The VR Classroom

A LTU student project

D7017E

Luleå University of Technology

Department of Computer Science, Electrical and Space Engineering

By:
Sophia Caspár *
Jesper Gladh †
Erik Karlsson ‡
Leo Ozolins-Carlson §
Dennis Persson ¶
Vikhram Ravi ‖
Isak Lindgren **
Claes Andersson ††

January 26, 2017

Github[4] - Webpage[7] - Facebook page[6]

*sopcas-1@student.ltu.se
†jesgla-2@student.ltu.se
‡erkkar-2@student.ltu.se
§leocar-2@student.ltu.se
¶denper-0@student.ltu.se
‖vikrav-2@student.ltu.se
**isalin-0@student.ltu.se
††calane-2@student.ltu.se

Abstract

This paper investigates the applicability of virtual reality for educational purposes. It presents a system for improving the learning experience in elementary school through various experimental approaches using the smart phone based virtual reality platform, Google Cardboard. The produced application explores several approaches for making use of virtual reality in various subjects, with a focus on grades 1-6.

The approaches taken in this project are primarily based on the Swedish curriculum as well as feedback received from actual teachers. The resulting work indicates that there is much untapped potential in regards to using VR to augment the traditional learning and teaching methods practiced today.

This project is part of a rapidly growing field of research, indicating that there is currently much interest in applying virtual reality as a tool for enriching the education experience.

Contents

1 Introduction

Digitization is becoming a more natural part of the world for each day that goes by. People can use different mobile applications for visiting their bank, shopping for clothes and reading books etc. If the way people live their lives have changed, then why shouldn't the way children learn in school change? During the last 20 years a lot has happened in the digitization of education, starting with having lectures of how to start a computer each week to today when all children work on their tablets during lectures.

This report is about taking the digitized education to the next level. The main purpose with the entire project was to explore the possibilities of using virtual reality as a complement during lectures in elementary school. Virtual reality could help many students to learn in a new way and hopefully revolutionize the education as we know it today.

1.1 Background

Virtual reality, VR, is starting to get very popular in the gaming industry and is predicted to change many industries in the future.[1] VR makes it possible to interact with computers in entirely new ways. One area where VR technology can be useful is in school where students could see and experience many different environments through virtual visits and through various experiences by using VR technology.

Today, Google are, for instance, focused on creating applications for Google Cardboard and other companies have used 360 degrees movies to illustrate how it is to be bullied in school. Many VR applications are created through various 3D game engines such as Unity or UnrealEngine.

There are a lot of different headsets to experience VR environments with, e.g. HTC Vive and Occulus Rift. Due to the increasing popularity of VR, more companies are releasing their own version of VR headsets, for instance Microsoft and Sony, which makes it easier and more available for the average household to buy.

However, most of the headsets are expensive and require even more expensive computers. But there are cheaper alternatives like Cardboard and Google platform Daydream (Google VR).

1.2 Assignment

The main goal with this project is to create a concept of how virtual reality can be used as a tool for children to learn and experience in new ways in school. Also, to show how technology can motivate learning and learn more about how technology can be used in learning situations. To achieve this, an advanced but easy to use 3D-application is going to be created.

1.3 Problem description

The project is very comprehensive and should include all aspects of a traditional project to get a successful result, from a thorough research to a near-finished product or concept. The main parts of the project is listed below.

- Put together what others have done in the world of VR.
- Talk to teachers about how technology can be used in school to get inspiration.
- Learn how to work with 3D tools to create VR applications, for instance Unity and Google Daydream.
- Create a few scenarios for how VR technology can and should be used in the classroom.
- Implement one or several of these scenarios.

The project is expected to result in:

- A market intelligence.
- A concept of how virtual reality can be used as a tool for education.

- Several possible scenarios for using VR in the classroom.
- Several small demo applications that show different applications.
- A larger VR application to show how it can be used in the classroom.
- Public documentation and promotional materials for end users.
- Technical documentation.

1.4 Delimitations

The project is directed towards children of ages between 5-13 because the curriculum for younger children includes more simplistic types of subjects and problems. Instead of focusing on actual advanced school assignments, the project can use simpler assignments and therefore focus on how to implement different learning scenarios with virtual reality.

2 Working model

The project group have been working in accordance with the SCRUM-model with help of Trello which is a collaboration tool that organizes projects into boards. By writing down the deadlines and different activities in Trello, each member in the group can work effectively with their individual part. This also prevents multiple members working on the same activity. Meetings on a weekly basis have been conducted with the members in the project group at least two times a week to follow up how the progress is going, if the goals are reached and to set new goals for next meeting. Meetings with the project owner has also been held regularly to update about the progress.

The project has been divided into four sprints and each sprint has separate objectives and tasks. The tasks and goals was separated into smaller sub-tasks and divided between group members with some kind of testable objectives in focus. This was to ultimately be able to check off each goal after an approved test.

2.1 Research

Before getting started on the actual development there were a lot of research needed. Some main areas of interest were:

- Health concerns with VR.
- Cost of VR.
- Resources needed for developing.
- What school subjects are suited for VR.

The research was mainly conducted with the help of research papers and articles found on the web and a workshop that let elementary school teachers try VR.

2.1.1 Teacher workshop

The idea with the workshop was to gather intelligence, ideas and experience from teachers working in elementary schools. The teachers tested both Google Cardboard and HTC Vive to get knowledge on what VR was all about. After that the teachers were split into smaller groups to discuss the technologies that they just tested and how the VR technology could be used in education and pedagogy. Then the groups presented what they came up with and the other groups could comment and discuss the ideas. Everything that were said during the workshop was recorded and summarized in a document for future reference. The summary can be read in appendix A.1. A selection of the thoughts that were discussed and was later used to base the project on are:

- What subjects children learn about in elementary school.
- Some ideas about how the technology can be used.
- What students in general find hard to learn.

2.1.2 Porting to HTC Vive and Oculus Rift

In the initial research phase, before it was decided to limit the application to the Google Cardboard platform, some time was spent investigating the possibility of porting the application to the HTC Vive and Oculus Rift.

As a part of this, a relatively simple game was ported from Google Cardboard to the HTC Vive, using the OpenVR framework (conveniently pre-packaged by Valve in the Unity store as SteamVR). OpenVR is an SDK for developing VR applications for both the HTC Vive and the Oculus Rift.

The actual porting process itself is relatively simple. The Unity components responsible for the camera needs to be replaced with the OpenVR equivalents. Then the function calls for various input events need to be converted to follow the OpenVR framework. The ported application works well, but there were a few unexpected complications. The HTC Vive and Oculus Rift both offer a much greater sense of presence, which in the case of this particular game, caused the user to experience nausea and disorientation. Porting an application between the currently popular VR platforms is relatively easy. However, given that the different VR platforms currently offer very different controller setups, great care needs to be taken when designing the user input.

It's relatively easy to port the Google Cardboard input method to the other platforms (i.e. controlling the application through your gaze). It's also relatively easy to map that same "gaze control" to instead work with the HTC Vive motion controller, by essentially giving the controllers themselves a kind of gaze (this is probably something that might also work well with the Oculus' future motion controllers).

In hindsight, porting our final application to use the OpenVR framework would probably only take 2-3 hours. However, as discussed in other sections, the primary obstacle is the cost and spatial requirements needed to implement this kind of setup for the school.

2.2 Project management and implementation

The project was split into four sprints. By working in sprints, the group could focus on short-term goals which made the progress more effective. The first sprint consisted of research, interview with the client and testing different VR-headsets and frameworks. During the second sprint a workshop was held for a couple of teachers to gather ideas and input about how to use VR in school. After that the concept plan of how the final application should work was created, and based on the plan an alpha version was developed to get more familiar with Unity.

The third sprint was the main development phase of the project which also included some testing. During this phase most of the progress towards the actual application was made and the different parts of the app all got more clear concepts in regards to gamification. The final sprint focused on testing and final touches of the application. The main goal during this sprint was to be fully finished with the application. Also to give a final presentation about the project and to be finished with all documentation.

2.2.1 Roles and responsibilities

Even though each member of the project group has equal responsibility to push the project forward, some members of the group got specific responsibilities and roles. This is to have someone that makes sure a specific part gets done in time. However, that doesn't mean that the person responsible for a part necessarily has to be the one doing it, only that the person responsible makes sure that part gets done. The different roles some of the project members have are the following:

Project leader: Sophia Caspár - Should have an overview and control over what every member in the group is working with and is kept informed on everything relevant to the project. The project leader is the contact person between the project group, the client and any third parties.

Secretary: Erik Karlsson - Is required to write protocols during meetings and be responsible for submissions and all written deadlines.

PR-Coordinator: Vikhram Ravi - Is required to keep the project's Facebook page and website updated.

Documentation-Coordinator: Claes Andersson - Should have an overview of the documentation of code and assets, and make sure that all parts of the project are properly documented.

Test-Coordinator: Isak Lindgren - Should have an overview of different tests that should be implemented in the project and makes sure that it is implemented.

Graphics-Coordinator: Leo Ozolins-Carlson - Is responsible of graphical part of the project.

Research-Coordinator: Jesper Gladh - Should have an overview of all research and be responsible of planning different workshops.

Code-Coordinator: Dennis Persson - Is responsible for connecting all different parts of the application into one.

2.3 Project planning

Before we started with the project, a project plan was created with overall goals for each sprint.

2.3.1 Sprint 1

The goal for this sprint was to get familiar with current VR solutions and the available hardware. This to get better understanding of the current state of the VR landscape and generate ideas for how it can be applied in the classroom. The first sprint was predicted to be research heavy, and in order to better divide the workload the following goals were set.

Table 1: Goals for sprint 1.

Id	**Name**	**Description**	**Goal**
100	Testing the hardware	Set up the hardware and install relevant software.	Each member in the project group must test HTC Vive and Google Cardboard.
101	Unity Research	Unity is the development environment chosen to develop the VR experiences for the project.	Each member in the project group must go through **at least** 1 Unity tutorial project.
102	VR research	It is important to know what others have done within VR and take others' research in account.	Each member in the project group must find research about VR.
200	Website and blog	The project must have somewhere to post updates and progress of the project.	Set up a website and create a Facebook page for the project.
300	Alpha version	Start production of an actual product prototype.	Present a simple application made in Unity.

2.3.2 Sprint 2

The main focus for sprint 2 was to talk to teachers and gather thoughts and ideas in order to start working on the actual concept of the project. For this sprint the project group had an overall goal to get started with the application and to have something more to present than the alpha from sprint 1.

Table 2: Goals for sprint 2.

Id	**Name**	**Description**	**Goal**
100	Teacher workshop planning	Plan the content for the workshop and come up with important discussion materials to get the most out of the workshop.	The planning of the workshop and an associated presentation must be finished.
101	Teacher workshop	Arrange a workshop with actual teachers to discuss their perspective on VR as a tool in school.	Collect feedback from the teachers about VR.
102	Teacher workshop summary	Summarize all ideas and thoughts from the workshop.	Have a finished summary from the workshop.
200	Concept plan	Decide in which direction the project should proceed and what the goal with the application is based on the result from the teacher workshop.	The project group must agree on a common concept of the project.
300	Application planning	Come up with ideas of actual applications to develop based on the concept.	The project group must decide on **at least** 3 smaller applications to include in the overall application.
301	Alpha applications	Start working on the different applications.	**At least** 3 alpha applications must be done for half-time presentation in the end of sprint 2.
400	Visit Samuraj	Visit the VR focused company Samuraj in Luleå for inspiration.	Gain inspiration from others working with VR and gain a better understanding for a professional workflow.
500	Report	Start working on the final report to prevent having everything left in the end of the project.	To have a body for the final report.
501	Half-time presentation	Planning and execution of the half-time presentation in the end of sprint 2.	The project group must present the alpha applications and the work so far in a presentation for the teacher.

2.3.3 Sprint 3

During the third sprint, the project group continued the development of the applications. There were a lot of smaller and more detailed goals for different tasks which made it easier to test each new function. However, the goals in table 2.3.3 are more generalized in order to have a better overview of the sprint.

Table 3: Goals for sprint 3.

Id	**Name**	**Description**	**Goal**
100	Use cases	To have a more distinct plan of the applications use cases must be defined.	The project group must define use cases for the application.
200	Gamification	The application must have some kind of gamification and goal with playing, like different tasks.	Implement a gamification part in the applications.
201	Score system	The players should be able to see progress when playing.	Implement a scoring system in the applications.
202	Menu	A player should be able to operate different settings inside the application	Implement a menu for each small application.
300	Testing	For each new function that is added on any of the applications a test must be done to validate the functionality.	Test all new features.
400	Report	Continue the work on writing in the report.	The project group must add content in the report.

2.3.4 Sprint 4

The goal with the final sprint was to complete the entire project and to connect all pieces into one big application. To prepare for and hold a final presentation where the entire project was demonstrated was a large part of the fourth sprint, also to finish up the final report.

Table 4: Goals for sprint 4.

Id	**Name**	**Description**	**Goal**
100	Beta version	Wrap up the development of all small applications	All small applications must be completed.
101	Merge applications	The applications should be merged into one big application.	Merge the applications into one working big application.
102	Final testing	The merged application must be tested to check that everything works.	Test the application.
200	Final presentation	Planning and execution of the final presentation of the project.	The project group must present the final application and the entire project in a presentation.
201	Demo	Instead of a live demo of the application, a video demo should be created.	Create a video demo.
400	Report	The final report must be finished in the end of the course.	The project group must complete the final report.

3 Specification

The following specifications should be fulfilled to ensure that the best user experience possible is acquired, and these are both for the user and the application.

3.1 Requirements specification

The essential goals for the entire application are the following:

- Good performance.
- High frame rate.
- Good graphic.
- Sense of gamification.
- Applicable to VR.
- Save game progress and score.

3.2 Systems and non-functional requirements

- User friendly - should be easy to understand how it works.
- Rapid scene transitions - Short waiting periods.
- UI elements should be large enough that it's legible on low resolution screens.
- Acceptable performance should *not* require a high-end smartphone.
- Camera responsiveness should not cause dizziness or discomfort.
- The application should be portable and support both Android and iOS.

3.3 User requirements

Table 5: User requirements for the VR classroom.

Id	**Name**	**Description**	**Dependency**
101	Navigation	Move around or start different applications.	
102	Interaction	User should be able to click on various objects.	101
103	Game selection	Initiates an application.	101, 102
104	Change settings	Customize settings to suit student needs.	101, 102
201	Save score	Keep track of score to see progress.	
301	Graphics	The whole application should have the same graphical layout.	
302	Good performance	Screen frame rate should not disrupt user while playing.	301
401	Easy to use	Comprehensible for younger students.	

3.4 Use cases

For this project there are two main actors that have been focused on while developing the applications.

- Student
- Teacher

There is also a secondary actor which only is relevant for a small multiplayer part in the Room application.

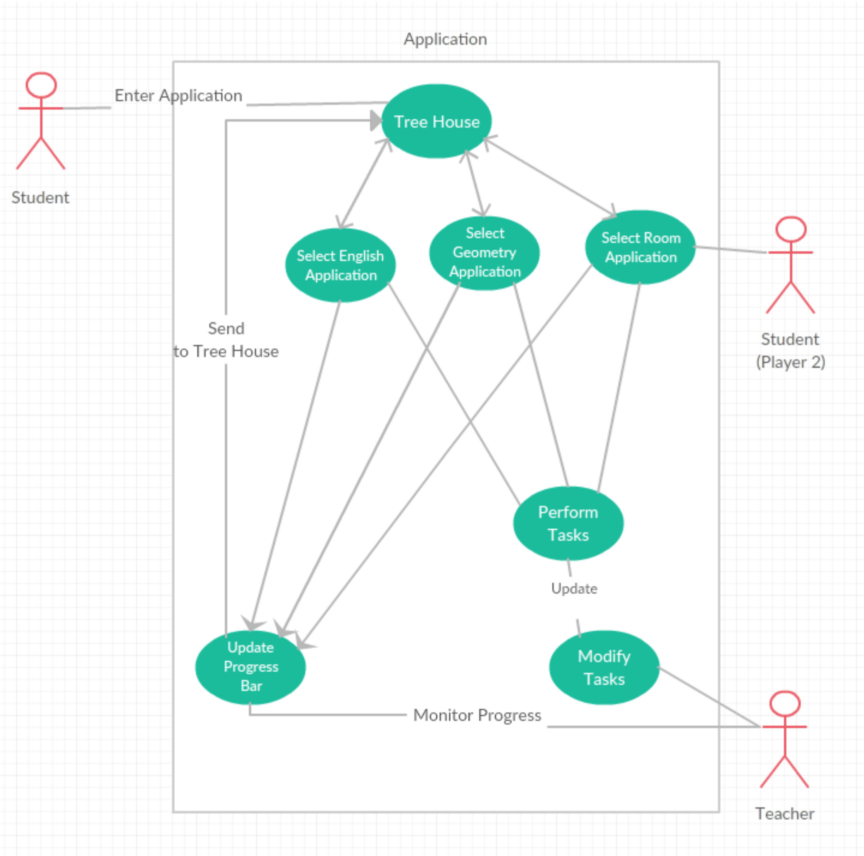

Figure 1: Use Case Diagram of the VR Classroom application

3.5 Testability

The application can be tested with both white-box and black-box testing techniques.

3.5.1 Black-Box Testing

In the table 3.5.1, there are some scenarios defined with the help of the user requirements for this application.

Table 6: Black-box scenarios.

Input	Output	User requirement
User rotate the head.	The scene displays the corresponding environment for that angle.	101
User tilts head against shoulder.	The tree house environment is displayed.	101
User uses reticle to click on object.	The object is clicked	102
User uses reticle to click on an object representing a game.	The game is started.	103
User changes settings in the teacher menu for a game.	The setting is set and the game runs with the new setting.	104
The user answers correctly in a game and gets a new high score.	The score is saved and displayed in the tree house.	201

3.5.2 White-Box Testing

White-Box testing is a method of testing the internal structures and workings of our application. In the table 3.5.2, there are some scenarios defined for every specific application.

Table 7: White-Box scenarios.

Input	Output	Application
Light spheres that take the user from the yard to the tree house.	Only the light spheres in front of and behind the user is visible.	Tree House
Item movement that follows ray-cast intersection with room wall.	The item moves along the wall and does not go through or behind the wall.	Room Application
Item should able to be moved around even after placing it in a fixed position.	The item can be clicked and moved.	Room Application
Instructions of the furnishing must correspond to the win condition.	Instructions must be the same as the win condition.	Room Application
Score must increase by an appropriate amount after finishing level.	Score manager should only increase if the level is finished and correct.	Room Application
Objects are loaded and placed in a randomized position.	The folder is connected to the correct level and the correct objects are placed.	English Application
Win condition calls upon the score function.	The score function increments the score correctly.	Geometry Application

4 The VR Classroom

The VR Classroom is the name given to the project and ultimately to the final product. This section aims to explain in-depth the specific functionality, concepts and ideas behind the different parts of The VR Classroom and the decisions that were made along the way.

4.1 The concept

The idea of The VR Classroom as a whole is to help teach kids within the school environment, and to be used by teachers as a tool for enhanced learning. As such the grounds for our concept is laid by the ideas and input provided to us by the active teaching staff consulted in the teacher workshop.

One significant and reoccurring concern that was brought up was the cost of the device. Further concerns touched upon ease of use, practicality and inclusion. The more advanced Virtual Reality headsets are fairly expensive and although a school handles a lot of money it was often seen as a very extravagant cost. The practicality of using such a VR headset in a regular classroom, containing around thirty students, also came into question. As maybe only one device could be provided and only a single student could use the device itself simultaneously, the device would be fairly excluding as opposed to including. Furthermore, advanced headsets often require special software running on a computer and involve a lot of cables and sensitive equipment, prone to breaking in an environment of children. Therefore the more advanced forms of VR, such as the HTC Vive and Oculus Rift, were deemed impractical and cumbersome in the school environment.

In light of this, the cheap and simple Google Cardboard seemed a much better and more inclusive alternative. The Cardboard, however, limits the functionality and performance of an application if compared to one running on the HTC Vive for example. This was deemed acceptable in relation to the difficulties the HTC Vive would bring in a classroom environment. There were also concerns regarding the need to provide cellphones for students to use with the Google Cardboard, as they cannot expect the students to bring their own equipment for classes. Although in reality most students do own a cellphone, and many schools in Sweden already provide their students with a Tablet, so the idea is not so far fetched in the project group's opinion.

4.2 The applications

With regards to the size of the project group, the decision was made to split up into three sub-groups working on separate applications for the project. As the input from the teacher workshop was so vast a few point were picked out for each application in order to cover different grounds and to provide the basis for each applications theme, interaction and educational purpose.

4.2.1 Tree house

The idea of the tree house application is to have a playful and eye-pleasing main menu that connects all the applications together, using a so called Low-Poly design. The user starts in a field and can in the distance see a tree with a small tree house. The application lets the user navigate throughout the scene with the help of light spheres that transports the user to the sphere that was clicked upon.

Due to limited options where to move, the user ultimately end up in the tree house. In the tree house the user can choose from three different objects to interact with, which each initiate a specified application. After returning to the tree house scene from an application, either by completing all tasks or exiting, the user can look at a scoreboard inside the tree house to check its progress and compare with earlier sessions.

Figure 2: Rendering of the tree house concept design.

Figure 3: The book that the user interacts with to start the vocabulary application.

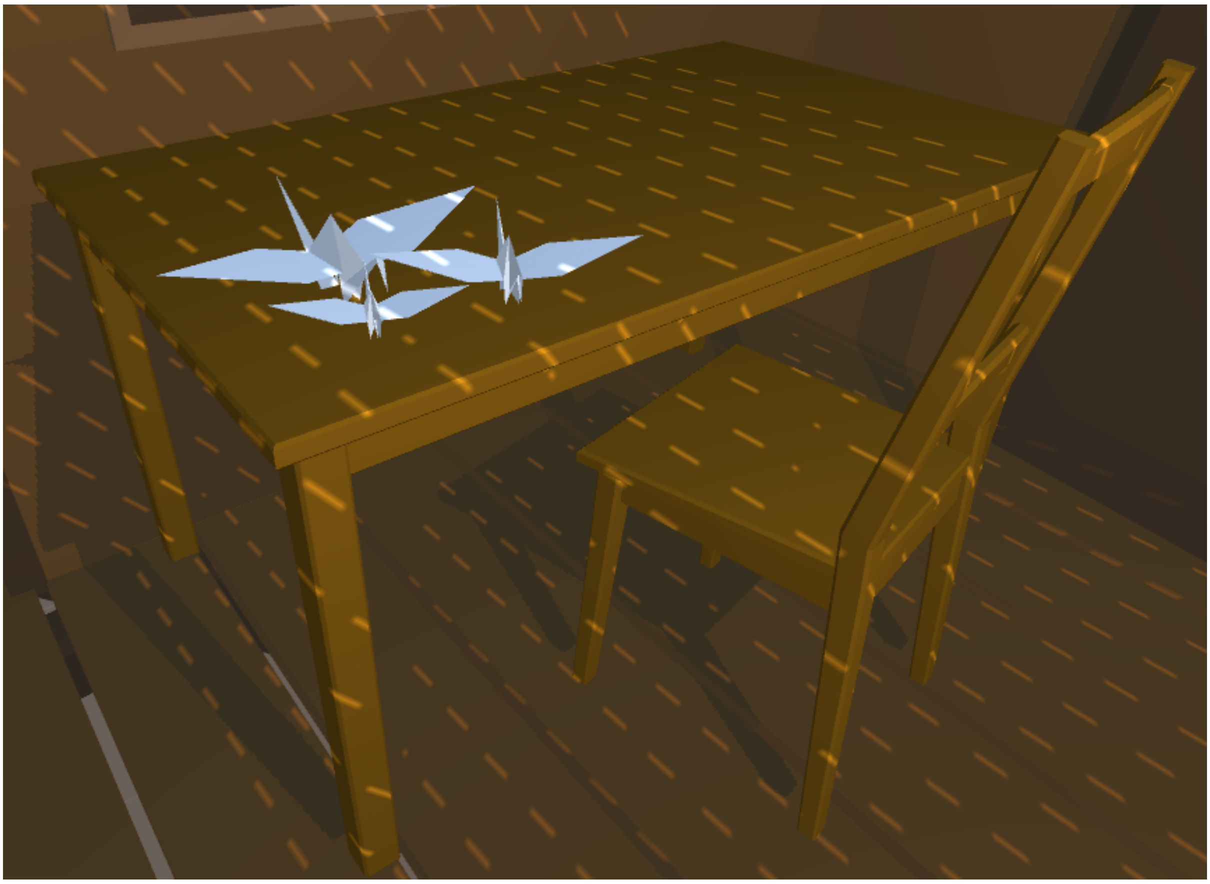

Figure 4: To start the geometry application the user has to interact with the group of paper cranes.

4.2.2 Room application

The application is built as an educational game to help teach kids different types of words as "next to", "above" and "behind". It also combines mathematical concepts as "twice as many", "1/3" and "half of". By using everyday objects in real life events we hope to make the learning more real for the user. It took inspiration from related educational projects using Minecraft with similar objectives.

There is also a multiplayer part in the application. It focuses on making the two users learn by communicating and also teach each other what you learn.

Design

The idea of this application is built on the popular escape-room concept where the player is stuck in an room and need to solve problems to progress to the next level.

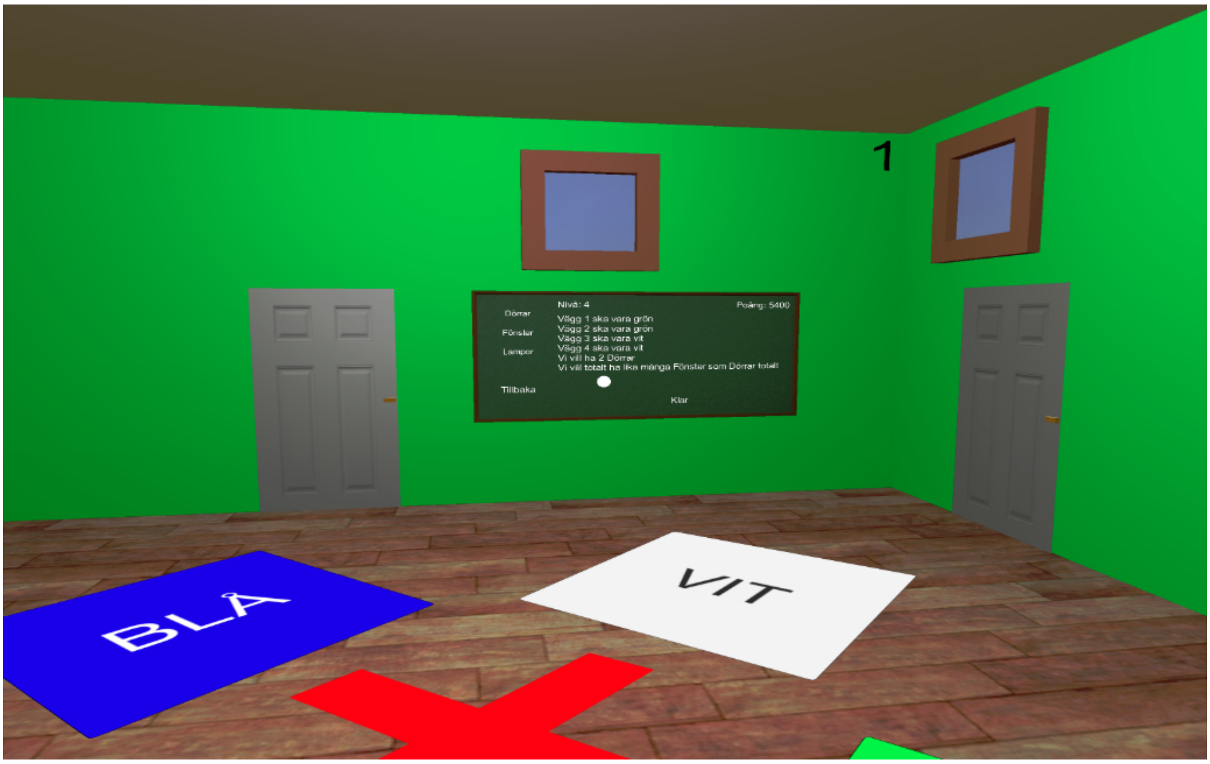

Figure 5: Fully decorated room in the Room Application.

It was pointed out that when placing objects in the room, there was a lot of moving your head back and forth and it would be better to be able to place multiple object before deselecting it. Due to lack of time, the group decided to decrease the number of objects that the level needed to be complete. Also if the room should have many different objects this could take even more time, if the user needs to deselect an object, every time a new one needs to be selected.

4.2.3 Singleplayer

The player starts in an empty room with only a blackboard on one of the walls. On the board, there's a set of instructions that the player must follow to advance to the next level. The user can select colors to paint the walls with by pressing buttons on the floor, or place different objects by selecting them from a menu and then placing them in the room. When the player feels that all of the given instructions are fulfilled they press a check button. If everything is correct the player will be awarded points and progress to the next level with harder instructions.

4.2.4 Multiplayer

Two players starts by selecting what roll they should have, observer or placer. The players are then loaded into separate room, one that is completely empty and one that is fully decorated. It's up to the one (observer) with the fully decorated room to describe to the other (placer) how the room should look.

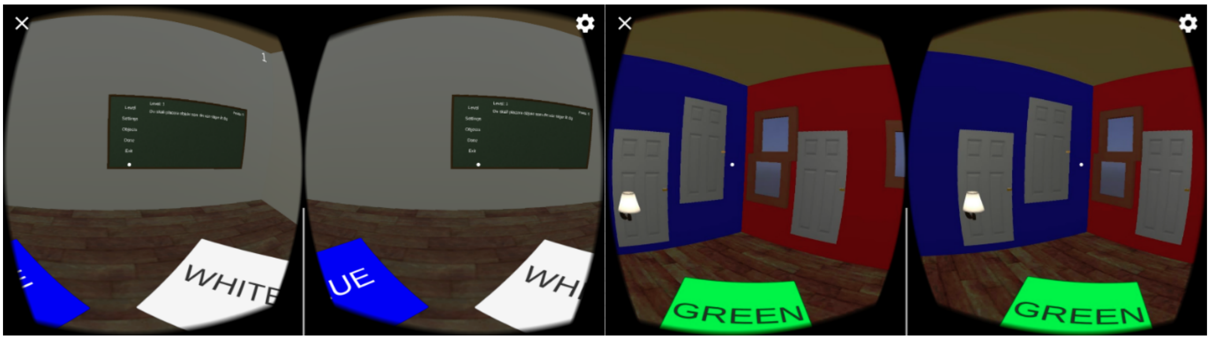

Figure 6: Multiplayer view of the two different rolls in the Room Application.

When everything is placed and colored they press a check button. If everything is correct the players will be awarded points and progress to the next level with a harder decorated room and the roles are switched.

4.2.5 English vocabulary application

The main idea for the application is to learn English vocabulary in a different and fun way. For those lacking the motivation for vocabulary study from a book or paper and pen, this offers a different more fun way of studying and also engages the student for a longer period of time. According to some research[2] children who play computer games increase their skill in English and this was the starting point for the development of the English vocabulary application. The application consist of two different parts, one where the student must pair an object with its correct description and the second where the student must spell an object.

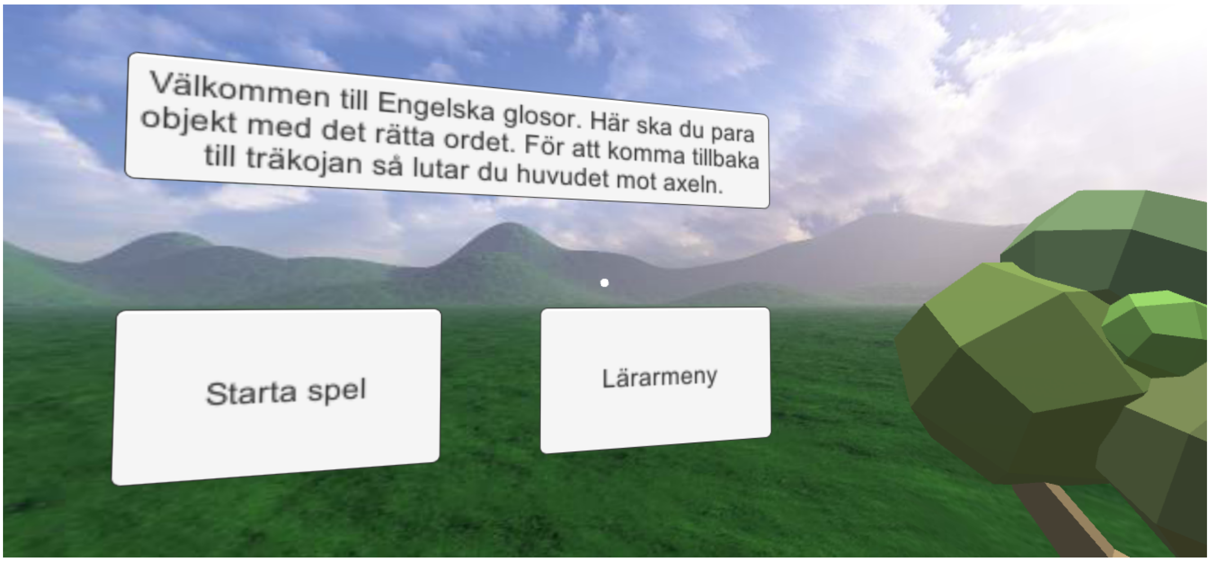

Figure 7: The starting menu for the English vocabulary app.

Design

During the planning of the application the idea was to have a fun and exiting world where there is a lot going on in order for the students to get an intriguing VR experience. The plan was that the student should really be able to interact with the world and to learn English vocabulary by experiencing the natural environment for the specific objects. Also to have the objects move around in the world in a realistic way.

However, due to lack of time, the group had to prioritize the functionality of the application instead of the design. The objects and environments that were used were all free to download and use from Unity Asset Store. The limits of the design therefore depended on which objects that were free from the Asset store.

The objects were placed randomly within a scope and distance in each level/world in order for the application to never look the same when started. The maximum distance between an object and the player was decided after testing different alternatives, it had to be short enough to be able to read all the answers without effort.

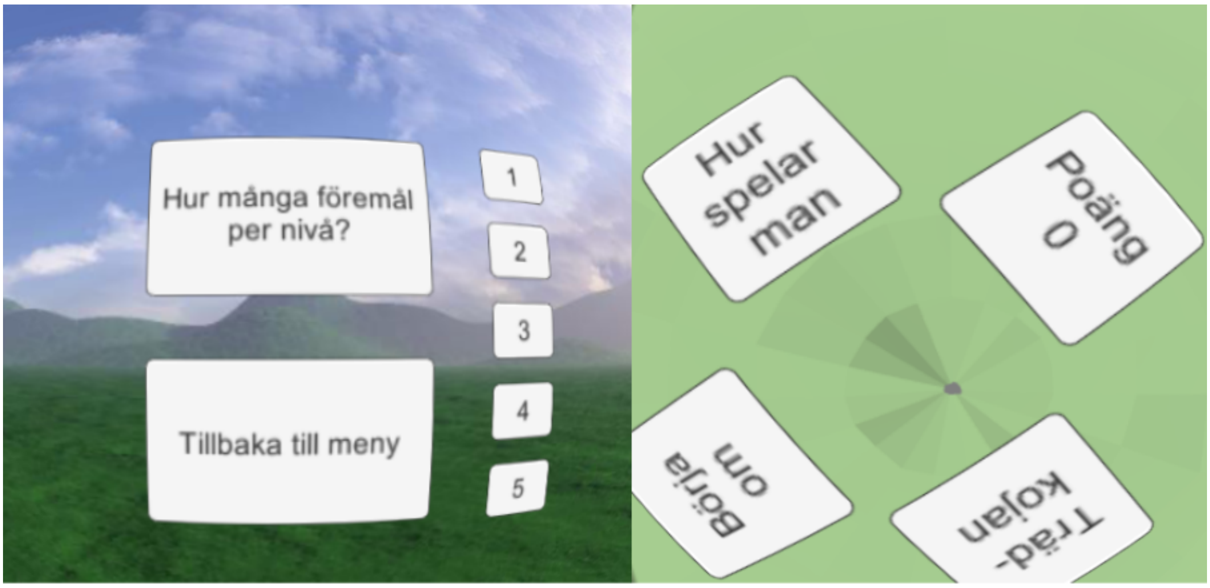

Figure 8: The teacher menu to change number of objects in each world, and a game menu to display score and to go back in the middle of the game.

Vocabularies

The first part of the game is focused on English vocabulary. The student must locate all objects in the level/world, click on them and then select the correct answer out of four options describing the object. Each level has a theme with similar objects. For instance animals, food and vehicles. This is because schools often have vocabulary tests based on a chapter or a theme.

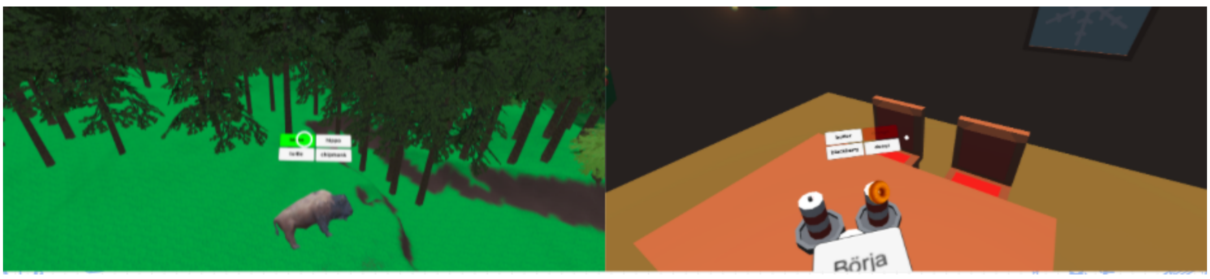

Figure 9: Displaying how it looks after selecting the correct answer and the wrong answer in the English app.

If the student answers correct on the first try for an object, the total score is increased with the maximum object score. If the student gets the wrong answer then the object score is reduced for each wrong try until the student answers correctly and the student increases the total score with the new object score. For instance, if an object is worth maximum 10 points and each wrong answers reduces the object score with 3 points, the student would get 4 points for that object if she answered it wrong 2 times before succeeding.

Spelling

The second part of the English vocabulary application is spelling. The student is loaded into a world with only one object on the side and a number of blocks beside to symbolize the number of letters in the word. On the other side the student sees letters falling down from the sky. The student must click on the correct letter in the right order to spell the object and get points.

Figure 10: The wall of letters and how it looks when selecting a couple of correct letters in the spelling level.

If the student selects the correct letter it turns green and the letter is placed on one of the empty blocks next to the object, the student also sees a green "flash" to know that she answered correct. If the student selects the wrong letter, then that letter turns red while raining down and the student sees a red "flash". If however the student selects a letter that is not correct at the time, but should be placed somewhere else in the word, the letter turns blue and continues raining down.
Due to the short time there is no individual score for the spelling of an object. Instead, the whole level is worth 100 points and for every letter not part of that certain object the score on that level decreases by 5 points.

4.2.6 Geometry application

The geometry portion of the application is intended to help the students gain a more intuitive understanding of geometric concepts such as shapes, sizes, volume and their mathematical applications. Several of the problems they will solve are designed to emphasize conceptual over numerical understanding. Since this is something that VR, as a platform, is uniquely capable of demonstrating. In the real world, it can be difficult to demonstrate 3D concepts on 2D surfaces such as a whiteboard or paper. In contrast, VR environments are specifically intended to be used to render 3D objects.

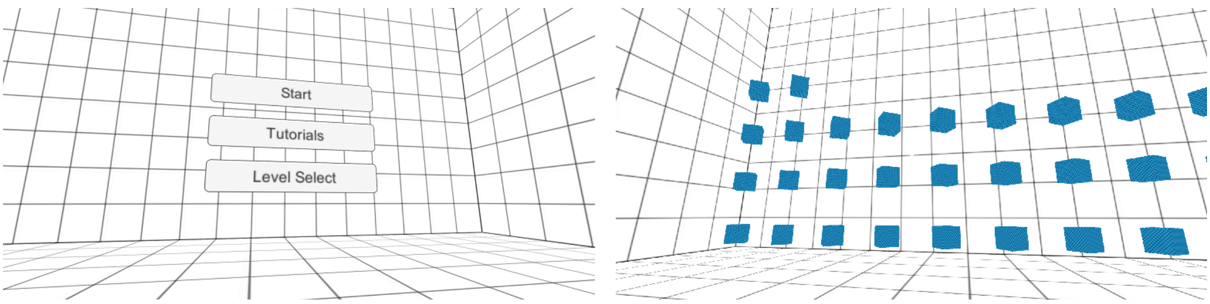

Figure 11: The Main Menu of the Geometry Application.

The hope is that by leveraging this capability, along with the gamification that's possible in a digital environment, we'll be able to better explain geometric concepts while also maintaining a higher level of engagement than traditional schooling environments.

Design

In the early planning stages of the geometry application, it was agreed to make use of a common design framework for textures and shapes.

Due to time and performance concerns it was decided to focus primarily on simple geometric shapes. In order to enhance the user's ability to correctly estimate relative size and distance of objects, the problems are presented against a grid like background. In that vein, most objects related to scale and volume, use a checkered texture to help visualize differences in size.

By presenting the user with consistently textured objects, the user will find the goal of the problems intuitive to understand (e.g. the green cube is generally considered the one we're "working with"). Another consideration was to generally attempt to keep the distance between the user and recurring objects at a constant distance, to aid the user in accurately estimating the scale of said objects.

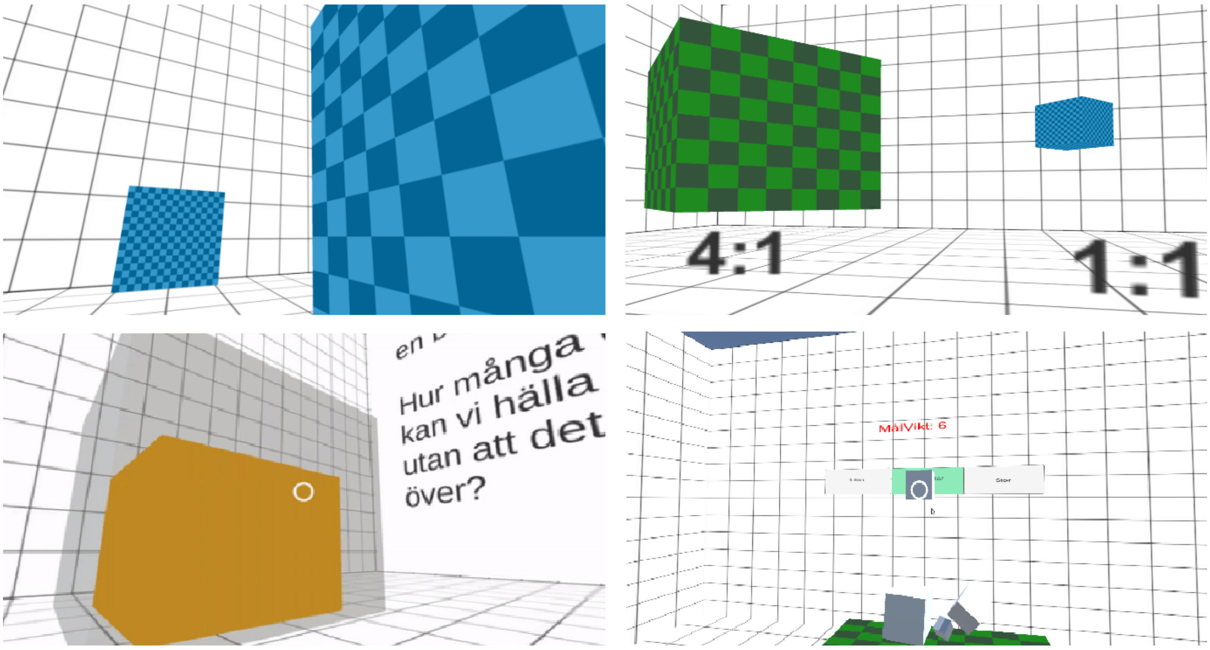

Figure 12: Different problem-sets from the Geometry Application.

Scale modification

This problem is intended to serve as an introduction to scale as a concept. The problem itself is not difficult, but lets the student focus on exploring differences in scale. Once the scene launches, the student is faced with two cubes of differing size. One of the cubes is blue and the other is green. The student is then prompted to alter the scale of the blue cube until it matches the green cube. By giving the students immediate feedback on their alteration to the cube's scale, both visually and numerically, the student has an easier time internalizing how scale functions.

Since this is the first problem the students will encounter, extra care was given to making sure that there's no pressure on the students to "get it right". As such the problem has no failure condition. The students may play around with the problem until they solve it. This problem-set is simple, but by letting the students themselves interact with the scaling of objects in a relaxed manner we're hoping to offer a quicker and more intuitive path to understanding scale as a concept than what is possible with the tools offered in the average classroom.

Scale recognition

Here the student is presented with four cubes of vastly different size. The cube straight ahead of them is colored green, while the other cubes are blue. Upon loading the level, the student is verbally prompted to select the cube that's X times larger/smaller than the green cube. Should the student want to hear the instructions again, they may do so by clicking on the green cube. In this problem-set, the student isn't presented with the numerical scale, but is instead supposed to develop a more intuitive understanding of scale by choosing the cube that reasonably corresponds to the prompt.

The idea here, is for the student to be able to pick the cube whose scale seems reasonable. To help the students gain confidence, the difference in scale is initially enormous. But if the student has repeated success in picking the correct cube the difference in scale shrinks, meaning that the problem becomes more difficult.

If the incorrect cube is selected, the student will receive verbal feedback telling them the scale they chose and wishing them better luck next time. Upon selecting the correct cube, the student receives a brief congratulatory verbal cue upon which they are send onward to the next scene.

Volume

Volume is another concept that can be difficult to convey using the 2-dimensional surfaces normally available in a classroom environment. Therefore, it was decided to investigate possible applications of virtual reality to better demonstrate volume. Here greater weight was given to engaging the students with the learning material through auditory and visual feedback. Further consideration was also given to the feedback received from the teacher workshops and the feedback we received after the presentation in October.

In this series of problem-sets the student is presented with both textual and verbal cues. They are asked to fill a glass container with milk. The unit describing the volume of the container differs from the unit the students are asked to answer in.

The students chooses their answer using a series of buttons followed by them clicking the container. This is then followed by verbal and visual feedback. The container fills to match the volume from the students' input. If they solved the problem correctly they are congratulated and transitioned to the next scene.

Much time was spent making sure that failure didn't feel like it was necessarily a negative thing. For example, if the student selects an input that overflows the container the visual representation of the container will also overflow flooding the floor with milk. If the student input doesn't correspond with the correct amount in general, the verbal feedback attempts to convey this in a very forgiving and sometimes even joking fashion while giving a hint as to what the correct answer should have been.

Weight recognition

This is a part of the geometry application, where the student is given a goal weight and with the help of different cubes they will have to achieve this goal by placing the cubes on a weight scale. The cubes vary in size and weight so the student must do some mathematical calculations before choosing the desired cubes. The student gets a point if he/she manages to achieve the goal weight but does not if the goal weight is exceeded.

This mini-game can be developed to be more complex and challenging for the students with a few changes. The size of the weight machine can be changed so that the student must make the difficult choice of choosing a few smaller cubes to achieve the goal weight rather than 1 big cube.

4.3 User Stories

As mentioned in previous sections there are two main actors that have been focused on while developing the applications.

- Student
- Teacher

4.3.1 Tree house

- Student
  1. The student is loaded into a landscape with a big tree in front them, carrying a tree house.
  2. The student can navigate in the landscape and the tree house through points that transports them to that points location.
  3. The student can interact with objects in the tree house to start the games. When hovering over object, a description of the game is shown in a pop-up bubble.
  4. The student can check it's high score on a board in the tree house.
- Teacher
  1. The teacher can check the student's progress through the scoreboard in the tree house.

4.3.2 Room application

- Student
  1. The student is loaded into a room where there's a blackboard on one of the walls. Instructions on this board tells the user what needs to be done to complete the level.
  2. Student selects a color using the buttons on the floor. It then selects the wall the student wants to paint in that color.
  3. If the instructions says an object (door, window, chair etc) needs to be placed in the room, the student can select different objects from the menu on the blackboard.
  4. When the student presses on an objects name on the board it's loaded into the room. The Student can then move the object to the right place by dragging it.
  5. When the student have fulfilled the instruction, a text spelling "Level complete" is shown on the board and the student can move on to the next level by using a specific door in the room.
  6. The next level is going to be harder with more difficult instructions to follow and more items to place out.
  7. All levels needs to be completed in a certain time or else the student won't get any points for completing the level.
- Teacher
  1. The teacher decides what the time limit for each level should be by using the teacher menu.
  2. The teacher decides how many points the student gets for completing each level before the time ends.

4.3.3 English vocabulary application

- Student
  1. Student looks around in the environment by turning his/her head.
  2. Student clicks on an object using the reticle and four different answers are displayed.
  3. Student selects the correct answer for the object by clicking with reticle. Correct answer turns green and the object and its answers disappears. Student increases his/her score with the object's value.

4. Student selects the wrong answer for the object by clicking with reticle. Wrong answer turns red and the object's value reduces.
5. Student answers on all objects on the level and a new level appears.
6. Student clicks on letters falling down to spell what kind of object is being displayed.
7. Student looks down to see his/her current score.
8. Student clicks on menu by using reticle to go back to the tree house (main menu).
9. Student reaches new high score which is saved and updated on scoreboard in the tree house.

- Teacher
  1. Teacher selects the number of objects per level in teacher menu by clicking with reticle.

4.3.4 Geometry application

- Student
  1. The student is loaded onto a tutorial scene where he/she is taught the basics about scales.
  2. The student completes the tutorial questions and is moved onto the first game scene.
  3. The student has to make box "B" the same size as box "A" by pressing one of the 2 buttons available.
  4. After a random number of question are completed, the student is loaded onto the next game scene.
  5. There are four boxes around the student and he/she has to choose the box which is the same size as the box in the middle of the scene.
  6. After a random number of questions are completed, the student is loaded onto the next game scene.
  7. Every time the student completes a question in any game scene, he/she is awarded points which is updated on the scoreboard in the tree house.
- Teacher
  1. The teacher decides between what sizes the boxes should randomize between and if its a smaller gap then it would make the box sizes close to each other therefore making the game scenes harder to complete.
  2. The teacher can decide how many points is awarded to the student when he/she completes a game scene.

4.4 Testing

White-box testing was used to make sure that all of the applications worked as intended by testing their internal structures. In the end of the project the applications were combined together with the tree house(main menu) which allowed the group to run black-box testing techniques to ensure that all applications send the correct data and the scripts from the different applications do not collide with each other.

4.4.1 Black-Box Testing

With the help of black-box testing, the application can be tested to see if the point system is working correctly. Due to the fact that the tree house acts as a menu in this application, each game instance will have to send packets with the points to the tree house where it will be stored and displayed. The test scenarios that was mentions in testability have been fulfilled and the outputs was the same as the predicted outputs in the table.

4.4.2 White-Box Testing

Tree house Due to the fact that the tree house acts as the main menu, there are not many internal structures that can be tested. When starting the application, the first thing that the player sees is the yellow spheres (way points) that moves the player from the field to inside the tree house. This is tested by making sure that the only way points that are visible is the one in front of the player and the one behind the player.

Room Application *Item movement.* A very central part of the application is moving items that follow a RayCast intersection with a wall. Items must appear where the player is looking. Items must snap to the edges of walls and not pass through. Items must not be able to be placed over existing objects. Items must be able to be picked up after being placed.

Instruction review. The instructions of the furnishing must correspond to the victory state of the game. Especially important as the instruction generator is quite intricate.

Seed. If two multiplayer sessions are started it is important that picking the same seed results in identical levels. Also important that this happens over every level and do not diverge.

Score. Must increase by appropriate amount when finishing level. Must not increase when acting as observer in multiplayer mode.

English Vocabulary Application All testing was mostly performed in one scene. Since it was designed to use a general script for every object (e.g. canvas), it was not necessary to test the script for every level. However, a test for every level had to be done to see if all the objects loaded from the different folders and all the words were correct. The last level had another concept to it and therefore the last level had to be tested separately. A lot of small tests were performed to see that every script worked as intended. After a milestone was achieved the application was built and installed on a mobile device to test.

Geometry Application Each small game in the geometry application has a function which handles the win condition and calls upon the function that will increment the points. The function that handles the win condition can be tested to check if there are any loopholes in the win condition set. The function that will increment the points is also tested to check if it's being handled correctly.

4.4.3 Testing strategy

When implementing new features and updating existing functions the group tested everything manually by rebuilding and playing the game. When reaching the new part implemented a step by step method was used to ensure that the wanted output was reached.

However, the manual testing was performed by the group developing the specific application. In order to improve the result, each group should test the other groups' application. This would also lead to more valuable input for the different groups when continuing the development.

To optimize the testing, Unit testing would've been more efficient. This allows us to test specific parts of the code to see if each function works as expected. For instance, if a function that handles the visualization of objects takes a number as input then the output should result in displaying that number of objects. This kind of testing saves time by running the functions directly instead of having to rebuild the project each time a new feature is implemented.

Another aspect of testing is stress testing where the tester should try to break the program and find errors. This is something that the project group didn't take into consideration while developing and therefore the applications might not be as robust as intended.

5 System design

A more technical description of the application.

5.1 Module diagram

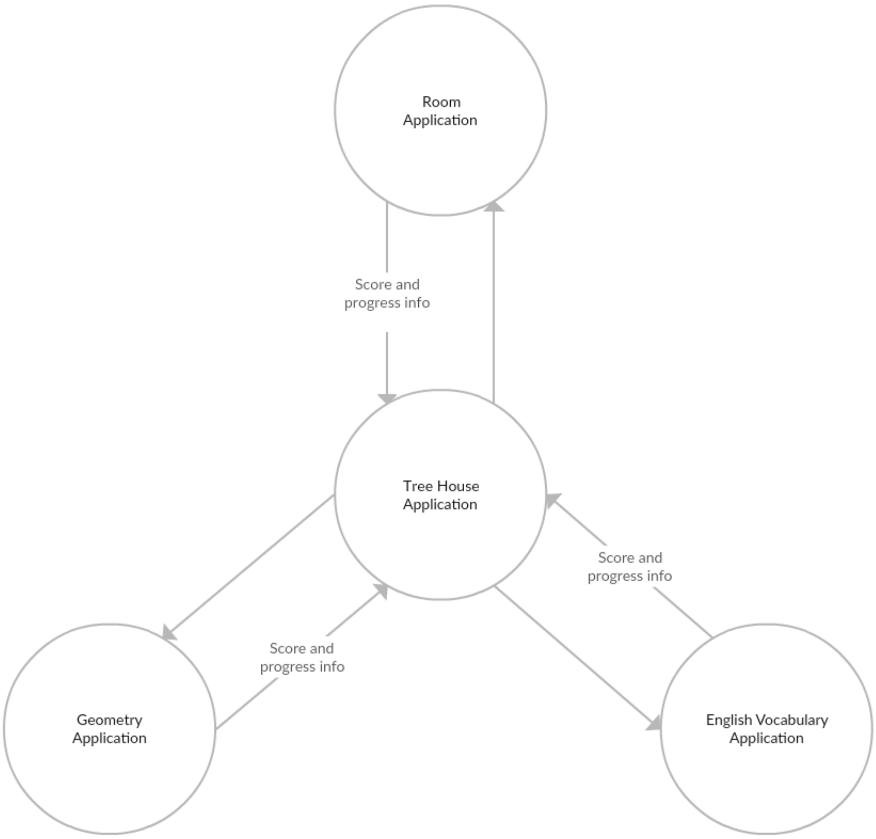

Figure 13: Module diagram of the VR Classroom application

5.2 Tree house

When loading the application, a script is initiated to check the user's current high score for each app stored in a local file on the device. If no file is located or it doesn't exist, the initial high-scores is set to 0. These values are visually represented on the blackboard inside the tree house. If a file with high scores is located, these values will be displayed on the blackboard.

Each time the user finishes or exits one of the applications initiated from the tree house, score and other information from that application is stored to the local file. This local file is then read when initiating the tree house and updates the values on the blackboard.

To navigate in the tree house application the user has light spheres that is activated through point and click. When clicked upon it starts a script which animates a forward movement. To make the user move in the desired path, only the two spheres closest to the user is shown. This limits the choices on where to move.

5.3 Room application

The room application works in a single Unity Scene. The basics of the application is a room scene which surrounds the player who's located in the center of the room. The player is presented with instructions and menu options on a blackboard. On the blackboard, items can be selected and spawned into the scene to be placed on a wall. A spawned object will follow the players gaze through a RayCast beam. Four distinct colors to choose from can be found on the floor. The initial idea was a sort of color picker menu to be accessed through some physical object like a paint can, but the placeholder button proved very simple and intuitive.

5.3.1 Singleplayer

At the start of the game, the player may choose to play in single- or multiplayer mode. When choosing singleplayer mode the game will generate a win state, i.e. a furnishing of the room that is deemed correct and will generate instructions of how to furnish the room from these instructions. Note, however, that it is not impossible for the game to generate instructions that are impossible to complete. For instance, place a lot of objects on the first wall that already has sparse of available space.

When a player feels done, the player press the "Klar" button and the game evaluates whether or not the room is furnished correctly. If not, the possible score is reduced by 20%.

5.3.2 Multiplayer

When multiplayer is selected, the player needs to choose if they want to start as the "Placer" or "Observer". It is assumed they have someone else to play with who picks the opposite. The roles switch after every level. The two players will also have to choose the same fruit in a menu. This fruit represents a seed which is used to emulate multiplayer behavior without an actual connection. When generating the level the game will, as with singleplayer, generate a win state. After that it will try to emulate this win state by going through each object and placing them on the walls randomly. As both instances of the game uses the same seed, both instances will place the object at the same positions on the walls. If placing an object at its randomly chosen position would result in it intersecting another object, the position is deemed invalid. In this case the game will try to place the object again with a new randomized location. If placing fails enough times the game will simply discard the object and change the win state to reflect this. As both instances run with the same seed, they'll both place the objects at the same location, and both disregard them if failing enough.

When all the objects and disregarding has taken place, the scene for the placer is cleared and leaves a blank room as for when playing singleplayer. The observer, however, get to see all the objects and colors placed in the room. But their ability to place objects or paint anything is disabled, as to leave the room static and only for observing. When the placer has placed all the objects the idea is for both of the players to press "Klar" and proceed to the next level, where the roles are switched and the process of generating the room is started over again.

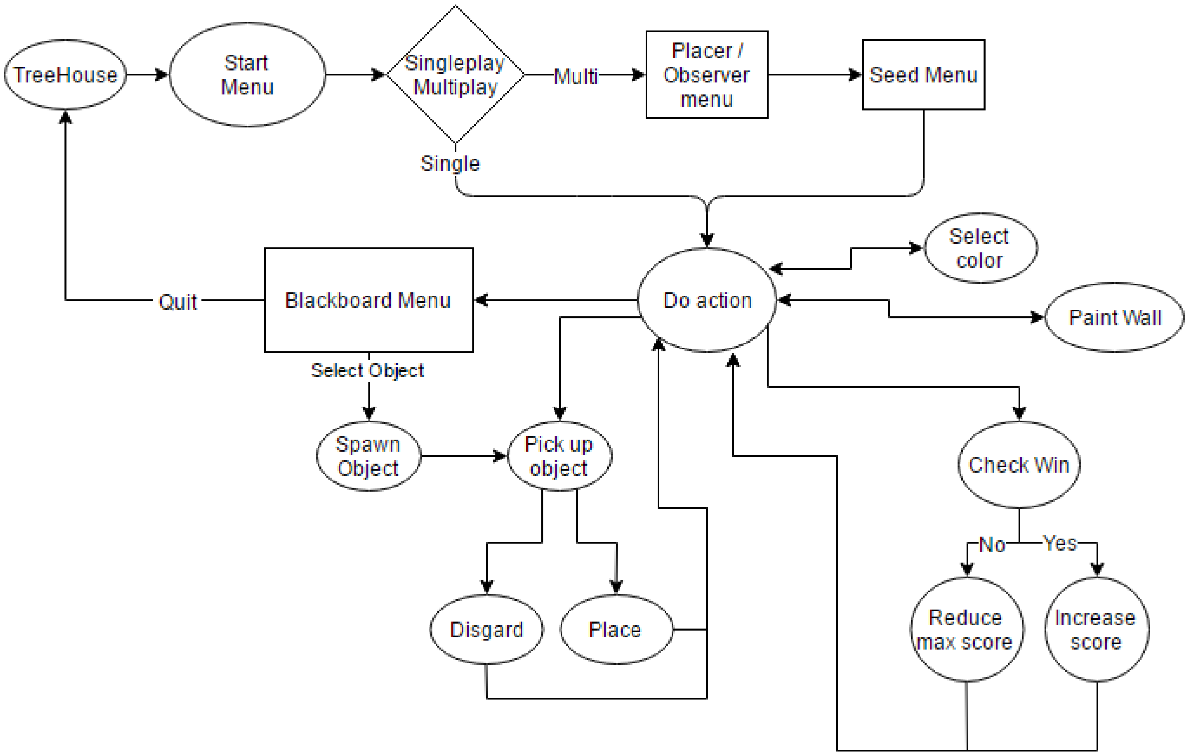

Figure 14: Overview of the Room application.

5.4 English vocabulary application

Each level in the English vocabulary application is a separate scene in Unity. All objects belonging to a scene is stored in an individual object folder. There are general scripts that applies for all objects and scenes. The application was designed this way to it makes it easy to add new objects by dragging and dropping the same scripts on the new objects. There are also a game object included in all scenes. For instance, a scoreboard to keep track on the score and our progress.

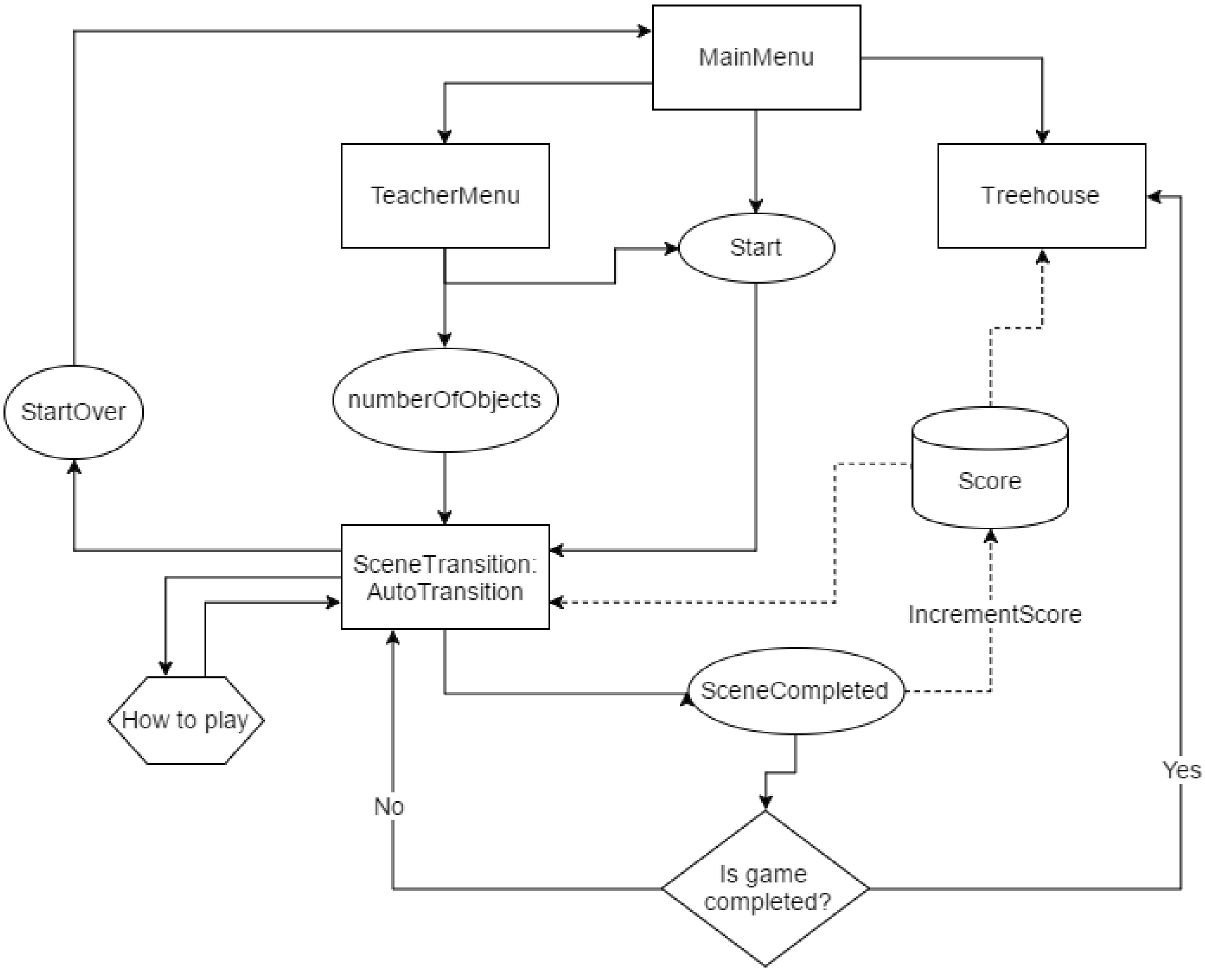

Figure 15: Overview of the English vocabulary application.

5.4.1 Loading a scene

Each level in the application is in a separate Unity file. To change the level in the application, a scene is loaded from Unity's building list which is a list containing all available scenes. The levels are added into the building scene in Unity before compiling.

When a level is completed, a new level is loaded from the list of scenes by calling the scenes' name from a function. The new level could also be loaded by using list index but that would require the scenes to be placed in the correct corresponding building scene index each time the application is rebuild.

5.4.2 Loading vocabulary objects

When entering a level, an initial script is called selecting a specified number of objects belonging to that particular level and displays the objects randomly in the level environment. Each object have a general script containing the object's score and properties. If the object is clicked by the reticle, the canvas script is called with another script that selects three different randomized words belonging to the level.

The forth word is the object's name. These four words are then placed randomly on four positions above the object.

If the wrong word is selected by the player, a script is called to color that particular box red and reduces the object's max score. If the correct word is selected, that box turns green and the score is sent to the scoreboard and then the object disappears. When all objects in the level are gone, the application loads the next level. When the last level is finished, the script holding the scoreboard writes the score to a local file on the device which then is read by the tree house which displays the high scores from the different applications.

5.4.3 Loading spelling objects

The letters are randomly selected and instantiated from a folder for the specific spelling level. Each letter on the letter wall disappears after it has fallen to a certain height. When a letter is triggered, the application calls a script to check whether this letter is a part of the displaying object's word. This triggers a red, blue or green color on the falling letter and a flash in the same color if the color is not blue. The object is selected randomly from all the level folders for variety. When the word is spelled, another script is called that writes the high score to a local file. If it doesn't exist, it creates the file and then loads the tree house scene.

5.5 Geometry application

The geometry part of the application is designed as several independent scenes, each completely self-contained. The scenes are then linked together using two frameworks, one for tracking the user's score and one for managing scene transitions.

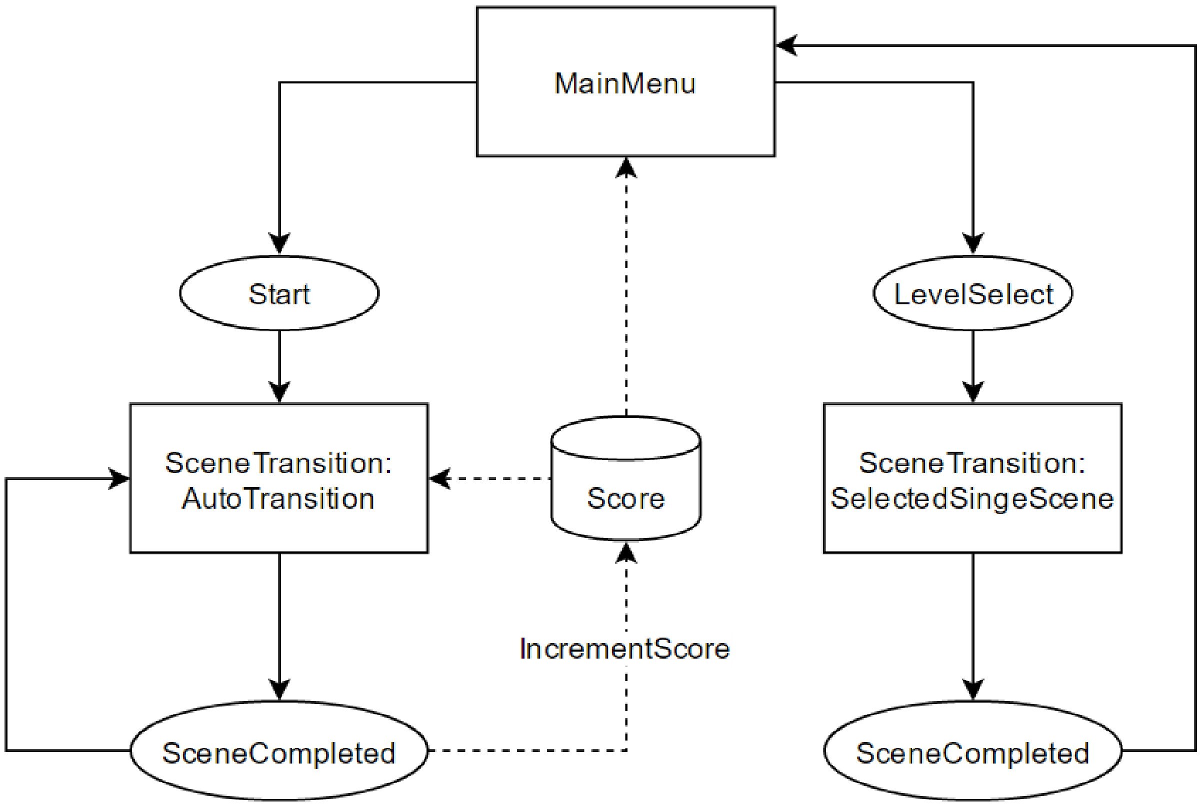

Figure 16: Overview of the geometry application.

5.5.1 Scene Transition Framework

Each scene presents the user with a single problem for them to solve. Once solved, either correctly or incorrectly, the scene calls the framework in order to issue a scene transition. All scene loading goes through this scene transition framework, even if the scene is simply reloaded.

The scene transition framework decides which scene should be loaded. In general, this will take the user's score into consideration. For instance, if the user's score is 0 it means that this is the first time they're starting the geometry portion of the application (since you get a point for the tutorial) and the user is sent to the intro scene. In case a user has selected a level manually from the "Level Select" menu, the scene transition framework will always send the user back to the main menu upon scene completion.

In all other cases (i.e. not manual level selection or tutorial) the scene transition framework will take the threshold set by the teacher in the admin menu into account when making a scene transition. This threshold is used to introduce additional problems over time at a rate set by the teachers.

For instance, if the threshold is set to 4 then every 4 points the user gains will result in an additional problem being introduced to the user.

5.5.2 Score Framework

The score framework functions as a global point of access for the scoring system in the geometry part of the application. It manages the incrimination of score at the successful completion of a scene, as well as the loading and saving to persistent storage. It's generally accessed during scene transition in order to modify the score upon successful completion of a scene to display the score to the user and in order for the scene transition framework to correctly choose the next scene.

6 Future development

In general, the application needs a more unified design where all the models has a low-poly design with simplistic textures. This would reduce the strain on the device using the application. The code also needs need a more unified design which would optimize the performance, making the application run smoother.

Some additions to the current applications could be:

Reward system To give the application a more gamified feeling, a reward system can be implemented to encourage the user to get better in each field.

Avatar Lets the user have a character that can be personalized using experience points or rewards.

Teacher menu Lets the teacher customize the entire game for each student directly from the tree house. In the current application the teacher can only change options in each game, which is time consuming and.

Cloud service Upload and store users' progress in the cloud to allow access from different devices.

In addition to the general goals for future development there are also some ideas for each small application.

6.1 Tree house

- Student
  1. Be able to interact with more objects that either starts an animation or a new application.
  2. Be able to save and retrieve progress in the cloud.
- Teacher
  1. A teacher menu where you can change game settings for all the apps.

6.2 Room application

- Student
  1. Be able to place items on the floor and ceiling.
  2. More items and colors.
  3. More advanced and comprehensive instructions.
  4. Prettier color picker menu.
  5. Unlockables. Spend points to buy more objects or new textures.
- Teacher
  1. A teacher menu where you can change game settings.

6.3 English vocabulary application

- Student
  1. More levels to play, with increasing difficulty.
  2. Add different kind of vocabulary like verbs and adjectives, not only nouns.
  3. Be able to use the different words and objects in sentences.
  4. Practice spelling by having an object with similar answers but different spelling.
  5. Practice listening to someone speaking English and then answer questions and interact with the enviroment. For instance listen to someone explaining directions to a location and then making the student click on the directions to move forward to the, hopefully, correct location.
  6. Put the objects in a more realistic environment, like walking around at the zoo or inn a safari and find animals.
  7. Unlock new levels after gaining experience and increasing score.
  8. Be able to select which level to start on if the student comes back later and doesn't want to restart from the beginning.
  9. Get hints or "buy" a letter if the spelling is too hard in the *matrix wall* level.
- Teacher
  1. Chose number of objects in the *matrix wall* level.
  2. Chose specific objects per level.
  3. Set a certain point-threshold to reach next level.
  4. Reward system.
  5. Be able to add new tasks and vocabularies.
  6. Customize the application in a better way to match this weeks' English chapter for instance.

6.4 Geometry application

There are many things that could be improved or further expanded upon in the geometry portion of the application. The largest portion of which is, of course, covering a larger portion of the curriculum. But there's still work to be done with the current systems and problem-sets.

Expansion of curriculum This would partially include expanding the application to cover grade 7-9 by introducing areas such as uniformity, symmetry, further geometric properties etc. But also to cover more of the current 1-6 grade curriculum, particularly how geometry relates to everyday situations.

Furthermore, it would be possible to expand on the number of tutorials as well as the non-interactive introductory portions of the application. This could include things such as animations showcasing how multiple smaller objects can be stacked to create larger objects, altering the size of objects to show practical use cases for scale (ex. Scaling up an ant to better study its body or scaling down a model of the solar system to gain a better understanding of how far apart celestial bodies actually are).

The current state of the application could use some general polish, but there are some specific things that could be added or improved.

Instruction system A more cohesive instruction system. Currently the geometry application features a mix of verbal and textual instructions. This could be improved upon to feature a more standardized way of conveying instructions. One way of doing this would be to feature a prefabbed object containing text that could be set at the start of the scene, and "onSet" calls the platform's text-to-speech system (this would need to be different for Android and iOS).

Set a trigger that monitors input and automatically repeats the instructions if no input has been detected after a set time.

Improve the fade in/out system used in the "Milk scene" to fade in a more user friendly amount of time. This system could also be implemented in the other scenes.

The main menu currently doesn't feature any instructions. This could be improved by having a short scene welcoming the student to the app, and giving him/her a brief overview of the menu.

Tutorials Currently, only the first problem-set features a tutorial sequence (the one with the two cubes). This should be expanded to cover each problem-set.

The tutorials currently, largely, only instructs the students in how to solve the problems. This was done with the assumption that the teacher will be going over the general concepts beforehand. But it could easily be expanded upon to also contain a brief overview of how the subject area itself works.

Due to a scaling bug, some text is rendered with very low resolution.

A system to have the user repeat the tutorials upon repeatedly failing a scene could be implemented. This would preferably be done in the scene transition framework. By tracking the last scene and if the score was incremented upon scene completion.

Main Menu The tutorial button in the main menu currently lets the user repeat the "introductory sequence", but should expand into a sub menu that allows the user to select the tutorial that they want to access.

The button/image that returns the user to the tree house was originally intended as a place holder, with the idea that it would be replaced with a low-poly model of the tree house.

The main menu could be expanded to include an options menu for things such as "replacing the milk with orange juice".

An idea that we didn't have time to implement was to have a "store" where the user could spend their points on small cosmetic upgrades such as changing the color of the cubes, altering the backgrounds, replacing the "score cloud" etc.

The score wall currently doesn't have a way of handling the score when it exceeds 100 points (it'll continue to grow outside of the grid background). This could be improved by, for an example, representing 100 points with a larger red cube etc.

The main menu also contains a relatively bare bones "Admin Menu". This could be improved to contain settings to modify the order in which the problems are presented to the student, as well as modify the threshold that adds individual problems to the pool of available problems.

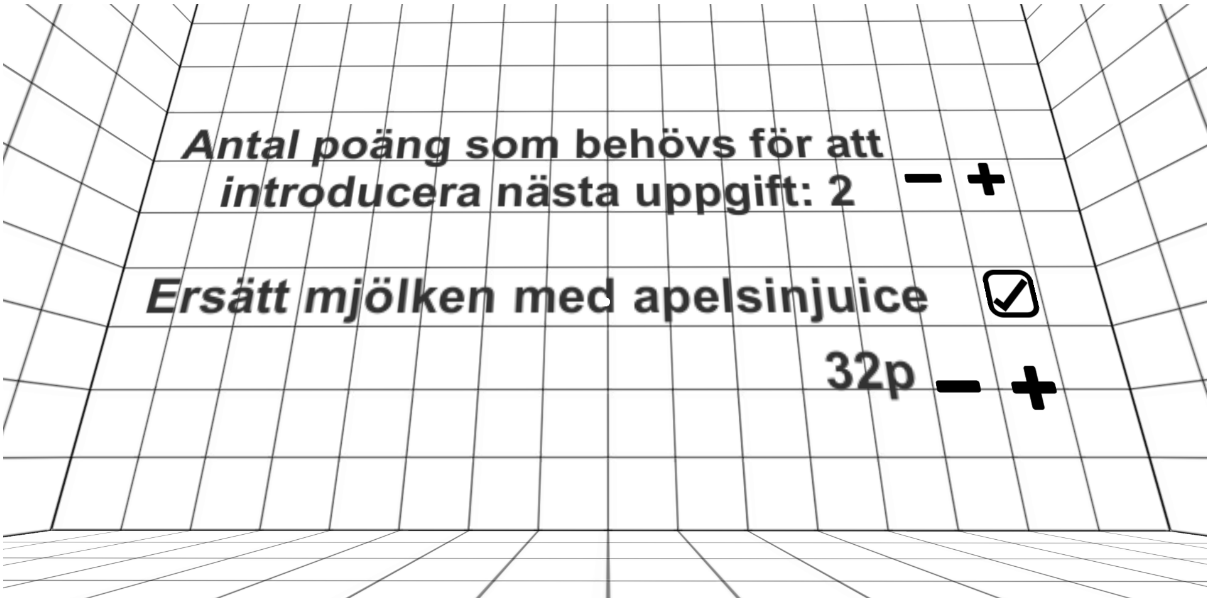

Figure 17: The Geometry portion of the application's Admin Menu.

Access to the Admin menu should also, in some way, be limited to the teachers as it contains controllers for altering the score and scoring system.

Currently the menu items are annotated in a mixture of Swedish and English, this should be converted to English or possibly converted to support both languages through a setting.

The problem-sets There are some states for which the controls in the 2-cube problem aren't moved correctly. omber

Due to the way the text-to-speech system is implemented at the moment, there's no way to have completely dynamic scenes. This could be fixed by implementing a way to access the native TTS systems of the different platforms.

The 4-cube problem could be expanded to also include states where you are supposed to pick smaller cubes.

The Milk problem doesn't fill the container if the input is 1 ml (due to division used in the algorithm controlling the milk level).

7 Sustainability

There are a lot of opinions on VR today and during the project the group came into contact with some, if not many of these questions. New technology is very often thoroughly investigated and debated, especially questions that don't have either right or wrong answers. The following section will contain some of these question formulations and what the group found out during the span of the project.

7.1 Ethical evaluation

The project did not follow any ethical guidelines because there was no need for it. The project is just a concept of how virtual reality can be used in school and should be viewed as such. The application is a tool for helping students in school and it does not record any personal data at the moment.

7.2 Financial evaluation

The cost of VR is still quite high for schools to implement as a full scale educational tool. Most schools have very limited resources and can't afford a set of phones and VR headsets for every student in every class even though this is one of the cheapest alternatives today. However, a small amount of headsets could be bought without a problem and be used as a complement in various subjects. Visiting a museum with VR instead of chartering a buss for a whole class saves both time and money. Things that was not possible before, like traveling to a different country, is made affordable with VR.

7.3 Social evaluation

Increased screen time is often seen as something that makes people less social and it's easy to "get lost inside the virtual world". VR is no exception and in some cases a user can get even more immersed with VR which disconnects them from the real world even more. VR is something that should be used in moderation and should never replace anything in any subject but should be used more as a complement. However, the possibilities to let children with various disabilities "visit" places with VR that they normally would not be able to can increase the general quality of education for them. Also, if a student lives far away from the school they can even use VR to be part of the classroom even if they are not really there. The project tried to tackle one of these problems with the inclusion of a cooperative multiplayer mode in one of the applications. Letting students work together for a common goal can teach them about teamwork and shows that games can be social tools as well.

8 Discussion

In an effort to show the possibilities with Virtual Reality for an alternative learning tool in school, a small application was developed to show what is possible in certain classes. We found that the geometry part of the application could be a large contribution in visualization and understanding of the 3D-space. One could also show the relation between a simple shape and volume. This in turn could open up for a more mathematical approach where you learn about mass, volume and density.

For the children that are playing a lot of games and having a hard time focusing on the traditional paper-pen learning in school, this could be a good alternative learning method. Since you are playing a game, although be it a learning game, these kids might not be as bored and might even engage in it making the learning experience more fun and remembered.

We chose to work with the Google Cardboard since we wanted to work with something teachers can use right now and not in a few years and by using Unity you can easily write your own virtual reality application. HTC Vive is too expensive today for a school to purchase for a whole classroom and you also need a lot of space for it to work.

The problem with Google Cardboard is that it might be too "simple" for it to be engaging. With HTC Vive you can look around, move around, the movements of your arms are responsive in game and everything seems to be more immersive. With Google Cardboard your only input is the rotations from your head and a little button on the side of it which acts as a click for a reticle in the middle of the cellphone screen. This makes the games for the Cardboard more basic and might seem boring to the more experienced gamer. The Google Cardboard, with it's lesser resolution and being more prone to lag, can also be a more nauseating experience than more expensive alternatives.

Another disadvantage on the Cardboard, as of today, is that the teacher can not see what the student is doing or how it's going without the student sharing the screen from his/her cellphone.

The application as a whole right now lacks a coherent design throughout. Some models in the game have very high polygon or texture resolution, while others are more basic. A number of objects in the game are simply cubes by themselves, or put together to mimic a real life object. Furthermore the different applications have different approaches to exiting the application and returning to the tree house. One does this by rotating the device 90 degrees to the right. Another by navigating a menu, and a third by pressing a picture of the tree house in its own main menu. This, of course, is a product of the different applications being developed separately and are almost completely disjoint. The only thing connecting the applications is the scoring system, which was discussed. However if removed from the tree house menu the games could be played entirely separately.

8.1 Problems and Challenges

This was the first time any of us participated in a group with this many members, which has meant that it was also the first time we'd needed to coordinate and divide the work up this efficiently. Overall, this has worked out fairly well, though there were a couple of points at the start of the project where it was difficult to divide the work in a reasonable way.

Before the project, almost none of us had any experience with Unity, game programming or virtual reality. The big challenge initially was simply to familiarize ourselves with the development environment and get a feel for what works and doesn't work in VR.

Another concern over the course of the project has been performance. Smartphones offer limited performance compared to traditional computing platforms. Particularly when it comes to graphics. Sometimes it was difficult to accurately estimate the performance required on the phones, since our development environments were run on our own laptops and testing was primarily done directly in Unity.

8.2 How does VR affect children?

Since VR has not been around for that long in practical purposes there are no long term studies on this subject. However, there was an article[3] that said there shouldn't be any side effects when we are talking about eyesight. Arguments were made that since you are so close to a screen and your eyes are developing it may screw with your eyesight. Counterarguments were made that since each eye have its "own" screen, the objects' placements could be made more to the center of each screen creating the illusion that the object on screen is far away meaning that the eye is in a relaxing state and not focused on a fixed point close to you i.e reading a book or looking at a computer screen.

The age restrictions that have been placed on products like the Oculus Rift and the Htc Vive are mainly based on the size of the product and that they simply are to large for kids to use. The age restrictions are based more on a juridical point of view than a technical.

9 Conclusion

Virtual reality is still a very young field, despite this it's already attracted much attention from the industry and the education sector. It's well positioned to serve as a unique teaching aid in the classroom and has the potential to completely disrupt the traditional classroom setup.

The interactivity and sense of presence offered by virtual reality enables both the teachers and students to experience learning environments that would've previously been unfeasible with the limited budget generally available to the public schooling system. But perhaps most importantly, it offers the ability to explore completely new ways of teaching and learning.

The application produced in this project serves a good initial step for practical testing of these kinds of systems in a real world environment. As well as a possible groundwork for a more extensive future application.

The ability to leverage the ever growing pervasiveness of smartphones could allow virtual reality to integrate with the current curriculum offered in elementary school at a relatively low cost.

Over time, the mounting popularity of virtual reality systems and experiences will continue to attract growing attention from the industry making it an increasingly attractive option in the field of education.

References

[1] Adams R. L. (2016). **Virtual Reality Is About To Revolutionize These Three Industries.** [online] Available at: http://www.forbes.com/sites/robertadams/2016/09/07/virtual-reality-is-about-to-revolutionize-these-three-industries/#2c7204487228 [Accessed 20 Dec. 2016].

[2] Sundqvist, P., & Sylvén, L. K. (2014). **Language-related computer use: Focus on young L2 English learners in Sweden.** ReCALL, 26(01), 3-20.

[3] Hill S. (2016). **Is VR too dangerous for kids?** [online] Available at: http://www.digitaltrends.com/virtual-reality/is-vr-safe-for-kids-we-asked-the-experts/ [Accessed 13 Dec. 2016].

[4] The VR Classroom. (2016). **Github repositories** [online] Available at: https://github.com/VCRLTU.

[5] The VR Classroom. (2016). **Finished project available on google drive** [online] Available at: https://drive.google.com/open?id=0B1jr1wDdhihYWEZhaGJTSktjMXM.

[6] The VR Classroom. (2016). **Facebook page** [online] Available at: http://bit.ly/2ivTQZ1.

[7] The VR Classroom. (2016). **Webpage** [online] Available at: https://vcrltu.github.io/.

[8] Google VR SDK Unity **Webpage** [online] Available at: https://github.com/googlevr/gvr-unity-sdk

[9] Google VR, **Get Started with Google VR in Unity** [online] Available at: https://developers.google.com/vr/unity/get-started

A Appendix

A.1 Summary teacher workshop

Background
The project group invited a group of teachers for a workshop to hear their thoughts and ideas about how you could implement Virtual Reality into the schools classrooms. They tried different applications on both the Google Cardboard and HTC Vive and then we discussed the experience. Below is a summary containing what was said under the workshop and this became our basis for our project.

- **Google Cardboard**

  Advantages

  - Visualize objects in another way.
  - New and exciting, which leads to motivation.
  - Opportunity to "visit" places.
  - Cheap to acquire.
  - Easy to set up and use.

  Disadvantages

  - External party can't see what the student is doing.
  - Access to cellphone might be hard.
  - Dizziness might be a problem.
  - The technology can become a bottleneck when hassling.

- **HTC Vive**

  Advantages

  - External party can see what the student are doing.
  - A lot more interactive.
  - Functionality on the controls was easy.

  Disadvantages

  - Expensive.
  - You need a lot of space.
  - Children with small hands might have a hard time reaching all the buttons on the controller.
  - Not that easy to setup.
  - You can get tangled up in the cable.

School topics
The teachers had a lot of ideas on how to implement Virtual Reality in certain topics and some of the proposal are listed below.

1. **English -** Meet a person i VR and talk to him/her and answer questions that the person asks.
2. **Civics/Geographic -** Travel to a location, ie Museum or the ocean, and experience different cultures in a more interactive way with your own eyes.
3. **Natural Sciences -** The possibility to see how an atom is built. See the life cycle of water, experience photosynthesis first hand.
4. **In general -** Concepts as before, under, behind, inside, a half, a fourth and so on can be easier to understand via visualization. Paint a picture from adjectives.

Remaining thoughts

Children with disabilities, i.e handicaps or autism, could have it easier to visit a certain place after seeing it through Virtual Reality first. i.e from a 360 camera recording or a 3d model of this place.

The game must differ from memorization and learning, the tasks must be random to a certain degree.

VR, specially through HTC Vive, gives a strong impression and might get students to better remember things they've learned.

It can be easier for children with concentration difficulties to learn if you're enclosed in "another world".

Many adults gaming mindset can be limited and it might be good to see how children think about gaming and learning.

Teachers design a plan for each lesson, so the application must also be able to follow the plan. You should not be able to do anything else that is not relevant for that lesson.

A.2 Installation instructions

Before being able to install the application, one must acquire the Unity project files. This one can be done at the project's Github Repository[4]. If using the git client the project can simply be cloned to a local directory. If you are using the Github web page to download the project directly you will retrieve a .zip file. This must be unzipped first. At this point the project can be opened in Unity by selecting "Open" and designating the VRclassroom folder as root folder.

Alternatively, if using an android device, you can acquire the android package file directly through `https://drive.google.com/uc?export=download&id=0B1jr1wDdhihYQjNFT1d4NUtTelU`, and thereby skip the entire section A.2.1.

A.2.1 Building

1. With the Unity project loaded, open File → Build Settings...
2. Make sure the correct scenes are added to *Scenes In Build*. Since the application is made out of three smaller ones each group had their own directory system therefore the scenes aren't together.
   - English Vocabulary scenes - *Assets/English/*
   - Room application scene - *Assets/RoomVR-master/Assets/Scene/*
   - Geometry application scene - *Assets/* and *Assets/Scenes/*

   The tree house scene must be on index 0, first in the list, since this is the scene it loads by default when starting the application.

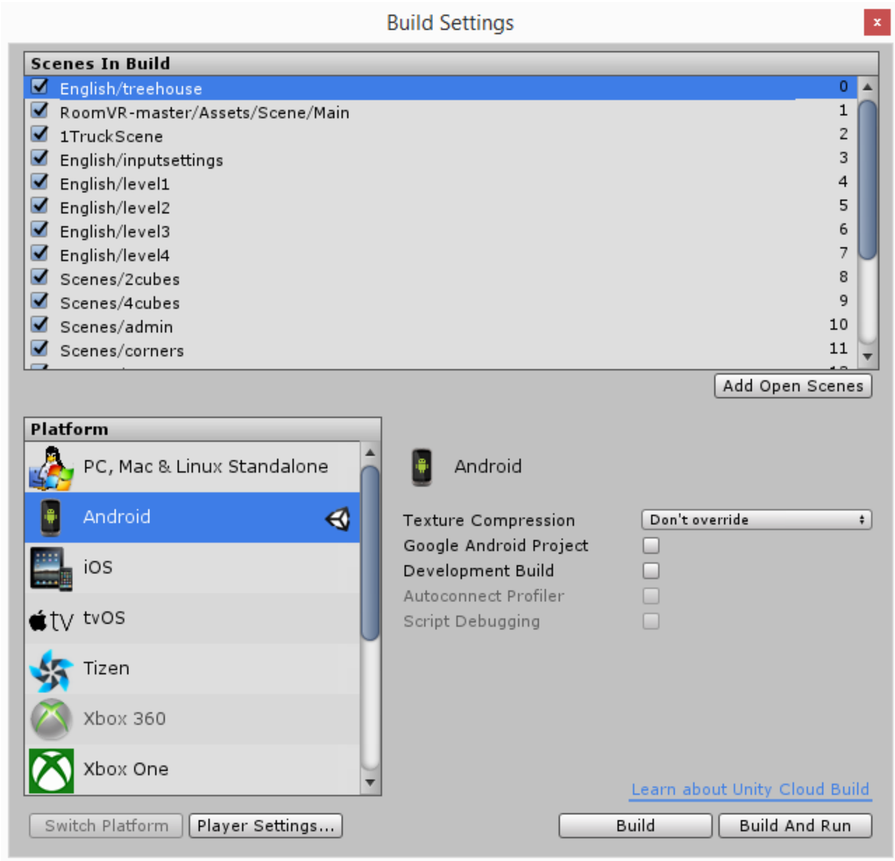

Figure 18: Screen of the scene builds

3. Under *Platform*, select your desired platform, either Android or iOS, then press *Switch platform*. This might take a while. In case you can't switch to that platform there will appear a button before you that takes you to the necessary sites to download the correct modules.

Figure 19: Screen of iOS module button.

Figure 20: Screen of switch platform button.

4. Press *Player Settings*. A new menu should appear in the Inspector window.

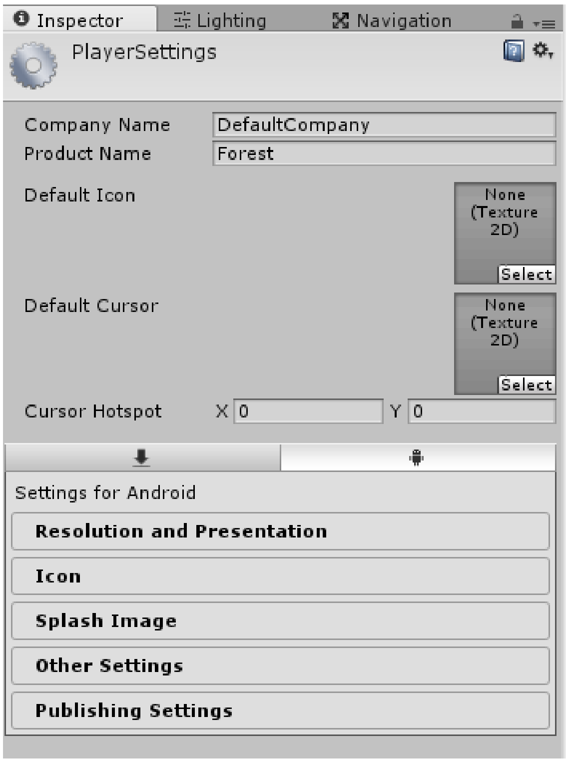

Figure 21: The window after "Player settings" is pressed.

5. In *Player Settings*, make sure that *Orientation* is set to *Landscape Left* under the *Resolution and Presentation* menu.

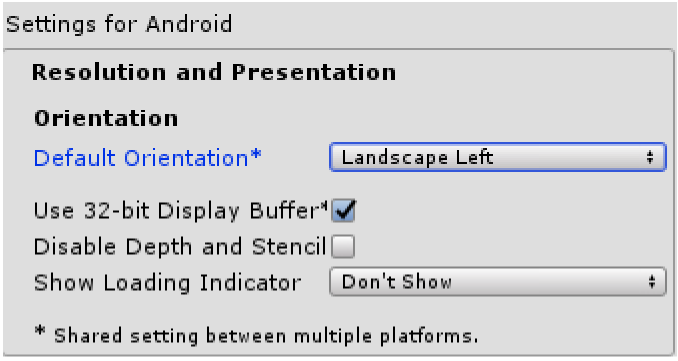

Figure 22: Resolution and Presentation.

6. In *Player Settings*, make sure that a unique *Bundle Identifier* is set under *Other Settings*.

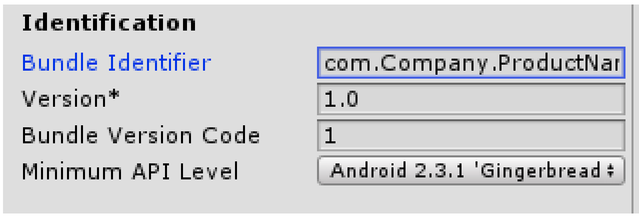

Figure 23: Screen of the bundle identifier

You may change company- and product name but you need to make sure that the bundle identifier matches the company- and product name, which on this case it does not.

Figure 24: Screen of the bundle name.

7. Press the *Build* button to build the project. Select a directory to save the file to.

A.2.2 Install on Android

1. When an Android Package (.apk) file has been generated it must be transferred to your Android device. This can be done in a number of ways. For instance
   (a) Cable transfer
   (b) Cloud service (Dropbox, Google Drive, etc.)
   (c) Emailing oneself
2. With the file on the Android Device, the file must be run and thus the application installed. Some modes of transfer allow the file to be run directly. Others place the file on the device, and must be navigated to using some File management application.

3. When the file is run, an alert will appear if .apk files from an unknown source are not allowed. The alert will most likely allow for a “one time only” exception to this rule for this file only. If any complications arise, .apk files from an unknown source can be allowed in the Developer Menu. Instructions on how to enable the Developer Menu can be found online.
4. After the file has been run the application should be fully installed and ready to be played.

A.2.3 Install on iOS

1. After building the application for iOS, go to the folder where the build-file was placed. Open Unity-iPhone.xcodeproj in latest version Xcode. If Iphone 6 or later, you need to install Xcode 8.2.1 (or beta).
2. Click on Unity-iPhone on the menu to the left to get the following screen.

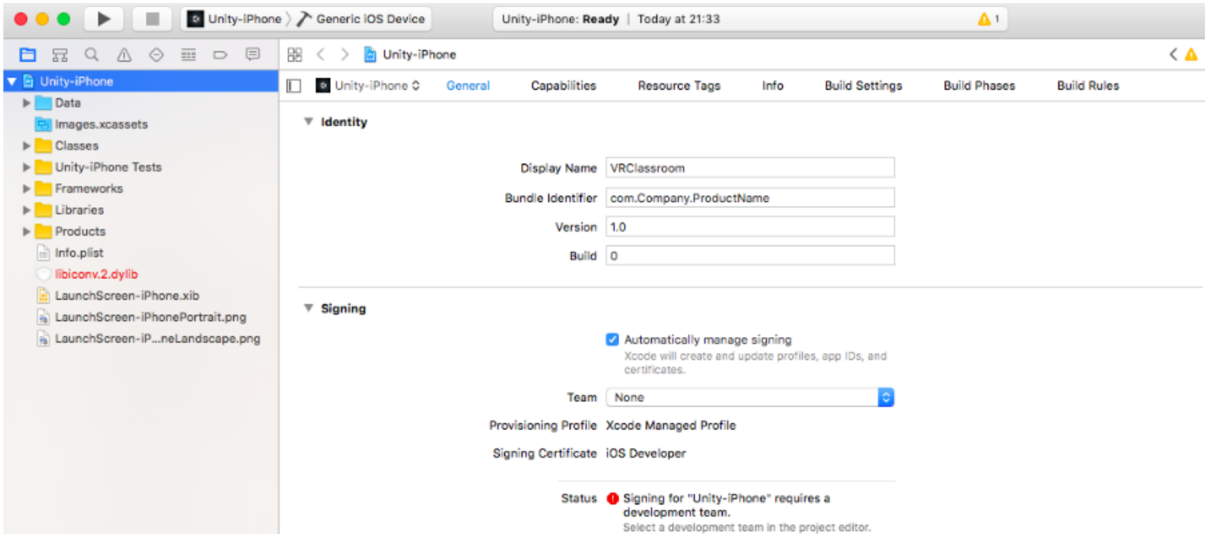

Figure 25: Screen after clicking on Unity-iPhone in Xcode.

3. To fix the error displayed next to *status*, you need to select a team. This is your Apple Id. If no team is in the list, just select *‘add an Account’* to add login with your existing apple-id.

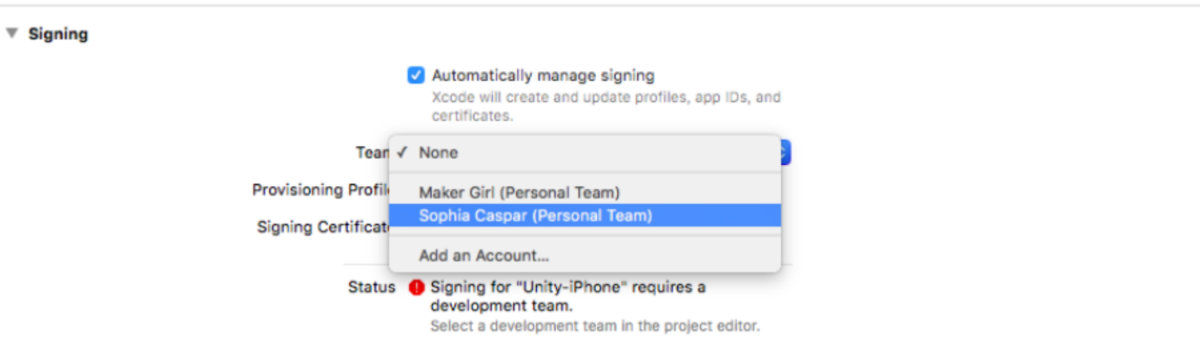

Figure 26: Screen to add a team for the application.

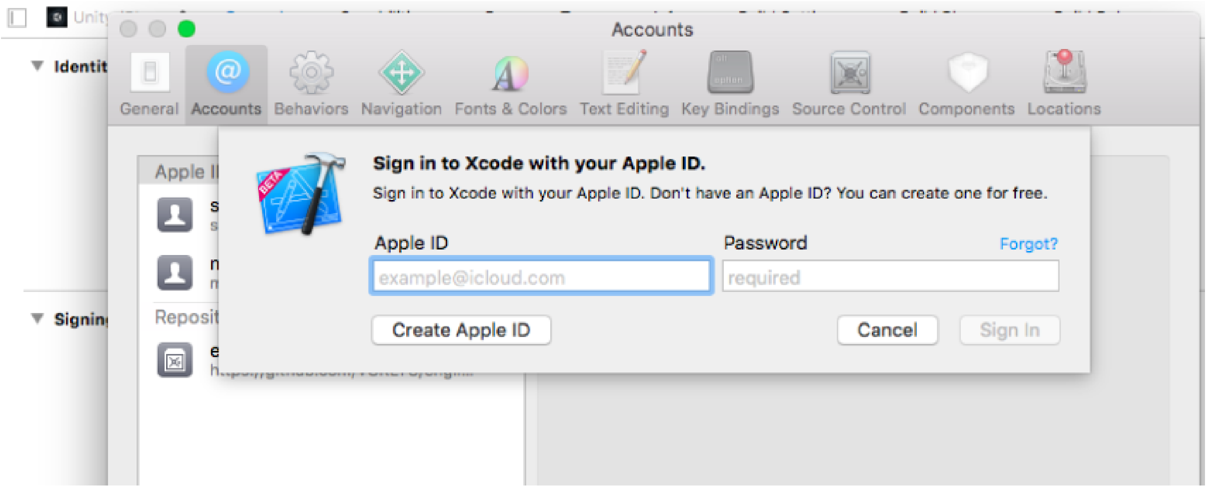

Figure 27: Add a new team by signing in with your Apple ID.

4. After you've selected a team you need to change the *Bundle identifier* from default in the identifier-section above the signing-section where you selected team. This should be in the form `com.name.appname` but can be named something else of your choice.

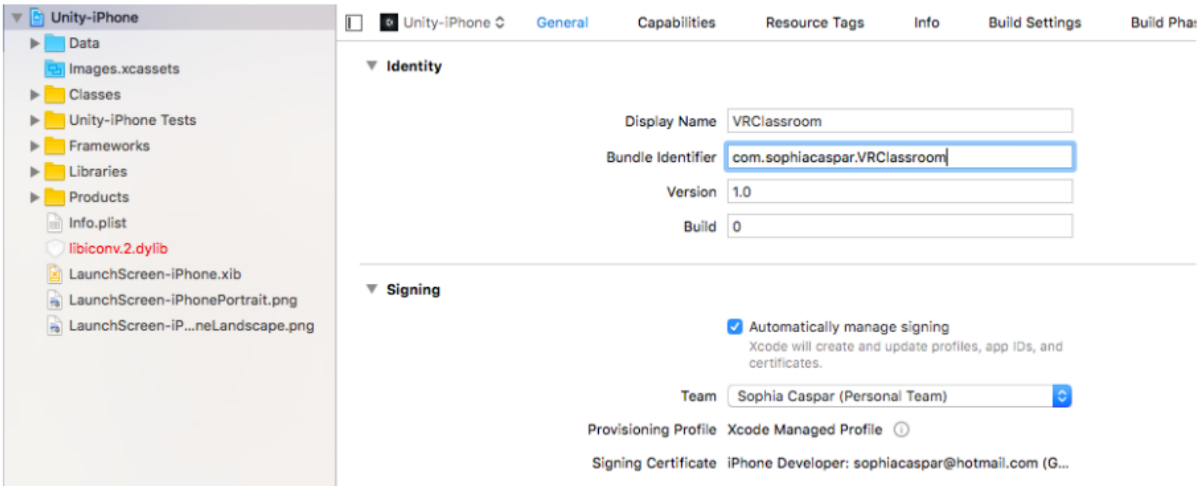

Figure 28: Screen to show where to add the Bundle identifier.

5. When all that is done, connect your iPhone to the computer. It needs to be unlocked when plugged-in. When connected, there are some files that need to process, so you'll have to wait until that's done. There is a blue bar increasing in the top of the window showing when the processing is finished.

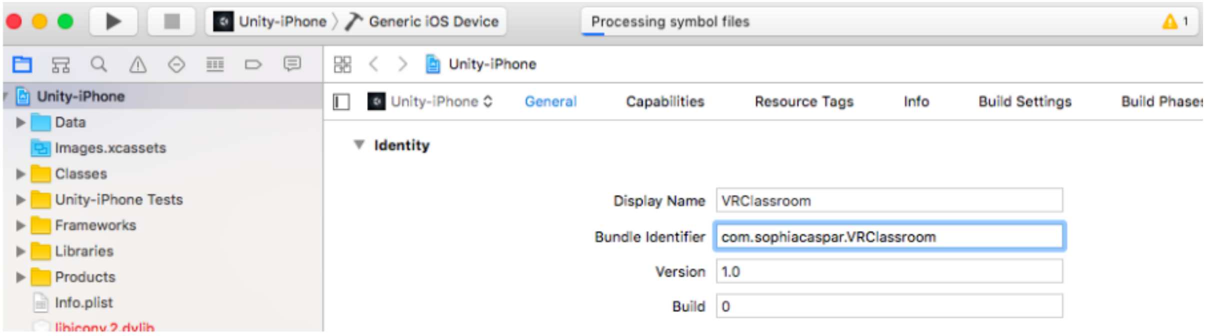

Figure 29: Xcode is processing files after connecting your iPhone.

6. Press play to build the app on the iPhone. When the build is done you might want to press stop and then unplug the phone from the computer and start the app via the phone. It is difficult to move around in the VR-world while connected to a wire.

Also, sometimes you need to go to your iPhone settings and allow your team to install unknown apps on your device. This is done on settings > general > Profiles or Profiles & Device Management, then click on the profile for the developer (most likely your apple-id) and then select allow.

A.3 Google VR SDK for Unity

This Software Development Kit (SDK)[8] is a framework for us to use in the VR development environment. By importing the SDK framework to a Unity 3D project, the SDK provides additional features such as User head tracking, Distortion correction for a VR viewer's lenses and more, when developing mobile VR application for Android or iOS for Google Cardboard or other VR headsets.

Importing the SDK to Unity is very easy. First download it from https://developers.google.com/vr/unity/download, then in Unity go to Assets > Import Package > Custom Package. Select the GoogleVRForUnity.unitypackage file you downloaded and click Open[9]. Some more configurations may be needed in the build settings depending on the users settings.