlipsync-unity/`

[2] `https://github.com/cmusphinx/pocketsphinx`

[3] `https://about.fb.com/news/2014/03/facebook-to-acquire-oculus/`

[4] `https://business.oculus.com/products/`

Bibliography

[1] American Psychiatric Association. *Diagnostic and Statistical Manual of Mental Disorders.* Fifth Edition. American Psychiatric Association, May 22, 2013. ISBN: 978-0-89042-555-8 978-0-89042-557-2. DOI: 10.1176/appi.books.9780890425596. URL: https://psychiatryonline.org/doi/book/10.1176/appi.books.9780890425596 (visited on 05/20/2021).

[2] Jeremy Bailenson. *Experience on demand: what virtual reality is, how it works, and what it can do.* First edition. New York: W. W. Norton & Company, Inc, 2018. 290 pp. ISBN: 978-0-393-25369-6.

[3] Jillian Sok Teng Boon and Daniel Shuen Sheng Fung. "Cyberage and Child Mental Health". In: *Child and Adolescent Psychiatry: Asian Perspectives.* Ed. by Savita Malhotra and Paramala Santosh. New Delhi: Springer India, 2016, pp. 161–178. ISBN: 978-81-322-3619-1. DOI: 10.1007/978-81-322-3619-1_10. URL: https://doi.org/10.1007/978-81-322-3619-1_10 (visited on 03/29/2021).

[4] Caroline Braet and Susan Bögels. *Protocollaire behandelingen voor kinderen en adolescenten met psychische klachten.* Boom, 2020. ISBN: 978-90-244-0894-8.

[5] Kimberly A. Brink, Kurt Gray, and Henry M. Wellman. "Creepiness Creeps In: Uncanny Valley Feelings Are Acquired in Childhood". In: *Child Development* 90.4 (July 2019), pp. 1202–1214. ISSN: 00093920. DOI: 10.1111/cdev.12999. URL: http://doi.wiley.com/10.1111/cdev.12999 (visited on 06/11/2021).

[6] Sharon L. Cohan, Denise A. Chavira, and Murray B. Stein. "Practitioner Review: Psychosocial interventions for children with selective mutism: a critical evaluation of the literature from 1990–2005". In: *Journal of Child Psychology and Psychiatry* 47.11 (2006). _eprint: https://acamh.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1469-7610.2006.01662.x, pp. 1085–1097. ISSN: 1469-7610. DOI: https://doi.org/10.1111/j.1469-7610.2006.01662.x. URL: https://acamh.onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-7610.2006.01662.x (visited on 04/01/2021).

[7] M. Gonzalez-Franco et al. "MoveBox: Democratizing MoCap for the Microsoft Rocketbox Avatar Library". In: *2020 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR).* 2020 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR). Dec. 2020, pp. 91–98. DOI: 10.1109/AIVR50618.2020.00026.

[8] Mar Gonzalez-Franco et al. "The Rocketbox Library and the Utility of Freely Available Rigged Avatars". In: *Frontiers in Virtual Reality* 1 (2020). ISSN: 2673-4192. DOI: 10.3389/frvir.2020.561558. URL: https://www.frontiersin.org/articles/10.3389/frvir.2020.561558/full (visited on 06/02/2021).

[9] Philip Lindner et al. "Creating state of the art, next-generation Virtual Reality exposure therapies for anxiety disorders using consumer hardware platforms: design considerations and future directions". In: *Cognitive Behaviour Therapy* 46.5 (Sept. 3, 2017). Publisher: Routledge _eprint: https://doi.org/10.1080/16506073.2017.1280843, pp. 404–420. ISSN: 1650-6073. DOI: 10.1080/16506073.2017.1280843. URL: https://doi.org/10.1080/16506073.2017.1280843 (visited on 04/01/2021).

[10] Steve Marschner and Peter Shirley. *Fundamentals of computer graphics*. Fourth edition. Boca Raton: CRC Press, Taylor & Francis Group, 2016. 734 pp. ISBN: 978-1-4822-2939-4.

[11] M. Mori, K. F. MacDorman, and N. Kageki. "The Uncanny Valley [From the Field]". In: *IEEE Robotics Automation Magazine* 19.2 (June 2012). Conference Name: IEEE Robotics Automation Magazine, pp. 98–100. ISSN: 1558-223X. DOI: 10.1109/MRA.2012.2192811.

[12] Mathias Parger et al. "Human upper-body inverse kinematics for increased embodiment in consumer-grade virtual reality". In: *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*. VRST '18. New York, NY, USA: Association for Computing Machinery, Nov. 28, 2018, pp. 1–10. ISBN: 978-1-4503-6086-9. DOI: 10.1145/3281505.3281529. URL: https://doi.org/10.1145/3281505.3281529 (visited on 04/08/2021).

[13] Mark B. Powers and Paul M. G. Emmelkamp. "Virtual reality exposure therapy for anxiety disorders: A meta-analysis". In: *Journal of Anxiety Disorders* 22.3 (Apr. 1, 2008), pp. 561–569. ISSN: 0887-6185. DOI: 10.1016/j.janxdis.2007.04.006. URL: https://www.sciencedirect.com/science/article/pii/S088761850700103X (visited on 03/29/2021).

[14] Raz Schwartz and William Steptoe. "The Immersive VR Self: Performance, Embodiment and Presence in Immersive Virtual Reality Environments". In: *Facebook Research* (Aug. 1, 2018). URL: https://research.fb.com/publications/the-immersive-vr-self-performance-embodiment-and-presence-in-immersive-virtual-reality-environments/ (visited on 04/17/2021).

[15] Mel Slater. *A Note on Presence Terminology*. Jan. 1, 2003. URL: http://www.cs.ucl.ac.uk/research/vr/Projects/Presencia/ConsortiumPublications/ucl_cs_papers/presence-terminology.htm.

[16] Mel Slater. "Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 364.1535 (Dec. 12, 2009). Publisher: Royal Society, pp. 3549–3557. DOI: 10.1098/rstb.2009.0138. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rstb.2009.0138 (visited on 04/17/2021).

[17] *Spectral flatness*. In: *Wikipedia*. Page Version ID: 1008558000. Feb. 23, 2021. URL: https://en.wikipedia.org/w/index.php?title=Spectral_flatness&oldid=1008558000 (visited on 06/03/2021).

[18] Misha Sra, Xuhai Xu, and Pattie Maes. "BreathVR: Leveraging Breathing as a Directly Controlled Interface for Virtual Reality Games". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. New York, NY, USA: Association for Computing Machinery, Apr. 21, 2018, pp. 1–12. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3173914. URL: https://doi.org/10.1145/3173574.3173914 (visited on 04/10/2021).

[19] Reuters Staff. "German cartel office extends probe of ties between Facebook and Oculus". In: *Reuters* (Jan. 28, 2021). URL: https://www.reuters.com/article/us-tech-antitrust-facebook-germany-idUSKBN29X1JB (visited on 06/18/2021).

[20] Els Wippo and Max Güldner. "Praten op school, een kwestie van doen". In: *Kind & Adolescent Praktijk* 2.3 (Sept. 1, 2003), pp. 88–94. ISSN: 1875-7065. DOI: 10.1007/BF03059477. URL: https://doi.org/10.1007/BF03059477 (visited on 04/01/2021).

[21] Jingbo Zhao and Robert S. Allison. "Real-time head gesture recognition on head-mounted displays using cascaded hidden Markov models". In: *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC). Oct. 2017, pp. 2361–2366. DOI: 10.1109/SMC.2017.8122975.