VCS2-VR71

Own proposal

Hugo Leenen

2018

Content

[The assignment 1](#_Toc504216882)

[Activities 2](#_Toc504216883)

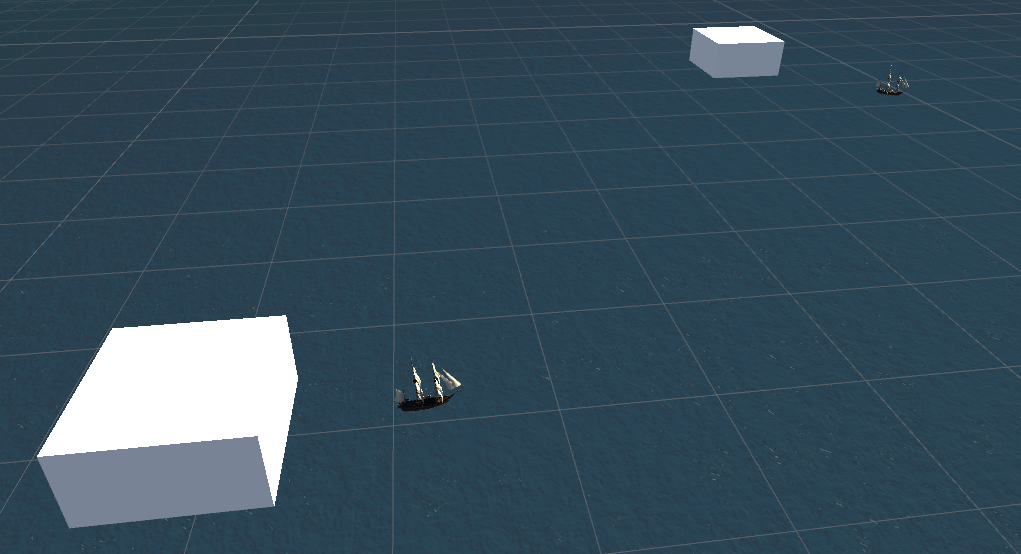
[Conclusion 4](#_Toc504216884)

[Hour registration 4](#_Toc504216885)

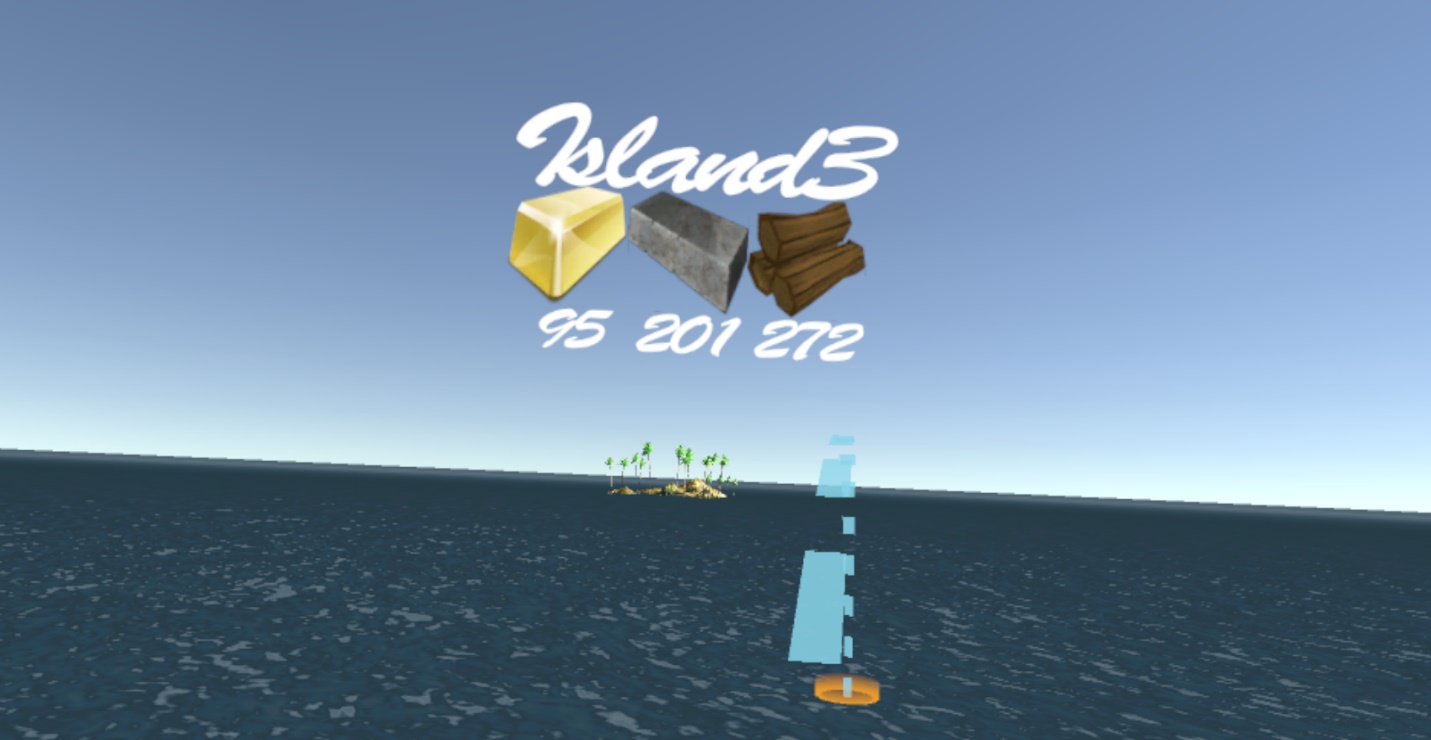
# The assignment

I want to create a virtual environment were the user is able to view and control different trade routes between islands. The purpose is to create different means of interaction for instance the user should be able to give a ship a name so texts needs to be put in. When you are in Vr there needs to be special solution because it is not easy to access a keyboard. This project is about trying out different input methods in Vr. And looking at ways to replace traditional input like mouse and keyboard with the Vive controllers.

# Activities

I started the project by realizing a basic prototype were a user can observe the different ships going from island to island. Very simple the ships randomly sail to an island and when they are there they go to another random island.

When I completed this I started with the Vr navigation aspect of the application as a Vr user you needs to be able to zoom in and out to get a better view of the world. A problem I tried to solve for a while is when you zoom out (by scaling the camera rig) and the user is standing on the outskirts of the play-area it seem as if the user is moved diagonally. This could be solved by “moving” the play area beneath the player but when I did this it caused a disorientation feeling for the user.   
The way the user can control his zoom level is by pressing the grip button and moving up or down.  
When the player does so the zoom level gradually changes overtime when the user lets go of the grip buttons the zoom level stays the same.

After implementing the zoom level I started to work on the teleportation. I did this before during the VED and the V3D courses. This process works the same as for an application that would not focus on world viewing. The user can hold right on the Vive trackpad a dotted line appears showing where you will teleport to. When the user lets go a blinking animation plays and the user will stand where he was pointing.

After I realized navigation around the world the player was able to see everything in the world.  
I imported some basic assets to make it look nicer and started to work on the UI.  
First I gave every island a name tag which I displayed above every island. I gave every text element a script that rotates is to the camera like we learned in the V3D course this works great for VR applications you can notice it but it’s always readable. Besides a name I decided every island will store 3 recourses: wood, iron and gold. The numbers are randomly determined it was the intention that you could order boats around to transport these resources. So the player should be able to look at what a ship was carrying so I added a similar UI layout to the ships as visible in the picture below.



A problem that the different UI elements caused was that they needed to be big to read while wearing the Vive headset. If you looked at a strange angle the different UI element would overlap and also were quite in your face, also the player would only look at UI elements and not the game.   
To solve this I decided to hide the different UI elements until the player would select one.  
This could be done by pressing left on the trackpad on the Vive controller. A red selecting line will appear and the user is able to click left again on the trackpad to select the island or ship.  
To hide the redline and disable the selecting mode (as I called it) by holding left on the trackpad.  
This was the first time I really worked with the trackpads but it was really easy.  
 When you press on the trackpad it gives a x and y value on a press with these values you can look if the press was on the left side or right side of the trackpad.

When I was able to select an island, or ship I decided to look into ways of changing the island name because this is something I am very interested in. A solution should not be used to write long text,  
but for instance to edit a text like an island or ship name. When the user selects and island or ship and clicks on the name of the ship or island a virtual keyboard should appear where the user can type the new name and confirm.

# Conclusion

I was not able to complete everything I wanted to make (that’s the problem of not having a Vive at home). But the process of designing an application in which you can move around more freely was successful. I had to think of solutions I didn’t have to think about before and that was exactly what I sought out to do. You can do very interesting things like gesture input controls for controlling the zooming level which I really liked. When I started I thought of a lot of restrictions the Vive controllers have (like inputting text) but when I was working on it I realized there is a lot of things the Vive controllers let you allow to do. Like displaying a teleport marker that shows you were you are going you could do that with normal mouse input but the Vive controllers make it way easier. I learned a lot about working with the Vive VR controller like the trackpad. I also really liked how I was able to display the UI information in 3d space.

# Hour registration

|  |  |
| --- | --- |
| Activities | Hours |
| Started working on a prototype were boats go from one island to the next. | 7 |
| Experimented with different ways of moving around in the world. | 5 |
| Learned about UI elements in Vr reading and testing things out described in [this](https://unity3d.com/learn/tutorials/topics/virtual-reality/user-interfaces-vr) document. | 3 |
| Tried different ways of displaying Canvas UI items to the player | 1 |
| Created a system where you can select an object so you don’t see the other UI elements. | 4 |
| Worked on a way of interacting with UI text so that you can input text while in VR. | 4 |
| Edited the selecting system so you can interact with different UI elements | 4 |
| Documentation | 2 |
|  | Total: |
|  | 30 |