HUMAN

New Technologies

Lisanne Bakker - Game Design

Tika Sara Kroon - Game Art

Akane Motomura - Game Design (Exchange)

Pepijn Kok - Game Development

Technology 1: Microsoft Hololens

The Design:

For the microsoft hololens, we thought of an experiment where we made a program that could recognize and display info about objects you see in the real world. So for objects like camera's, or things like boxes, lockers, doorways, or any object the hololens could recognize, it could display simple info about it. We got this idea from the "WatchDogs" game's display, where the player can hack into anything and display info about people, streetlights, interfaces, and so on.

The Execution:

When we started we borrowed a hololens and tried to set it up. We searched online for a way to make and upload projects to the hololens. On the microsoft hololens website there's a simple setup introduction and on youtube we found a video which could help us with making games in unity and uploading them : https://www.youtube.com/watch?v=OYx4qIqi0oI .

This video made some difficult arguments and we had some trouble uploading to the hololens. Apparently, you need to have windows 10 personal edition to run the right system for the hololens. Luckily, personal edition is free for students, so that wasn't a problem. Next you need to activate some systems in your windows, like Hyper-V, to work with the hololens. and at last, you needed to have the latest of visual studio community edition. The rest of the setup was fairly easy. You could make a game in unity and build it in a "Universal Windows Platform". Then you had to open your unity project in visual studio, and then run it in a remote debugger, input the code you get in the hololens developer tab in settings, and then the game is run on the hololens. After returning the hololens we downloaded a simple hololens emulator so we could work without the hololens for convenience.

For the rest of the concept we didn't actually make a system that scans objects, but we did found a very interesting video from a "NDC conference" in Sydney, August 2017: https://www.youtube.com/watch?v=LxEl7DN5uRg .
In this video, Steven Carter explains the possibilities of using multiple recognition programs to achieve what we are trying. The best solution for us is to use Vuforia because it's a reasonable good system that is also free for recognition to about 1000 objects. To use Vuforia we'll have to create an account, and get a free developer key to upload our objects to the database. These object are what the hololens will have to recognize with it's camera, and we're gonna use the HoloToolKit SDK for that.https://github.com/Microsoft/MixedRealityToolkit-Unity/releases . Next we'll have to import the Vuforia Unity Package, and the Target Database to add target prefabs from that database to our scene. With this the hololens will be able to scan the objects we set in the Vuforia database, and we'll be able to display information about those objects.

Technology 2: Tobii Eye Tracker

The Design:

Next we thought of making a game where your eyes are the only thing that control the player and the environment.

This concept was taken from real life human “eye interactions”. For instance, what social interactions already exist between humans using the eyes alone? This can be winking, staring, avoiding eye contact etc. We wanted to implement this in dialogue the user would have with NPCs. If a player does not know what it means, you can move your eyes in a circle and the NPC will know you are either annoyed or confused. Winking could add the possibility to flirt with a character. Sadly, we could not make the winking possible for the developers of the tobii eye tracker have disabled functionality that allows detecting winks.

The Execution:

When making the game, we first had to download a sdk for unity, which included some templates to work with the tobii. but other than the hololens, this was all you had to do. The tobii gives one simple Vector2 of where the eyes stand on the screen, which we could use in any way possible. We could make a prototype without the tobii by using the mouse, because that too is just a Vector2 on the screen. We made a system where the player could look left and right by looking at the sides of the screen. they could also look up to look a little up and down, and if they looked at specific spots on the ground, they could teleport to the place they looked at. of course this won’t always happen by looking at the ground. The player has to look at given places on the ground to teleport there, to make some restriction of where the player is allowed. For the rest of the game the player can interact with anything by looking at it for a given amount of time. But, when it came to prototyping with the tobii we found a slight problem. the tobii does give it’s Vector2 right, but it’s accuracy is a little off. because your eye is not as stable as a mouse, the teleport might glitch a bit, or the player might move around too fast without them wanting to move that fast.

Task Distribution:

Lisanne Bakker:

- Level Design
- Interaction Design

Tika Sara Kroon:

- 3D modeling: Modeller of the bookcase, books, pencil, pencil holder and human model

- Tried to fix animations with miximo but in the end it didn't work

Akane Motomura:

- 3D modeling
- thinking about how the player can move in the game

Pepijn Kok:

- Game Development

Problems, Solutions, Exploration:

Extra's and Future Expectations:

For both concepts we made some experimentation and we created some games, but we also thought of maybe a way to use both technologies together. a small concept we thought of was an ar escape room, where the hololens creates a room of an already scanned area, and it makes an escape room scenario. and in combination, the tobii eye tracker checks if your eyes are on any object so it knows you're looking at the object. And if it knows what object you're looking at, it might be able to manipulate or animate that specific object, the moment you look at it.

Both the hololens and the tobii eye tracker are great objects that still need some work. The hololens needs to have a more simple way of uploading programs, and i might be simpler to use if it had a sdk like the tobii eye tracker. The developer kit for the hololens can be improved, but for the tobii eye tracker, the system has to be more accurate if you want to use it with precision.

We also tried using the manus vr gloves, but we sadly couldn't the software to run the gloves and to create games and apps, because to use them you had to pay for the software. A bit of a shame though because we had high hopes of using it in correlation with some vr projects. We thought the leap motion hand tracker might be "handy" too, but just like the tobii eye tracker, the Leap motion was not precise and very glitchy.