**Ordinary Wizard Level (OWL’s)**

**Team 8**

**Section 22**

**Braeton Buckley, Jennifer Shin, Chris Yee**

**Abstract**

Our project is a Harry Potter themed survival game, where the player must use their wand to defend themselves against hordes of skeletons in Gringotts. The skeletons spawn randomly from the fog surrounding the player, with the amount of skeletons spawning increasing over time. In order to cast a spell, the player must swing their wand (right controller) rapidly in a large motion and then point their wand at where they want to aim. The game uses various rock meshes, particle effects, and sound effects to help immerse the player into the game.

**Introduction**

Our project is about allowing the player experience what it would be like to be a wizard/witch in the world of Harry Potter. The game is set in Gringotts bank, which was recently under attack. Trapped and alone deep in the bank’s caverns, the player must use their magical abilities to survive as long as possible. Using the Oculus controllers, the player can wield their wand to cast spells against the incoming skeletons. They can also look at their spell book in their left hand, which lists their score and health. Surrounded by fog and incoming skeletons, the player must survive as long as they can before they are ultimately overrun.

The game uses a unique mechanic in order to cast spells. We implemented a method similar to the “swish and flick” method of using a wand in the books by forcing the player to make exaggerated movements to cast their spells. Also, we spent a lot of time in looking for free assets and sound effects to help the player feel more immersed in the game. The spell the player casts, incendio, is a blue plasma explosion particle effects combined with a fire explosion. There are various fire effects scattered around the terrain, and there are multiple rock meshes to simulate a cavern. We also added sound effects to the spell casting to add more impact to each attack. Along with the background music and the surrounding fog, we placed a lot of effort in making the atmosphere of the game match the setting we wanted.

There are multiple games from the Harry Potter universe, but very few are in first person, much less VR. Thus, allowing the player to virtually hold a wand and cast spells allows the player feel like an actual wizard/witch. Especially when combined with the particle and sound effects, the spells feel powerful to cast. VR also allows the player to feel some fear with getting attacked; the skeletons are the height of the player, and they run up next to you and start swinging their weapons in the player’s face. This occasionally causes the player to panic and rapidly swing their remote in order to cast spells at a faster rate, which due to us using VR, can lead to “gorilla arm” fatigue.

**Related Works**

Our project was originally inspired by the Harry Potter series, hence the player being a wizard and the title “Ordinary Wizarding Level” being a reference to the name of the standardized test wizards take in the world of Harry Potter. We originally wanted to make many aspects of our game from the world of Harry Potter, like make the environment of our game look like Gringott’s and travel through it with a cart, have the player be able to casts different kinds of spells like “stupefy” and “wingardium leviosa,” and have different kinds of creatures such as dragons or dementors in our game. However, we were probably a little too ambitious, and ended up making a generic wizard game, or an “Ordinary Wizarding Level.” The environment was cavelike, like Gringott’s, but there was no cart, the spell was a lightning fireball never before seen in Harry Potter, and the only kinds of enemies we had were skeletons, but we were at least able to stick to the theme.

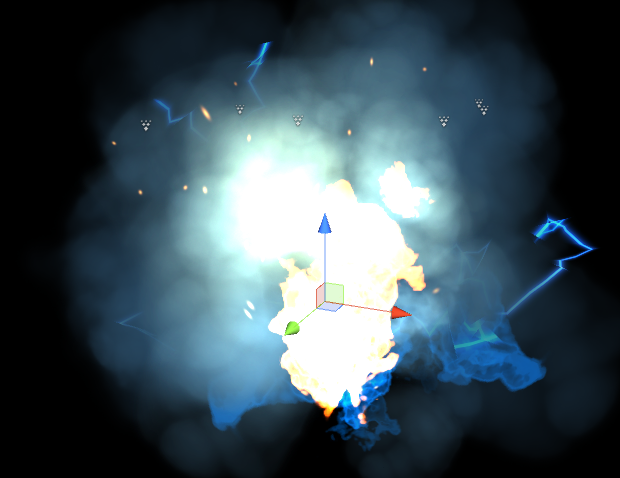
We took our inspiration from many games, namely the Fantastic Beasts VR game, and the other Harry Potter games on PC and console. Our original design to draw hexes in order to cast spells, but after not being able to figure out how to track if the user drew a specific symbol and then not being able to make a gesture recognition asset work, we had to scrap the idea. We then took inspiration from the movie-based games made, namely The Deathly Hallows games, where the characters cast powerful spells with a small flick of the wand. We felt for a survival type game, this type of combat was best for our game.

Elven Archer, a game we were able to play as a demo on the Oculus, also inspired our project in some ways. This game was one of our first games aside from the tutorial levels on the Oculus. We liked how it took some amount of skill in order to aim your bow and hit the enemies running past you. Though more related to our original idea of a cart going through Gringotts and firing at enemies you pass by, we took the idea of hordes of relatively weak enemies and skill-based aim into our final product.

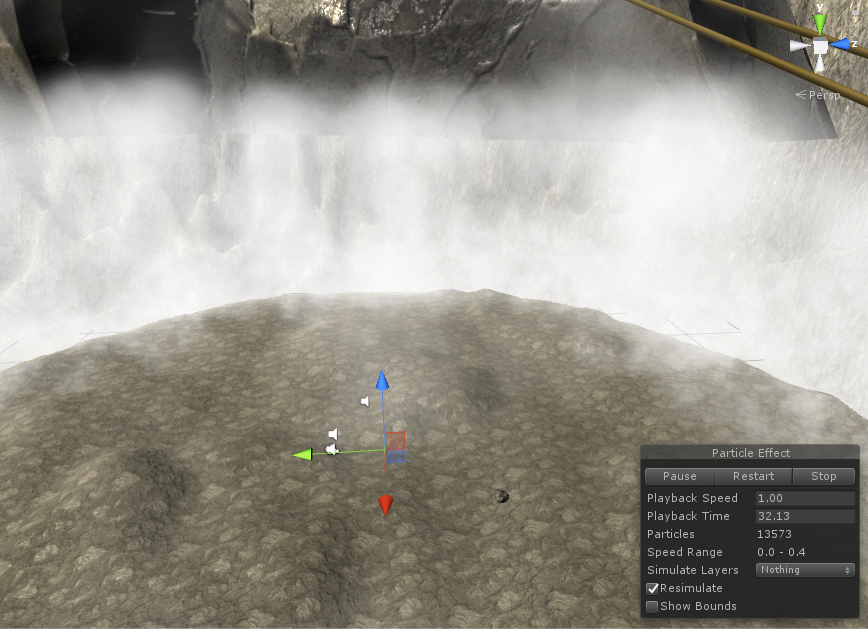
When looking at some of our classmates’ projects, we can see that the core aspect of our game is not very original. It is a basic VR FPS where the player stands in the center while enemies appear from all sides and walk toward the player. Several VR games like this exist, and it makes sense, because they are exciting to play and relatively easy to make. The aspect that makes our game unique is that the weapon is a wand, so in order to attack the player must make a wand-like movement, flicking the wrist, in order to cast a spell. We had many problems with programming this movement, which will be talked about in other sections, but in the end we were satisfied with the result, as were many of the people who tried our game.

**Design**

**Spellcast Effect**

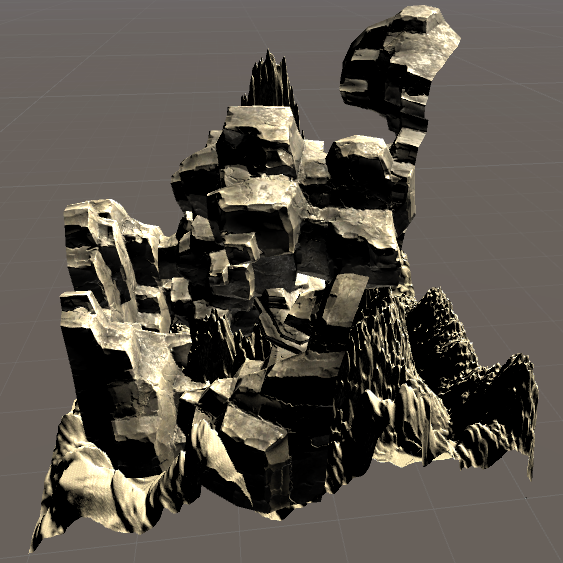


**Fog Effect**



**View Above Player**

**Outside View Of Entire Terrain (Player is placed inside)**



Our game was played entirely using Oculus controls, mainly because we assumed those were the requirements of the project, and designed our game with that requirement in mind. Either way, our game was optimized for the Oculus control system, and it would not have worked with keyboard or mouse. The core aspect of our game, as mentioned previously, was being able to cast spells using a wand, and this was much more enjoyable and interactive using Oculus motion controls. Using a keyboard and mouse to cast spells would result in this game being exactly the same as any other FPS game.

We only used one button in our game, the right hand secondary trigger (index finger button), no other buttons were needed. This button, when pressed, would begin recording the motion of the right hand controller. When released, speed would be calculated based on distance over time from the point where the button was pressed to the point where the button was released. Left hand motion was tracked, as the player was holding a book that would display the player’s health and score. The player could only rotate, which is taken care of by the Oculus headset, so there was no need for the use of the thumbsticks. If there were more types of castable spells or other in-game actions, more buttons may have been required. We did try to have the “wingardium leviosa” spell, which was working, and could move objects the wand was pointing at. In the end, this feature was removed because it was too hard to hit enemies with this spell unless the objects were obnoxiously huge, and this spell was made obsolete by the flame shooting spell. If we were able to implement more spells, we had ideas to cast different spells by turning the page of the book before casting a spell, or drawing different symbols in the air with the wand, but in the end decided that these would more difficult to implement.

Although we did not put in too much active effort into following the Oculus Best Practices guide, we subconsciously followed most rules out of common sense. We used default rendering settings because we were not advanced enough to do things like make our own distortion solution. Head-tracking was always active, and the display responded to user’s movement at all times. Positional tracking was not broken because the player could only rotate and not move around, and no one who played the game tried to look under the terrain. All moving objects in the game moved with a constant velocity, so there was no acceleration to disorient the user. Neither was there any camera zooming to disorient the user. The health and score UI was affixed to a book in the player’s hand, which was easy to read. Sound was automatically modified to match object distance using Unity’s algorithms, and all players used the Oculus headset to play sound. Overall, there was minimal problems with users feeling disoriented or experiencing simulator sickness.

There were some problems we did have though. Because we increased the spawn rate of enemies over time, when there were too many enemies rendered at once, latency would continue to increase until the game became unplayable. We left this in the game since this did not occur until playing the game for a long time. It could also become disorienting if the player casted too many spells too quickly, resulting in many swirling fire and lightning particles being displayed in front of the user. We left this in since this was caused by the player’s own actions, and they could simply decide to cast spells slower if it was disorienting them. Finally, since the enemy skeletons did not use gravity, they simply floated above or went slightly underneath the bumpy terrain as they ran towards the player. This was not noticeable at a distance, and only if the user was paying close attention when the skeletons got close to the player, so it was also left in.

It was important for the player to have enough space around them since they were constantly flicking their wrist and would otherwise risk hitting someone or something. As mentioned before, if the player played for a very long time to the point where too many skeletons were spawning, or if they casted too many spells too quickly, the game could start to lag. Overall though, the game performed with minimal issue.

**Implementation**

Our project was implemented using basic Unity features and scripting. To create the player character, we used a free asset to have a local avatar that tracks the user’s oculus controller and displays their hands. We adjusted the position of the fingers to make it look like they were holding the wand and book in each hand, and didn’t add the feature to allow the local avatar to track what fingers are pressed down to prevent dropping the wand and book. All enemies would run towards the player’s position and hurt the player every three seconds if they came into contact. The wand has a small sphere which acts as a fire point, and it shoots out an invisible sphere with particle systems attached to it. There are spawn areas places around the player, so the player must physically turn around to defend themselves.

Our main spell script, Stupefy, works by calculating the distance/time ratio from when the user presses the trigger to when they release it. We compare that ratio to a constant, and if they pass the attacking “threshold,” then a sphere prefab with the particle effects is instantiated, with its velocity set in the direction of the wand. The enemy scripts deal with time, and spawn or attack based on how long it has been (for example, the 3 enemies spawn every 10 seconds).

**Open Source Code and Assets**

- Brackeys ray-casting tutorial videos

- Oculus Integration - Local Avatar(Oculus hands)

- DLD - Arnis sticks

- Robotic Rainbow Studios - Grimoire Style Book

- Editable VR hands

- BSHGAME - Skeletons pack

**-** TRIPLEBRICK - Free Rocks

- Kanistra Studio - Modular Railing Set

- Unity Technologies - Unity Particle Pack

- Marcelo Fernandez Music - RPG game music

- Little Robot Sound Factory - Fantasy Sfx

One of our main issues was that during the demo, the aim on the user’s left side was off. It may have been with our scripts, but the aim this changed a lot based on the sensors. Since it was so unpredictable, we decided to simply turn off the spawn points in those directions for the player. Another thing that we had trouble fixing was where the skeletons attacked. Based on the player’s height and where they stood, the skeletons could attack an area around the player, not the player itself. We had to adjust an invisible cylinder that we used as the hitbox for the player everyday. Lastly, we found that the aim of the spell could be extremely difficult since when the player releases the trigger and points where they want to shoot, the player’s physical would shake slightly due to the physical action of moving their arm fast and then suddenly stopping their arm at where they want to aim. The player would have to be extremely calm to aim correctly, which we found to be awkward for the player to do while being charged at by many skeletons. To fix this, we move the fire point a little bit in front of the wand, to reduce the “cone of error” aim from the player.

**Lessons Learned**

Making this game was eye-opening to how VR games are made and we learned a lot. One of the big things we learned is how crucial it is to make the game not nauseas, or else it’s literally unplayable. An obstacle we faced is that we couldn’t get the main game mechanic of our game to work: drawing spells in the air. It proved to be too difficult for us in the short amount of time we had. Thus, we decided to change our path and do the closest thing we could to drawing; we made sure the player was moving their wand when casting spells. How we did this is we simply calculated the distance and time at which they moved the wand. Moving the wand fast enough in a time frame allows the player to cast a spell.

To our surprise, many people liked our game and it made us very happy. They especially liked the particle effects of our fire ball, music and sound effects, the whole magic aspect, good graphics, and how our game is not as nauseas compared to other student’s games. We think our game was successful with the audience because it is unique; other student projects did not involve magic and did not have the flashy particle effects which caught everybody’s eyes. The parts they didn’t like is the repetitiveness of our game and how it gets boring over time. To improve this, we would want to add more spells and challenges. For example, we could add a blizzard ball; the player would have to cast a blizzard ball to kill blue enemies and fire balls to kill red enemies. Also, we were going to add a “Wingardium Leviosa” spell where the player can move rocks on the ground to hit enemies. However, we were not able to solve the bugs that we faced with this spell and scrapped it from the final game.

On the topic of the workload, we did most of the work together. However, in our own time, Chris focused on the map, Braeton on the enemy AI, and Jennifer on the wand mechanics. Still, 70% of the game we worked on together.