ArachnoTherapy VR

Project Experience Report

Jacob Chapman, Roxanne Harrison, Jacob Sauer

ENSE 477 - Capstone Project Winter 2022

April 9, 2022

List of Figures

Figure 1 Project MVP Progression 9

List of Tables

TABLE 1 List of Members 3

TABLE 2 What went well and why. 10

TABLE 3 What can be improved and how. 10

Background & Contextualization

Team/Project Name: ArachnoTherapy VR

Date of Review: April 9th, 2022

Participants:

NAME	JOB TITLE	ROLE IN TEAM
Jacob Chapman	Back-End Developer	Business Lead
Jacob Sauer	Full-Stack Developer	Lead Developer
Roxanne Harrison	Front-End Developer	Scrum Master

Table 1 - List of Members

It is often difficult to treat mental health disorders, especially those that affect anxiety and confidence. An example of such a disorder is a *phobia*, which is defined as an irrational fear of, or aversion to, a particular activity, object, or situation [1]. Individuals who suffer from phobias often attempt to avoid anxiety-inducing situations; depending on the frequency at which these situations may occur, this behaviour can be severely detrimental to one's quality of life. However, phobias are by no means untreatable. Therapists and psychiatric professionals commonly deliver *exposure therapy* to assist patients in overcoming their fears.

Exposure therapy involves intentionally exposing an individual to an anxiety-inducing stimulus, ideally in a safe and controlled environment [1]. This form of therapy is often effective in treating phobias and other anxiety disorders because the environment in which the therapy occurs is *in vivo*; that is to say, it strongly resembles organic real-world situations. As a result, the processes of eliminating incorrect perceptions, developing new and more accurate perceptions, acclimating to the point at which panic attacks no longer occur or are infrequent, and developing confidence in one's ability to overcome their fears can be easily facilitated [2]. Considerable research supports the efficacy of exposure therapy in treating phobias, including a 2020 literature review of 33 phobia exposure therapy studies by Thng et al., and a 2007 meta-analysis of exposure therapy's ability to treat childhood anxiety disorders by In-Albon & Schneider [3] [4].

However, despite the prevalence and significance of its benefits, *in vivo* exposure therapy is not feasible for every situation from which phobias and other disorders arise. For example, it would be cost-prohibitive and time-consuming for aviophobic patients to travel by commercial airline as part of their treatment plan. Similarly, it is likely impossible to recreate conditions of warfare for individuals suffering from combat-induced post-traumatic stress disorder (PTSD). Finally, it is extremely difficult to control the behaviour of living stimuli such as dogs, cats, or spiders for the purpose of exposure therapy. In such situations, alternatives to *in vivo* exposure may be more desirable.

Furthermore, even if exposure therapy *in vivo* were applicable to every scenario, the toll of the COVID-19 pandemic has resulted in an increasingly high demand for treatment of mental health disorders. According to a 2021 survey of 1,141 psychologists by the American Psychological Association, 84% reported have an increase in demand for treatment since the start of the pandemic, 62% reported an increase in referrals, 68% reported an increase to their preexisting waitlist, 41% felt that they were unable to fulfill the needs of all potential patients, and 46% indicated that they were experiencing symptoms of burnout [5]. As a result of these trends, remote or virtualized methods of delivering psychiatric care are being explored as potential replacements for in-person appointments and therapy sessions.

An alternative to *in vivo* exposure therapy, whose potential has been documented for decades but never capitalized on in a widespread or meaningful manner, is *virtual reality exposure therapy* (VRET or VR exposure therapy). VR demonstrates considerable potential as a tool in the area of exposure therapy for three key reasons [6]. First, since VR systems display environments that the brain interprets as being four-dimensional, and these virtual environments often respond to the user's perceptual actions in a realistic manner, users often report feeling a sense of "presence" when immersed in them [7]. Presence is a prolonged subconscious interpretation that the virtual environment is reality, and as a result of this interpretation, VR users often execute the same actions and convey the same emotions that they would in the real world — even though they understand, on a conscious and cognitive level, that the VR experience is a simulation. This is believed to be the mechanism by which new behaviours can be learned from VR experiences. Secondly, there is a nearly infinite design space for the setting and content of a VR therapy experience, which means that situations for which *in vivo* exposure would be logistically impractical, such as air travel or warfare, can, in theory, be simulated adequately in VR. Thirdly, VR applications can easily be distributed around the world, and if coping strategies and positive reinforcement are provided within the applications themselves, then they could potentially be used as a form of remote care in order to alleviate psychologists' excessive workloads. Finally, and perhaps most importantly, the content of VR therapy is entirely risk-free, and if the experience is implemented well, the intensity of said content can be controlled. This is a notable advantage of VR exposure therapy over exposure *in vivo* with respect to treating phobias of living creatures, as it provides a straightforward mechanism for patients to personalize their therapy experience. For example, assume that Patient A and Patient B have a phobia of dogs. Perhaps Patient A is comfortable with a dog wagging its tail and running around, but Patient B is only comfortable if the dog sits still. A VR exposure therapy application could potentially ensure that the virtual dog only ever exhibits the sitting behaviour for Patient B, but enable access to additional behaviours for Patient A, or other patients who may be further along in their treatment for the phobia. A real dog's behaviour is not nearly as easy to control in this setting.

Problem Definition

For the reasons listed in the previous section, virtual reality has demonstrated potential efficacy in the treatment of several phobias and anxiety disorders. Indications of this efficacy, however, are largely confined to academic studies, with only minimal success having occurred in the commercial sector. Based on the team's existing knowledge of VR theory and literature, we attribute this lack of success to two key reasons: first, the interactions present in most VR therapy experiences are insufficient or unrealistic, and second, the experiences often do not contain modules for in-game progression or customizable stimulus intensity. The former detracts from the perceived "plausibility" or "realism" of the VR environment, which negatively affects the sensation of presence that facilitates learned behaviour, while the latter fails to accommodate patients with different degrees of anxiety or different paces of learning. We sought to produce an application that would resolve both of these issues and, at a proof-of-concept level, potentially represent a blueprint for commercially accessible and successful VR exposure therapy experiences. As a result, realistic hand-object interactions and a system by which to modulate the intensity of the anxiety-inducing stimulus were considered paramount objectives for our project.

<u>Golden Circle</u>

Why? In an age where demand for mental health treatment is steadily exceeding capacity, the accessibility and efficacy of treatment options for anxiety and phobias need to be improved.
How? We wanted to build an application that allowed for level-based progression, realistic interactions, control over a stimulus' intensity, and built-in guidance from a virtual therapist character.
What? ArachnoTherapy VR is compatible with all Oculus hardware, improves upon existing limitations of *in vivo* exposure, and can theoretically be self-administered.

<u>Envisioned Customers</u>

As ArachnoTherapy VR is a dedicated exposure therapy experience for individuals with a fear of spiders (arachnophobia), said individuals were always our envisioned northstar customer. The overarching metric by which our project would be judged in a professional setting is its ability to assist arachnophobic patients in overcoming their anxiety. Consequently, the vast majority of our design decisions were influenced by their potential psychological effects on this user base.
We also identified two carryover customer groups; one of these groups exists as a result of higher-level design practices, while the other exists as a result of our lower-level development practices. For one, therapists and other phobia-focused psychiatric professionals could conceivably benefit from ArachnoTherapy VR and projects similar to it, as the remote nature in which VR experiences can be delivered would allow for work to be offloaded. As well, this carryover customer group would benefit further from the inclusion of coping strategies, assumption-breaking facts, and real-time encouragement in the experience. The second carryover group is patients with phobias of other living creatures that cannot be easily controlled in a real-world setting. The spider A.I. module was written in such a way that it could be relatively easily reused for other animal A.I. systems. This opens the door for potential follow-up applications, such as EntomoTherapy VR for fear of insects, or OphidioTherapy VR for fear of snakes, if they were to be requested.

Development Methodology

Early on in the project we split responsibilities into three streams; Roxanne with the front end and design work as well as project management, Chapman with the documentation, and Sauer with the backend development. This allowed for a clear division of activities without a need for frequent check-ins or risk of overlapping effort or missing objectives. By following an iterative development approach, each room was designed on the front end, backend coding then came after, followed by testing before moving on to the next room. This process then repeated until all rooms were completed, with simultaneous documentation and project progress tracking.

Each room sprint cycle lasted two weeks with the exception of Room 4 and 5 which were longer due to the complex nature of the spider creation panel and multitude of rooms in the room 5 series. The team had two weekly scrums for short project check-in and alignment and to ensure all deadlines were adhered to prior to the in class scrums. By following a fail-fast fail-forward approach the team was able to finish the fall semester ahead of schedule, allowing for more time troubleshooting and finalizing the therapist character and tutorial room in the winter semester.

Figure 1 shows the MVP progression timeline planned for the project finalization (winter 2022). Having these milestones set early on ensured that we had critical functionality in place at each deadline in order to make steady progress. Throughout the course of the project this stayed fairly static but we did make some adjustments based on user and stakeholder feedback. This included adding the tutorial room, shortening the ‘about exposure therapy’ and scrapping the ‘optional check-ins’ so as not to overwhelm the user with too much content. Therapist character scripting, the implementation of the coping strategies, and layout of the spider creation panel also went through different iterations as more feedback was received. In the end our early start, thorough planning, and ability to remain agile and adapt to feedback received allowed us to be successful and produce an impressive end product.

Early in the design process, we selected Unity as the game development engine for ArachnoTherapy VR, as members of the team have extensive experience working within it, and it contains a powerful and developer-friendly tool for interfacing to Oculus hardware. In addition, Unity’s predominant backend scripting language is C#, whose class-based file structure closely resembles that of Java. Due to this similarity, we saw an opportunity to apply many of the software development principles that we’ve learned throughout our degree. A more detailed explanation of these principles can be found in our code quality review, but the most important ones are outlined below:

1. *Avoid per-frame computation whenever possible*

It is tempting for Unity developers to place all of their logic for a particular class in its Update() function, which is called once per frame as part of a global superclass called MonoBehaviour, simply to ensure that the logic will actually execute. However, due to the frequency at which Update() is called, the inclusion of too many or too complex calculations can be detrimental to the project’s runtime. Further compounding this issue are the facts that the Oculus Quest's processing capabilities are rather underpowered, especially in comparison to expensive gaming PCs, and that much of the system's computational power needed to be reserved for rendering frontend aesthetics. Therefore, we needed to ensure that our code base ran as efficiently as possible, so instead of relying on per-frame updates, we implemented an event-driven logic system for every non-continuous process in ArachnoTherapy VR.

For example, *Grabbable.WhenReleased()* computes translational and rotational velocity vectors for grabbable objects when they are released from the user's hands, while *RoomParameterGrabbable.SetGrabStatus()* invokes *RoomController.UpdateRoomConditions()* whenever the user grabs a grabbable object that is also a room parameter. Rather than running a process in which these functions are called once per *frame*, our implementation is structured in such a way that these functions only need to be called once per *event*. Both functions correspond to an event listener for the Trigger button on the Oculus Touch controllers, and thus, only ever need to be invoked when the state of this button changes. As a result, the inefficient velocity calculations are only ever performed on the exact frame in which a grabbable object is dropped, and *RoomController.UpdateRoomConditions()* is only invoked on the exact frame in which a room parameter grabbable is grabbed.

Some processes, such as the application of translation vectors to the player camera in *Movement.cs*, do require per-frame updates, but the vast majority of the application logic can be executed on a per-event basis. While not identical to the difference between constant runtime (i.e. O(1)) and linear runtime (i.e. O(N)), the difference between per-frame and per-event execution is analogous to it, and if the former outweighs the latter, the negative effect on the application's runtime is not inconsequential.

2. *Minimize the use of hard-coded constants*

Hard-coded constants are often detrimental to the integrity of a code base, as they inhibit its flexibility across intra-project situations and reusability in other projects. Instead, using variables whose values can be altered across multiple instances of the same class is considered best practice. For example, whenever an *AbstractButton* object is interacted with by the user, its position must be toggled between a "pressed" vector and a "released" vector. Since all of the buttons are located in different spots around the virtual environment, all of these vectors have unique values. An extremely unclean option for implementing this would be to hard-code position vectors into the script for every specific button, and use if-else statements to ascertain which position vector should be applied. Instead, we simply created public 3D vector variables in the *AbstractButton* script that were then instantiated at compile-time via the Unity Inspector window.

Some hard-coded constants are present in our code, as we did allow exceptions for when the value itself was justifiable or easily understood. If a spider collides with a wall, for example, it chooses a random orientation around its local Y-axis between -180˚ and 180˚, and rotates to it in order to simulate the action of turning around. These two integers are hard-coded, as from our point of view, they are rather obvious limits for a random rotation function. Furthermore, were we to extend this module to other creatures, we would almost certainly implement the exact same behaviour for whenever those creatures collided with walls. As such, reusability would not be affected.

3. *Manage complexity through structural and behavioural design patterns*

The code base of ArachnoTherapy VR contains numerous dependencies between modules, and since we separated event-listening logic from event-executing logic in order to minimize rigidity, these dependencies were unavoidable. Furthermore, many concrete subclasses of *AbstractButton.cs* and *AbstractGrabbable.cs* – representing objects with which the user can, and should, interact – possess unique individual properties that require additional backend logic. In order to manage these two sources of complexity, we sought to employ the use of structural and behavioural software design patterns whenever they were applicable.

The ArachnoTherapy VR code base also contains creational design patterns, including the Singleton for static spider creation interface data, and the Factory for instantiation of spiders. These design patterns were powerful for their specific scopes, and we always intended to use them if we felt that they were necessary, but in order to connect inter-module logic efficiently, design patterns of the structural and behavioural categories are especially essential. As such, these categories were emphasized as a tenet of our development process.

Examples of non-creational design patterns used in ArachnoTherapy VR are the Facade, which encapsulates concrete button event logic from the buttons that invoke it, and the Model-View-Controller (MVC), which separated the functionalities of the spider creation interface. The Facade pattern ensured that the button classes would only ever call an abstract event execution function, and that if additional button event types were needed, they could be added to *AbstractButtonEvent*'s existing descendants without jeopardizing the functionality of the underlying button logic. Meanwhile, the MVC pattern ensured that no GOD classes would be created for the spider creation interface, as the data-containing *SpiderCreatorModel* class would only ever execute its functions if they were invoked by a specific *Controller* class. Furthermore, all *Controller* classes derive from *AbstractButton* and maintain a reference to a specific *SpiderCreatorModel* instance, which meant that there was no need to modify existing user-button interaction logic.

4. *Follow the principles of SOLID object-oriented design*

As specified by Robert C. Martin, the tenets of SOLID object-oriented design, which dramatically improve a code base's reusability, extendability, and integrity, are as follows:

- **S**ingle Responsibility Principle
  - A class should only have one reason to exist, one reason to change, and one primary responsibility. Directly related to this principle is the issue of GOD classes, which we previously discussed when explaining the spider creation interface architecture.
- **O**pen-Closed Principle

  - Software entities should be open for extension, but closed for modification. In other words, the addition of new behaviour to a software entity should never compromise the functionality of existing behaviour.
- **L**iskov Substitution Principle
  - Concrete subclasses must be substitutable for their abstract parent classes. In ArachnoTherapy VR, for example, an *AbstractButton* object is able to call *AbstractButtonEvent.ExecuteEvent()* and *VideoButtonEvent.ExecuteEvent()* without inherently knowing the difference between the two.
- **I**nterface Segregation Principle
  - Classes and subclasses should not be forced to depend on unused or unnecessary features. The separation of spider creation interface functions into individual classes is an example of this principle being followed, as each controller class handles one function, which reduces coupling and more easily facilitates extension.
- **D**ependency Inversion Principle
  - Software entities should depend on abstractions, rather than concretions. For example, the object interaction module *Grabber* in ArachnoTherapy VR maintains references to *AbstractGrabbable* and *AbstractButton*, but does not contain any logic for their concrete subclasses. This principle dramatically simplifies the logic in *Grabber*, as it never needs to distinguish the exact type of grabbable with which the Oculus controller is interacting. Consequently, the *AbstractGrabbable* framework is rather easy to extend if new types of grabbables are desired, as additional concrete subclasses can simply depend on the abstractions of the parent class.

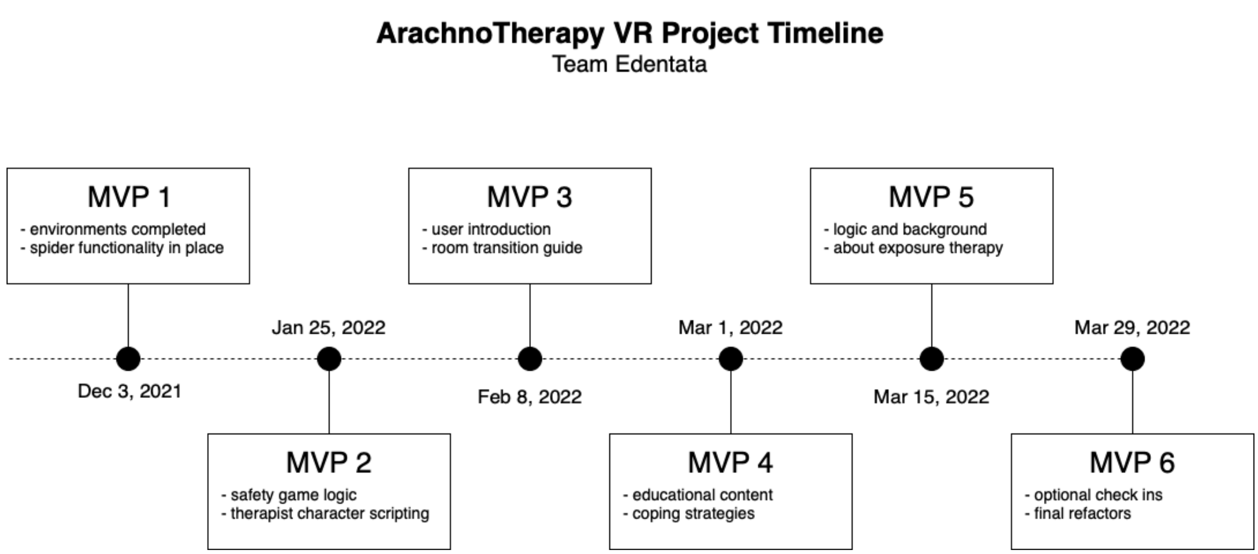

Figure 1 - Project MVP Progression

Reflection

Project progression.	Planning all critical functionality and incrementally building on the product allowed for steady and effective progress.
Room iterations.	Following the same process for each room allowed for seamless additions from a testing scene into a master scene without duplication of effort.
Psychology influence.	Meetings with Dr. Nick Carleton were scheduled on a bi-weekly basis for the entire semester so we had consistent check-ins and feedback in every stage of the process. Changes were made quickly whenever feedback warranted and were reviewed in the subsequent session.
User testing.	Having an Oculus headset loaned from the aRMADILo Lab allowed for each member to perform user testing frequently and provide their feedback as well as beta user feedback to make changes and identify bugs in a timely manner.
Tutorial room.	Lots of user feedback and struggle early in development was around the interactions needed in the environment. By adding the tutorial room, users were able to learn about VR itself and the controls if they were quite new before starting into an anxiety inducing environment.

Table 2 - What went well and why.

Version control.	Plastic SCM migration caused some project issues - in the future this could be improved by using the same version control throughout the project without having to change midway through. This was not in our control, unfortunately.
User customization.	Functional ‘nice to haves’ could be improved in terms of customizability for the user. This may include profiles, saving progress, or selecting a therapist voice of choice.
Unity organization.	Having hundreds of assets and materials in the Unity project resulted in some disorganization. The project itself could be improved by better organizing imported objects into their necessary folders and further grouping material components in individual folders.
Therapist intervention.	Future MVPs could include the ability for a live therapist to remotely watch a user’s session or talk them through the experience instead of a virtual therapist. This may be more relevant for people with extreme

	arachnophobia.

Table 3 - What can be improved and how.

Resources

Unity Scripting API: https://docs.unity3d.com/ScriptReference/index.html

OVRInput Scripting API: https://developer.oculus.com/documentation/unity/unity-ovrinput/

Wolf Spider Animated (Unity Asset):
https://assetstore.unity.com/packages/3d/characters/animals/insects/wolf-spider-animated-29330

Tarantula Animated (Unity Asset):
https://assetstore.unity.com/packages/3d/characters/animals/insects/tarantula-animated-186189

Animated Spider (Unity Asset):
https://assetstore.unity.com/packages/3d/characters/animals/insects/animated-spider-22986

Facade Design Pattern: https://www.tutorialspoint.com/design_pattern/facade_pattern.htm

MVC Design Pattern: https://www.tutorialspoint.com/design_pattern/mvc_pattern.htm

Many other objects in ArachnoTherapy VR, including the television, book objects, and bedroom furniture, were also found on the Unity Asset Store as free downloads.

References

[1] Yetman, D. & Saripalli, V. (2021, June 21). Exposure Therapy. Healthline Media a Red Ventures Company. Retrieved April 8, 2022, from: https://www.healthline.com/health/exposure-therapy.
[2] APA Div. 12 (Society of Clinical Psychology). (2017, July 1). What is Exposure Therapy? American Psychological Association. Retrieved April 8, 2022, from: https://www.apa.org/ptsd-guideline/patients-and-families/exposure-therapy.
[3] Thng, C. E. W. et al. (2020, March 19). Recent developments in the intervention of specific phobia among adults: a rapid review. F1000Res., 2020 (9). doi: 10.12688/f1000research.20082.1.
[4] In-Albon, T. & Schneider, S. (2006, December 1). Psychotherapy of Childhood Anxiety Disorders: A Meta-Analysis. Psychotherapy and Psychosomatics, 2007 (76), pp. 15-24. doi: 10.1159/000096361.
[5] Bethune, S. (2021, October 19). Demand for mental health treatment continues to increase, say psychologists. American Psychological Association. Retrieved April 8, 2022, from: https://www.apa.org/news/press/releases/2021/10/mental-health-treatment-demand.

[6] Maples-Keller, J. L., Bunnell, B. E., Kim, S. J., & Rothbaum, B. O. (2017). The Use of Virtual Reality Technology in the Treatment of Anxiety and Other Psychiatric Disorders. Harvard review of psychiatry, 25(3), 103–113. doi: 10.1097/HRP.0000000000000138.

[7] Sanchez-Vives, M. V. & Slater, M. (2005). From presence to consciousness through virtual reality. Nature Reviews Neuroscience, 2005(6), 332-339.