**Driving School**

Unity Seminar - Project Document

Members: Eyal Politansky, Liraz Inbar

Advisor Workshop: Moshe Sulamy

# Introduction

Our project is a virtual reality driving school simulation game. The game will simulate driving a car and incorporate real-world driving elements - roads, lanes, traffic signs, pedestrians, other vehicles, etc. The project aims to teach people learning to drive and help them prepare for driving lessons and practical tests - without leaving the house!

## Purpose Of The Project

Today, driving lessons are not accessible to all young people who want to get a license, both from the economic point of view and the psychological point of view - they do not prepare driving students for extreme situations that they may find themselves in the future.

Both during COVID-19 and the war, we've understood the significance of the ability to learn from home.

Our application is designed to simulate driving while defining the driving conditions in advance (lighting, weather, various distractions), in addition to practicing the theory content (such as signs and markings identification), and a driving instructor who can give comments. The VR game will give a real driving experience and allow tracking and measuring the player's responses, including where he looks and when.

The application provides an immediate response from anywhere and wherever the student wishes. It has added value to educational content - such as traffic rules, traffic signs identification (theory), vehicle operation, etc.

By harnessing the capabilities of VR technology and the Unity game engine, we deliver a compelling and educational driving simulation experience that prepares users for real-world driving scenarios while providing an engaging gameplay experience.

Main Features

In our application, the driving instructor can create a personalized route for the student based on his abilities.

The student will receive feedback on any mistakes made during their driving.

Before junctions, the student will be given an instruction from the teacher which turn to take.

The student can practice various kinds of routes and improve his abilities.

## Target Audience

Our target audience is people who want to learn to drive to obtain a license, students of driving refresher and preventive driving courses, and people who want to gain confidence on the road or for fun.

## Market Analysis

There are some driving games on the market today. Still, our game has 4 main key differences:

1. Teacher - In our game, a digital instructor sits next to you during all the lessons and improves the feeling you are in a real driving lesson.

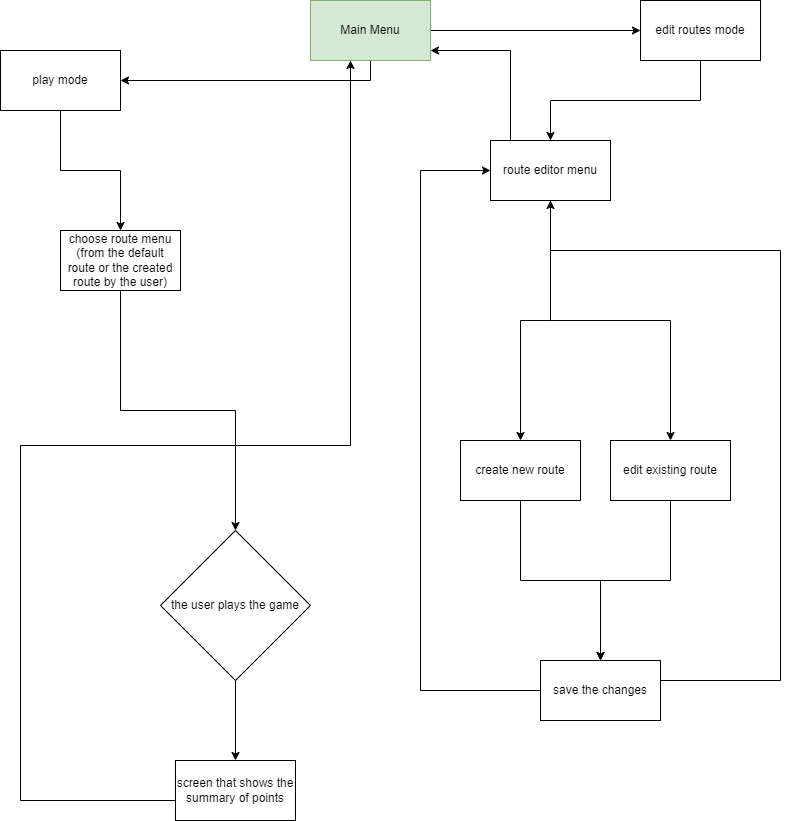
2. Feedback And Scoring - At the end of every lesson the player is given feedback on how his driving was in the lesson. He gets a score on every fault or good action he did with explanations that he can learn from and improve his driving for the next lessons.

3. Editor - The teacher can create the roads and the route the student will drive in, and thus can focus on specific driving difficulties the student encounters and wants to practice.

4. VR Experience - Our VR game gives a real driving experience by tracking and measuring the driving actions of the player in a precise way - moving the wheel, gearbox, winkers, and looking at the mirror with the VR simulates the real moves and actions of the driver as in a real car.

Architecture

The User Flow

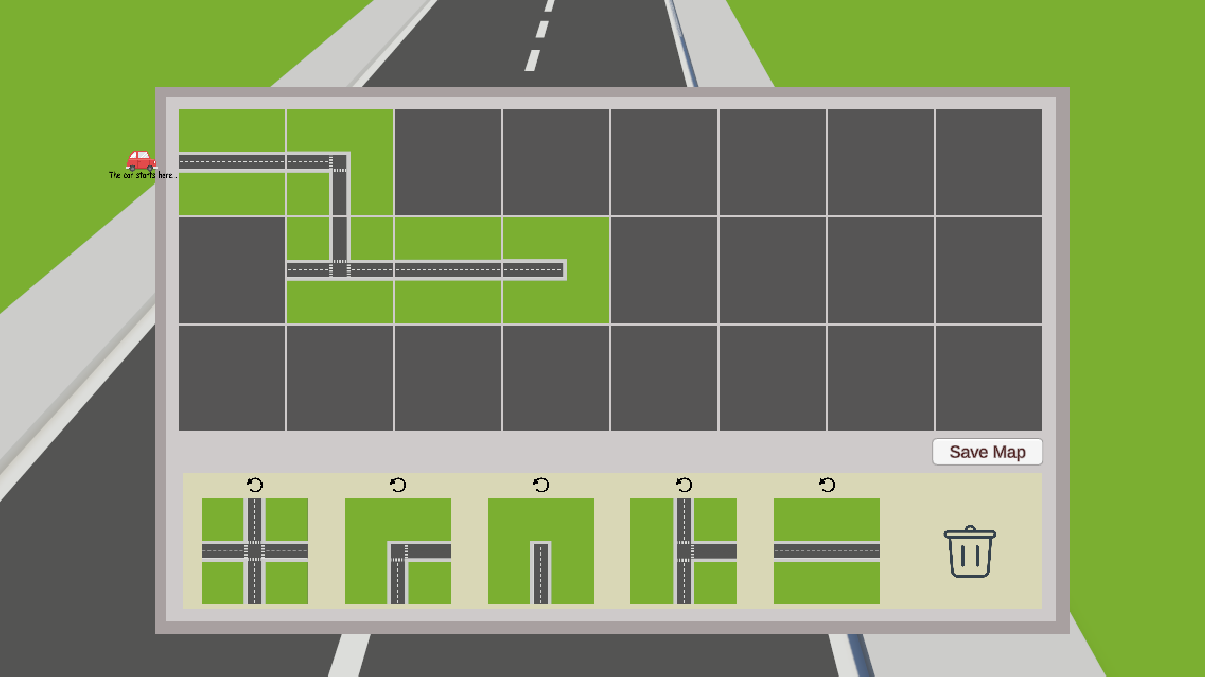


User guide

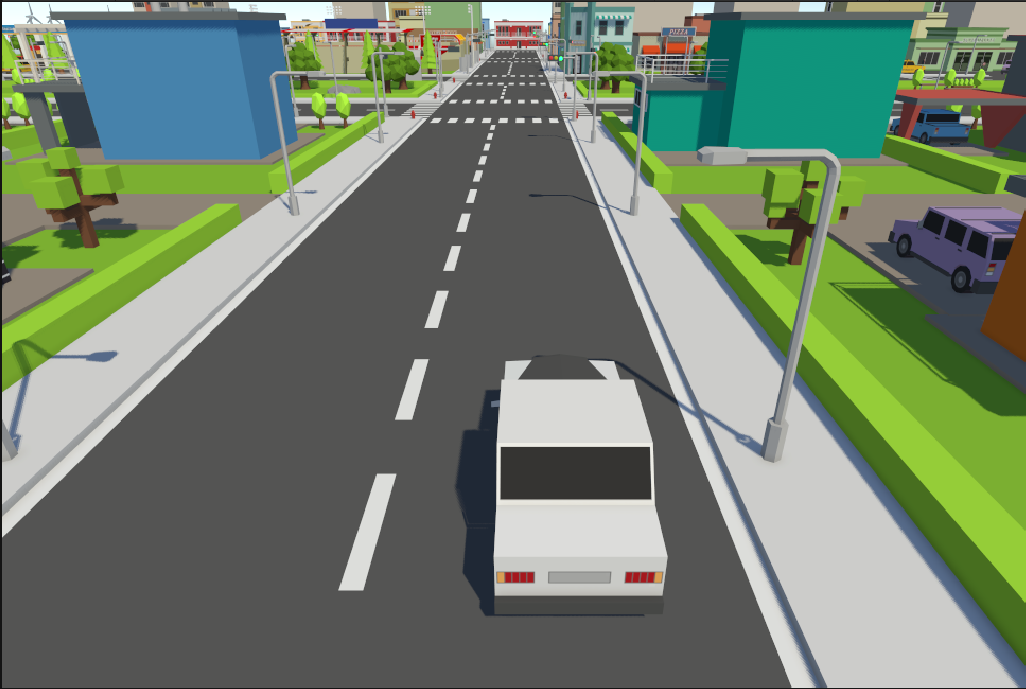
Main menu:



Route Editor Screen:



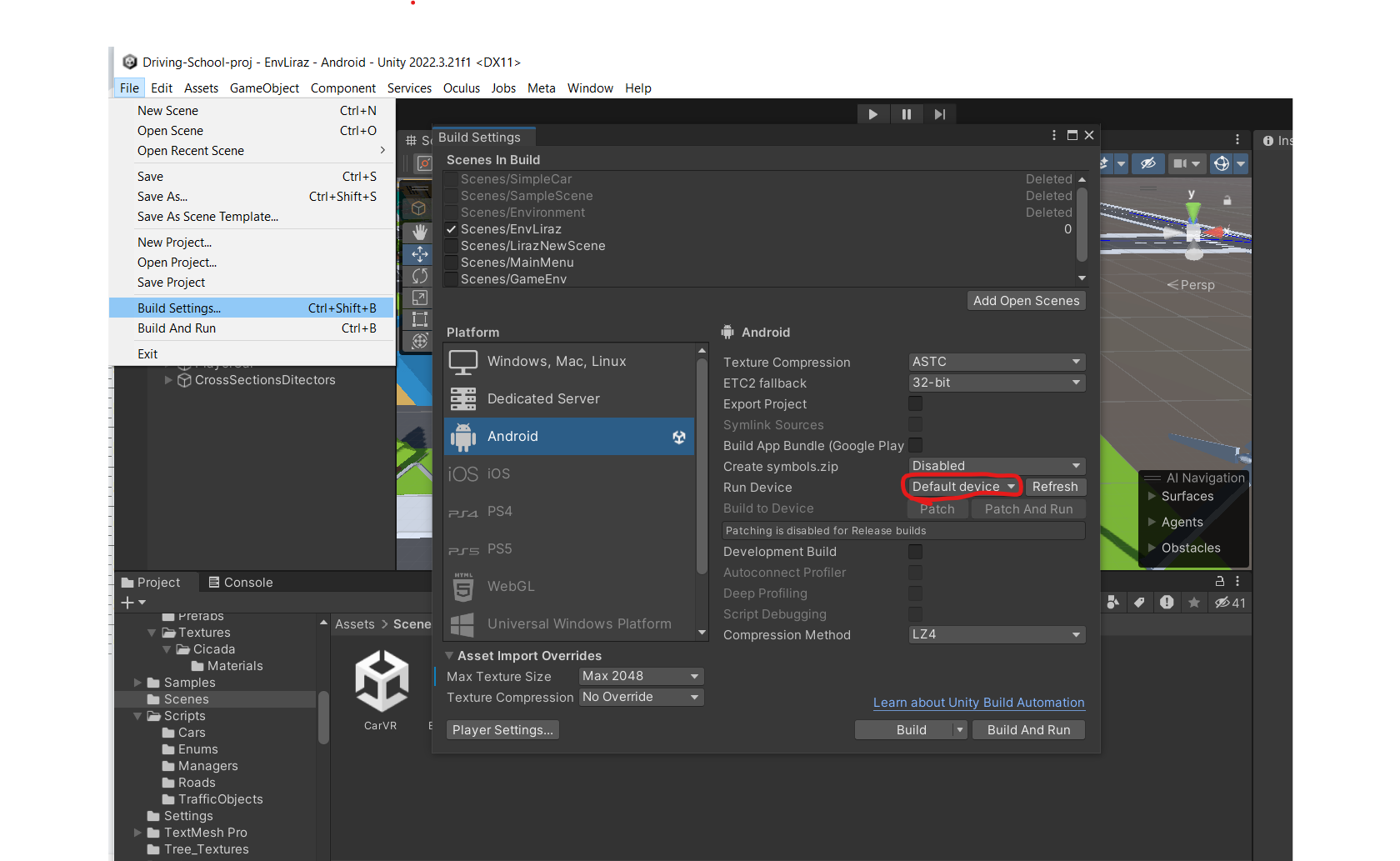
Default route view:



The player point of view via the VR:



How do we run the project:



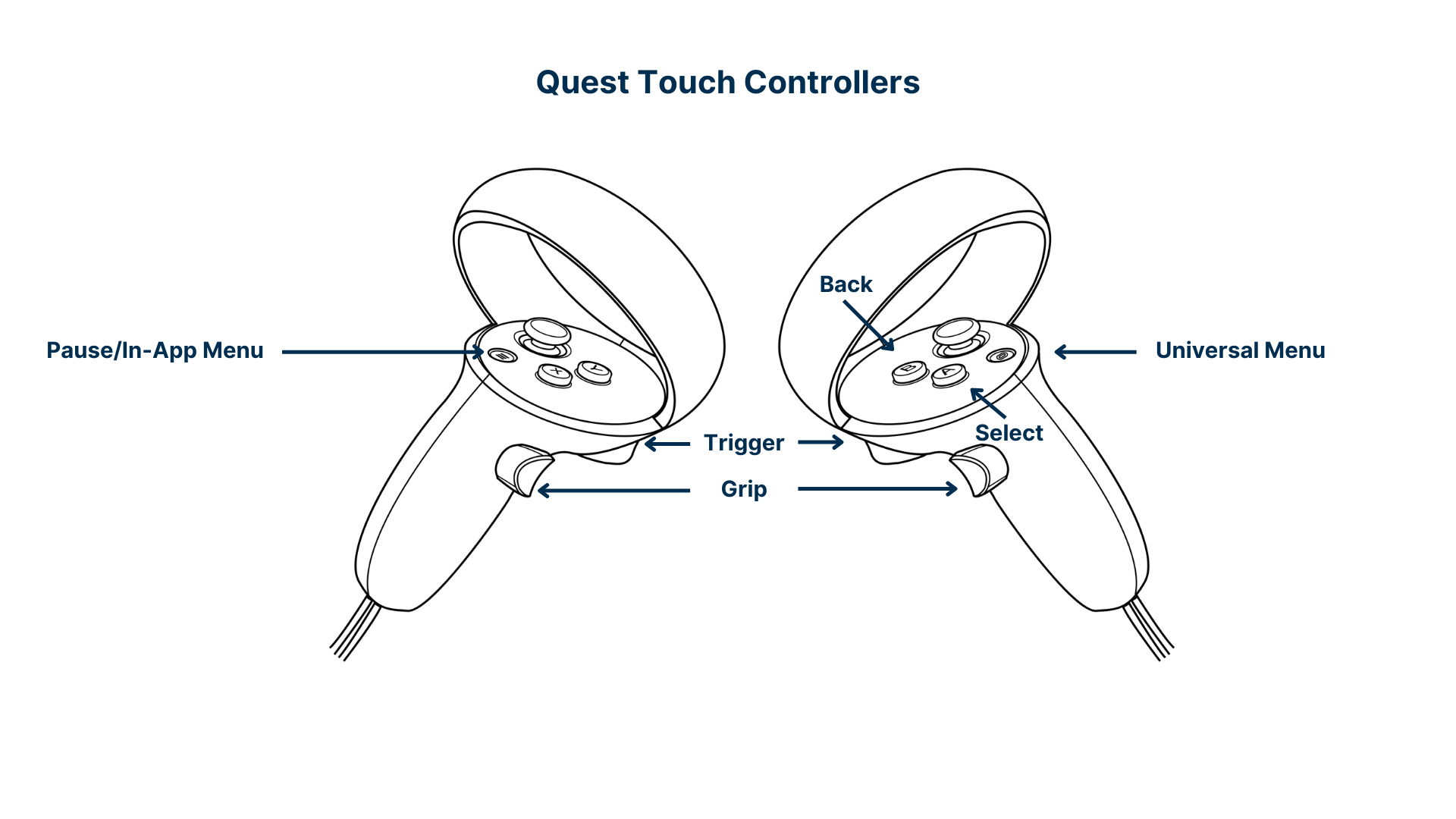
After connecting the VR glasses with USB, select in the red box the option of the VR glasses, and then build and run.

After the application is loaded on the VR glasses, you will be in the point of view like the picture above.

Both the grip buttons control the steering wheel and the gear stick,

The left trigger controls the break, and the right trigger controls the gas.

In order to start driving, you need to switch to drive mode using the gear stick.



For your convenience, there is an option to run the game not in VR mode - run the MainMenu scene. and then to control the car use the arrows buttons.

## Testing

During all the development of the project, we strictly tested and fixed our game and code. Mainly we test the game manually. We are testing via three main aspects - Unity editor, IDE and VR glasses.

Unity Editor: First we conduct our testing inside the Unity Editor. This is our main development framework and it has the most variety of tools for testing. We mostly use the play mode for testing - and change and fix the game while observing it closely to detect defects - game stuck or acting in an unexpected manner.

Additionally, in order to test the autonomous cars and traffic rules, we implemented a code on a car in the game to be played as the main car like in a real game run, but without the VR glasses. We can control the car via the keyboard and mimic the flow of the game faster and easier.

IDE: We are writing the game code on Rider IDE, that is specialized on Unity C# code. We get its suggestions on writing more readable, clean and right code, and test it with the embedded debugger tool.

VR Glasses: Our game is running on VR glasses, so this is very important to test it on it as well. In order to do so, we managed to find a way to transfer the current game build to the glasses and test it on them in a few quick steps. With the glasses we can check the real feel of the game, and pay extra attention to details that can’t be detected in the Unity Editor.

Also, we let our close friends and family play the game with the glasses and give us sincere feedback about it - if it feels realistic, if there are bugs, and if it’s fun!

## Issues and Future Work

The challenges we encountered during the project included, for instance, the operation of autonomous vehicles running in the background, which needed to be aware of each other and obey traffic signs and traffic lights to avoid collisions. To address this, we employed a technology in Unity called Raycast, which helps in detecting objects from a distance and behaving accordingly. This allowed us to simulate the behavior of autonomous vehicles and as a result simulate city routes as real as possible.

Another issue we faced was conveying the realistic sensation of driving a car through VR glasses. To make it feel authentic, we equipped the virtual vehicle with all the elements found in a real car, such as a gear stick, mirrors, and more. This enhancement significantly improved the realistic feel of the driving experience.

Currently, there can be situations where too many autonomous cars are generated, causing them to lose control. Additionally, there isn't always a smooth experience when using VR glasses from the driver's perspective, especially when moving the head.

Our vision for the future is, of course, to first address the current edge cases and provide users with a high-quality and comprehensive experience. We have several "nice-to-have" features we are interested in developing that could enhance the user experience, such as:

* Adding the option to upload created routes to the cloud
* Implementing a parking test
* Adding a radio to the car
* Including an avatar of a driving instructor to create a more realistic feel.
* Enhancing sound effects
* Adding an option to record the session