Seasons: Seasonal Adventures

Ahmed Abdalla
Dept. Computer Science
Hunter College
New York, USA
ahmed.abdalla36@myhunter.cuny.edu

Sheng Dong
Dept. Computer Science
Hunter College
New York, USA
sheng.dong23@myhunter.cuny.edu

Lemuel Gjannie Cataquiz
Dept. Computer Science
Hunter College
New York, USA
LemuelGjannie.Cataquiz32@myhunter.cuny.edu

Amir Hasan
Dept. Computer Science
Hunter College
New York, USA
amir.hasan34@myhunter.cuny.edu

David Olutalabi
Dept. Computer Science
Hunter College
New York, USA
david.olutalabi94@yorkmail.cuny.edu

***Abstract*—The purpose of this project was to make a playable simulated experience of the outdoors which would help improve mental well-being especially when people are heavily deprived of such experiences in reality after being on lockdown due to the pandemic for so long. We used Google Cardboard VR and Unity to make this application. Each scene within this application represents a season with its own game mechanics.**

***Index Terms*—VR, Google Cardboard, Open-world, First-Person**

I. Objective and Goals

The main objective of this project was to create an interactive virtual reality (VR) simulation of being outside in nature to help alleviate stress from quarantine or otherwise. The experience aims to soothe/relax the player and improve their mental well-being. This effect was achieved by giving users the ability to interact with the virtual natural environment during the four seasons of the year keeping in mind VR concepts such as immersiveness and presence. As a team, we each decided to make games or activities to do in each season and scene. For the winter, we wanted the user to throw snowballs or catch snowflakes with their tongue. For the fall, to chase animals or catch the leaves as they fall. For the summer, to jump and collect fruits on platforms. For the spring, to clean the environment, collect flowers and plant new ones. We were able to accomplish these goals by using a collection of free assets from the Unity Assets store.

II. Rationale for choice of VR project

The rationale for this project was of course the need/want for outdoor interaction. Given the current state of things, we figured that by utilizing the seasonal changes in correspondence with the four scenes per group (five in our case), it would be appropriate. As such we set each season to have a general theme for our minigame which ranges from relaxing flower picking in spring to a more robust and action driven 'platformer' where the player tries to collect all the fruits. Though set in a 'real world' setting, we did not want that to be a constraint as the goal of this game is to provide mental well-being. We instead wanted the player to have options as to what they want to enjoy via relaxation or a challenge.

III. Review of VR application area

Matthew P. White et al. in their "A prescription for "nature" – the potential of using virtual nature in therapeutics" explains how nature-themed VR experience can be beneficial to one's mental health. Though exposure to natural settings is "associated with reduced general mortality, improved mental health, increased physical activity, and better birth outcome," such settings may not be accessible to everyone, so experiencing nature in VR is an alternative way to get at least some of these health benefits [1]. As the article explains, immersive VR experiences with nature themes can help with depression and chronic pain, reducing anxiety and boosting mental health by providing therapeutic experiences of relaxation and distraction. VR nature experiences can also help those with neurological disorders or victims of stroke who have limited mobility by focusing on "motor control, balance, gait, and strength." While more research and studies are needed to truly explore the benefits and costs of using VR nature as an alternative to experiencing real nature, the authors conclude that the applications of immersive VR experiences in mental and physical health should not be ignored and thought of as just a "gimmick." In a study done in 2020, the usage of VR provided greater "nature connectedness" and "positive emotion" than a nature experience provided on a television [1].

IV. Methodology

A. Interaction

In the early stages of this project, a quick inventory check made it clear that our platform will mainly revolve around Google Cardboard for its cheap and effective model. Obviously when compared to fully fledged VR headsets and controllers, it lacks, but for its price point, it is unmatched. We also intended for our application to be built on iOS since most of us had an iPhone. One of our group members had a Macbook capable of deploying the project through Xcode. In terms of code collaboration, we utilized Unity Teams which provides version control similar to using Github. We were able to commit and pull changes independently without having to work simultaneously and to test features. We were also able to revert the project to a working state whenever issues came up after making changes.

We began by having one master scene with basic assets as a 'baseline' to plan what we wanted to do and how we were going to do it, with the perspective being in first person since it was ideal for VR and the experience we were each trying to create. In the scene, the player views the environment through the eyes of an in-game player object that can interact and be affected by its surroundings. After solidifying our approach and understanding, we began importing several free assets and testing them in our own individual scenes. One major issue we then ran into was the build time. To build on our phones, the actual game needed to be capable of running smoothly on any iOS build which severely limited the game's design and features. Too many interactions and/or assets meant insanely long build times and the possibility of crashing when playing. This became the main driving force behind our decision to keep the game in first person, and to avoid graphically intensive assets and adding too many scripts.

As we neared the end of our second month, we began to realize various problems with both our Unity version and the assets we imported. The Unity version we began development with (2019.4) was incompatible with the Google VR SDK (which is deprecated and only buildable on 2019.1¿). This forced us to use the new Cardboard SDK which lacked many of the features that were provided with the GVR SDK. In-class demonstrations relied on the older deprecated SDK and the newer Cardboard SDK had little to almost no documentation. To make matters worse, all online tutorials/guides we found were based around the deprecated SDK. Another issue was assets, namely that we could not use high-poly assets as they made compilation and build time insanely long. These were two problems that we had to tackle very early.

We tackled the first problem by completely removing the old SDK from our project and relying solely on the new Cardboard SDK. While the old SDK had cool features like an ingame emulator, reticle, and a good raycasting system; it was simply not feasible to downgrade our current Unity version and risk breaking everything. Recreating ingame emulation and a reticle was quickly achieved through other means like scripts and UI objects, but a custom ray casting system proved to be troublesome. In the end, we were all able to figure out ways to either use proximity interaction, or boolean interactions with the Cardboard SDK. These interactions will be discussed more later in the paper.

The second problem was a bit easier to resolve. We just had to delete large assets and only use low poly assets so as to not overwhelm our scenes with too much to process. This meant that the scenes would look emptier than we would've liked, but this was necessary to prevent crashes, performance issues or long build times.

As we neared the finish line for Seasonal Adventures, we were surprised to face far less technical problems than we anticipated. This is due to a combination of early design choices and troubleshooting on our part. By limiting what we are adding in each scene whether its assets or interactions, the game never suffered any performance issues. Good performance ensured that we were able to deliver our work on time and without cutting anything out. For us, having a stable and smooth frame rate far outweighs any stunning visuals, especially when dealing with VR.

B. Design Requirements

Having the game played on Cardboard VR meant only one button thus we kept this constraint for our scenes. The first interaction we needed was movement which we developed two methods for. The first was automoving where the player would move continuously at a fixed speed towards whatever direction he or she was looking at upon loading into a scene. The second was trigger movement, which meant that the player's movement is activated upon pressing the trigger button and is stopped when the button is pressed again. Both methods of movement rely on head tracking and we incorporated both of them in our respective scenes since each were suitable for certain situations. Fig.1 shows the player moving towards a rabbit in one of the fall scenes while their head is tilted to look in their direction.

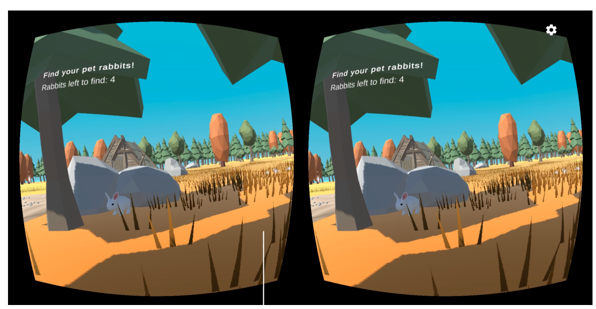

Fig. 1. Player moving towards rabbit in fall scene

Other interactions in our scene came in the form of moving towards and then triggering or colliding with objects and pressing the trigger button in certain situations.

C. Input Device

This game was built and meant to be played on iOS mobile devices with a Cardboard VR headset. As such, attempting to build and test the game on other platforms will likely result in issues.

V. Game Development

A. Assets Choice

We kept the low polygon theme throughout this project. We consciously made this decision after realizing how important performance was compared to visuals. Having low frames is excruciating to deal with, even more so when this project is built primarily for Cardboard VR. We also wanted assets that were not too advanced or complex to use. We used free low-poly assets from the Unity store with a nature theme, with our overall scene landscape from a pack called Low-Poly Nature Assets. We also utilized some useful prefabs from the pack to decorate and improve the look of our scenes.

B. Designing of Scripts

The project currently has over a dozen scripts ranging from basic movement, camera free-look, trigger button pressing and collision interactions. Our most essential scripts are of course the movement and camera scripts which allow the player to both traverse and look around each scene. Algorithm 1 showcases the trigger movement player script which is one of two scripts/methods we used to implement movement.

Algorithm 1: Player Move

```
function start ();
Input : movement switch, speed
function update ();
if (movement) then
    move player/camera forward by delta time *
      position;
end
```

Another algorithm involved ball traversal physics which was more geared towards the Summer and the Winter scene. Namely, the accuracy and overall ‘feel’ of the ball when thrown as seen in the Summer scene. Implementation began using the basics of physics which is also conveniently applied in Unity via distance, x-axis, y-axis, gravity, and position. Vertical motion was achieved using V = g.t whereas horizontal motion was achieved using X = V.t; both of which worked, but not as well as it should have. By trying to replicate real world physics, the game ended up feeling too stiff and static, even if the idea was to make it dynamic. As such, the final implementation of the ball physics ended up being a simple yet effective one. Please note the following is only a snippet of the algorithm, as ball holding was combined with throwing into one. For further reading, please refer to the Github under Scripts/Summer/PlayerGrab.cs.

Algorithm 2: Player Grab/Throw

```
function update ();
if player is holding the ball and trigger is pressed then
    release gravity;
    safety boolean trigger set to off;
    Prevent position sticking to player;
    Ball Velocity = currentCam.rotation *
      forwardMomentum * handPower;
    Safety boolean trigger set to on;
end
```

Instead of using outside gravity, Unity’s built-in gravity was used alongside the built-in velocity of the player. This natural velocity combined with the natural gravity of the scene makes it so that the ball will travel in a somewhat dynamic yet natural pattern.

VI. Game Guide

A. Control and Navigation

The game is controlled via trigger or auto movement combined with head-tracking. The player begins in the main scene of our game which is basically a portal hub and can then navigate to the other scenes by walking over and colliding with the portals. There are five different portals leading to five different scenes.

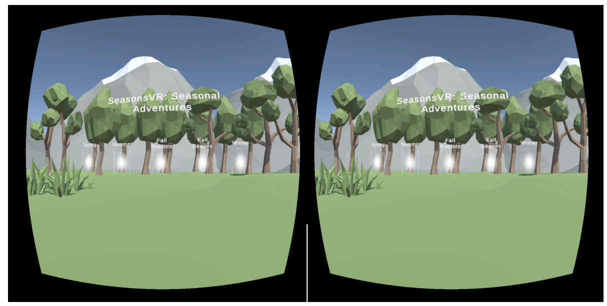

Fig. 2. The main scene at the start of the game

The player can activate and stop movement in some scenes but will be constantly moving in others. Speed and acceleration of the player is increased in some scenes when certain conditions are met. For instance, in the spring scene, apples increase movement speed and in the spring, fruits increase both movement speed and jumping power.

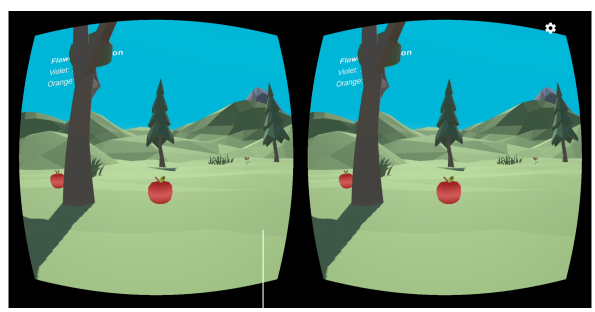

Fig. 3. Apples in the spring scene

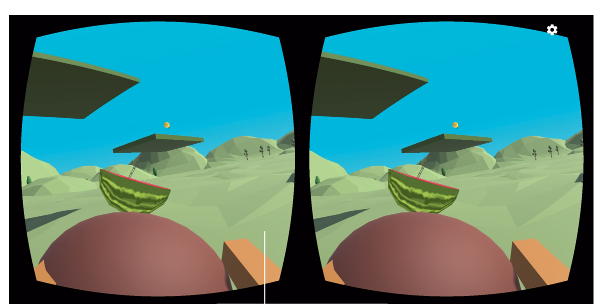

Fig. 4. Player looking at a fruit in the summer scene

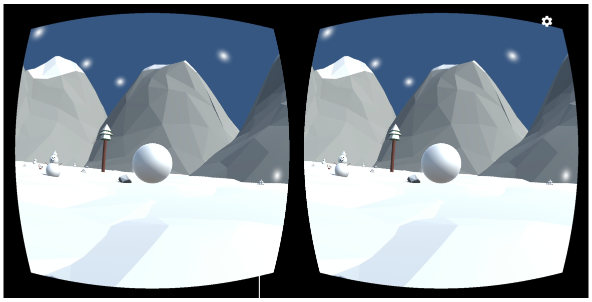

Fig. 5. Holding and aiming a snowball in the winter scene

B. User Interface

In some scenes, a user interface was provided so that the player is able to track important information related to their activities. For instance, in the spring, the UI tells the player how many flowers they have yet to collect and in one of the fall scenes, it tells the player how many rabbits they still need to catch.

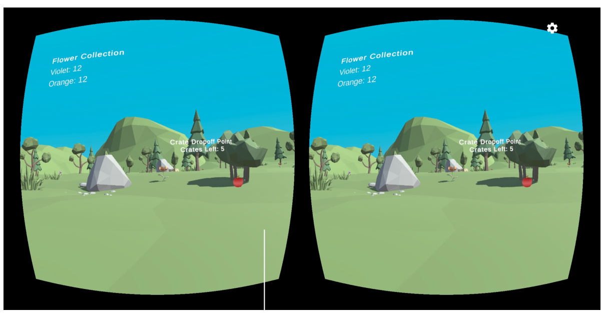

Fig. 6. Spring scene UI elements

VII. User Evaluation

A. Peer evaluation

Throughout this project, we as a group met periodically to give progress reports on our scenes and talk about any difficulties we were facing. In our first meeting, we brainstormed what kind of interactions we wanted for our individual scenes and then we moved onto working on our individual scenes while providing updates and getting feedback from each other on development. For the final demonstration, we were able to make five interactive scenes as per professor instruction and feedback during the midterm presentation.

B. Results and Analysis

There were some rough roads we had to go through while working on each of our scenes. For our summer scene, a major challenge was to get ball physics to behave correctly. This meant getting the ball to launch and fly in a trajectory that is not linear, via hit scan type feel, but rather a more dynamic manner. Various methods and formulas were added and tested to try and exemplify complexity of code but this was not always a good idea. Complexity does not always mean effectiveness, even more so when dealing with something like physics. Instead of convoluted and complex equations, a linear dropoff point for the traversal of the ball became the best solution to this problem. For our spring scene, it was important to understand the difference between triggering and colliding with objects in 3D space, sometimes you want rigid objects that react to gravity but sometimes you also want to keep them fixed in an area so we had to be careful about what attributes we assign to our objects. The other problem we faced was that most of our teammates had an iPhone as a mobile device so we decided that we are going to build and test the game exclusively for iOS devices, the caveat is that we need a Mac device in order to deploy our application onto an iPhone device. The Macbook that one of our teammates had couldn't handle big asset imports. Big asset imports such as large texture files and models with large amounts of polygons made the computer freeze and made it impossible to build and run the project. As a result we set a limitation on how big our project can be and what type of assets we can import. Regardless, we were still able to create a sufficient experience in each scene.

VIII. Discussions and Conclusions

A. Conclusions

In conclusion, this report summarizes how we developed an interactive game in nature to help alleviate stress. While working on this project we were able to get a deeper understanding of VR and how it can be applied in improving mental health and wellness. Additionally we learned a lot about creating VR experiences through Unity. We're excited and eager to learn more about the different use cases of VR in different sectors for the benefit of humanity.

B. Future Work

For future work, we would like to add even more interactions in our scenes, to improve the scenery and the visuals and to possibly create a version of the game for better VR hardware. We want to turn our game into an online multiplayer experience where the player can meet other players and explore nature or do activities with them. The possibilities are endless.

IX. Resources

Our project files can be accessed through the Github repository: https://github.com/ahmdaa/SeasonsVR A live demo of our project is hosted on youtube through this link: https://www.youtube.com/watch?v=Z1Fc1ePmoJU

References

[1] W. P White, Nicola L. Yeo, Peeter Vassilijev, Rikard Lundstedt, Mattias Wallergard, Maria Albin, and Mare Lohmus. A prescription for "nature" – the potential of using virtual nature in therapeutics. In *Neuropsychiatric Disease and Treatment*, 2018.