Develop Hardware Familiarity (VR)

- ❖ What are our options when it comes to VR headsets/hardware?

When doing research for VR hardware/VR headsets, we found that there are quite a few different options when it comes to picking a headset, however all of the headsets can be categorized into 3 separate groups:

The first is more of an original design, something that's middle of the road when it comes to price and quality, and is generally regarded as the most commonly sold. Not necessarily the "best selling", but one most people think of when they hear about a "VR Title". Hardware and headsets in this tier usually cost anywhere from $800 to $1500 depending on the seller, quality, and additional attachments (storage space) that the headsets come with. Unlike mobile headsets (another topic), these headsets don't typically run on their own, and in addition to their hefty price tag require a PC capable of running VR games and environments.

Next is probably the most well sold when it comes to headsets and that is the mobile headsets, specifically the line of Oculus headsets. What sets these apart from the previous category are two things. First, the price. These headsets generally run between $300-$400 depending on the speed/size of your storage capacity. Because of this, it tends to be the most commonly bought headset due to the lower barrier of entry. The other reason is due to it running games/software natively (the reason for this is due to these headsets having their own independent software store (the Facebook/Meta store) as compared to a universal store (like Steam) that will run games regardless of the headset.

Lastly there are the exclusive and elusive headsets. These headsets offer the advantage of higher refresh rates and wider/deeper FOVs. However these devices lack a reasonable price tag and practical mobility, and due to that reason there wasn't much consideration taken into developing and/or testing for our current project iteration.

- ❖ What VR headset is best suited for our needs/our game?

For our purposes, cost and mobility are probably our two biggest factors. Cost for obvious reasons, but mobility due to how our current project is designed around movement. While I (Tyler) have two headsets of different styles (tethered and untethered), I believe going with the Oculus Quest 2 is our best option, as the combination of mobility, being able to run onboard .exe files, and the low consumer cost makes the most sense for us.

Desktop to VR Integration

- ❖ How do we plan on translating our MVP to VR?

Currently, our game exists as a 2D "proof of concept", where the main concepts are there but the implementation doesn't achieve any of our future goals (story, generation, tone/mood, VR functionality).

We plan on implementing it in multiple phases. Now that we have a "proof of concept", our next step is to start designing the 3D models and assets used in our planned 3D environment. Then we'll start piecing the environment together while taking into consideration VR constraints over typical gaming constraints. Finally we'll switch the controls from binary Keyboard/Controller inputs into motion based controls/inputs.

- ❖ What would be the biggest change/challenge for this?

Quite possibly the hardest challenge is going to be developing our product into not only a 3D environment, but also an environment developed for the use of motion controls. This is because with motion controls, it's not like a keyboard/controller where the input is binary. With motion controls every "hit" has to take into account the angle, speed, direction and force with which the user is hitting the "object".

Motion controls are something that we plan on tackling in a much later sprint, but as it stands it does remain something we objectively would like to include. If we were to look at our primary focus when it comes to "Desktop to VR integration". is to develop a fully 3D environment that works with our existing binary controls.

Hardware Research (VR)

As compared to "Hardware Familiarity", our research for "VR Hardware" deals more with the input of the controls as they are the tools to which you interact with the game world. Currently (and the simplest way) is to use binary inputs on a keyboard/controller, however in the future the plan is to translate our input style to a hit detection system that takes advantage of the speed/degree/angle of a user's movement.

While motion controls are a topic that we plan on doing further research on as the project progresses into a more complete 3D environment, our research provided us a good understanding of what the requirements are for motion controls, and how we can design our logic/assets/environment around it.

Software Research (VR)

- ❖ How do you best develop in a 3D space in VR, with potentially 2D planes having to adjust to face the player at all times?

Unity has worked hard to make VR feasible in their development suite. They offer at least some support for these platforms.

ARKit

ARCore

Microsoft HoloLens

Windows Mixed Reality

Magic Leap

Oculus

OpenXR

PlayStation VR

For all of these platforms, the best way to develop the 3D space (when creating Euclidean spaces) is to use a normal Unity project in the 3D mode. If space warping or mapping is needed, this problem becomes more complicated, but neither is needed currently for this project

The process of 2D planes always facing the player is called bill boarding and it can be implemented many ways but the most elegant way to implement this feature is by not having the spite update its vector in the axis that it is to rotate upon when compared to the camera orientation.

An example of bill boarding implementation by online user Robertbu:

```
var v = transform.position - Camera.main.transform.position;

v.y = 0.0;

transform.rotation = Quaternion.LookRotation(v);
```

❖ How do you attach the Unity camera to the VR headset?

Because Unity has built in VR settings, all we have to do is enable VR mode, and this will automatically:

- Render to a head-mounted display
- Track head input (movement and angle)
- Set the camera to the VR