جامعة نيويورك أبوظبي

NYU | ABU DHABI

Computer Programming for Engineers

ENGR-UH 1000

Final Project: *Fight-or-Flight*

Participants: Amina Kobenova, Adilet Majit, Sayazhan Sagynay
NYU IDs: ak7588, am10044, ss12782

1. **Introduction**

What are you scared of? Is it heights? Spiders? Or perhaps, public speaking? As much as we seek for ourselves to be one-of-a-kind, our fears seem to be universal. British Journal of Psychiatry states that certain phobias such as acrophobia (fear of heights), arachnophobia (fear of spiders) and claustrophobia (fear of confined or crowded spaces), for instance, are quite widespread. In attempt to treat various fears, modern medicine borders with Virtual Reality, which has great medical potential according to Chris Brewin, a professor of clinical psychology at University College London. Practicing a technique called exposure therapy, experts make patients gradually and consistently face a specific fear. Most studies agree that patients are substantially more comfortable with virtual reality immersion treatments. They promote clinical interaction and medication adherence. Simulating a fear exposure, the user is able to explore the roots of the phobia within the constraints of VR.Virtual reality provides many benefits, such as: accessibility, privacy, facilitation of care, patient safety, financial gain, and more:

- **Enhanced security**

Virtual Reality Exposure means introducing the patient to a virtual environment that includes the expected stimulus instead of placing the patient in a real environment or making the patient visualize the stimulus. The actual environment can be unpredictable. The therapist controls the virtual environment by means of the computer keyboard, ensuring exposure to the programmed situations.

- **Better Treatment and Simple Scheduling**

Deliberately designed to support anxiety disorder exposure therapy, the VR software and virtual environments models provide more efficient treatment along with the convenience of scheduling. As it usually requires leaving the office of the therapist and consequent extended sessions, Standard exposure therapy can be very expensive. Within the standard therapy hour (usually 45-50 minutes) and within the confines of the therapist's office, virtual reality programs can achieve the same exposure. In the same week, multiple sessions may be scheduled.
Because counseling is performed in the therapist's office, there is no greater risk of running into colleagues, meeting difficult situations and being in public should the disclosure go too far than with any other form of unlimited potential.The extended appointments required for traditional exposure therapy will not be covered by many insurance companies. This may not be a problem with an exposure-assisted VR device. The sessions are conducted in the office of the psychologist and can usually be done within 45-50 minutes.

- **Limitless interactions of distresses**

Having VR allows the therapist to monitor ultimate stimulation for optimal viewing. For example, only one take-off and landing per flight is bound by real-world limitations.VR training helps the therapist to thoroughly manipulate conditions, such as landing the digital aircraft, in order to best match the patient.

In light of such superiority and clinical potential of VR, our team has designed a VR fear-oriented project, called *Fight-or-Flight*, which presents a training platform that will expose its audience to two most common phobias(arachnophobia and acrophobia) in order to slowly eliminate the fear. It will allow the therapists to control the environment entirely, making VR extremely convenient. A medium between reality and simulation, it will allow battles with phobias in a safer way.

***Fight-or-Flight* identifies the following objectives:**

- To take advantage of VR as potentially an extremely powerful tool for the psychiatric community.
- To utilize up-to-date modern technologies to benefit its creators.
- To create a sense of presence for total immersive experience.
- To use our current skillage to attempt to tame one of the most ever-lasting problems of medicine and psychology.
- To provide patients with a strong feeling of enhanced security to stimulate the clinical progress.
- To showcase the feasibility of implementation and of search of solutions to eliminate phobias for various groups: college students, teachers, graduate school students and even high school students.
- To learn a cross-platform game engine - Unity 3D engine.
- To master the application of Oculus Rift VR platform.
- To acquire a knowledge of a general-purpose programming language - C#.

2. **Project Development**

The project was made for the Oculus Rift VR platform in the Unity 3D game engine. Our application mainly consists of three scenes: a menu from which the user selects a feared phobia and two phobia scenes-acrophobia and arachnophobia respectively.

1) *Menu* — the UI Scene.

This scene's aim is to get a user choice and then move the user to a chosen scene.

Image 1: *Menu* Scene.

There is a script attached to the Scene (to the UI elements, in particular) to make it easier for the user to navigate between two other scenes — Game1 and Game2. We have used a core Unity class called SceneManager to count the scenes and operate between them. In the script, the public void PlayGame1()function gets the active UI scene and increments the active scene index by 1 to redirect to the second scene, which is Acrophobia simulation. Likewise, the public void PlayGame2()function increments the active scene index by 2 to redirect to the third scene, which is Arachnophobia simulation. The script is written as follows:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
public class MainMenu : MonoBehaviour
{
 public void PlayGame1()
    {

SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
    }

 public void PlayGame2()
    {

SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 2);
    }

public void QuitGame()
    {
        Debug.Log("QUIT!");
        Application.Quit();
    }
}
```

In the scene of the User Interface, adding each function to a corresponding button was a challenge for our group, as Unity did not read the script correctly and none of our functions would work initially. To resolve this issue, we had to refer to a YouTube tutorial and only then could we see the error: our UI GameObjects were nested in two UI panels, which made it difficult for Unity to attach the script to a particular panel. We were able to attach a script to the *MainMenu* panel by removing the unnecessary panel and making that feature a corresponding UI component. Enhanced usability of the UI scene makes switching between the scenes simpler for the user.

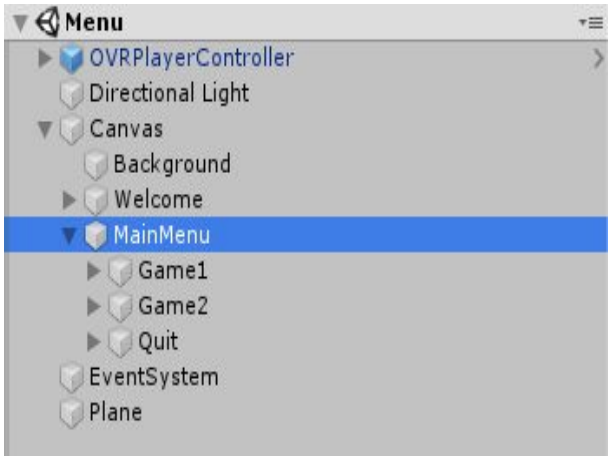

Image 2: *Menu* Scene correct layout.

2) *Game1* — Acrophobia Scene.

The goal was to make a virtual reality where the patient faces his/her fear of heights to overcome acrophobia in the most challenging way - by walking down high-altitude tightrope.

We took an intimidating picture for the basis of a woman walking down the rope at an altitude and simulating her surroundings and conditions.

We began with adding a Terrain game object in Unity3D. We added mountains, hills and water component for a lake to simulate a realistic environment as shown in the picture above. We then used the Environment Standard Assets to add models of grass and trees and paint the field with rough, rocky, cliff textures. The surroundings are set.

The next step is to add a platform that the user will walk along.To do this, we simply added 3D cube objects and placed them at high height from the ground in the middle of the terrain.

We added the OVR Player Controller from the Oculus Integration Assets Package to allow the patient to feel the full immersion of walking at altitude. The controller has scripts that allow the user to move in the real world and adapt to position themselves in a virtual reality environment, or the keyboard can be used as an option for walking.We put the OVRPlayerController on the platform, which allows the patient to challenge his / her acrophobia.

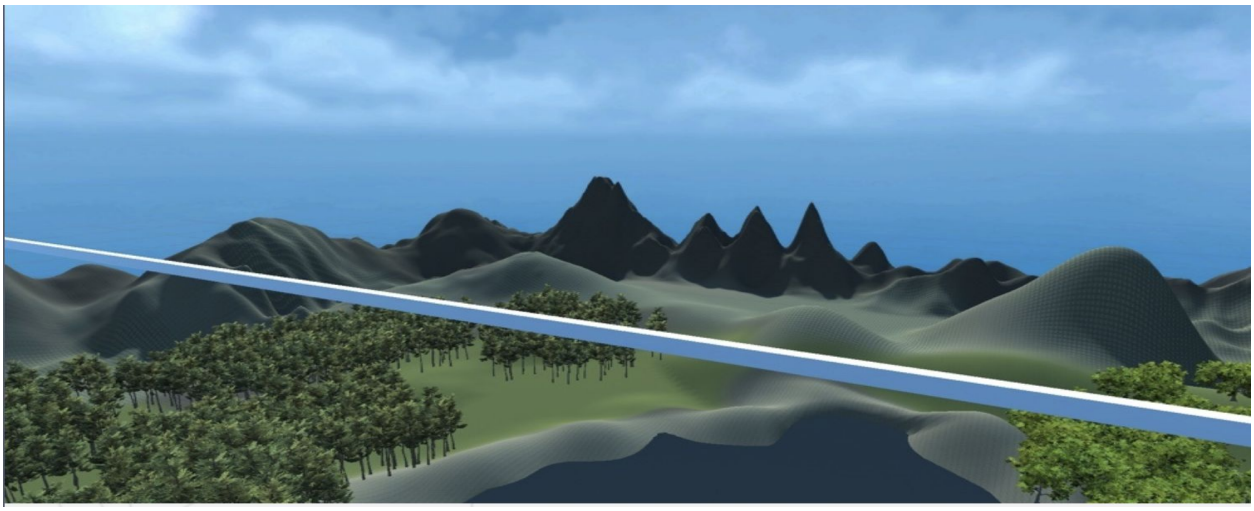

Image 3: *Game1* Scene Snapshot.

Nevertheless, it was necessary to address the case of falling out of the platform. To this end, under our walking platform, we added another platform (we used cube object), made it transparent and used it as a trigger.The idea is that if the user falls off the platform, he/she will fall right down on the invisible, trigger platform, and return to the platform.

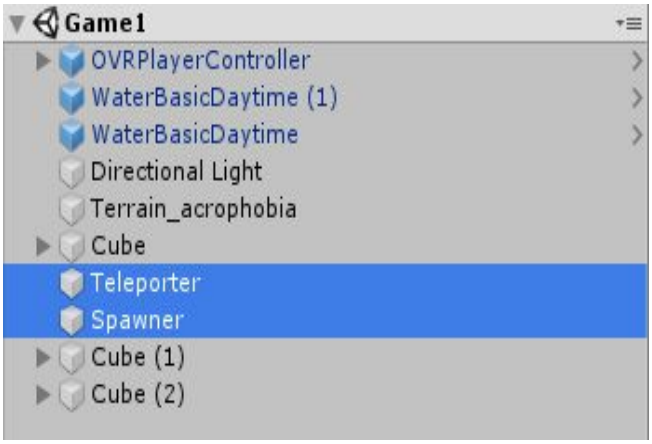

Image 4: *Game1* Scene layout.

We wrote a simple C # script for the teleport feature and added it to the platform-trigger object.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class TeleportPlayerBack : MonoBehaviour
{
    public GameObject teleportTarget;
    public GameObject thePlayer;


     void OnTriggerEnter(Collider other)
    {
        thePlayer.transform.position = teleportTarget.transform.position;

    }
}
```

Once the user has ' entered ' the trigger (Teleporter object to which the script has been added), the user's position has been transformed into the position of teleportTarget, which is a Spawner game object placed on the platform, so that the user is returned to the platform.

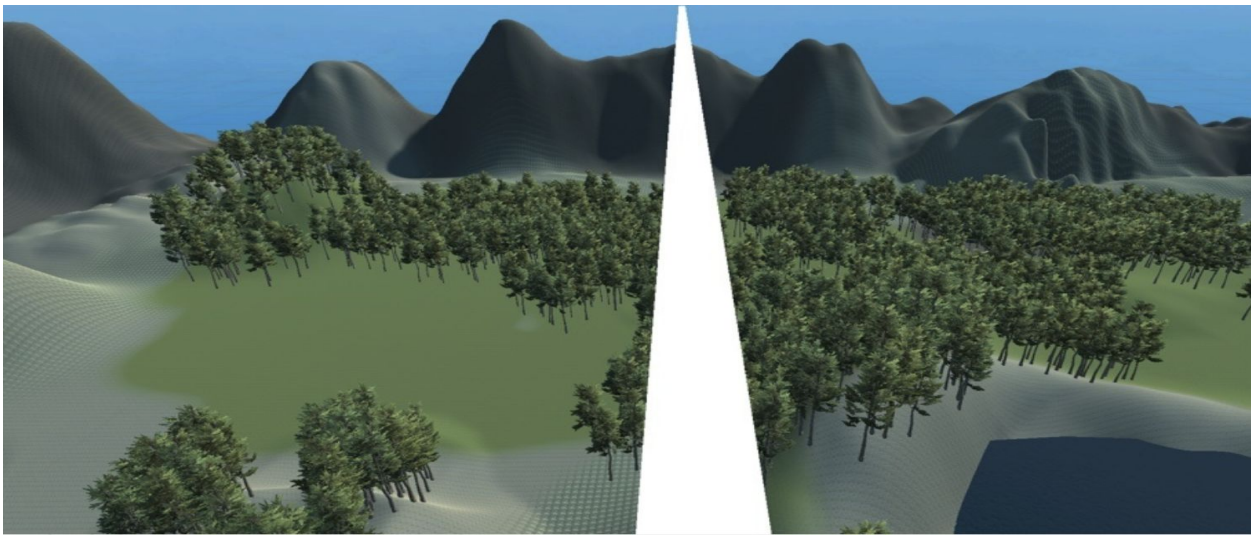

Image 5: *Game1* Scene Snapshot.

The simulation is ready for patients to confront their height fear.

3) *Game2* — Arachnophobia Scene.

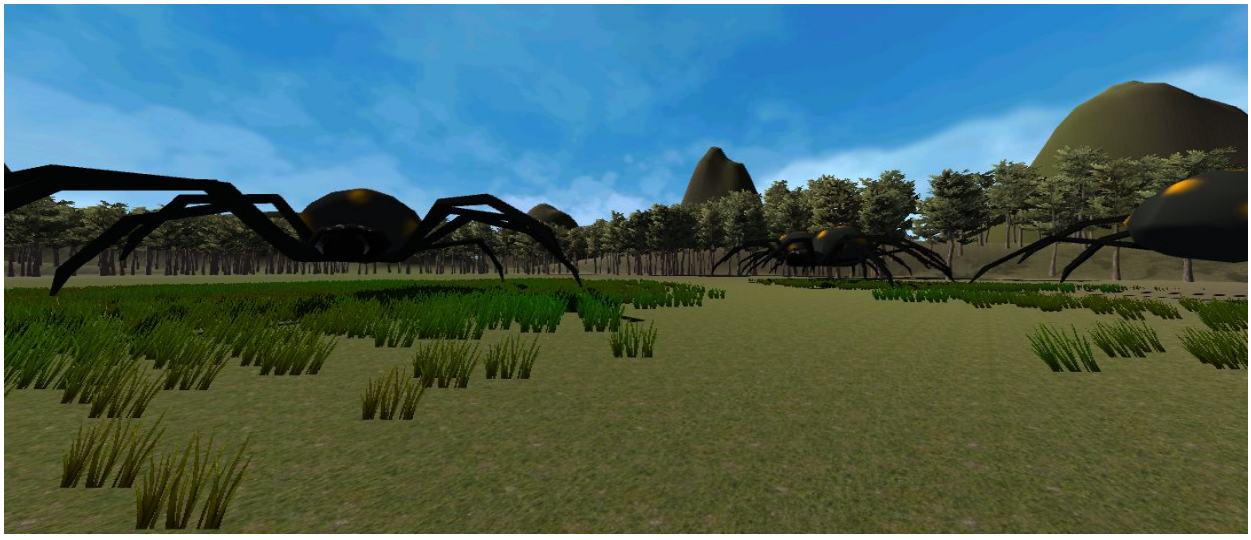

Image 6: *Game2* Scene Snapshot.

The goal was to create an environment of virtual reality where the user is constantly surrounded by large spiders; wherever the user goes, spiders move around him/her.
We started by creating a terrain with trees and mountains where the user and spiders will move, adding grass and textures. We added here OVR Player Controller from Oculus Integration as well, so that the user can easily move across the terrain using the VR platform.

Next, we downloaded and imported Spiders prefabs from the Assets Store. After creating an Animation Controller, we linked the animation “walk” to the Entry and put it under Controller in the Animatore section. By clicking on our Spider model, we changed the Animation Type to Humanoid, put it on loop and applied all the settings.

Image 7: *Game2* Scene layout.

We added a simple script to each spider object for Spiders to constantly surround the user.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player : MonoBehaviour
{
    public float radius = 10f;
    public float speed = 0.1f;
    public Transform target;
    float theta = 0f;
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        theta += Time.deltaTime * speed;
        theta %= 360;
        float x= radius * Mathf.Cos(theta);
        float z = radius * Mathf.Sin(theta);

        transform.position = target.position + new Vector3(x, 0, z);
    }
}
```

3. **Results and Evaluation**

Our team faced various challenges in implementing this project, ranging from debugging C# scripts to properly using Oculus VR Asset in Unity 3D. *Fight-or-Flight*, consisting of three scenes, presents a user interface menu to get a user selection and then redirects the user to a selected scene.

Some of the challenges faced by our group during the implementation of the project were creating animations and writing scripts for these animations for the phobia simulation scenes. As each scene has a C# script attached to it, we had to debug and adapt it to a corresponding scene.

When it comes to immersing the user in the environment that causes acrophobia, we had to find a way to put the user on the rope and trace the movements of the user in the event that he/she "falls off the path." To overcome this challenge we created an invisible cube, which we then placed underneath the tightrope at a reasonable height for it to work as a trigger that puts the patient back to the initial position to try again.

On the other hand, when doing the arachnobia simulation, we were challenged with creating the appropriate animations for them to move both seamlessly and realistically.

4. **Conclusion and Future Work**

Despite all the challenges faced by our team during the development of Fight-or-Flight, we were able to familiarize ourselves with a new programming language C #, a Unity 3D game engine platform, and a VR Headset Oculus Rift in a relatively short time frame. With the knowledge gained from *Computer Programming for Engineers* and other online resources (YouTube tutorials and Unity Learn), we were able to create an application that addresses the top phobias in the world. Now that we have developed the necessary skills to create VR applications, in our future work we can see the potential improvements. In other words, a stronger UI-Oculus interface can be rendered using the joystick or other remote input / selection tool. When it comes to phobia immersion, by making vivid graphic animations and using special hardware such as touch sensors, the user experience can be improved and made more realistic.

Reflection on Learning

Since introducing *Fight-or-Flight*, our team has grown to find Virtual Reality as a promising resource for dealing with a number of different conditions, with the most powerful evidence for use in exposure therapy for patients with anxiety disorders, cue exposure therapy for patients

with substance use disorders, and diversion for patients with acute pain involving painful procedures.

Notwithstanding some of the challenges we faced along the way, our group not only gained a deeper understanding of the ever-lasting phobias problem and its implications sequences for overall health, but also challenged ourselves to push our boundaries to find a solution to this problem through our knowledge of Unity 3D engine and C # to the best of our ability. Namely:

- How to apply even the most basic set of programming skills to solve multiple major issues.
- How to operate and utilize Unity #D engine to bend and tailor it to a specific environment.
- How to practice coding in C# in collection with Unity.
- How to integrate Oculus Rift VR platform with Unity.
- How to self-educate and learn across different educational channels and video tutorials on our own.
- How to work in a cooperative environment.
- How to cleverly coordinate timings to meet the schedules of each participant.
- How to fully understand current worldwide issues that require trained programmers to alienate them.