The Cosmic Perspective:
Teaching Middle-School Children Astronomy
Using Ego-Centric Virtual Reality

PORTFOLIO

Jan Detlefsen

Supervisor: Rasmus Stenholt

Master Thesis
Spring 2014
Medialogy
Aalborg University

CONTENTS

1. SOFTWARE & EQUIPMENT

This chapter describes the software and equipment used for implementing the Virtual Reality application. The equipment used is consumer-grade and affordable which is important for cost-effectiveness. This will make it easier to justify its use in educational settings. The application is developed in Unity 3D which is a game editor -and engine. The application makes extensive use of the Space Graphics Toolkit which is elaborated in section 3.

1.1. OCULUS RIFT

The head-mounted display used in the VR application is the Oculus Rift HD prototype. The Oculus Rift HD features rotational head-tracking and stereoscopic display. This allows the user of the VR application to look around naturally and sense depth within the virtual world. The following technical specification is extracted from the Oculus Rift Development Kit 2 (Oculus VR, 2014), since the Oculus Rift HD prototype does not feature a list of technical specifications but should have specifications approaching the DK2. The display has a resolution of 1920 x 1080 pixels which yields a resolution of 960 x 1080 pixels pr. eye. Persistence is as low as 2 milliseconds. Persistence is a measure of how long a pixel remains lit and low persistence is extremely important in VR (Abrash, 2014) due to vestibulo-ocular reflex (VOR). The VOR stabilizes the eye on a fixation point during head movement by performing eye movement opposite the direction of the head movement and this is made more difficult by the directional blurring of long persistence. Head-tracking is performed by the Rift using gyroscope, -accelerometer -and magnetometer sensors and has an update rate of 1000 Hz. The Oculus Rift HD weights about 0.5 kg.

FIGURE 1: THE OCULUS RIFT HD PROTOTYPE

The Oculus Rift implements stereoscopic vision by dividing the screen vertically with one half of the screen for each eye. Each of the two pictures is then distorted as seen in Figure 2 in order to fit the view of the lenses and provide a wider field of view. All this is automatically implemented by the Oculus Rift SDK which also features plug-ins for Unity3D and other game development software. The Oculus Rift plug-in for Unity3D limits the frame rate of the VR application to 60 frames per second (FPS).

FIGURE 2: OUTPUT PICTURE OF THE OCULUS RIFT DISPLAY VIEWED WITHOUT LENSES. THE DISTORTION FITS THE VIEWING AERA WHEN SEEN THROUGH THE LENSES FITTET INTO THE DEVICE

1.2. MICROSOFT KINECT

The Microsoft Kinect is used to implement skeletal motion tracking for use in positional tracking of the users head and hands. Without positional head tracking the user's point of view would be positionally fixated, like having your head on a stick, and this would break the connection between the user's movement and the movement of the point of view within the application. Hand tracking is used for letting the user interact with 3D user interfaces within the application. Using the Kinect allows the user to move naturally between 0.8 and 3 meters away from the Kinect and approximately 3 meters sideways. However, the Kinect's tracking area is cone shaped as seen in Figure 4 and tracking accuracy drops when approaching these limits. The virtual environment in the VR application is designed to facilitate the user to stay within the trackable area of the Kinect. The Kinect updates the positional tracking 30 times per second (FPS) which is half the frame rate of the Oculus Rift. Early in development the Kinect plug-in was revealed to limit the frame rate of the entire application to 30 FPS which imposed noticeable lag in the display. This problem was solved by moving the Kinect functions to a separate thread in the source code.

FIGURE 3: THE MICROSOFT KINECT

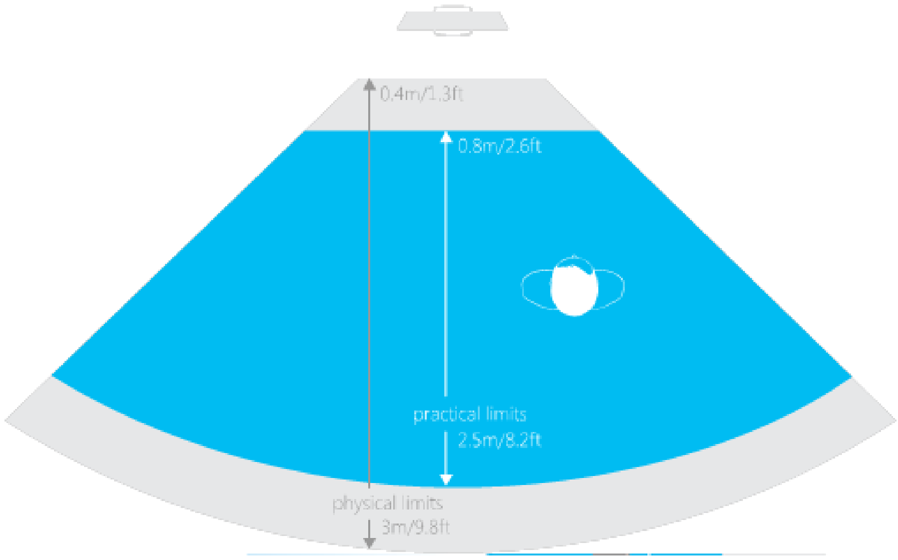


FIGURE 4: HORIZONTAL TRACKING AREA OF THE KINECT **(MICROSOFT CORPORATION, 2014)**

2. PLANET ROOM

The Planet Room is designed as a quickly developed proof-of-concept of the final application. The purpose of the Planet Room is to make a preliminary test of how middle school children react towards Virtual Reality and to get an indication of the effectiveness of VR as an educational tool. The results from the preliminary test can be found in section 6.1.

The Planet Room allows the user to walk within the tracking area of the Kinect, but does not include any other form of travel. All the planets are visible from the position of the user and all planets are scaled correctly relative to each other. The Moon is placed in orbit of the Earth in order for the user to experience the size of this familiar object. The Sun is placed outside the room itself since it is too large to fit inside. The user must therefore look backwards in order to see the Sun. The planets and the Sun use the Space Graphics Toolkit for graphics which is elaborated in chapter 3. The pillars serve an aesthetic purpose, but also allows for better depth perception. The pillars provide size comparison since they are all the same size and also provides motion parallax since the user is able to change the position of his point-of-view due to the Kinect. The pedestals on which the planets are placed also have equal sizes to each other and provide a line from the planet to the floor clearly showing its position. The Planet Room uses cast shadows for aesthetics and as an additional depth cue. Since the Planet Room does not change position or angle from the light source (the Sun) the shadows are baked for increased realism. The texture of the floor also provides a depth cue as the user can see less detail from the sections of the floor which is further away. (Ware, 2004) (Johnston, 2014)

The terrestrial planets are placed in front of the user and allows him to walk up and examine them up close since they are small enough to fit inside the tracking area of the Kinect. When the user is looking at Jupiter and Saturn the terrestrial planets, Uranus and Neptune also act as a foreground and provides parallax for increased depth perception. The planets have realistic tilt and rotates in order to visualize this tilt. The Moon orbits the Earth and always faces the same side towards the Earth in order to show that the Moon is tidally locked to the Earth. Rotation is not relatively correct since one day on Venus equals 243.02 Earth days (NASA, 2014). This would either make the rotation of Venus to slow to be noticeable or the rotation of the other planets to fast.

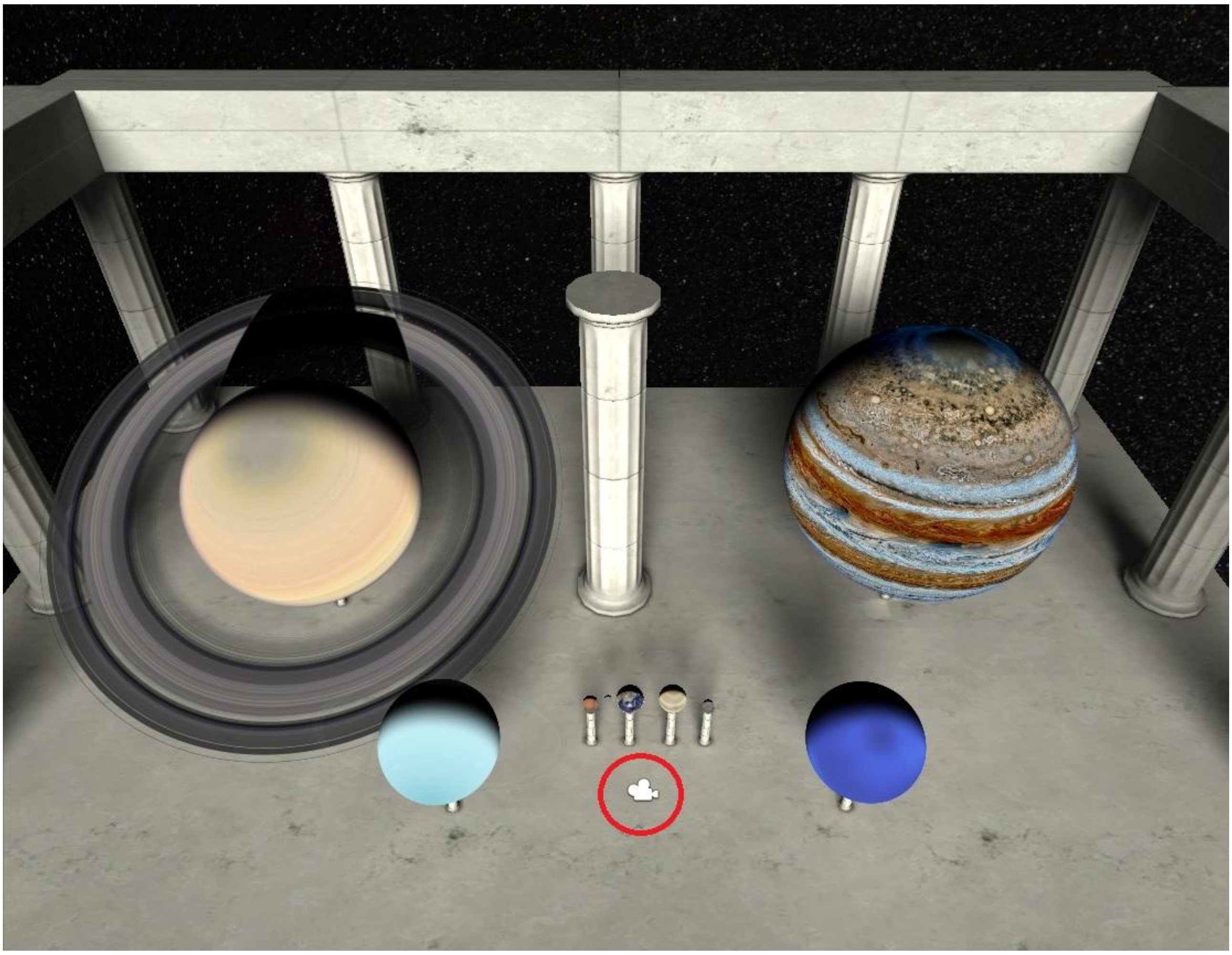


FIGURE 5: THE PLANET ROOM AS SEEN FROM ABOVE. THE USER IS PLACED AT THE RED CIRCLE LOOKING IN THE DIRECTION OF THE CENTRAL PILLAR

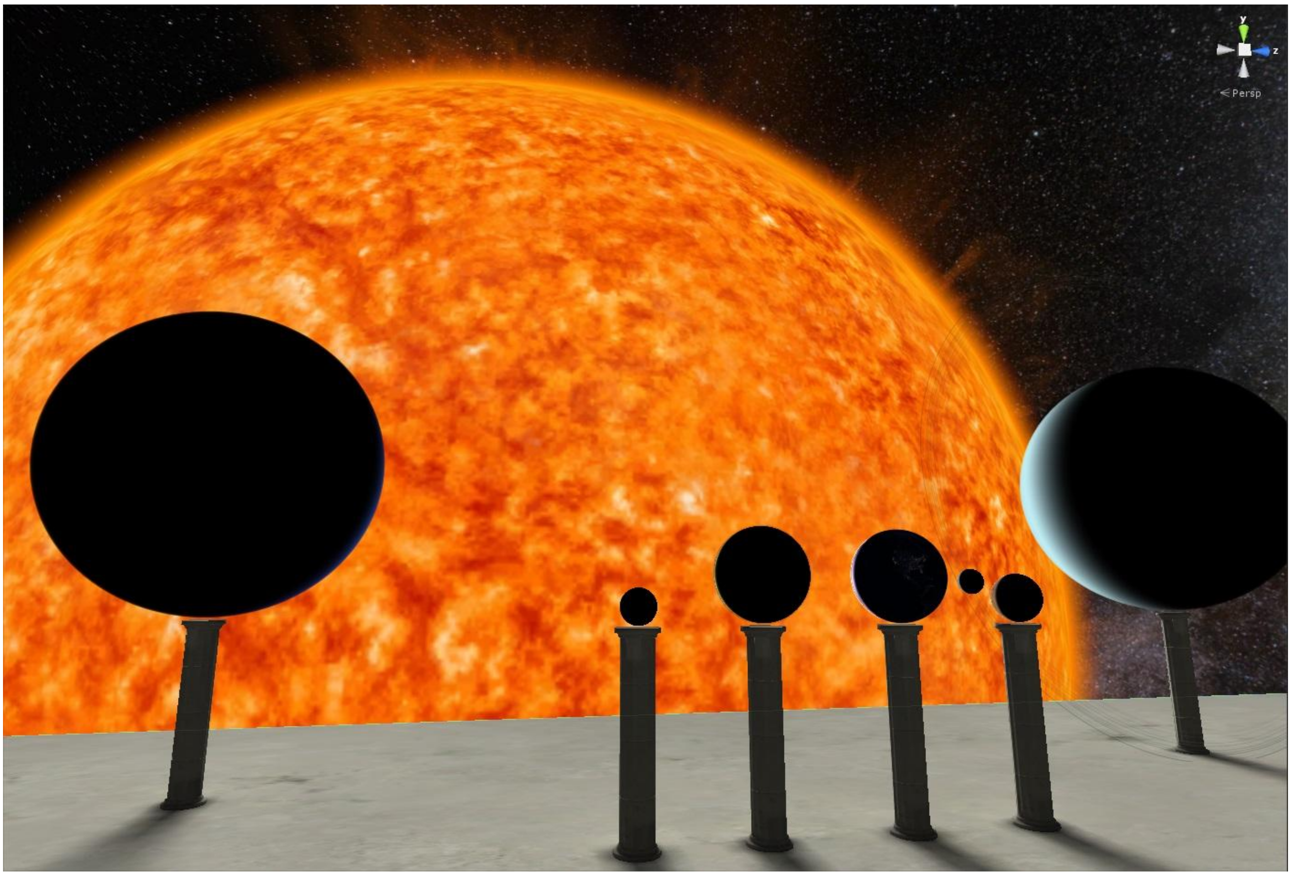


FIGURE 6: THE SUN IS PLACE OUTSIDE THE PLANET ROOM BEHIND THE USER

3. SOLAR SYSTEM

The Solar System in the final VR application is made to simulate the true scale of the planets and moons and the distances between them. The graphics of the application is also made to be as realistic as possible. The graphics in the Solar System of the VR application is implemented using a package for Unity 3D called Space Graphics Toolkit. It provides realistic and customizable graphics for the creation of astronomical objects such as planets, stars, asteroids and dust clouds. It also provides meshes (mostly spheres) with different properties and numbers of polygons.

The stereoscopic display provides a challenge when choosing the scale of the Solar System since the coordinate system in Unity3D is limited by floating-point precision. This means that making a model of the Solar System with true relative distances between the planets is not possible unless the planets are extremely small. With a stereoscopic display the user is able to percept distances and will easily sense the small scale of the planets. This is not desirable when making an ego-centric simulation of a spaceflight. The floating-point precision also means that making the user very small does not work either, since the decimal precision will become too small. This will result in jittery motion of the user and all surrounding geometry such as the spaceship. The application does not feature the true relative distances between planets. However, since the planets are not visible to each other, it is possible to fake these distances by use of travel duration since both planets would be out of sight for most of the travel duration. In the planet-moon systems the celestial bodies are visible to each other and distances cannot be faked the same way. However, in this case the distances are small enough to be true scale relative to the size of the planets and moons. Chapter 5 will elaborate on travel between planets and within planet-moons systems. We found that the optimal solution was to scale the Earth to 10 meters radius and scale the user to 1/10th of his normal size. This also implies scaling the in-game interpupillary distance (IPV) by that amount. This effectively makes the Earth seem like a 200 meter wide sphere. In this scale all the planets can be placed far enough away to reduce disparity-based binocular cues (Ware, 2004) (Johnston, 2014) while keeping the planets large enough in the field of view for making out details.

FIGURE 7: UNITY3D IS NOT ABLE TO HANDLE THE ASTRONOMICAL DISTANCES REQUIRED FOR A TRUE RELATIVE MODEL OF THE SOLAR SYSTEM WHEN IMPLEMENTING A STEREOSCOPING, EGO-CENTRIC SPACESHIP SIMULATION

The application uses real spherical images of the night sky as background. This allows the user to see the night sky as it would appear without any light pollution. The normals of the sphere's polygons are inverted which allows the texture to be seen from inside the sphere. The radius of the sphere is equal to the rendering depth of the virtual camera and the center of the sphere is always the same as the location of the user.

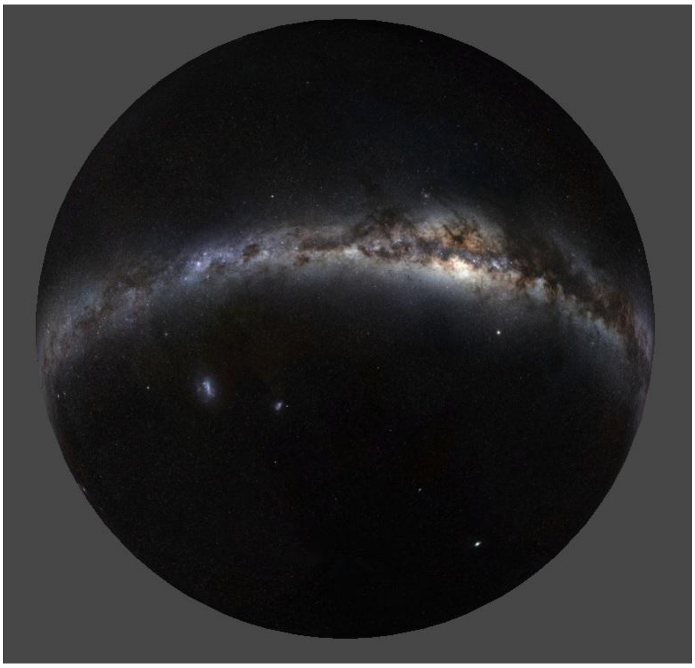
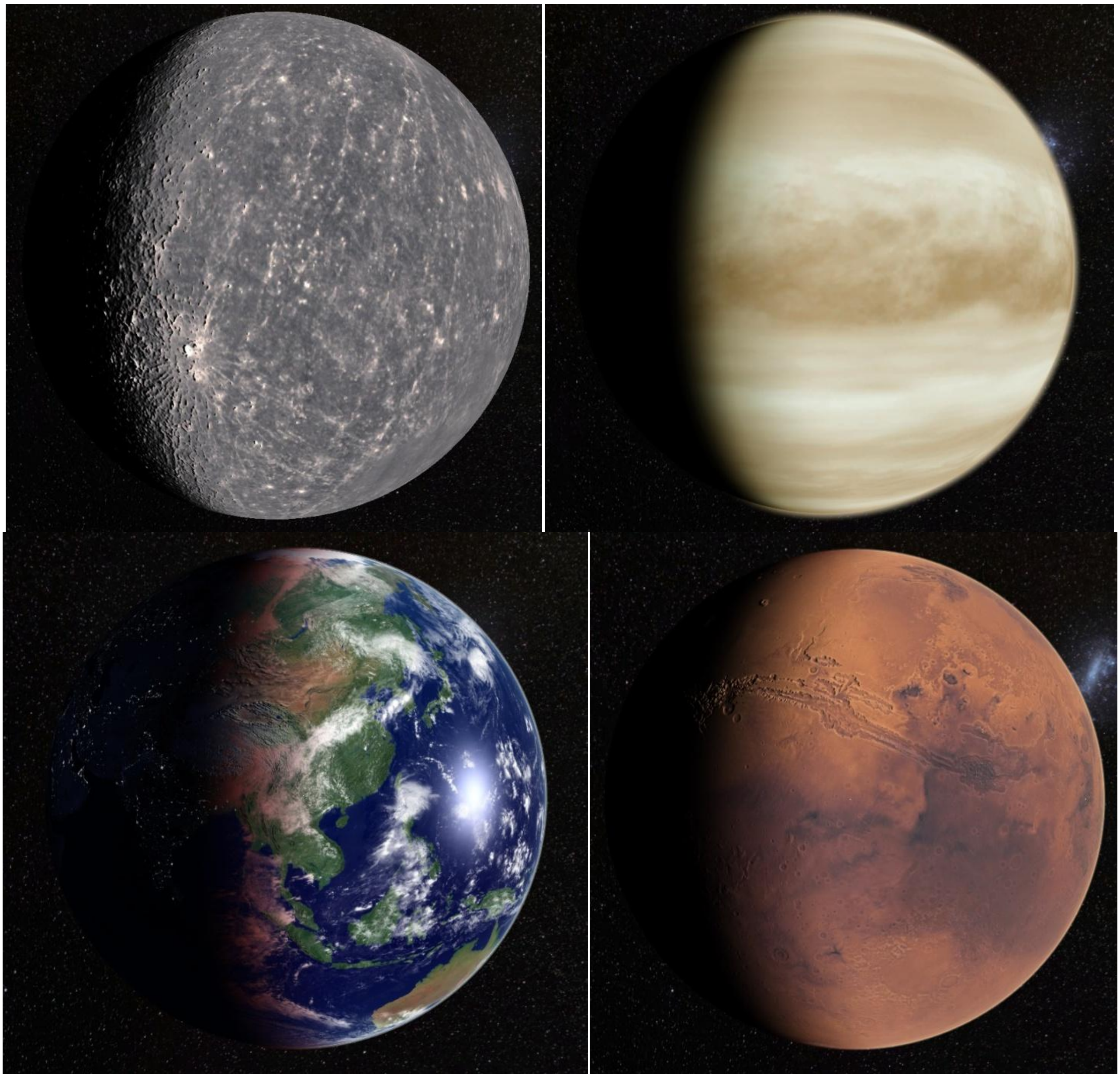


FIGURE 8: THE SKYSPHERE DEPICTING THE NIGHT SKY WITH THE MILKY WAY

3.1. PLANETS

The VR application features the eight major planets of the Solar System and up to four of the largest moons for each planet. The Space Graphics Toolkit provides rich customization options for both terrestrial planets and gas giants. All textures are made from actual photographs, and height maps when available, of the planets. The planets also feature atmosphere, clouds, reflective and night texture when needed. Venus is made to appear as it naturally would with its all-covering cloud layer. The planets have correct tilt and rotates slowly in order to better visualize the tilt. The planets in the application are aligned linearly and does not orbit the Sun. We believe this makes it easier for the user to understand the order and distances of the planets.



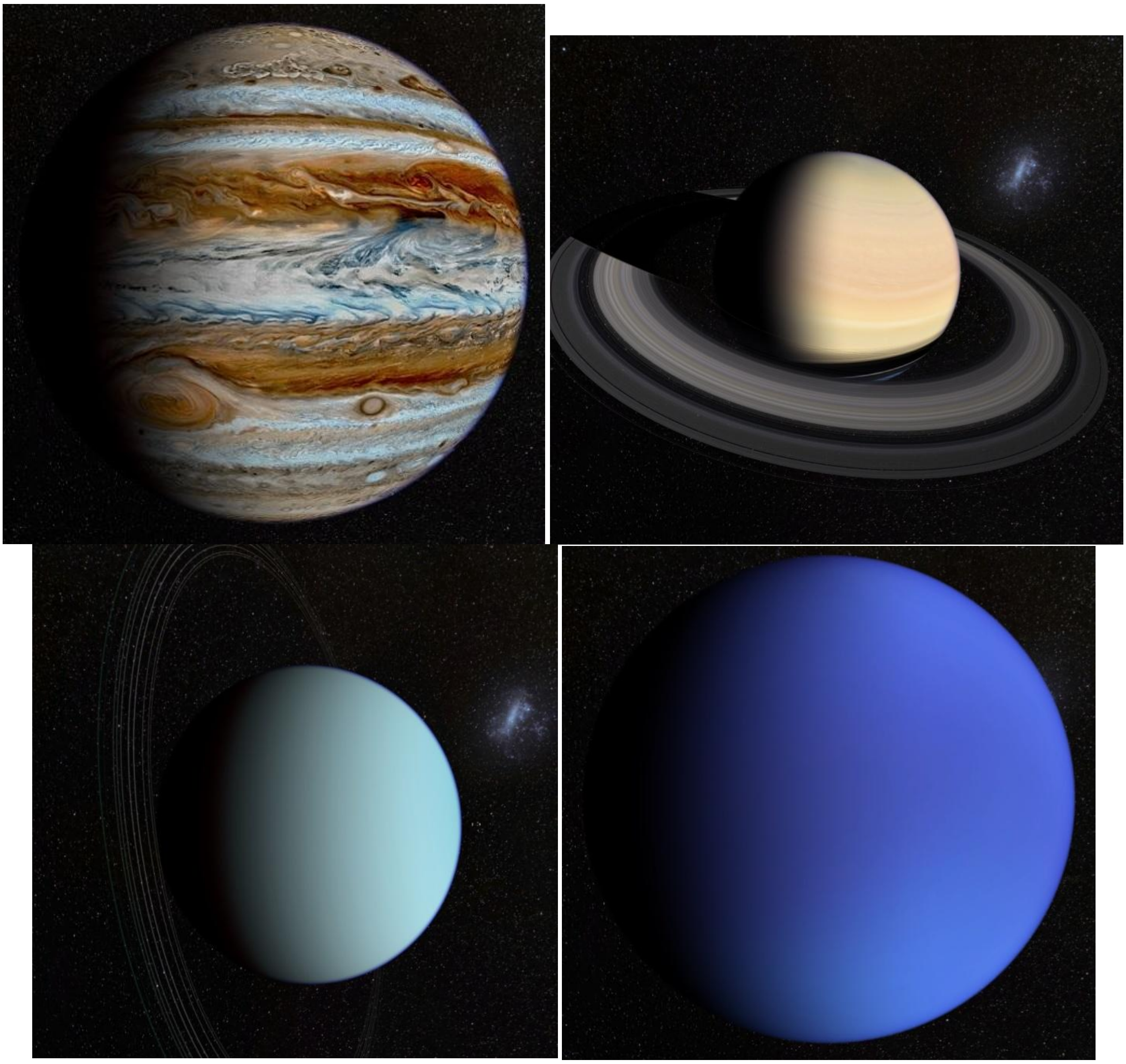


FIGURE 9: ALL THE PLANETS AS THEY APPEAR IN THE APPLICATION.

FROM TOP LEFT: MERCURY, VENUS, EARTH, MARS, JUPITER, SATURN, URANUS AND NEPTUNE

3.2. MOONS

The moons are simple to create since they only need a texture and normal map (if available), Titan being the only exception with its thick atmosphere. Some of the moons of the outer planets have not had their entire surface photographed and textures from other moons are used in this case with a bit of artistic freedom. In the application the moons do not orbit their parent planet, but are instead aligned linearly to make it easier for the user to understand their order and distance.

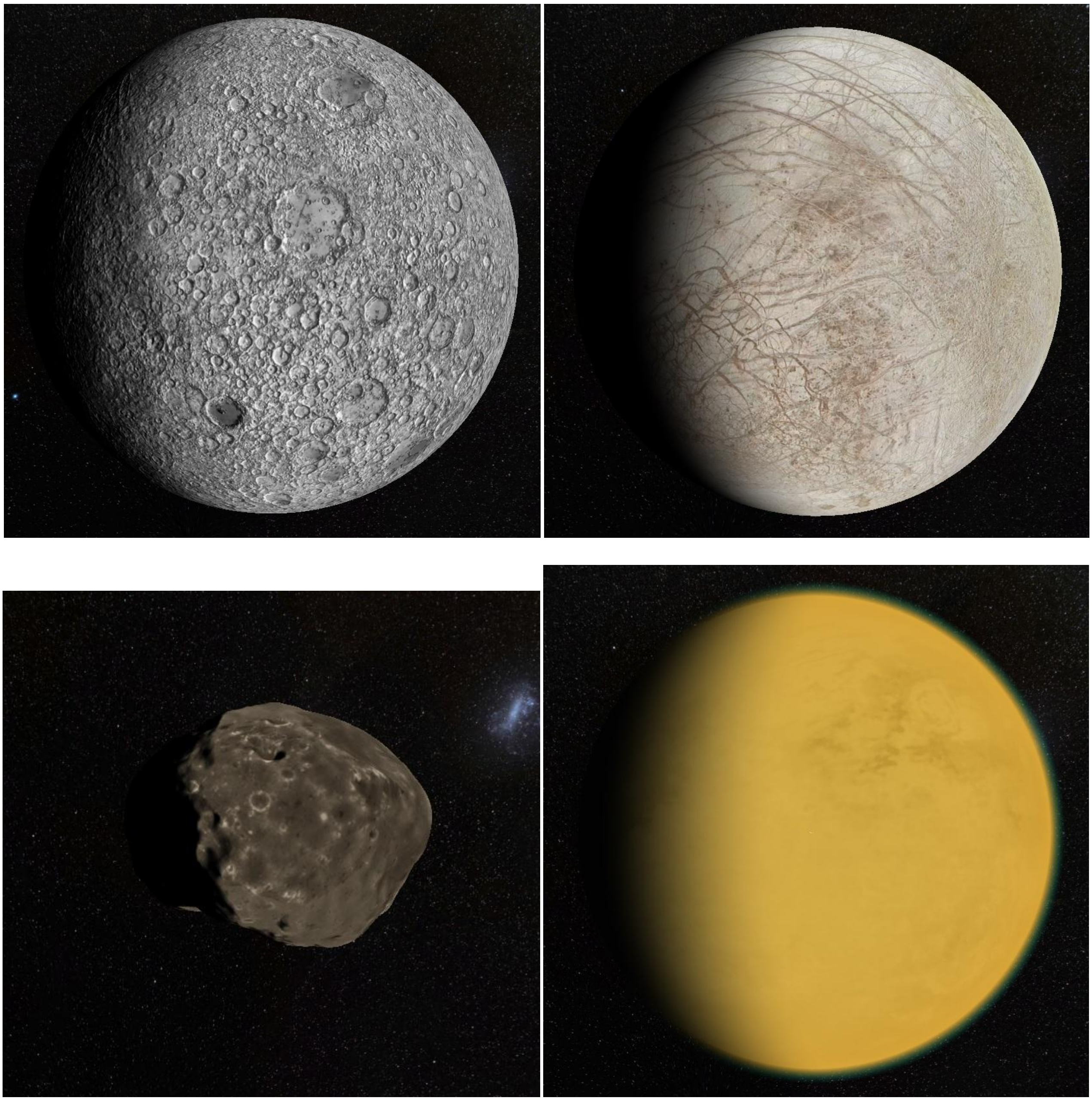


FIGURE 10: SOME OF THE MOONS. FROM TOP LEFT: THE MOON, EUROPA (JUPITER), DEIMOS (MARS) AND TITAN (SATURN). TITAN IS THE ONLY PLANET WHO HAS AN ATMOSPHERE

3.3. The Sun

The Sun is the only object that is visible at all times. However, the Sun is only depicted as an orange sphere of fire when the user is actually visiting the Sun. When the using is visiting other object, the Sun is depicted as a lens flare. The lens flare increases and decreases with distance and does so dynamically while traveling between planets. This adds to the sense of distance. The Sun's corona moves slightly over time in order to illustrate the dynamic nature of the corona. When visiting the Sun the 3DUI displays Jupiter next to the Sun in order to visualize the enormous size of the Sun.

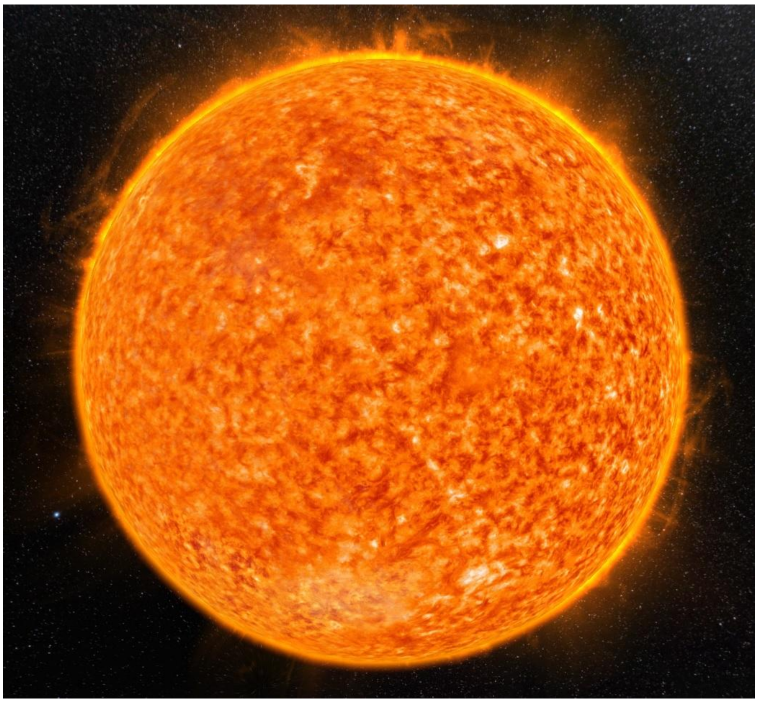


FIGURE 11: THE SUN WITH THE CORONA AS SEEN WHEN THE USER IS VISITING

4. Spaceship

4.1. Concept Art

When designing the spaceship it is considered important that the user is able to view as much as possible of the outside world. Since the application uses the Kinect for positional tracking, it is important that the cockpit limits the user's area of movement in a natural way. Also, the deck of the cockpit must allow enough space for a 3D user interface to be implemented. Figure 12 shows the different concept art produced on the basis of these requirements.

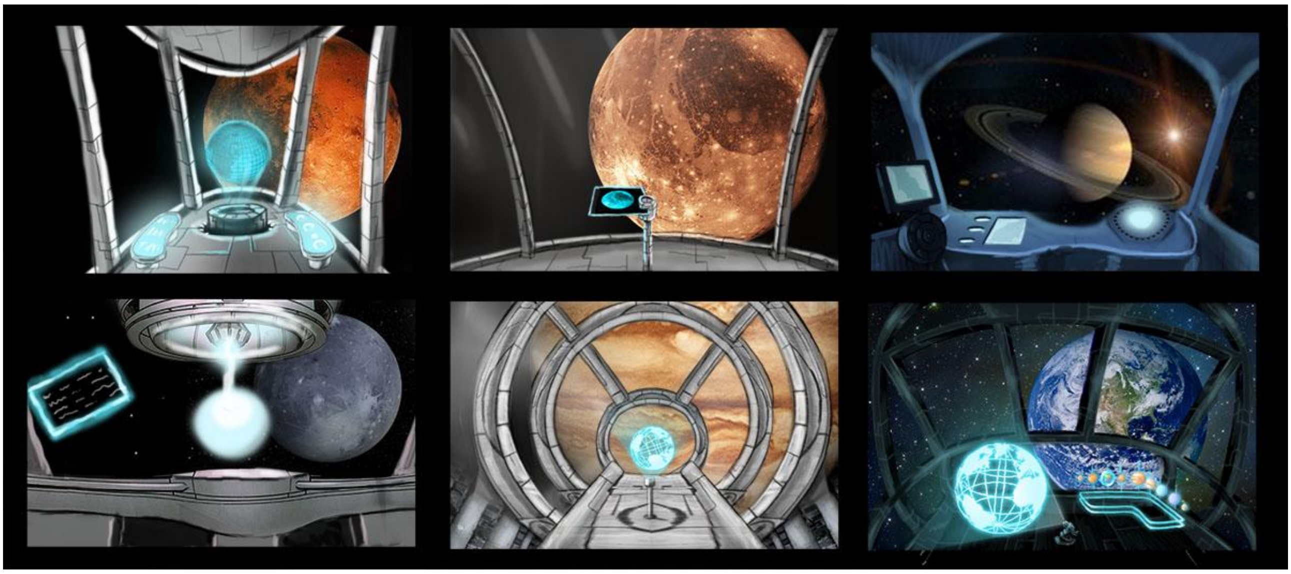


FIGURE 12: CONCEPT ART FOR THE COCKPIT OF THE SPACESHIP

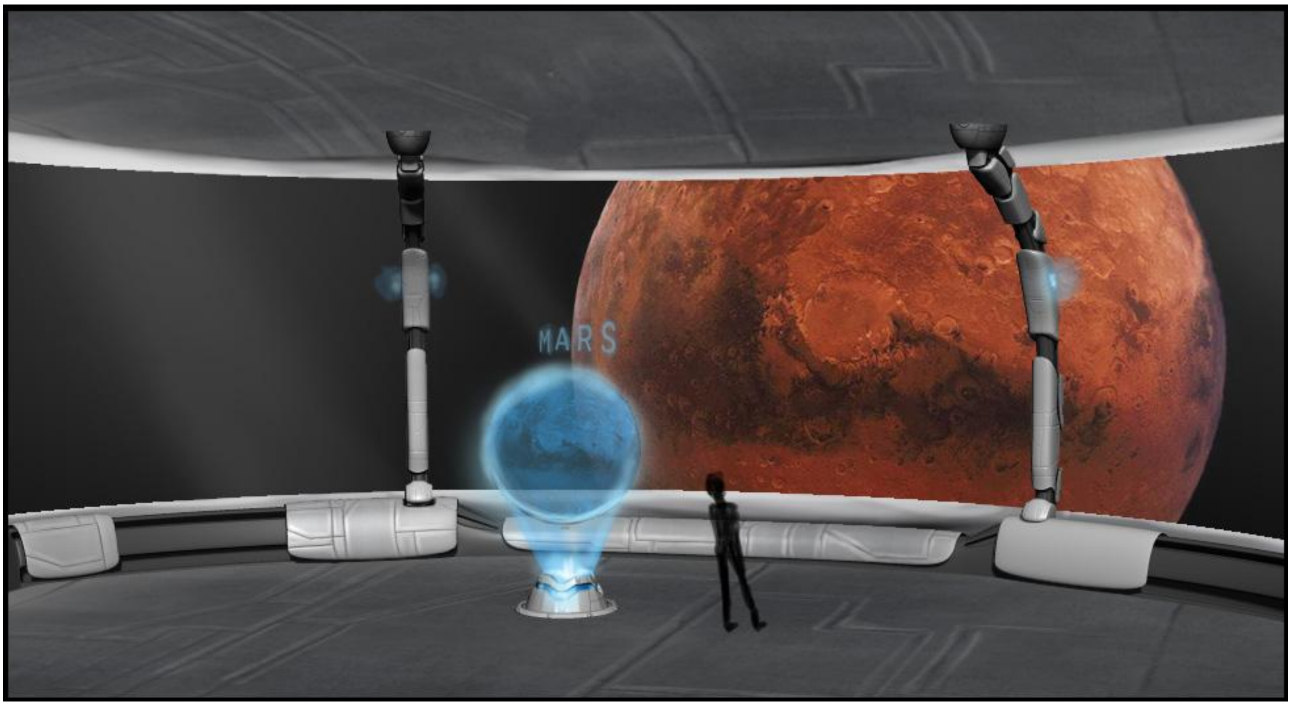


FIGURE 13: VISUAL MOCKUP OF THE COCKPIT WITH HOLOGRAPHIC PLANET INSIDE

4.2. FINAL DESIGN

The upper-center drawing of Figure 12 was chosen as a reference for development of the final cockpit design. Since the user is never outside the spaceship, any exterior design is not critically important. Figure 13 shows a mockup of the cockpit with some of the 3D elements used in the final design. The roof is removed in the final design, for a better view of the sky and the Milky Way. A monitor is added in the front windows of the spaceship showing the user his current position. The monitor also informs the user of the status of the spaceship allowing him to stay aware of what is happening at any time. While the spaceship is in motion and traveling the text in the monitor fades up and down to illustrate this progress. The final design uses tiles in the floor to provide depth cue (Ware, 2004) and for aesthetics. The windows also have "pillars" in order to provide additional depth cues and together with a low railing it acts as a natural constraint for the area in which the user can walk.

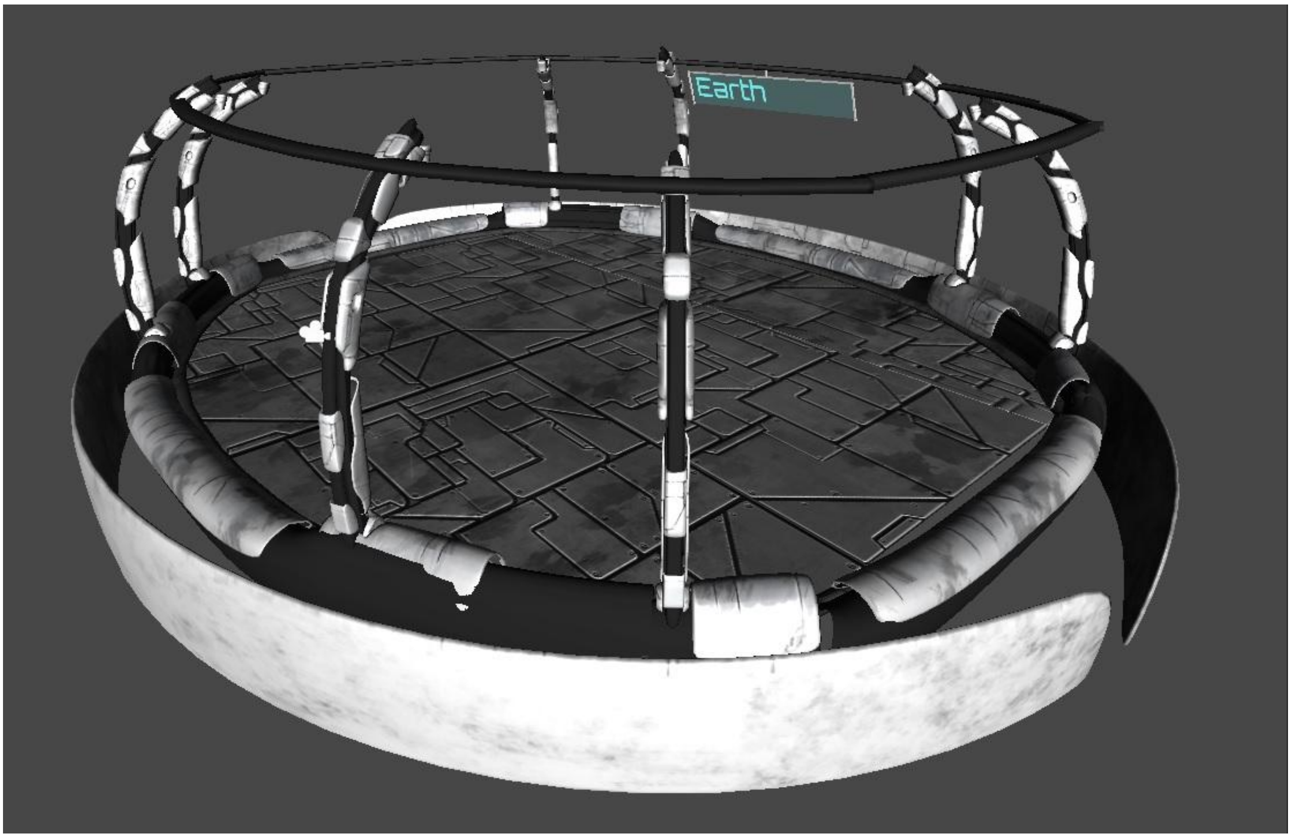


FIGURE 14: THE FINAL DESIGN OF THE SPACESHIP

4.3. 3D USER INTERFACE

The spaceship features a 3D user interface (3DUI) which the user can interact with through the Kinect. The 3DUI consist of a representation of all the planets and their name in correct order and correct relative sizes. The preliminary test shows promising results in learning outcome by scaling the planets correct relatively to each other and thus it seems appropriate to implement this feature in the final application. By selecting a button the user can also choose to view the current planet-moon system. When viewing the current planet-moon system, the parent planet is smaller than would be relative correct due to size limitations. Only in the Earth-Moon system are the sizes relatively correct. In the remaining system the parent planet has been scaled down in order to allow larger moon models while allowing the host planet to be seen. The moons have correct relative sizes to each other and larger moon models allows for better size comparison since the size differences will be more visible. When the user has arrived at a new planet the 3DUI will present the current planet-moon system by default.

The planet models are semi-transparent and only colored in monochrome cyan, heavily inspired by holograms from science fiction works such as Star Wars. However, there is a purpose to this beside futuristic aesthetics. According to the Information Gap Theory proposed by George Loewenstein, curiosity can be triggered by being exposed to a part of a whole (Anderson, 2011). By depriving the user of the information of color, we hope to trigger the user's curiosity towards the planets.

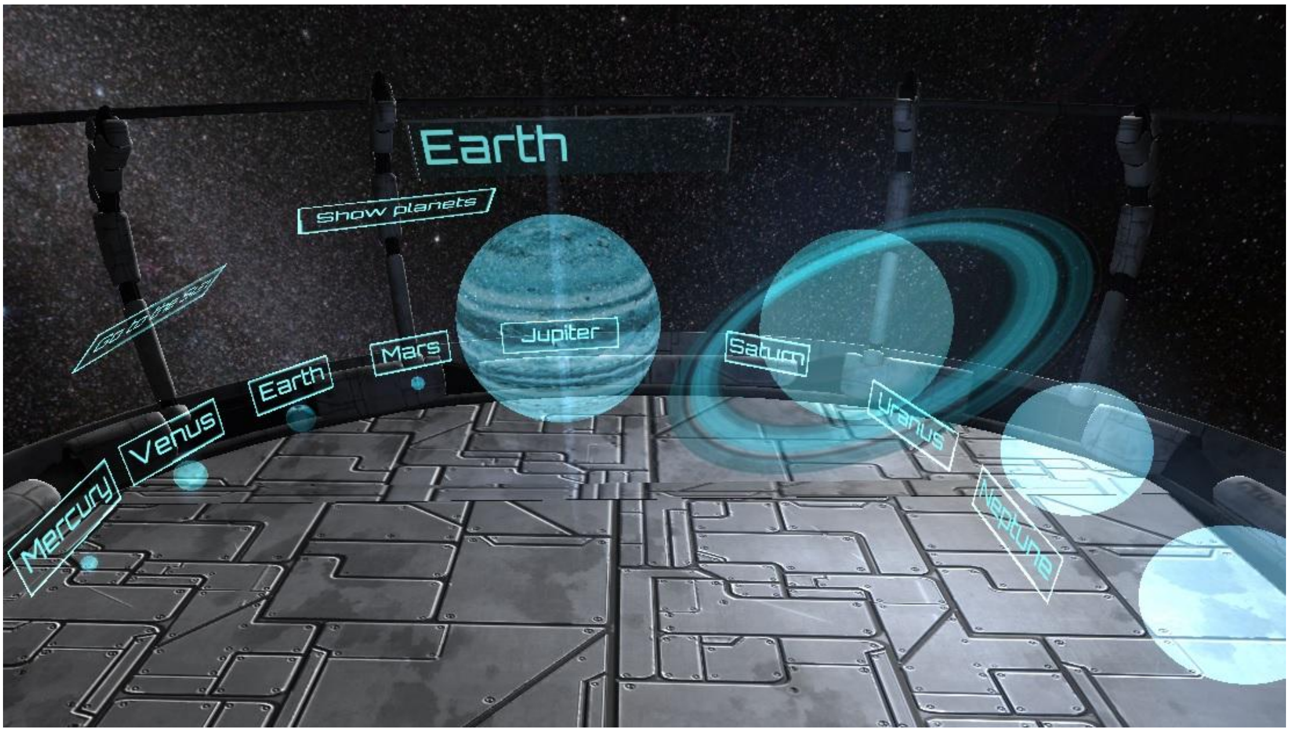


FIGURE 15: THE 3D USER INTERFACE SHOWING THE SOLAR SYSTEM. THE TOP RIGHT BUTTON SWITCHES BETWEEN THE SOLAR SYSTEM AND THE CURRENT PLANET-MOON SYSTEM. THE TOP LEFT BUTTON LETS THE USER VISIT THE SUN.

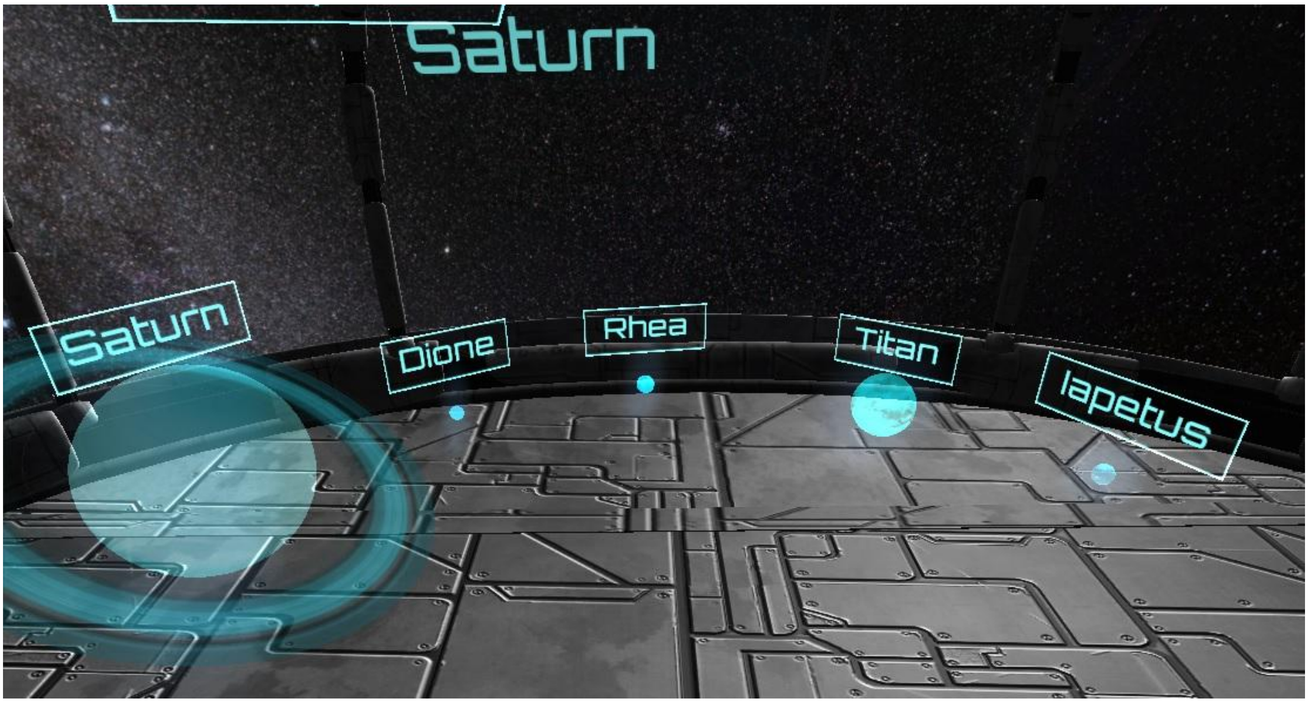


FIGURE 16: THE 3DUI SHOWING SATURNS PLANET-MOON SYSTEM WITH THE FOUR LARGEST MOONS. THESE ARE SELECTABLE DESTINATIONS.

The name of the planet are placed within a rectangle in order to indicate that it can be selected. When selecting the nametag of a planet or moon the travel sequence will start and the user will be translated to the selected destination. The planets are placed in a half-circle so that all the planets have the same distance from the user's start position and thus the sizes are easier to compare. It also allows the planets to be placed in the correct order and also allows for an ergonomic height of the name tags. The user selects a nametag by touching it with either of his hands which are represented by grey spheres. This method is intuitive and natural since it is a direct mapping of the real world, however, the user might accidentally select a nametag if he walks through one of the planet models. A two-hand selection-confirmation method was considered in which a confirmation button would appear while one of the user's hands where touching a nametag. The user would then confirm the selection by pressing the confirmation button with the other hand. However, since the user must sometimes face sideways to the Kinect when selecting some planets, one arm would occlude the other making the method unsuitable. A timed selection method was also considered in which the user's hand must keep touching the button for a short duration. However, The Kinect often jitters the virtual hands which results in the hands occasionally ending up outside the button and aborting the selection countdown. This proved to be very frustrating and in the end, the isomorphic direct touch method was deemed most suitable due to its simplicity. When selecting a button, a sound is played as a feedback to the user.

5. TRAVEL

5.1. USER-CENTRIC IMPLEMENTATION

Unity3D's game engine uses single-float procession which limits the distances that can be used by the coordinate system within the virtual world. The greater the positional coordinates of a game object becomes the less precision it can retain. When traveling large distances away from the origin of the virtual world the surrounding geometry becomes increasingly jittery due to lack of precision. One way to solve this problem is to keep the user in the center of the virtual world by translating the virtual world instead of the user. There will be no difference from the users perspective and precision is preserved close to the user. Objects that are far away from the user will have less precision, but this will not be noticeable to the user since they are far away. This user-centric transformation of the virtual world is a very important concept in the travel implemented of the VR application.

5.2. TRAVEL BETWEEN PLANETS

Traveling between planets takes advantage of the fact that the planets are not visible to each other. The user does not translate and instead the departure planet is moved away and afterwards the destination planet is moved in. This keeps the user in the center of the virtual world while maintaining the illusion of movement. In between departure and arrival the user is subjected to travel time proportional to the real world distance between the locations. Simply teleporting the user instantly would not allow for the experience of the distances. In order to reduce travel time to an acceptable level, the spaceship "warps" hundreds of times the speed of light which reduced the travel time from the Sun to Neptune to 30 seconds. The warp is visualized through motion opposite the travel direction (motion parallax) which provides a visual cue of directional motion (Ware, 2004). A deep, rumbling sound is added to reinforce the feeling of high-speed travel. A "sonic bang" is played when entering and exiting warp to provide the user feedback of these events. While traveling between planets the 3DUI disappears and the user is presented to a visualization which displays his current location between the planets. This visualization uses the correct relative distances. Hopefully, this further adds to the feeling of traveling. It also lets the user anticipate the arrival at his destination. When the user arrives, the spaceship is rotated towards the planet. The spaceship is not parked in equal distance at all the planets. The planets only differ little in their retinal size. The planet must be close enough for the surface to be detailed while not close enough for the surface to look pixilated. Unity3D has a texture limitation of 4k which limits how close the player can be to the planet before spotting pixelation. Furthermore, The size visualization is already taken care of through the 3DUI. The algorithm for traveling between planets is shown below.

1. Spaceship aligns toward destination planet.
2. Current planet moves away opposite the travel direction
3. Spaceship enters "warp" and an animated warp bubble appears surrounding the spaceship
4. When travel time has passed the spaceship exits warp
5. Destination planet moves towards the user
6. The spaceship aligns towards the planet

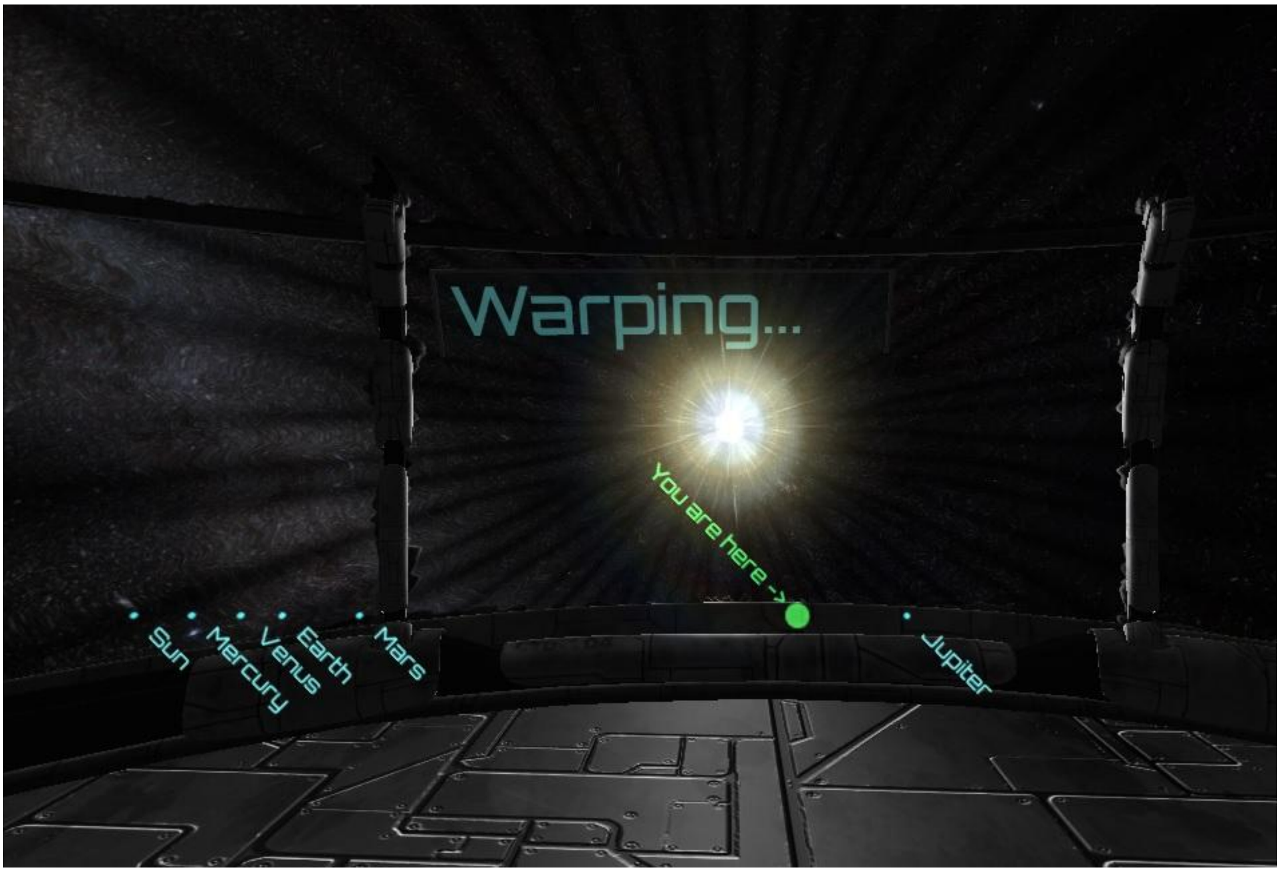


FIGURE 17: THE SPACESHIP TRAVELLING BETWEEN PLANETS. THE MONITOR PROVIDES THE USER WITH STATUS INFORMATION. THE DISTANCE VIZUALIZATION IS PRESENT.

5.3. TRAVEL BETWEEN MOON

The travel mechanism is more simple when traveling inside a planet-moon system. All the objects in the planet-moon system stays present in the virtual world since the coordinate system of Unity3D can handle the distances in a planet-moon system. In this scenario the world is translated instead of the user. When the user chooses a destination, the planet-moon system is moved by the opposite vector of the vector between the user and the chosen destination as seen in Figure 18. travel speed uses ease-in and ease-out. When the user has arrived, the spaceship rotates to face both the moon and the parent planet in order to let the user sense the distance.

$$v_p = v_t * -1$$

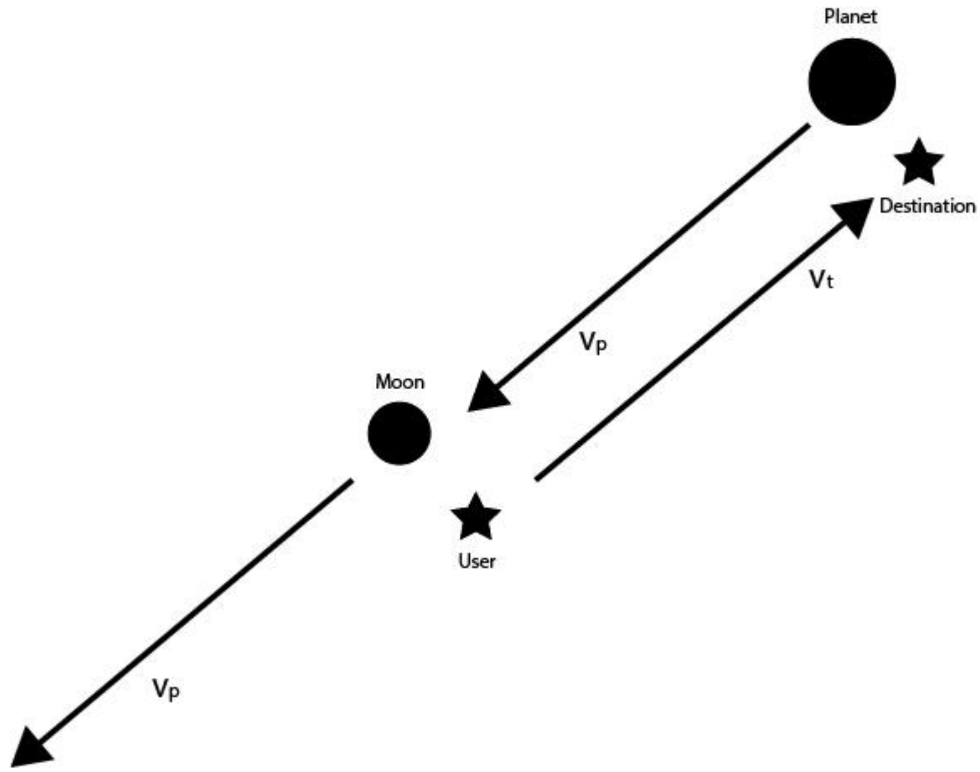


FIGURE 18: THE PLANET-MOON SYSTEM IS MOVED BASED ON THE VECTOR BETWEEN THE USER AND THE SELECTED DESTINATION. THE USER STAYS IN THE CENTER OF THE VIRTUAL WORLD WHILE RETAINING THE ILLUSION OF TRAVEL

6. Test Results

6.1. Preliminary Test

Six children participated in the preliminary test. The children were divided into pairs. Each pair made a drawing of the Sun and the eight planets both before and after they used the application. The children were instructed to draw the planets and the Sun in correct relative sizes. Each session ending with an unstructured interview of the children. Video recordings of the test sessions and the interviews are found on the attached DVD.

Pair/team		
1	Female, age 12	Female, age 12
2	Male, age 11	Female, age 13
3	Male, age 13	Male, age 12

FIGURE 19: PAIR 1'S DRAWING BEFORE THE TEST

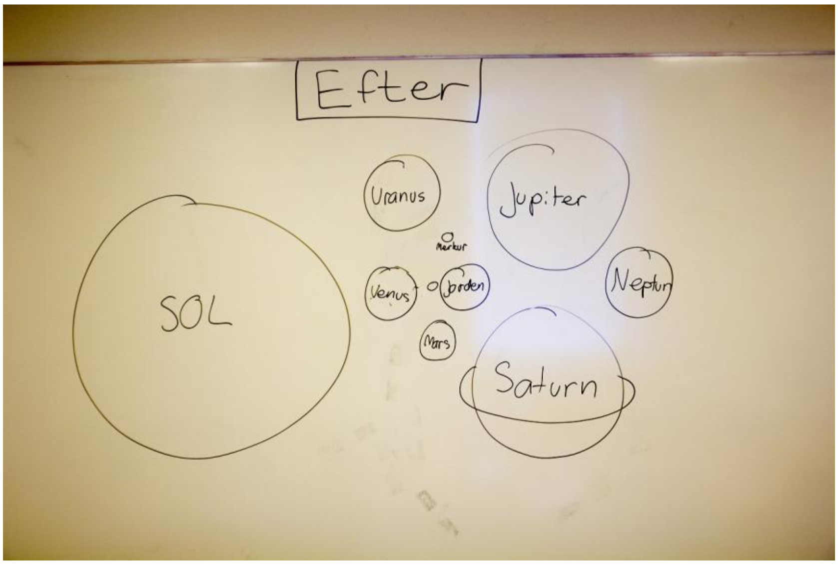


FIGURE 20: PAIR 1'S DRAWING AFTER THE TEST

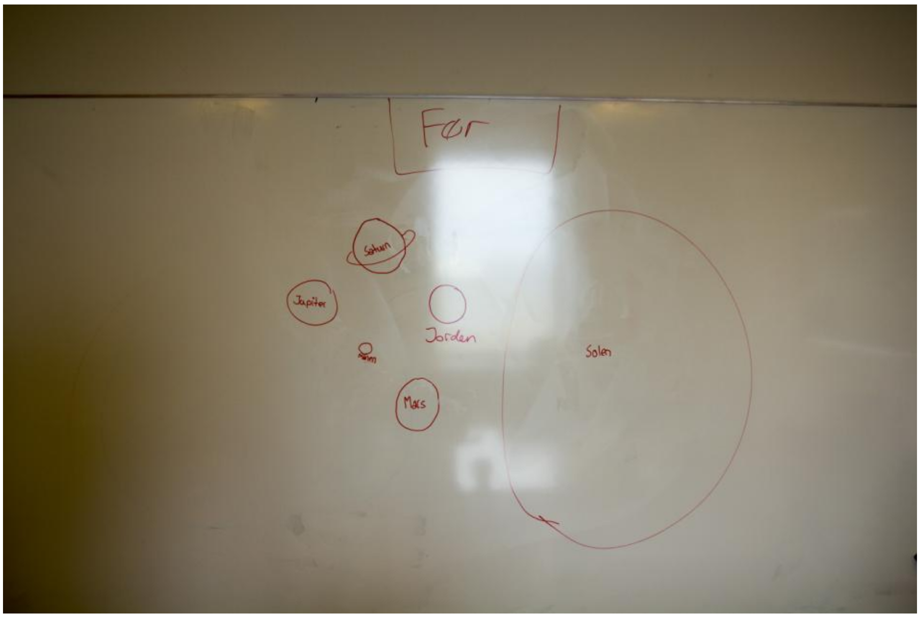


FIGURE 21: PAIR 2'S DRAWING BEFORE THE TEST

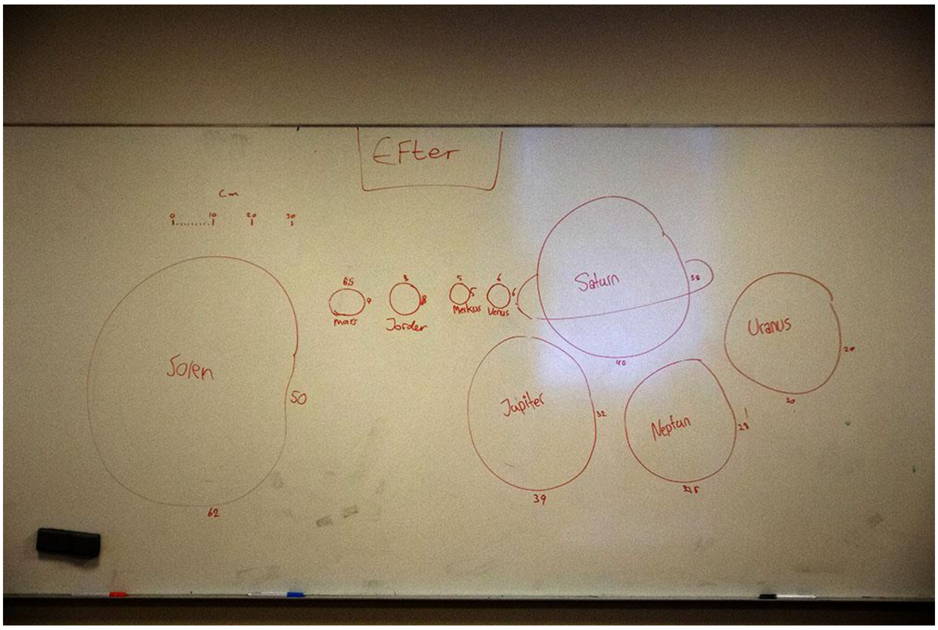


FIGURE 22: PAIR 2'S DRAWING AFTER THE TEST

FIGURE 23: PAIR 3'S DRAWING BEFORE THE TEST

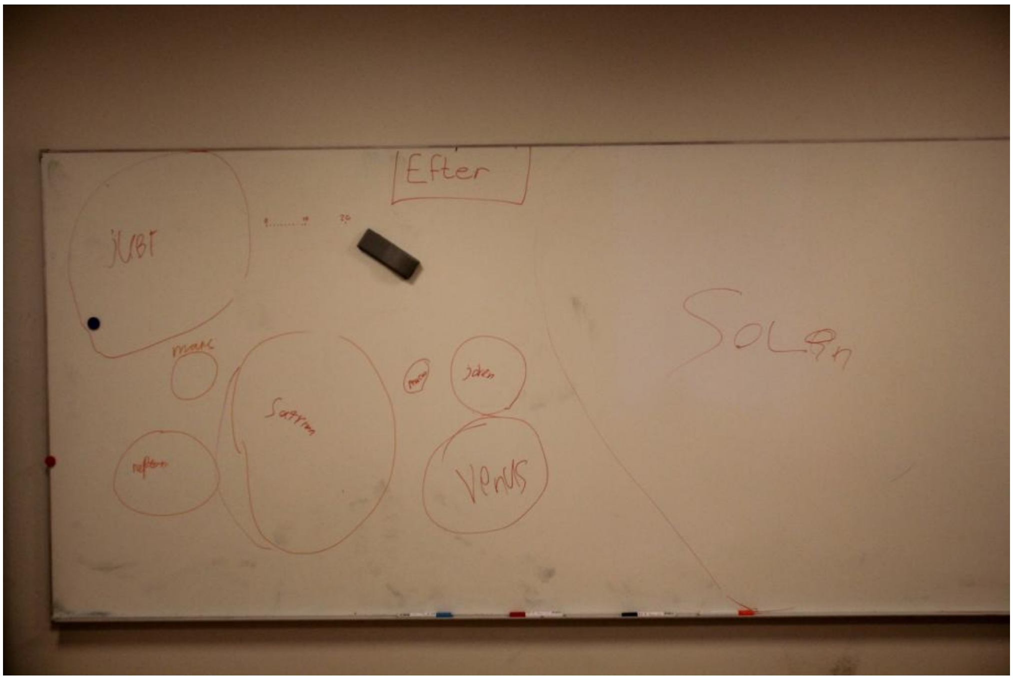


FIGURE 24: PAIR 3'S DRAWING AFTER THE TEST

6.2. Final Evaluation

The raw data from the evaluation of the final application can be found on the attached DVD. The data per participant includes:

- one video recording
- one screen capture video recording
- one log of the users interaction with the 3D user interface
- one audio recording of the interview before the test
- one audio recording of the interview after the test

Furthermore, the DVD includes a spreadsheet containing all the quantitative data from the drawings, interviews and questionnaire. Graphs are also included in the spreadsheet. Also included are the empty templates for the Engagement Sampling Questionnaire, size drawing and distance drawing.

The Engagement Sampling Questionnaire (Schnoenau-Fog, 2011) was chosen over the MEC Spatial Presence Questionnaire (Voderer et al., 2004) and Presence Questionnaire (Witmer & Singer, 1998). Both of the rejected questionnaires contain questions which we believe are too complex for middle school children. The ESQ is far more simple and allows for running evaluation of engagement throughout the test session.

Bibliography

Abrash, M. (2014). What VR could, should, and almost certainly will be within two years. Steam Dev Days.

Anderson, P. S. (2011). *Seductive Interaction Design: Creating Playful, Fun and Effective User Experiences.* Berkeley: New Riders.

Bowman, D. A., Kruijff, E., Jr., J. J., & Poupyrev, I. (2004). *3D User Interfaces: Theory and Practice.* Addison-Wesley Professional.

Johnston, A. (2014, May 23). *Depth Perception*. Retrieved from UCL Division of Psychology and Language Sciences: http://www.psychol.ucl.ac.uk/alan.johnston/Depth.html

Microsoft Corporation. (2014, May 22). *Microsoft Developer Network*. Retrieved from Skeletal Tracking: http://msdn.microsoft.com/en-us/library/hh973074.aspx

National Aeronautics and Space Administration (NASA). (2014, May 24). *PDS: The Planetary Data System*. Retrieved from Venus: Planet Profile: https://pds.jpl.nasa.gov/planets/special/venus.htm

Nielsen, J. (2014, May 24). *10 Usability Heuristics for User Interface Design*. Retrieved from Nielsen Norman Group: http://www.nngroup.com/articles/ten-usability-heuristics/

Oculus VR. (2014, May 22). *Development Kit 2*. Retrieved from OculusVR: http://www.oculusvr.com/dk2/

Schnoenau-Fog, H. (2011). Hooked! - Evaluating Engagement as Continuation Desire in Interactive Narratives. *ICIDS*, 219-230.

Voderer, P., Wirth, W., Gouveia, F., & Biocca, F. (2004). MEC Spatial Presence Questionnaire (MEC-SPG).

Ware, C. (2004). *Information Visualization - Perception for Design (2nd edition).* San Francisco: Elsevier.

Witmer, B. G., & Singer, M. J. (1998). Measuring Presence in Virtual Reality Environments: A Presence Questionnaire. *Presence*, 225-240.