



UNIVERSITY OF THESSALY
SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

**DESIGN AND DEVELOPMENT OF EDUCATIONAL
SOFTWARE USING VIRTUAL REALITY**

Diploma Thesis

Kletsas Georgios

Supervisor: Korakis Athanasios

Volos 2021



Πανεπιστήμιο Θεσσαλίας

Πολυτεχνική Σχολή

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

**ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΕΚΠΑΙΔΕΥΤΙΚΟΥ
ΛΟΓΙΣΜΙΚΟΥ ΜΕ ΧΡΗΣΗ ΕΙΚΟΝΙΚΗΣ
ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ**

Διπλωματική Εργασία

Κλέτσας Γεώργιος

Επιβλέπων: Κοράκης Αθανάσιος

Βόλος 2021

Εγκρίνεται από την Επιτροπή Εξέτασης στις 11/06/2021:

Επιβλέπων

Κοράκης Αθανάσιος

Αναπληρωτής Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και
Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος

Αργυρίου Αντώνιος

Αναπληρωτής Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και
Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος

Μπαργιώτας Δημήτριος

Αναπληρωτής Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και
Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

This page is intentionally left blank

ΠΕΡΙΛΗΨΗ

Η εκπαίδευση έχει αλλάξει πολύ ανά τα χρόνια. Οι τεχνικές της έχουν εξελιχθεί από θεωρητικές σε όλο και πιο πρακτικές, δοκιμάζοντας καινούριες προσεγγίσεις, χρησιμοποιώντας νέα εργαλεία, παρεχόμενα από τη ραγδαία ανάπτυξη της τεχνολογίας. Ένα από αυτά τα εργαλεία, που χρησιμοποιείται ευρέως σε κάθε φάσμα της ζωής του ανθρώπου, είναι η Εικονική Πραγματικότητα (ΕΠ). Αρκετή έρευνα έχει πραγματοποιηθεί σχετικά με την επίδραση της στις μεθόδους εκμάθησης και έχει αποδειχθεί τόσο επιτυχημένη, που εμφανίζεται πλέον μία προφανής κλίση προς ψηφιακούς τρόπους διαπαιδαγώγησης. Παράλληλα με την ΕΠ, μια νέα τάση εμφανίστηκε, δείχνοντας το δρόμο για την παιχνιδοποίηση των εκπαιδευτικών μέσων. Ο όρος «Σοβαρά Παιχνίδια» έχει εμφανιστεί εδώ και καιρό, ξεκαθαρίζοντας ότι η γνώση και το παιχνίδι πρέπει να συνεργαστούν.

Η παρούσα διπλωματική είναι μια προσπάθεια να διευρυνθεί το ψηφιακό εκπαιδευτικό υλικό και να δείξει ένα δρόμο για τη σύμπτυξη της Εικονικής Πραγματικότητας και των Σοβαρών Παιχνιδιών, δημιουργώντας εφαρμόσιμα μαθήματα. Το ανεπτυγμένο λογισμικό έχει ως δημογραφικό στόχο παιδιά ηλικίας δέκα με δεκαπέντε ετών, με ενδιαφέρον στην Επιστήμη των Υπολογιστών, αλλά θα μπορούσε να χρησιμοποιηθεί και από τον οποιοδήποτε που μόλις εισάγεται σ' αυτό το χώρο. Είναι χωρισμένο σε τρία μέρη, με το καθένα στοχευμένο να διδάσκει ένα συγκεκριμένο θέμα. Μέσω της ΕΠ, κάθε παίκτης μεταφέρεται σε έναν διαφορετικό κόσμο και εμβυθίζεται στο έπακρο χάρη στην παρεχόμενη τεχνολογία. Μια τεχνολογία που έχει έρθει να τα κατακτήσει όλα.

This page is intentionally left blank

ABSTRACT

Education has changed a lot over the years. Its techniques have evolved from theoretical to more practical, trying new approaches with tools provided by the rapid technological advance. One of those tools, that is widely used in every aspect of a person's life, is Virtual Reality (VR). A lot of research has been conducted on how it can affect the methods a person can learn and it turned out so successful, that there is an obvious inclination to more digital approaches in education. Alongside VR, a new trend has emerged, showing the way for a "gamification" of educational means. The term "Serious Games" has appeared for quite a while now, pointing out that knowledge and games should go side by side.

This thesis is an attempt to expand the digital educational material and show a way to merge Virtual Reality and Serious Games into applicable lessons. The developed software has a target demographic of kids, aged between ten and fifteen years old, interested in Computer Science, but could also be applied to every person who is just now being introduced to this setting. It is split into three parts, each one specified in teaching a subject. Through VR, every player is transferred to a different world and is immersed to the fullest, thanks to the technology provided. A technology that has come to take everything by storm.

This page is intentionally left blank

Acknowledgements

I would like to thank my family and friends who believed and supported me in every step of the way, always motivating me for greatness. I am also grateful to my supervisor Professor Korakis Athanasios for giving me the tools and the support needed to implement this thesis, along with his guidance to surpass every obstacle.

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism.

The Declarant

Georgios Kletsas

14/06/2021

Table of Contents

1 Introduction	1
2 Background	3
2.1 Virtual and other realities	3
2.1.1 Virtual Reality	3
2.1.2 Augmented Reality	4
2.1.3 Mixed Reality	4
2.2 Serious Games	5
2.3 Game Engines	5
2.3.1 Unity Engine	6
2.3.2 Unreal Engine	7
3 Design and Development	8
3.1 Hardware	8
3.1.1 Computer Specifications	8
3.1.2 VR Headset	8
3.1.3 VR Controllers	12
3.2 Software	13
3.2.1 SteamVR	13
3.2.2 Unity Editor	13
3.2.2.1 Toolbar	14
3.2.2.2 Scene View	15
3.2.2.3 Hierarchy Window	16
3.2.2.4 Inspector	16
3.2.2.5 Game View	17
3.2.2.6 Project Folder	17
3.2.2.7 Console	18
3.2.2.8 Lighting Settings	18
3.2.2.9 Animation View	19
3.2.2.10 Timeline	19
3.2.2.11 Unity Asset Store	20
3.2.2.12 Scripting	20
3.2.2.13 Build Options	21
3.3 Application	22

3.3.1 Scenes	22
3.3.1.1 Main Menu Scene	22
3.3.1.2 Inside a PC Scene	25
3.3.1.3 Algorithms	35
3.3.1.4 Networks	43
4 Conclusion	48
5 References	49

Figures

Figure 1 X Reality	3
Figure 2 Virtual Reality	3
Figure 3 Augmented Reality.....	4
Figure 4 Mixed Reality.....	4
Figure 5 Serious Game - America's Army	5
Figure 6 Game Engines	6
Figure 7 Unity Engine Logo	7
Figure 8 Unreal Engine Logo	7
Figure 9 Google Daydream View.....	8
Figure 10 Oculus Logo	9
Figure 11 Google Cardboard	9
Figure 12 HTC Vive Headset	10
Figure 13 HTC Vive Controllers	10
Figure 14 HTC Vive Base Stations	11
Figure 15 HTC Vive Controls	12
Figure 16 Steam VR HOME	13
Figure 17 Unity Editor.....	14
Figure 18 Unity Toolbar	14
Figure 19 Unity Toolbar Transform Tools	14
Figure 20 Unity Toolbar Game View Tools.....	15
Figure 21 Unity Toolbar Miscellaneous	15
Figure 22 Unity Scene View	15
Figure 23 Unity Hierarchy Window	16
Figure 24 Unity Inspector.....	16
Figure 25 Unity Game View	17
Figure 26 Unity Project Folder.....	17
Figure 27 Unity Console Window.....	18
Figure 28 Unity Lighting Settings	18
Figure 29 Unity Animation View	19
Figure 30 Unity Timeline	19
Figure 31 Unity Asset Store	20
Figure 32 Unity Scripting (C#/JavaScript).....	20
Figure 33 Unity Build Options	21
Figure 34 Main Menu Scene	22
Figure 35 Main Menu Scene - Prompt	23
Figure 36 Inside a PC Scene.....	25
Figure 37 Inside a PC Scene - Environment.....	25
Figure 38 Inside a PC Scene - Companion.....	26
Figure 39 Inside a PC Scene - Components	27
Figure 40 Inside a PC Scene - Quiz Board.....	29
Figure 41 Inside a PC Scene - Test Environment.....	30
Figure 42 Algorithms Scene	35

Figure 43 Algorithms Scene - Kitchen Counter	36
Figure 44 Algorithms Scene - Milk Mechanic	42
Figure 45 Networks Scene.....	43
Figure 46 Networks Scene - Environment	44
Figure 47 Networks Scene - Test Tablet	45

Abbreviations

2D	Two Dimensional
3D	Three Dimensional
API	Application Programming Interface
AR	Augmented Reality
CPU	Central Processing Unit
HDD	Hard Disk Drive
HDMI	High-Definition Multimedia Interface
HMD	Head Mounted Display
LAN	Local Area Network
MR	Mixed Reality
NIC	Network Interface Card
PC	Personal Computer
PSU	Power Supply Unit
RAM	Random Access Memory
SDK	Software Development Kit
SSD	Solid State Drive
USB	Universal Serial Bus
VR	Virtual Reality
WAN	Wide Area Network
XR	Cross Reality

This page is intentionally left blank

1 Introduction

Year 2020 showed people not to take anything for granted. Covid-19 pandemic challenged humanity in many areas, with one of them being a way to work, teach and learn remotely. Virtual reality was there to provide a solution. From virtual conferences, to classrooms, to anything one can imagine, it created the opportunity not only to surpass the difficulties but to seize a new future.

The timing was also at its best. In previous years virtual reality applications were only accessible to specific individuals and teams who could afford it. The military is a perfect example that used VR to train pilots and its staff in general, simulating real life situations. Nowadays, with the development of smartphones and Head-Mounted Displays (HMDs), it is affordable by a much larger crowd to access VR. The trick now is how they can use it and what its applications would be.

The current VR development is seen in a plethora of disciplines like healthcare, architecture, tourism and others [1]. As stated, education is one main area that is explored but has yet a lot more ground to cover. Research has shown that learning through VR can benefit the students in six ways:

- **Visual learning:** They can now observe the subject that is being taught, rather than having to imagine how it would be, or see it in a picture.
- **Practical learning:** They learn by doing not by saying how it would be done.
- **Scalable experiences:** Anything can be created in a virtual environment, cutting the cost of physically possessing everything that needs to be shown.
- **Sense of place:** Having every student on his own inside every space, giving the required room to experiment and acclimate better.
- **Emotional Reaction:** Having the opportunity to design the class however they want, the teachers are given an opportunity to trigger emotional reactions from the students, making them remember and instill all the knowledge.

- **Creative Development:** The students are given a new kind of freedom that is absent from the dangers of real life. Within that freedom they can explore new possibilities fearlessly.

These six ways are guaranteeing a more unique educational experience for every student [2].

VR also encouraged people with disabilities to be more included, giving them the opportunity to be virtually anywhere. Various applications have been implemented, from touching museum artifacts for the visually impaired, to being able to listen to movies with translation to sign language, for the hearing impaired. However, this field has a lot of challenges and many things yet unexplored [3].

Development of VR applications has also become a lot simpler. Free-to-use game engines and easily integrated HMDs have given every developer the means to create his own application. From games, to educational software, to training simulations, there is a clear growth in the area that there wouldn't be if it weren't for the low cost of the whole package.

Serious Games is also something that needs to be explored and VR is a great way to do that. There have been computer games that are educational in the past but through this new technology a lot of groundbreaking results could surface by putting students into a whole new world.

The purpose of this thesis is to present an example of an educational process, different from the status quo and encourage even more developers into taking action, in order to have a more immersive, accessible and successful education.

2 Background

2.1 Virtual and other realities

X Reality (XR) or Cross Reality is used to define an extended reality, furthering the one that we already have. Through XR, other terms have surfaced like Virtual Reality, Augmented Reality and Mixed Reality.



Figure 1 X Reality

(source: <https://medium.com/@northof41/what-really-is-the-difference-between-ar-mr-vr-xr-35bed1da1a4e>)

2.1.1 Virtual Reality

Virtual reality is a simulated reality that uses a computer designed environment. That environment can be similar to the real world or not. To access VR, a Head-Mounted Display (HMD) is needed, along with an application that is tailored for the specific experience.



Figure 2 Virtual Reality

(source: <https://oannes.gr/virtual-reality/>)

2.1.2 Augmented Reality

Augmented Reality (AR) is an enhanced reality that uses the real world and adds elements to it. To access AR, there are a lot of ways, from HMDs, to eyeglasses, to even smartphones and through them you can observe the added elements from a specified application.



Figure 3 Augmented Reality

(source: <https://gr.depositphotos.com/stock-photos/augmented-reality.html>)

2.1.3 Mixed Reality

Mixed Reality (MR) is a mix of real and virtual environments. AR is a subcategory of MR, with the first taking place in the real world while the second can also be in a virtual. The ways to access them are quite similar.

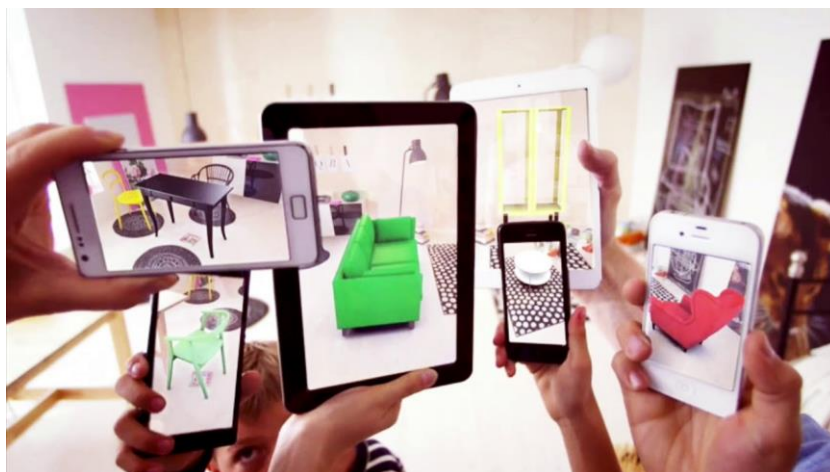


Figure 4 Mixed Reality

(source: https://www.youtube.com/watch?v=v8wtHZDVdlc&ab_channel=MozillaHacks)

2.2 Serious Games

Serious game is a type of game that its purpose is not only to entertain but to educate too. They have been used in many disciplines like education, defense and city planning, proving that fun and knowledge can coexist successfully. A primary example is a game called America's Army, a game created by the U.S. Army to train its soldiers [4].



Figure 5 Serious Game - America's Army

2.3 Game Engines

A game engine holds the software needed to create games. Some of the components most game engines have are:

- A program that holds the game logic (programming)
- A rendering engine to project 2D or/and 3D graphics
- An audio engine that handles the sound department
- A physics engine that is responsible for implementing the game physics required for each game
- An artificial intelligence system
- An animation module to modify and create animations

[5].

The most famous game engines nowadays are Unity Engine and Unreal Engine among others, with these two being free-to-use and maintaining huge communities.



Figure 6 Game Engines

(source: https://www.youtube.com/watch?v=bGkaeYEB4gU&ab_channel=Etredal)

2.3.1 Unity Engine

Unity Engine, the engine used for the development of this thesis' application, is a game engine released in June 2005 [6] and has been greatly praised for a lot of things. Some of them are:

- Beginner friendly user interface
- Cross-platform capabilities (with Unity someone can ship to a lot of platforms like PC, Mobile, PlayStation with little to no change)
- Giving the opportunity to independent developers to learn and create games, a privilege that was given to few individuals in the past
- Huge community and availability on tutorials

Unity uses C# and JavaScript as main programming languages, however there is also visual scripting with a framework called Bolt. It can handle 2D and 3D graphics and VR integration has become as easy as possible.

Made with



Figure 7 Unity Engine Logo

2.3.2 Unreal Engine

Unreal Engine was showcased for the first time in 1998 [7] as a first-person engine that was used to create a game called Unreal. Since then it has evolved a lot, with huge titles created with it. Its most significant advantages are:

- Detailed graphics
- Almost anything in game logic can be created with Blueprints (Visual Scripting)
- Uses C++ making it very performant

Unreal engine mainly uses C++ and Blueprints (Visual scripting) for programming game logic. It can ship to most platforms like PC, Mobile, PlayStation, really easily.



Figure 8 Unreal Engine Logo

3 Design and Development

Various tools were used for the implementation of this thesis. Hardware and software architecture will be discussed, along with the design and development for each lesson.

3.1 Hardware

For the hardware part of the implementation some specific components were required: a computer, a headset, its controllers and two base stations.

3.1.1 Computer Specifications

The computer used to develop the application needs to have certain specifications, because VR development is quite a ‘heavy’ operation. Minimum requirements are:

- NVIDIA® GeForce® GTX 1060 or AMD Radeon™ RX 480
- Intel® Core™ i5-4590 or AMD FX™ 8350
- 4 GB RAM or more
- HDMI 1.4, DisplayPort 1.2 or newer
- 1x USB 2.0 or better port
- Windows® 7 SP1, Windows® 8.1 or later, Windows® 10

3.1.2 VR Headset

A VR headset is a head-mounted device through which a person can enter virtual reality. A few categories of headsets are:

- **Standalone:** These headsets do not need any extra software or hardware to function, thus they can be used by themselves. Some examples are the Oculus Mobile SDK and the Google Daydream.



Figure 9 Google Daydream View

(source: <https://venturebeat.com/2019/10/15/google-discontinues-daydream-vr/>)

- **Tethered:** This kind of headsets require another device to project virtual reality (PC, Console) and the most mainstream examples are Steam VR, PlayStation VR and Oculus PC SDK. These are used a lot for development.



Figure 10 Oculus Logo

- **Mobile:** This kind of headsets requires a smartphone to work, which provides the display. These are fairly cheaper and an example is Google Cardboard.



Figure 11 Google Cardboard

During this application's development, the HTC Vive was used, which is a Tethered headset. Its components are:

- **Vive Headset:** The actual headset with two OLED panels for the person to see



Figure 12 HTC Vive Headset

- **Vive Controllers:** Two controllers, each for one hand, to track movements and execute actions



Figure 13 HTC Vive Controllers

- **Vive Base Stations:** Two black boxes that track the user's position across a room (Lighthouse Tracking System)



Figure 14 HTC Vive Base Stations

Its specifications are:

- Headset
 - Screen: Dual AMOLED 3.6'' diagonal
 - Resolution: 1080 x 1200 pixels per eye (2160 x 1200 pixels combined)
 - Refresh Rate: 90 Hz
 - Field of View: 110 degrees
 - Safety Features: Chaperone play area boundaries and front-facing camera
 - Sensors: SteamVR Tracking, G-sensor, gyroscope, proximity
 - Connections: HDMI, USB 2.0, stereo 3.5 mm headphone jack, Power, Bluetooth
 - Input: Integrated microphone
 - Eye Relief: Interpupillary distance and lens distance adjustment
- Controllers
 - Sensors: SteamVR Tracking
 - Input: Multifunction trackpad, Grip buttons, dual-stage trigger, System button, Menu button
 - Use per Charge: Approx. 6 hours
 - Connections: Micro-USB charging port

- Tracked Area
 - Standing / Seated: No min. space requirements
 - Room-scale: Up to 15m² (an area of approximately 3.5m x 3.5m)

[9]

3.1.3 VR Controllers

The controllers are used to interact with the VR world and are tracked by the base stations. They have various buttons and other components, some of them configurable for the user.

Components:

1. **Menu button:** Shows the VR menu
2. **Trackpad:** Usually handles movement (Continuous or Teleportation on press)
3. **System button:** Shows the steam menu for integration with SteamVR
4. **Status Light:** Shows the status of the controller (idle, needs charging, connecting etc.)
5. **Micro-USB port:** Connection with computer or charger
6. **Tracking sensor:** To track motion of hand
7. **Trigger:** A button for various actions configured by the user
8. **Grip button:** Also configurable for the user

[10]

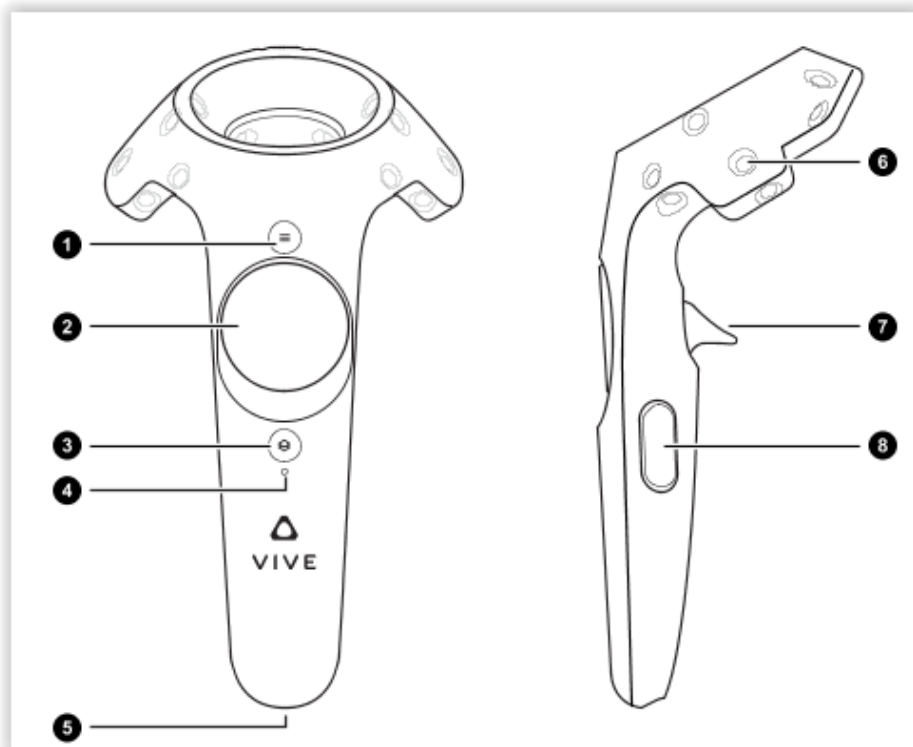


Figure 15 HTC Vive Controls

3.2 Software

The primary software applications used for this application are Unity Editor and Steam VR. Others were used for sound editing, Sprite editing and 3D modeling but were not that important in the implementation.

3.2.1 SteamVR

SteamVR is software created by Valve in order to use room-scale VR instead of being stationary. It uses OpenVR (also created by Valve), which is a software development kit (SDK) and an application programming interface (API), so that someone is able to access VR. Through it the use of the HTC Vive becomes accessible.



Figure 16 Steam VR HOME

3.2.2 Unity Editor

In order to use Unity Engine one must know how to navigate inside Unity Editor. Its most important windows are:

1. **Toolbar**
2. **Scene View**
3. **Hierarchy**
4. **Inspector**
5. **Game View**
6. **Project Folder**
7. **Console**

Other important features of this Editor are the Lighting Settings, the Animation System, the Timeline system and the Unity Asset Store.

Some important terms to acknowledge before navigating the Unity Editor are:

- **GameObject:** Every object inside Unity is called a gameObject
- **Asset:** Sounds, art, scripts and all these files that can be used inside Unity

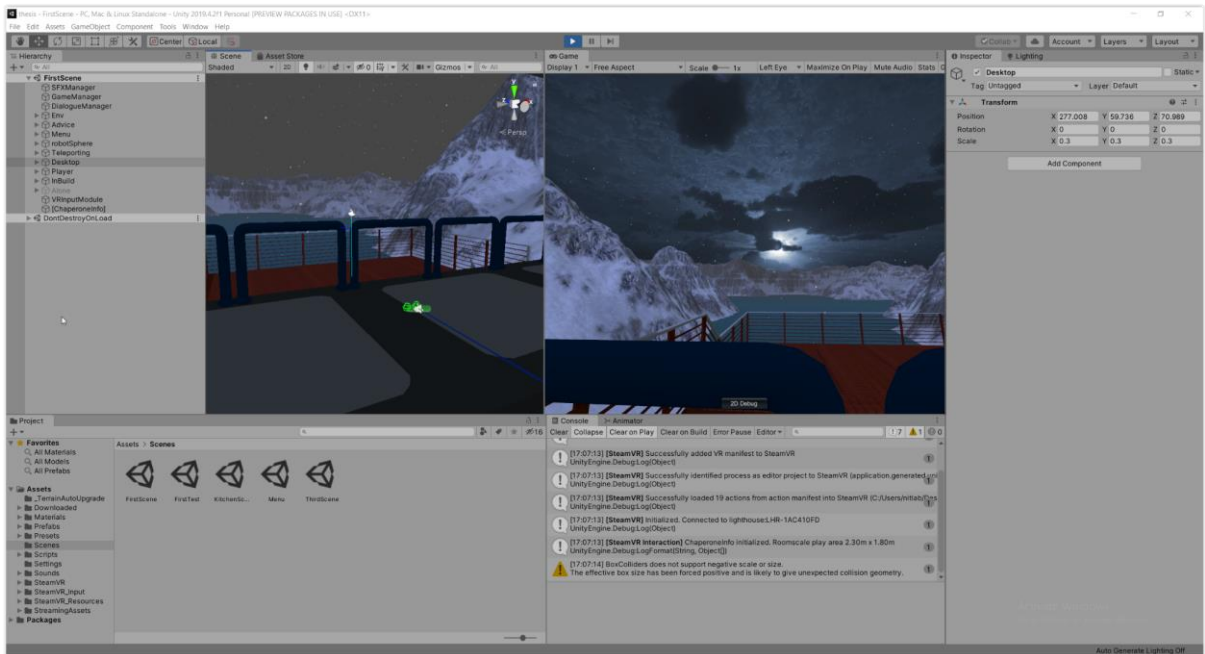


Figure 17 Unity Editor

3.2.2.1 Toolbar

The toolbar consists of groups of buttons that affect some of the windows of the editor.



Figure 18 Unity Toolbar

- The Transform Tools that help edit (Move, Rotate, etc.) a gameObject in the Scene View



Figure 19 Unity Toolbar Transform Tools

- The Play, Pause, Step buttons for the Game View



Figure 20 Unity Toolbar Game View Tools

- The Unity Collaboration drop-down menu
- The Cloud Button to access Unity Services
- The Account drop-down menu to access Unity Account
- The Layers drop-down menu to control the visibility of objects in the Scene View
- The Layout drop-down menu to change the arrangement of Unity Editor's layout



Figure 21 Unity Toolbar Miscellaneous

3.2.2.2 Scene View

Through the Scene View the developer is able to manipulate the placed gameObjects, add new and apply changes at a scene in general.

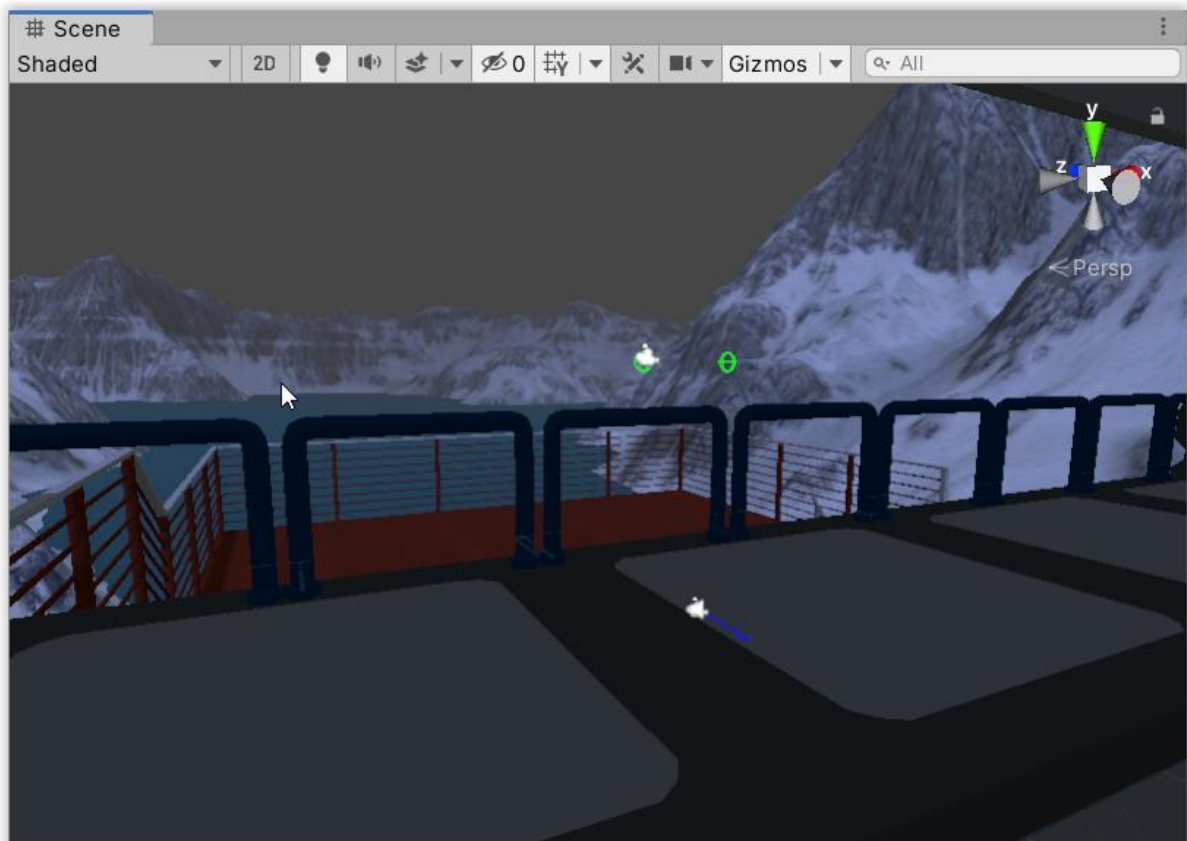


Figure 22 Unity Scene View

3.2.2.3 Hierarchy Window

The hierarchy shows all the gameObjects the developer has in a scene. Inside this window someone can find the relationship between them (parent-child) and can also study their modifications and new creations at runtime.

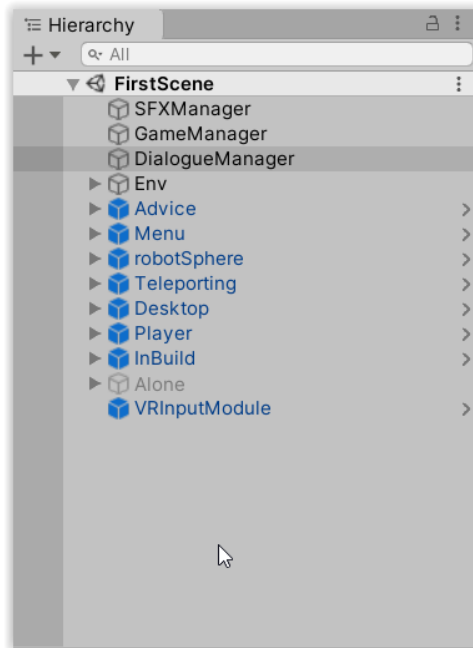


Figure 23 Unity Hierarchy Window

3.2.2.4 Inspector

Through the inspector, the developers can modify the components of any gameObject or asset, inside or outside a scene. They can add, edit or remove components that add functionality.

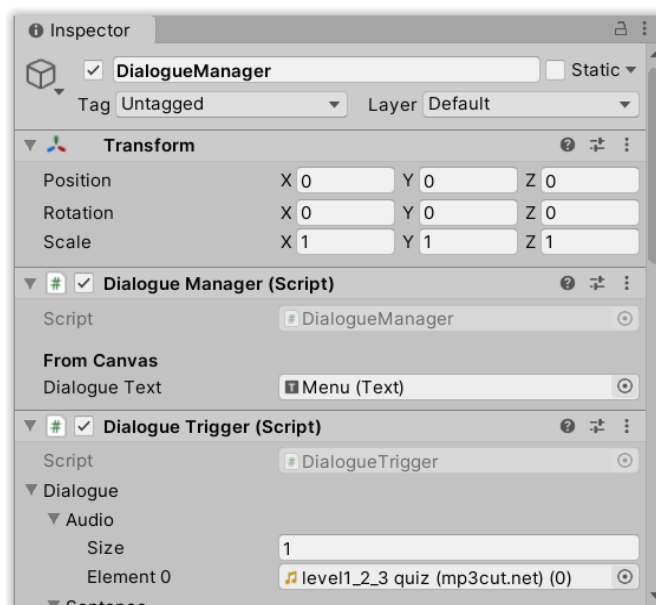


Figure 24 Unity Inspector

3.2.2.5 Game View

The Game View is the player's view after hitting the Play button on the toolbar. This shows the view of the in-game camera(s). At least one camera is needed.

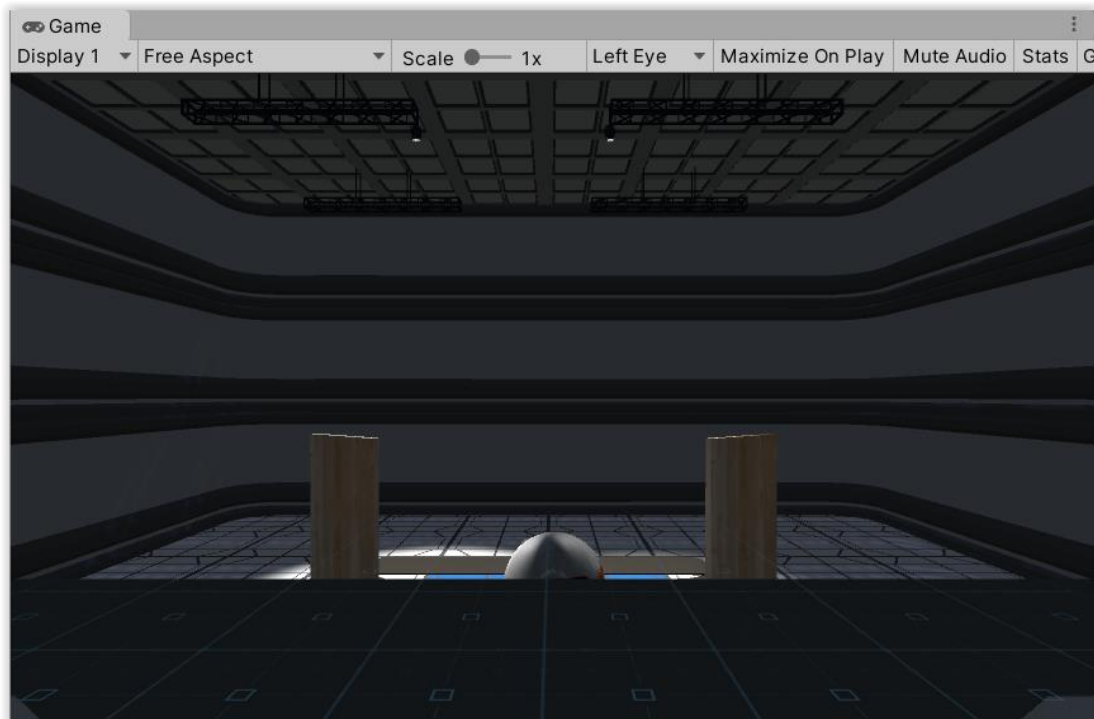


Figure 25 Unity Game View

3.2.2.6 Project Folder

As the name suggests, the project folder contains all the assets of the project and can be modified however the developer needs.

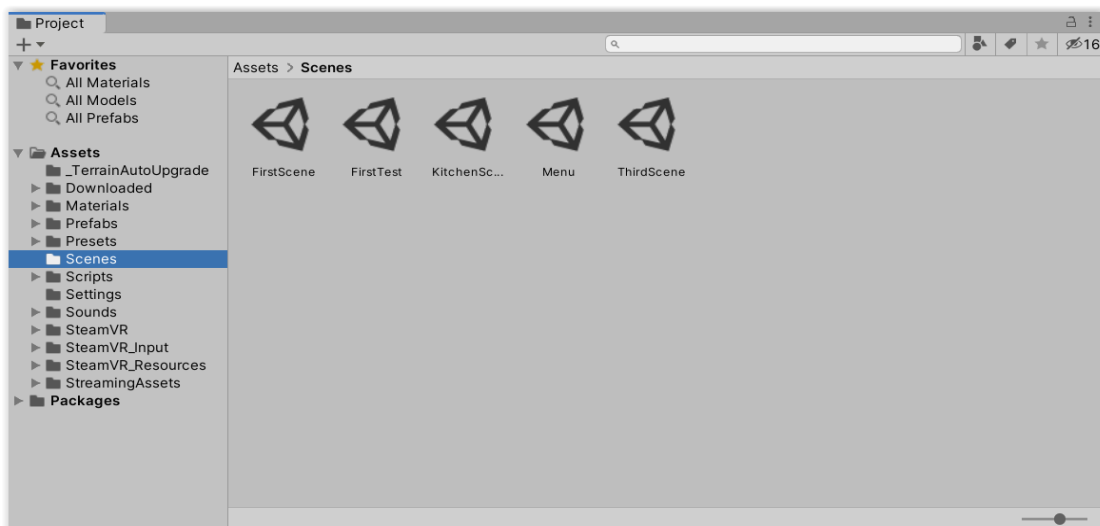


Figure 26 Unity Project Folder

3.2.2.7 Console

Through Console the developers can see all the debugging information provided either by Unity or by them. From errors, to warnings, to specific messages, everything that can help the debugging process can be found there.

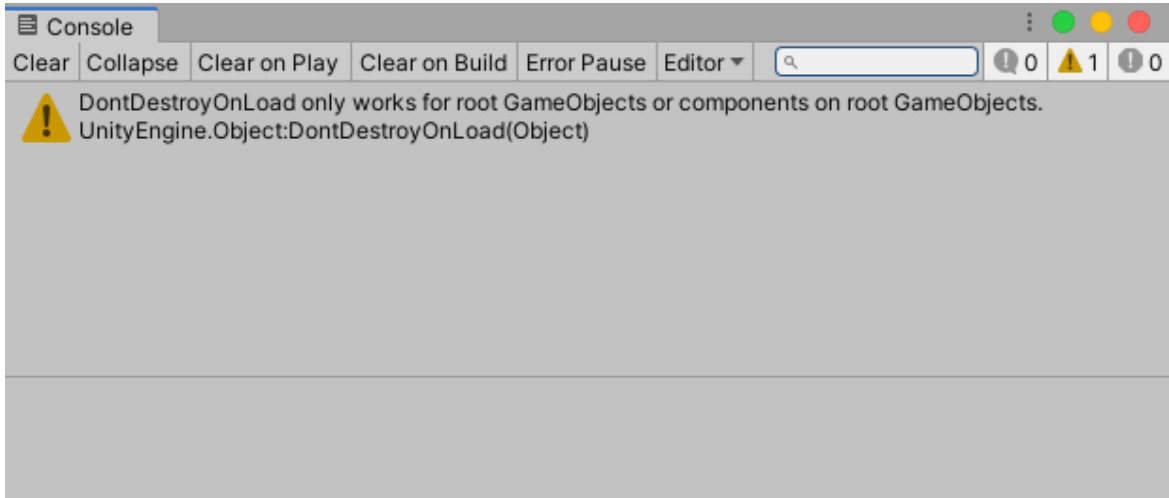


Figure 27 Unity Console Window

3.2.2.8 Lighting Settings

As the name suggests, through lighting settings the developer can set the lighting modifications he needs for a game along with other things like fog and skybox.

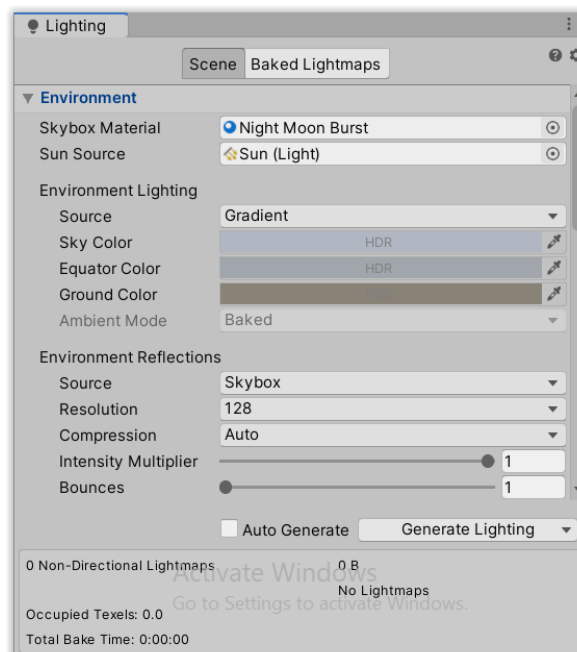


Figure 28 Unity Lighting Settings

3.2.2.9 Animation View

Within the Animation View the developer can create and modify animation assets, in order to attach them to gameObjects or to the Timeline.

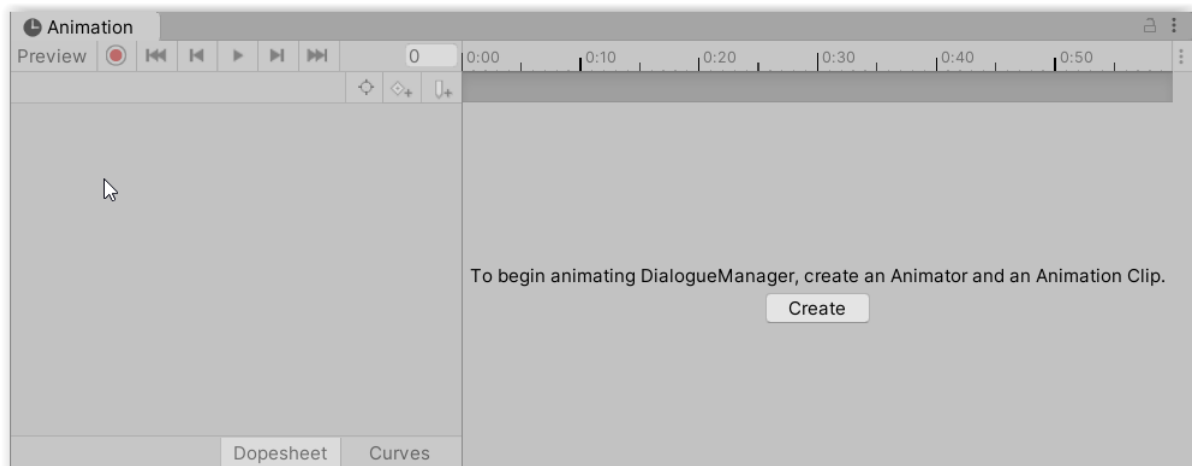


Figure 29 Unity Animation View

3.2.2.10 Timeline

The Timeline Feature enables the ability to add music, animations and other assets visually to create premade scenes to add to your application.

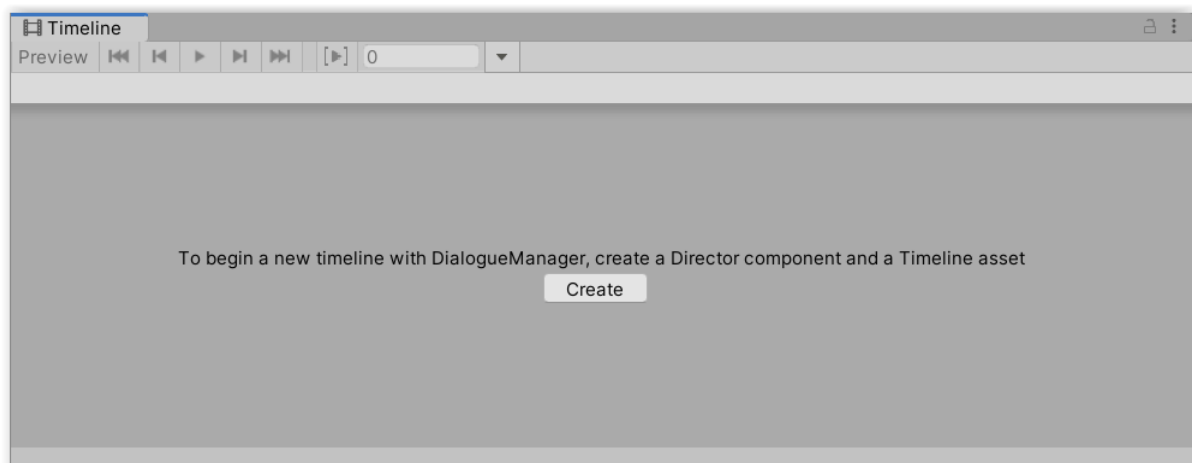


Figure 30 Unity Timeline

3.2.2.11 Unity Asset Store

Unity Asset Store is a huge asset library, filled with ready to integrate elements that every developer can get either free or with some cost. Also, every developer can apply to upload an asset to the asset store.

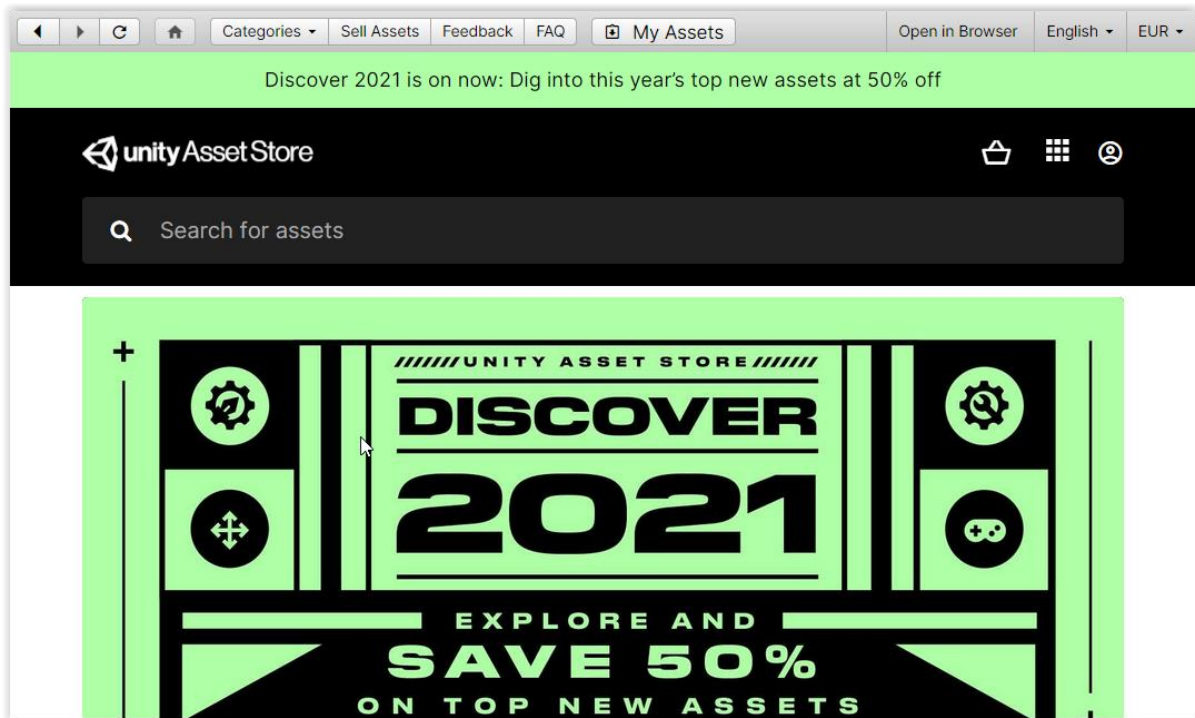


Figure 31 Unity Asset Store

3.2.2.12 Scripting

In order to add logic to an application, one must add a script to a gameObject. That script can be written either in C# or JavaScript; however Visual Scripting (Bolt) is a trend nowadays. To use Unity Engine's library the developer should add it as a namespace and to be able to have playability inside the game, every script should derive from MonoBehaviour (base class for all Unity scripts). The most common editor for scripting in Unity is Visual Studio.



Figure 32 Unity Scripting (C#/JavaScript)

3.2.2.13 Build Options

Through Unity, developers can now export their applications to many platforms, with little to no change. Some of the supported platforms to build for are:

- iOS
- Android
- Windows desktop
- Mac OS desktop
- Linux desktop
- WebGL
- Playstation 4

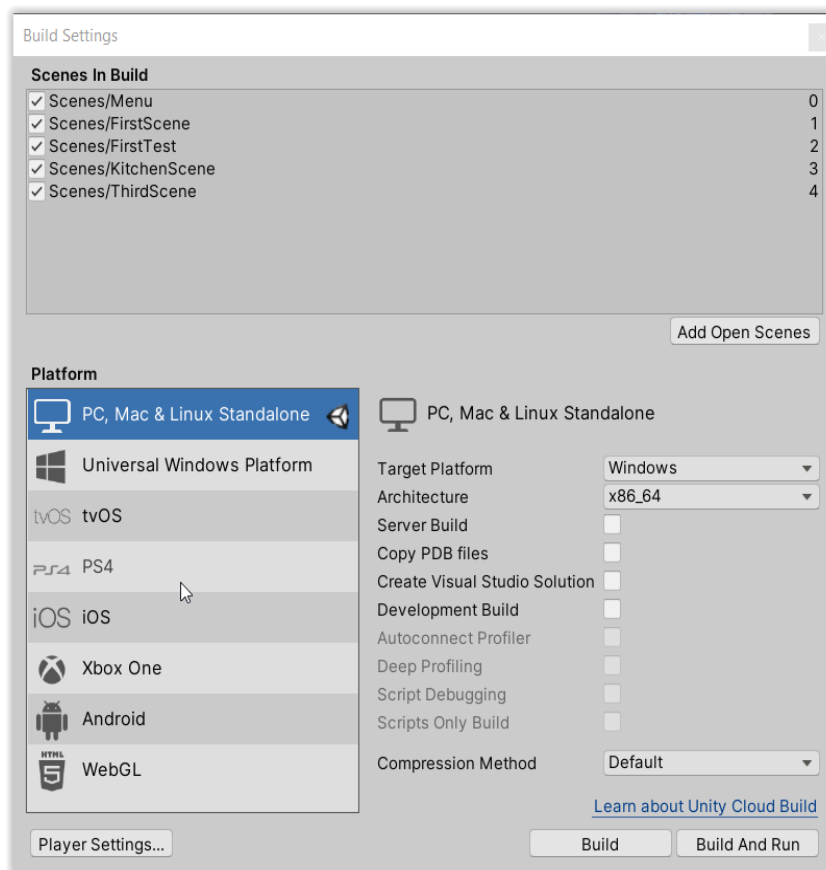


Figure 33 Unity Build Options

3.3 Application

The application implemented consists of three levels. In Unity, levels are also called scenes and that's how they are going to be presented. Each one has a specified subject. They are in no particular order so the players can play whichever they want first.

3.3.1 Scenes

The application's scenes are named after their teaching material:

1. **Inside a PC:** With this lesson the players get to see the components inside a computer and get acquainted with their uses.
2. **Algorithms:** Through this lesson the players are taught algorithmic terms
3. **Networks:** In this lesson the players learn about networking components and their functionalities.

There is also a main menu scene, to help the navigation between all the levels. We'll start with that.

3.3.1.1 Main Menu Scene

The main purpose of this scene is to go from scene to scene before starting and after ending a level.

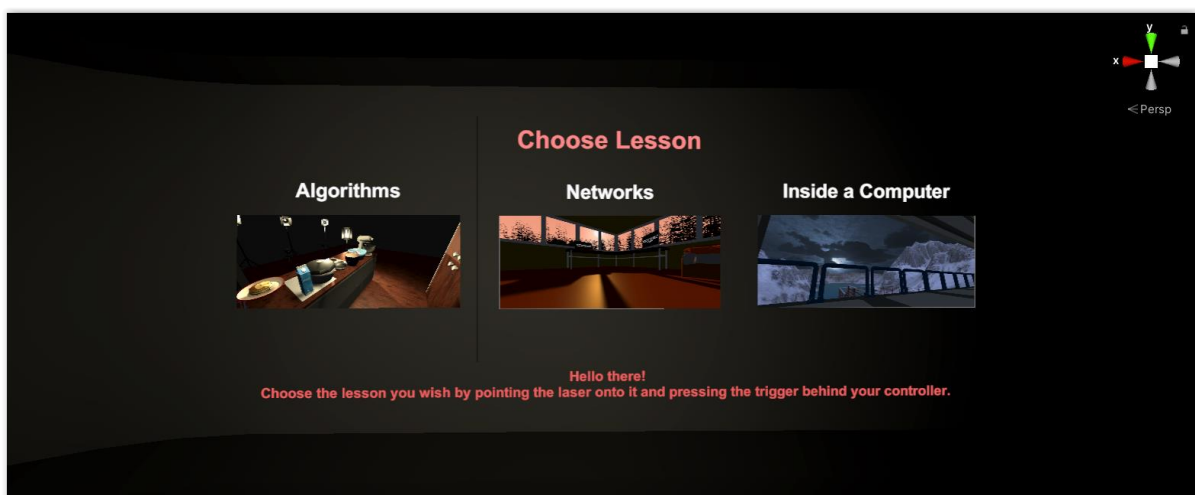


Figure 34 Main Menu Scene

- **Pre-Design**

What was intended is to put the players inside a dark room so that the most significant thing they can see is the levels' pictures to point and click. Also, it was needed that wherever they started, they would know where the buttons would be.

- **Design**

Via Pre-Design it was clear that there should be indication where the buttons would be, in case the players started with their backs on them. So there is a text on the back of the room notifying the player to turn around to see them.

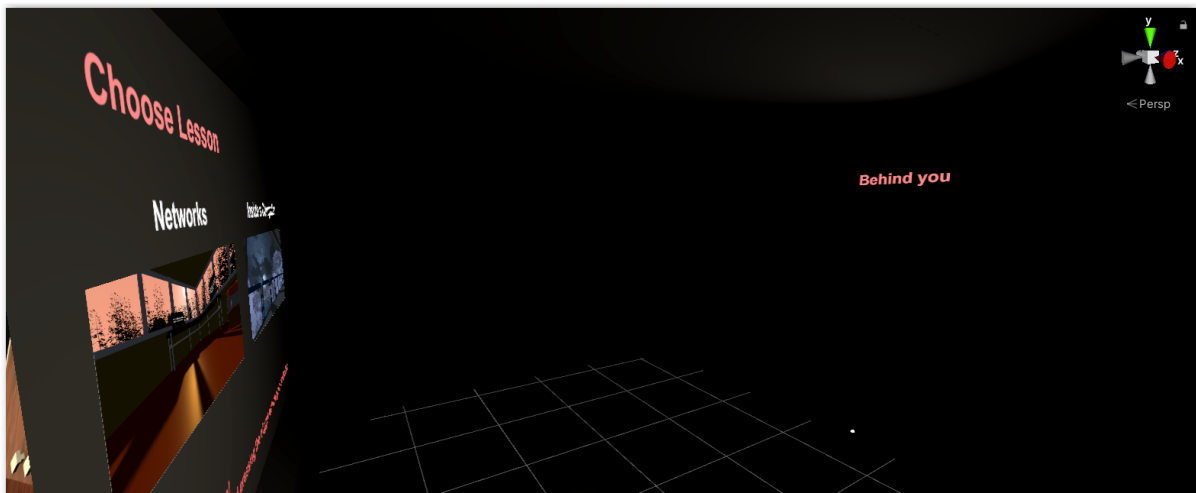


Figure 35 Main Menu Scene - Prompt

One more thing that was needed was to signal to the player what button to press and when. The use of VR is not so common, so there should be clear instructions for the buttons at all times. That's why there is an indication to press the trigger button on the controller. To tackle the problem of when this should be pressed, there is a line starting from the controller and when it hovers over a level button, this changes its opacity so that it's darker, to show that it can be pressed.

- **2D/3D Art**

The room has an industrial look however the dark lighting doesn't really show it. It's just an enclosed space for the player. There are texts on two sides of it, and three level buttons on the one.

Downloaded 2D/3D art assets for this scene:

- Snaps Prototype | Sci-Fi / Industrial¹

¹ <https://assetstore.unity.com/packages/3d/environments/sci-fi/snaps-prototype-sci-fi-industrial-136759>

- **Sound**

An atmospheric theme was chosen as background music for the main menu, to draw the player into a different world. It fits the darkness of the room.

Downloaded sound assets for this scene:

- Background Theme 1²

Companion lines (voice helping the player):

- ➔ Hello there!

Choose the lesson you wish by pointing the laser onto it and pressing the trigger behind your controller.

- **Programming**

Three scripts were needed for the specific level:

1. **Pointer:** This script was created to cast a straight line from the player's right hand to wherever it pointed, making it easy to use the user interface, by pointing and pressing the trigger button.
2. **VR Input Module:** This one was used to change the default input module, to a VR one. It worked alongside the Pointer script and everything that was pointed and triggered by the last, was executed by the first.
3. **Menu:** This script is responsible for switching through the scenes. Basically, it changes the scene based on the button that is pressed.
4. **Player script (from SteamVR):** It handled room-scale movement and in general the whole package of VR movement and interaction.

² <https://freesound.org/people/Speedenza/sounds/251530/>

3.3.1.2 Inside a PC Scene

The goal of this scene is to present the components that a computer holds while also talking about their unique functionalities.

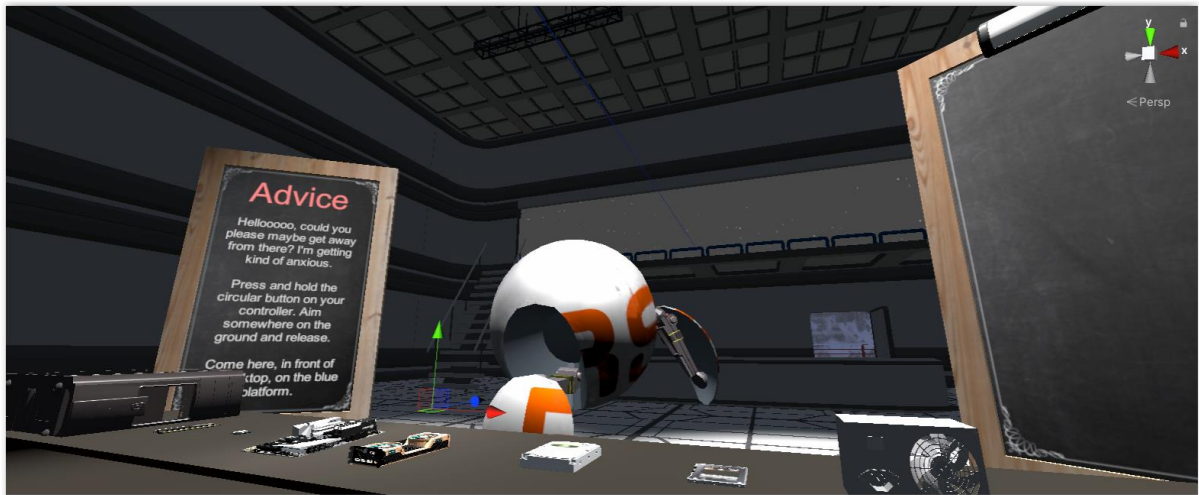


Figure 36 Inside a PC Scene

- **Pre-Design**

The thoughts while creating every VR scene were to make them as immersive as possible. Through research, some commonalities were obvious among the most acclaimed VR experiences, with one being that big spaces really give an essence of reality. So for this level, a snowy mountain along with a lake was chosen as an external environment and a factory-looking space for the internal.

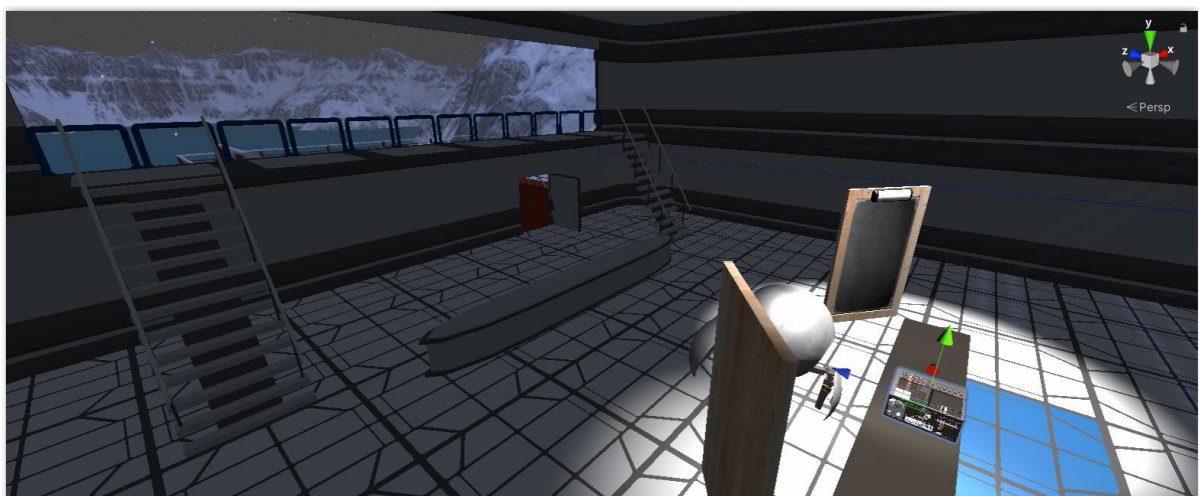


Figure 37 Inside a PC Scene - Environment

Also, listening to someone talking and handling instructions was something that most developers used to immerse players into a new world, so this was implemented too. A

voice actor was used as a companion for the player in every scene, who was only shown in this one, as a big spherical robot.

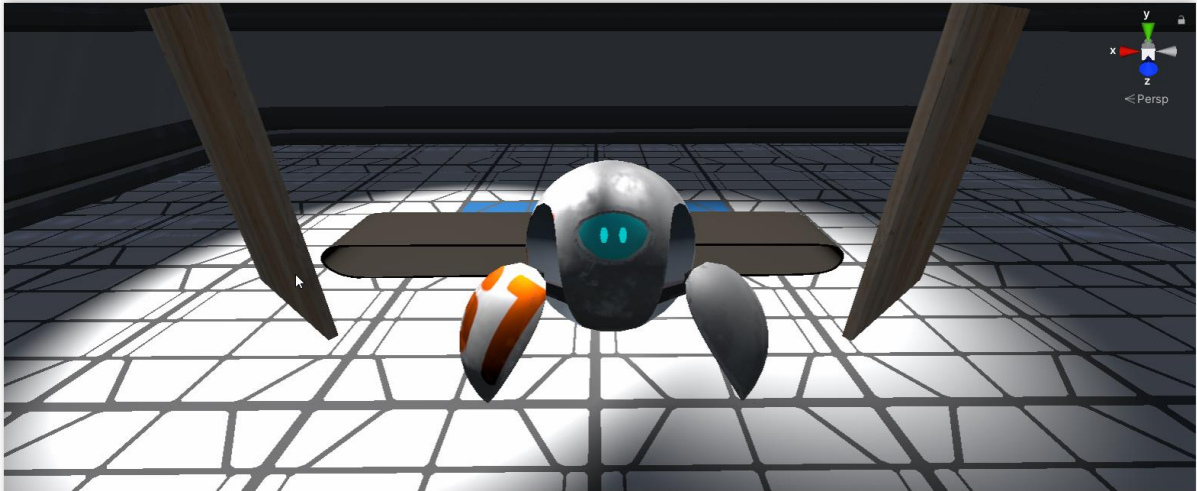


Figure 38 Inside a PC Scene - Companion

Then the idea was that the players should see and grab with their hands every component, so that they know their look and then read and hear its use. After learning all that, a test was needed to check their newly acquainted knowledge. These led to a few specific design decisions.

- **Design**

In order to acquaint the players with the controls, there is an indication on the controller of the button used to teleport from area to area. There were two available methods for moving, continuous and teleportation, with the second being picked because the other one was causing motion sickness. So the place was designed as such. Along with the indication, the companion encourages the players to get on a specific blue platform by “pressing the circular button on their controller, pointing to the ground and releasing”, another hint on how to teleport. The particular color of the platform was used to distinguish it from the others and make it stand out so that the player wants to get there intuitively.

After getting to the blue platform the players are presented with a desktop filled with the components for a quick look and then they are split so they can navigate them one by one. The companion urges the players to touch each object and press the trigger button to pick them up. That way he is able to describe each component to them thoroughly, as many times as they want. Everything the companion says is also written on a blackboard at the

left of the player, while there is also another blackboard on the right that is used for an upcoming quiz. The items and their descriptions are:

- **Computer Case:** A computer case is the enclosure that contains most of the components of a personal computer.
- **RAM:** Random-Access Memory or RAM is a form of computer memory that can be read and changed in any order. The stored data is lost when the computer is shut down.
- **Motherboard:** The Motherboard is the main circuit board of a computer and makes everything work together. Attached to it, one can find the CPU, memory RAM expansion slots and other components.
- **PSU:** A Power Supply Unit (PSU) converts mains AC to low-voltage regulated DC power for the internal components of a computer. Basically, it supplies power to all components.
- **HDD:** A Hard Disk Drive or HDD is a data storage device that stores data even after the computer is shut down.
- **SSD:** Solid State Drive or SSD is a data storage device, similar to a hard disk drive (HDD), but a lot faster at reading and writing.
- **CPU:** A Central Processing Unit (CPU) is the electronic circuitry within a computer that executes instructions that make up a computer program.
- **Sound Card:** A Sound Card is an internal expansion card that provides input and output of audio signals to and from a computer under control of computer programs.
- **Video Card:** A Video Card is an expansion card which generates a feed of output images to a display device (such as a computer monitor).

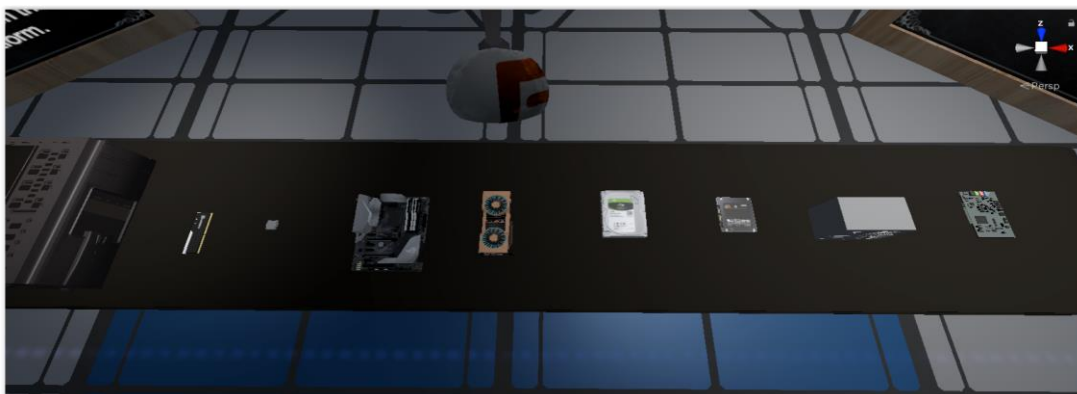


Figure 39 Inside a PC Scene - Components

Upon grabbing every item, the player is prompted to hit a button so that the quiz is started. He does that by pointing the laser to a 'Continue' button on the right blackboard and pressing the trigger button. Then the quiz starts to gauge the knowledge that the players got until this place. The questions and their options are:

- Which component is responsible for the output of a computer monitor?
 - Video Card
 - CPU
 - HDD
- Which of these components can store data even after the computer is shut down?
 - HDD
 - RAM
 - Case
- Which component is responsible for executing instructions that make up computer programs?
 - CPU
 - RAM
 - SSD
- Which of these components is responsible for supplying other components with power?
 - Power Supply
 - Video Card
 - Motherboard
- Which of these components can read and write data faster?
 - HDD
 - Video Card
 - SSD
- Which component is the enclosure that contains most of the components of a personal computer?
 - Case
 - HDD
 - RAM
- Which component is the main circuit board of a computer and makes everything work together?
 - Motherboard

- Video Card
- Power Supply
- Which component provides input and output of audio signals to and from a computer?
 - Sound Card
 - Case
 - RAM



Figure 40 Inside a PC Scene - Quiz Board

At every correct answer the player continues to the next, while at every wrong one another questions comes up and the other is held to be asked later. This was to be certain that the players knew the correct answers at all questions by the end of the quiz. This was to test if they understood the functionalities.

Finishing the quiz the players are transferred to another room a lot more enclosed and darker. Also, the companion is heard whispering through a speaker in there, in order to improve immersion. The players now have to listen to the descriptions of the objects and throw the right components inside some vents, after picking them up from some shelves. At every correct item the companion asks for the next one, while at every wrong one, he asks for it again, by indicating that he got the wrong one. After three errors the test is to be taken again. This test is a way to gauge whether the players can distinguish the components based on their use and their appearance.

The prompts on this test are:

- Get me a computer memory that can be read and changed in any order.
- Get me the electronic circuitry within a computer that executes instructions.
- Get me a Hard Disk Drive or HDD.
- Get me the main circuit board of a computer that makes everything work together.
- Get me a Solid State Drive or SSD.
- Get me the component that generates a feed of output images to a display device.
- Get me the component that provides input and output of audio signals to and from a computer.
- Get me the part that supplies power to all components.



Figure 41 Inside a PC Scene - Test Environment

Upon finishing the test the player is transferred back to the main menu to choose the next level.

- **2D/3D Art**

As said above, the external environment of the scene consists of a snowy mountain along with a lake. It's also snowing so there are particles for that too and a night sky gives a feeling of loneliness. As the scene starts the players are up on a platform looking at the mountain lake and a wooden porch in front of them. Behind them is the internal environment.

The interior looks like a factory with a production line having a desktop in the middle (and afterwards every component alone). At the center of the space is the companion/robot and at each of his sides there is a blackboard, one for everything that the companion says and another one for the quiz. Also, on the right blackboard there is a light to indicate correct and wrong answers during the examination.

The 2D/3D art assets used for this scene: (add links)

- Snaps Prototype | Sci-Fi / Industrial
- Robot Sphere³
- Spotlight and Structure⁴
- MotherBoard + Components⁵
- RTX Titan⁶
- Garage Props Pack⁷
- SciFi Light 11⁸

- **Sound**

A few sounds were used in this scene mainly to create atmosphere. There was some snow/mountain ambience sound along with more serious background music for the test to accompany the darkness. There were also two sound effects for correct and wrong while the player was doing the quiz, for better indication.

Downloaded sound assets for this scene:

- Correct Bell⁹
- Wrong Choice¹⁰
- Lightly snowing¹¹
- Ominous Background¹²

³ <https://assetstore.unity.com/packages/3d/characters/robots/robot-sphere-136226#content>

⁴ <https://assetstore.unity.com/packages/3d/props/interior/spotlight-and-structure-141453>

⁵ <https://sketchfab.com/3d-models/motherboard-components-3bc94057328243d4b341a55f59160f8a>

⁶ <https://sketchfab.com/3d-models/rtx-titan-f7e8fb69cce14f53a2d9159898fd89e5#download>

⁷ https://assetstore.unity.com/packages/3d/props/industrial/garage-props-pack-132664?aid=11011Gxj&pubref=132664&utm_source=aff

⁸ <https://sketchfab.com/3d-models/scifi-light-11-ba0baa5f0f514726ae81436674eddac1>

⁹ <https://freesound.org/people/Fupicat/sounds/538147/>

¹⁰ <https://freesound.org/people/unadamlar/sounds/476177/>

¹¹ <https://freesound.org/people/florianreichelt/sounds/453167/>

¹² https://freesound.org/people/Zat_Dude/sounds/402439/

Companion lines:

→ **Main Level**

- ◆ Hellooooo, could you please maybe get away from there? I'm getting kind of anxious.
Press and hold the circular button on your controller. Aim somewhere on the ground and release.
Come here, in front of the desktop, on the blue platform.
- ◆ Great! In this lesson we'll learn about what is inside a computer.
Whenever you're ready press one of the grip buttons on the back of your controller.
- ◆ Alright then, go ahead and grab the items here by pressing the trigger on the back of your controller.
I'll explain each and every one of them.
Keep your eyes and ears open though, there is a quiz later!

→ **Components**

- ◆ A computer case is the enclosure that contains most of the components of a personal computer.
- ◆ Random-Access Memory or RAM is a form of computer memory that can be read and changed in any order. The stored data is lost when the computer is shut down.
- ◆ The Motherboard is the main circuit board of a computer and makes everything work together. Attached to it, one can find the CPU, memory RAM expansion slots and other components.
- ◆ A Power Supply Unit (PSU) converts mains AC to low-voltage regulated DC power for the internal components of a computer. Basically, it supplies power to all components.
- ◆ A Hard Disk Drive or HDD is a data storage device that stores data even after the computer is shut down.
- ◆ Solid State Drive or SSD is a data storage device, similar to a hard disk drive (HDD), but a lot faster at reading and writing.
- ◆ A Central Processing Unit (CPU) is the electronic circuitry within a computer that executes instructions that make up a computer program.

- ◆ A Sound Card is an internal expansion card that provides input and output of audio signals to and from a computer under control of computer programs.
- ◆ A Video Card is an expansion card which generates a feed of output images to a display device (such as a computer monitor).
- ◆ Awesome! Now that you picked up everything, click on the button on the right to proceed to the quiz.
- ◆ Very well, let's go to a more closed space now. I'll need your help for an experiment.

→ Quiz

- ◆ Which component is responsible for the output of a computer monitor?
- ◆ Which of these components can store data even after the computer is shut down?
- ◆ Which component is responsible for executing instructions that make up computer programs?
- ◆ Which of these components is responsible for supplying other components with power?
- ◆ Which of these components can read and write data faster?
- ◆ Which component is the enclosure that contains most of the components of a personal computer?
- ◆ Which component is the main circuit board of a computer and makes everything work together?
- ◆ Which component provides input and output of audio signals to and from a computer?

→ Test

- ◆ Okay now, we'll have to do this silently, because people are sleeping here. I'm going to need you to put the items on the shelves inside those black vents.
Just throw them in! Ok let's start.
- ◆ Get me a computer memory that can be read and changed in any order.
- ◆ Get me the electronic circuitry within a computer that executes instructions.
- ◆ Get me a Hard Disk Drive or HDD.
- ◆ Get me the main circuit board of a computer that makes everything work together.

- ◆ Get me a Solid State Drive or SSD.
- ◆ Get me the component that generates a feed of output images to a display device.
- ◆ Get me the component that provides input and output of audio signals to and from a computer.
- ◆ Get me the part that supplies power to all components.
- ◆ No not that! Try again.
- ◆ Not quite right. Find the right one.
- ◆ Well done! Everything went as planned! See you on the other side!
- ◆ Hmm that didn't work... We'll have to do it again.

- **Programming**

A number of scripts were written in order to get all the functionality needed. Except for some standard ones, for example the Player script for VR was provided by SteamVR along with the one that handled Teleportation, a few more were needed to handle some other things. Specifically:

1. **Player script (from SteamVR):** It handled room-scale movement and in general the whole package of VR movement and interaction.
2. **Teleporting (from SteamVR):** It handled teleportation for movement from place to place.
3. **Interactable (from SteamVR):** To make a gameObject interactable for VR, to be able to pick it up and rotate it with the controller.
4. **Throwable (from SteamVR):** To make a gameObject follow physics while the player can throw it.
5. **Pointer:** Explained before, to point and select.
6. **VR Input Module:** To handle input from the controllers.
7. **Game Manager:** To change from full desktop to specific parts and start the quiz.
8. **Dialogue Manager:** To handle incoming dialogues from the companion.
9. **Dialogue Trigger:** To trigger new dialogues.
10. **Dialogue:** A class created to specify a dialogue (audio and lines).
11. **Blue Platform Script:** To check whether the player has reached the blue platform and check for input of Grip Buttons to continue to the next dialogue.

12. **Part Script:** In some cases the VR controllers were malfunctioning and caused some grabbed gameObjects to disappear and this script was responsible to return them to their previous place.
13. **Pick up item script:** To play the audio every time the players pick up a component and change the text on the blackboard. Also, to check when everything was picked up in order to go to the quiz.
14. **Choice script:** To handle the quiz. Specifically to show every question randomly, along with its answers and check the correctness of the option the players chose. Also, to check when every question was answered to proceed to the test.
15. **Test Manager:** To handle the test. In specific, it changed the request of the companion and checked for correctness when the players put components into the vents. Also, handled errors for either restarting the test or moving to the main menu.

- **VR Advantages**

In this lesson the students try to grasp the functionalities of every component while also seeing their look and scale. This could also happen in a real classroom however it's not always possible to have all the components or add new ones. This scalability is really easily implemented in a virtual environment, facilitating the process.

3.3.1.3 Algorithms

The purpose of this lesson is to teach some basic algorithmic terms and give an intro to how a problem is solved through them.



Figure 42 Algorithms Scene

- **Pre-Design**

To describe what an algorithm is, a good example should be chosen. In this scene a recipe was used, because it has all the ideas that an algorithm has, like specific orders, values, repetitions etc. All the algorithmic terms were explained in parallel with a recipe, making the lesson fun and comprehensible.

In order to put the players to interact with ingredients, they were put on a film set, where they were prompted to do a cooking show. In that show, they had to follow a recipe to create pancakes, while the companion explained the algorithmic terms.

- **Design**

Starting the scene, the players see in front of them all the ingredients and tools they are going to need, while also they are given the controls needed to proceed to the next line of dialogue. Then comes the definition of what an algorithm is by the companion, explaining how there is going to be a parallelism with recipes. This states up-front what the lesson is about and how it's going to be taught.



Figure 43 Algorithms Scene - Kitchen Counter

The algorithmic terms needed to be described were:

1. **Assignment statements:** Explained as the ingredients of a recipe. For example: 2 eggs assign how many eggs the recipe needs.
2. **Input statements:** This was explained as the user perhaps saying how many eggs the recipe needs, so the recipe takes the number of eggs it needs from an input.

3. **Branching statement:** Explained with the option to have salty or sweet pancakes. If someone wants the first then he/she puts salt in the mix, otherwise he/she puts sugar.
4. **For loop statement:** Explained as how many times the player should break an egg and put it in the mix.
5. **While loop statement:** Explained as when should the player stop pouring milk into the mix.

It was required from the player to manipulate various items on the kitchen counter, so it was important that all of them were able to be grabbed, even the ones that shouldn't be used. This increased immersion and gave truthfulness to the world.

The actions that the player had to do were fairly straightforward. These were:

- Pick up sugar cubes from a bowl and put them inside the mix
- Pick up the milk and pour it in the mix until it becomes yellow
- Pick up the eggs needed, break them by hitting them on the pan and put them in the mix

The players finish the recipe the first time with the help of the companion and then, in order to test their knowledge, they are prompted to do it again "on air". This time very little help is provided by the companion and the recipe has minor changes to challenge the players' adaptability. At the end of the show they are transferred back to the main menu.

- **2D/3D Art**

For this scene a lot of detailed 3D objects were needed to make the kitchen set more realistic. From the kitchen counter, pan and mixer, to all the ingredients like eggs, pancakes, broken eggs, cartons of milk and sugar cubes, everything had to have a sense of realism. For the milk itself, a line renderer was used which is something like a thick line with some particles showing up where it lands.

Regarding the lighting of the room, there are some spotlights above and in front of the player pointing at him, making it look like a show, especially with the cameras next to the lights. When the show/test starts, the lights change color to indicate the different nature of the recipe.

The 2D/3D art assets used for this scene: (add links)

- Pancake¹³
- S**t happens¹⁴
- Milk Carton¹⁵
- modern kitchen Free 3D model¹⁶
- photography studio objects Free 3D model¹⁷
- Egg¹⁸
- Mixer¹⁹
- Sugar Cubes²⁰

- **Sound**

A background jazz music was picked for this scene to accompany the kitchen set and the show. It sets a lighter mood for the players to make the experience stress-free.

Downloaded sound asset for the scene:

- jazz_music_loop.mp3²¹

The lines of dialogue used by the companion are:

- ➔ Alright! So, in this lesson, we're going to learn what an algorithm is and its critical elements. Press the Grip Button on your controller to move to the next sentence.
- ➔ An Algorithm is a list of instructions and rules that are needed to be done, in order to complete a task. A recipe to create pancakes is an algorithm. That's what we will do now.
- ➔ Let's take a look at the recipe and try to link it to algorithmic terms.
- ➔ Look at the ingredients for our pancake mix:
 - 1 cup of flour
 - 2 cubes of sugar
 - 1 pinch of salt

¹³ <https://sketchfab.com/3d-models/pancake-f533087557cb47549c5e6ab5554005ce>

¹⁴ <https://sketchfab.com/3d-models/st-happens-9df0cb94b17a4221bd48ba9d8f225b11>

¹⁵ <https://sketchfab.com/3d-models/milk-carton-713643151cce41798f6dfb98bbff09cb>

¹⁶ <https://www.cgtrader.com/free-3d-models/interior/kitchen/modern-kitchen-a55f9ecb-5edb-4edf-be7a-2a761550739f>

¹⁷ <https://www.cgtrader.com/free-3d-models/electronics/video/photography-studio-objects>

¹⁸ <https://sketchfab.com/3d-models/egg-04c322d857504a80983d4bf4f4b56ba9>

¹⁹ <https://sketchfab.com/3d-models/mixer-382319dab129471aa4985d0f599bd48d>

²⁰ <https://sketchfab.com/3d-models/sugar-cubes-c47bbf11407043b6938fa638a68c2851>

²¹ <https://freesound.org/people/anechoix/sounds/456797/>

2 eggs

Half a carton of milk

→ The ingredients can be matched to an algorithmic term called assignment statements.

→ They look like this:

flour <- 1 cup

sugar <- 2 cubes (if they are sweet pancakes)

salt <- 1 pinch (if they are salty pancakes)

eggs <- 2

milk <- half a carton

→ Every ingredient can be called a 'Variable'. The name suggests that the value of every ingredient is subject to change. We may want less eggs for example. So after the first assignment we can do:

eggs <- 1

Then the 'eggs' variable has 1 as value.

→ These can also be provided by the users. To get them from the users we have to ask them.

This can be done by input statements.

→ Input statements look like this:

input flour, sugar, eggs ... etc.

Then the user is prompted to put each of them and then every variable has the desired value. So we understand that a variable has that type:

Variable Name <- Value

→ Now that we have all our ingredients assigned, let's start our recipe.

→ We have this bowl of flour with the exact dosage that we need. This has the required flour inside (1 cup).

→ Next we want to add in the bowl either the sugar or the salt.

→ We can check that by a branching statement. A branching statement looks like this:

if some condition is true, then

statements

else if other condition is true, then

statements

else

statements

end if

→ In our case we want to determine what we will put in the mix. Our branching statement looks like this.

if we are making sweet pancakes, then

 put sugar in the mix

else if we are making salty pancakes, then

 put salt in the mix

end if

→ We are making a sweet pancake mix so we need to put two sugar cubes in.

 Find 2 sugar cubes and put them in the bowl.

→ Awesome. Now we'll take a look at loop statements.

→ Loop statements, as the name suggests, are the ones that will be called repeatedly, until a condition is reached.

We'll start with a while loop.

→ while some condition is true do

 statements

end while

What that tells is that while something is true, you will execute the following statements.

→ In our recipe we want to pour milk into the mix, until it has taken a yellow color.

That translates to:

while mix color is not yellow do

 pour milk

end while

 Now pour milk until the mix is yellow.

→ Nice job! Another loop statement that we will learn about is the 'For Loop'.

→ The For Loop tells us that a statement is to be executed for a certain amount of times.

To explain it better let's take a look at the example.

→ for eggs bigger than zero do

 break egg

 put broken egg in mix

 eggs <- eggs - 1 (remove 1 from the variable 'eggs')

end for

So for our number of eggs, we are going to break each one and put it in the mix. Every time we do that, we subtract the number of eggs we have to put in the mix. That way when the variable 'eggs' reaches zero, we get out of the For Loop.

- ****Put the required number of broken eggs in the mix.****
****To break the eggs, hit them on the pan.****
- Great! Now we have put everything we need in the mix. Now it's time to renew the bowl and get ready for your test.
- We will be on air in a minute, the cameras are rolling. You have to make the recipe as we discussed. The ingredients and the algorithm will appear on the board, along with instructions. Press the Grip Button whenever you're ready. Good luck!
- 5 4 3 2 1. Welcome, today we are making a sweet pancake mix! We have a cup of flour in our bowl. Let's begin by adding our next ingredient.
****If we are making a sweet pancake mix, what will you put in the mix?*****
- After the salt, we need to put our next ingredient. This will make our pancakes nice and juicy.
****Add the milk.****
- And now we must break some eeeggs!
****Break the eggs and put them inside.****
- Very well! That was it for today ladies and gentlemen, I hope you make those pancakes yourself! Let's move to another show!

And there were some extra lines to indicate that the player made a mistake:

- We don't need that right now...
- This ingredient is for another time.
- No, the recipe might not work. Be careful!

- **Programming**

Various scripts were used in this scene to handle adding ingredients to the mix and check their correctness in dosage and accuracy:

1. **Player script (from SteamVR):** It handled room-scale movement and in general the whole package of VR movement and interaction.

2. **Teleporting (from SteamVR):** It handled teleportation for movement from place to place.
3. **Interactable (from SteamVR):** To make a gameObject interactable for VR, to be able to pick it up and rotate it with the controller.
4. **Throwable (from SteamVR):** To make a gameObject follow physics while the player can throw it.
5. **Dialogue Manager:** To handle incoming dialogues from the companion.
6. **Dialogue Trigger:** To trigger new dialogues.
7. **Dialogue:** A class created to specify a dialogue (audio and lines).
8. **Egg:** To check the collision and transform a normal egg to a broken one.
9. **Mix:** To handle what the recipe wants next and if the player is putting in the right ingredient. Also triggers the dialogues every time a right or a wrong one is added.
10. **Spill:** To handle the pouring of the milk, along with the time needed to pour until the mix is yellow.
11. **Stream:** To create the line renderer that is used as the milk.

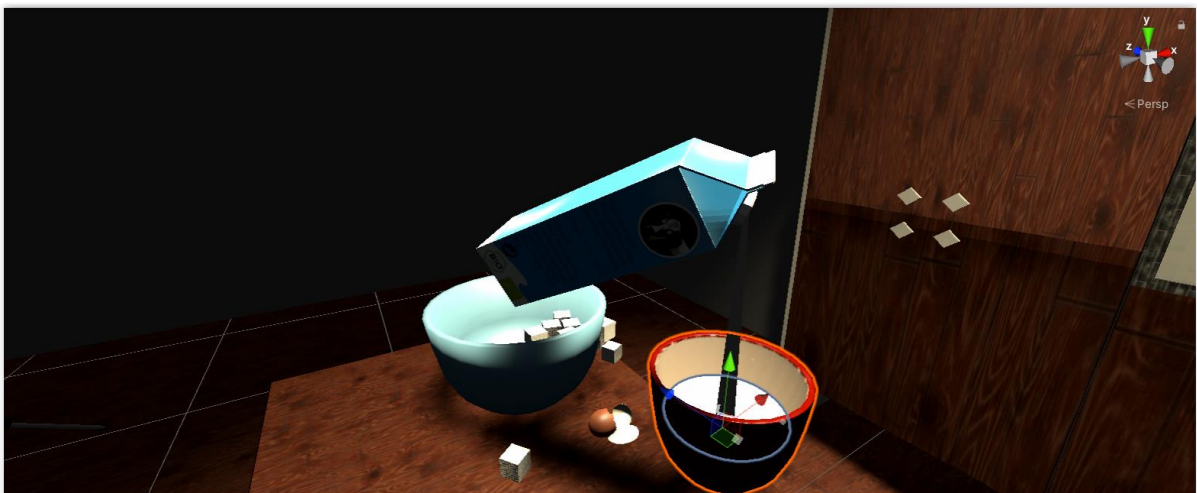


Figure 44 Algorithms Scene - Milk Mechanic

- **VR Advantages**

While someone can take students to a kitchen and try to teach them this lesson, it's not always applicable and certainly not scalable. Through the virtual world it's obvious that a more immersive experience can occur without the cost of resources.

3.3.1.4 Networks

The thoughts behind this scene were to teach the fundamentals of networking. Specifically, show the most important components of house networks and how they work.

- **Pre-Design**

Similarly to the ‘Inside a PC’ scene, a big space was used as a background, with a forest and a lake giving a feel of realism. The scenery reminds autumn with the sun glazing in the sky. The players start inside a watchtower, with some network components on top of a desk and a bed behind them. All these are to make them feel a sense of ownership and the reality of being a communications manager.

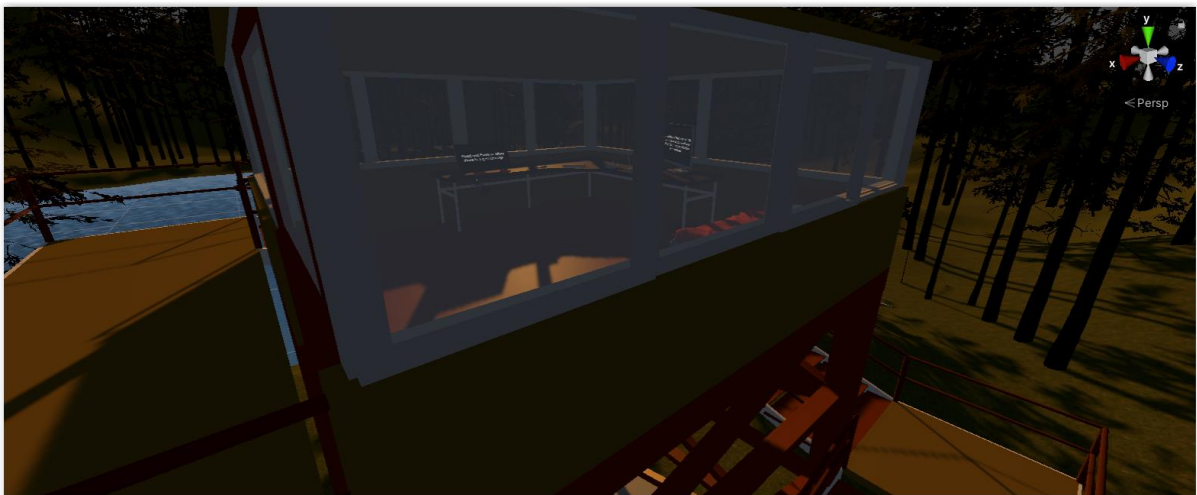


Figure 45 Networks Scene

The idea is that the players can see and interact with the components; the companion should teach them their functionalities, along with some other important terms and then test their newly acquainted knowledge with a true or false examination.

- **Design**

The important parts of networking that were needed to be taught were some hardware components and a few other terms. For the first ones, on the desk inside the tower there is a switch, a router, an access point and a Network Interface Card (NIC), while for the second ones the companion explains the definitions of a network, a Local Area Network (LAN), a Wide Area Network (WAN) and wireless connectivity.

In particular:

- That's a switch. A switch is a hardware device that connects multiple devices, on a wired computer network, creating a LAN.
- That's a router. A router is responsible for sending and receiving (routing) data between networks. Most routers today work as Access Points too.
- That's an access point. An Access Point does the same job as the Switch, connecting devices creating a LAN, however with wireless connection with the devices. Wireless connectivity happens through electromagnetic waves. Most routers today work as access points too.
- That's a Network Interface Card (NIC). An NIC is a component inside the computer that enables wired and wireless network connection.
- What is a network you ask? A network is defined as a group of two or more computer systems, linked together, in order to exchange information. Easy right?
- LAN or Local Area Network is a network in a small, distinct area. Like a house for example.
- WAN or Wide Area Network is a network of different LANs (networks). Like a university. The internet is a network of networks too. Just a very big one.



Figure 46 Networks Scene - Environment

After all that, a test is given to the players in order to gauge their knowledge, which has a true or false format.

The questions are:

- If two devices are connected to the same Switch then they are in the same LAN (Local Area Network).

- The internet is not a network.
- A router handles the routing of data between networks.
- The internet is a Local Area Network (LAN).
- A computer needs a Network Interface Card (NIC) in order to connect to networks.
- If two devices are connected to the same Access Point, then they are in the same LAN (Local Area Network).
- Routers today tend to work as Access Points as well.
- A computer can connect with and without a wire to a network if they DO NOT have a Network Interface Card (NIC).
- Wireless connection is possible through electromagnetic waves.

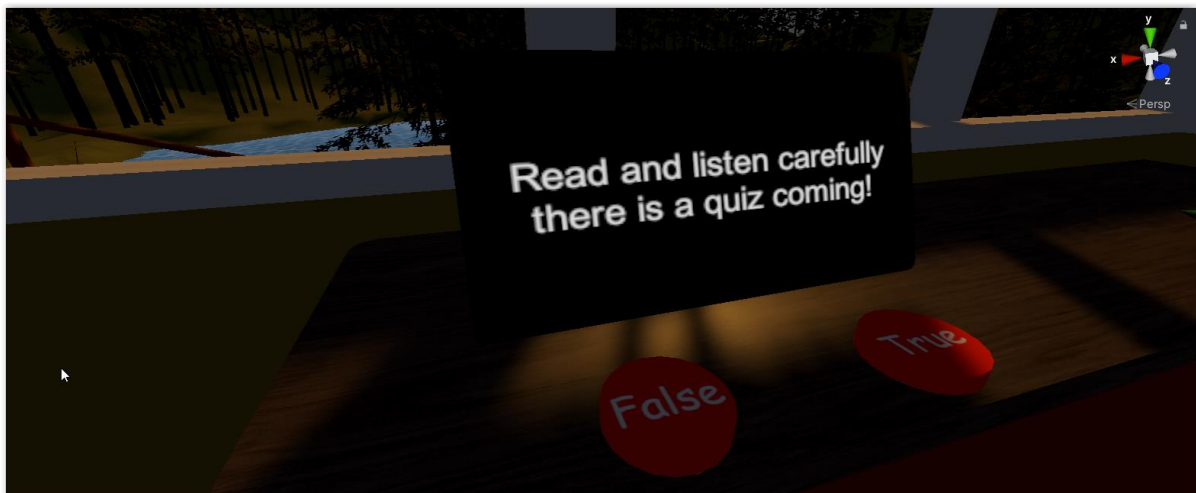


Figure 47 Networks Scene - Test Tablet

After answering all the questions they are returned to the main menu.

- **2D/3D Art**

The assets for this scene needed a lot of detail to understand the physical form of the hardware components. The terrain, with the trees and the lake had less detail as they are in the background. The bed and the bed sheets have a good level of detail to provide some realism, while there is also a computer screen to see what the companion says and a tablet to show the test questions.

The lighting was mainly handled by a directional light that acted as a hot sun. It illuminates the components and gives a feel of a relaxing noon to the players.

The downloaded 2D/3D art assets:

- Lv2 Watchtower²²
- Messy Bed²³
- Router²⁴
- Desk Asset²⁵
- Junpiper EX3300 Metro Ethernet Switch Free 3D model²⁶

- **Sound**

A forest ambience was selected for this specific scene to accompany the environment and spark a relaxing feeling to the players.

- Kampina forest spring009 190322_1321.wav²⁷

Apart from the definitions, the companion says the following lines of dialogue:

- ➔ Look at this view! Magnificent right? Ok let it inspire us, for today we will learn about networks! Hit your Grib Button when you're ready.
- ➔ Let's start from a few definitions!
- ➔ Let's look at some network categories.
- ➔ Now that that's out of the way, pick up the components on the desk and I'll explain their uses.
- ➔ Great. Now that we have cleared everything out, it's time for our quiz.
Push the True or False button to answer.
- ➔ Very Well! Good job! That would be all for this lesson.

- **Programming**

Only one new script was created for this specific scene and it is to handle the true or false test.

All the scripts needed:

1. **Player script (from SteamVR):** It handled room-scale movement and in general the whole package of VR movement and interaction.

²² <https://sketchfab.com/3d-models/lv2-watchtower-e532b302807f464f9193a2a4e0d30fc1>

²³ <https://sketchfab.com/3d-models/messy-bed-b49dc1778b0b430cabdbad327d6e2e0d>

²⁴ <https://sketchfab.com/3d-models/router-ac86d8ae65a54f4aa99d7d624f71e5f4>

²⁵ <https://sketchfab.com/3d-models/desk-asset-ca972bf129344a73a3fe59a0eef78164?fbclid=IwAR2fNcknCFHfW/Nx62kdshM1BuYosaK0qt8U3vNeb1zlyexbLOQ4sexqoh2Q>

²⁶ <https://www.cgtrader.com/free-3d-models/electronics/computer/junpiper-ex3300-metro-ethernet-switch>

²⁷ <https://freesound.org/people/klankbeeld/sounds/546179/>

2. **Teleporting (from SteamVR):** It handled teleportation for movement from place to place.
3. **Interactable (from SteamVR):** To make a gameObject interactable for VR (to be able to pick it up and rotate it with the controller).
4. **Throwable (from SteamVR):** To make a gameObject follow physics while the player can throw it.
5. **Dialogue Manager:** To handle incoming dialogues from the companion.
6. **Dialogue Trigger:** To trigger new dialogues.
7. **Dialogue:** A class created to specify a dialogue (audio and lines).
8. **Pick up item script:** To play the audio every time the players pick up a component and change the text on the screen.
9. **True or false script:** To add the questions of the test, project them on the tablet and check the correctness of the players' answers.

- **VR Advantages**

This lesson could have happened to a different place in the real world too; however it could not provide the feeling of the virtual. As in the previous lessons, adding and removing components becomes easier and creating a specified experience for the students gives a newly acquainted freedom to the teachers.

4 Conclusion

It's obvious that education as we know it will evolve to something better and quicker. Classrooms will be replaced with virtual environments and theoretical knowledge will be substituted with a practical more interactive approach on things. That said everything will work for the best, making learning and teaching effortless and fun.

Through this thesis, an extension of virtual educational material is available for use. A lot of work and knowledge were put into taking the field a step further and building ground for next applications to come. Of course, there were some things that were left out or weren't implemented yet. These all concern future work done upon this application and more immersive features created. Ideas that have come to mind involve new levels that teach more in depth, or other fields of Computer Science, while there is a possibility to create scenes for totally different subjects (Math, Physics, etc.).

5 References

- [1] S. Thompson, "VR Applications: 21 Industries already using Virtual Reality," *VirtualSpeech*, 11-Dec-2020. [Online]. Available: <https://virtualspeech.com/blog/vr-applications>. [Accessed: 07- Jun- 2021].
- [2] N. Babich, "How VR Education Will Change How We Learn & Teach | Adobe XD Ideas", *Ideas*, 2021. [Online]. Available: <https://xd.adobe.com/ideas/principles/emerging-technology/virtual-reality-will-change-learn-teach/>. [Accessed: 07- Jun- 2021].
- [3] P. Chang, "Inclusivity Of VR And AR Accessibility For The Visually And Hearing Impaired", *ARPost*, 2021. [Online]. Available: <https://arpost.co/2018/09/18/inclusivity-of-vr-and-ar-accessibility-for-the-visually-and-hearing-impaired/>. [Accessed: 07- Jun- 2021].
- [4] "America's Army", *Americasarmy.com*, 2021. [Online]. Available: <https://www.americasarmy.com/>. [Accessed: 07- Jun- 2021].
- [5] "How Do Game Engines Work?," *Interesting Engineering*, 29-Oct-2019. [Online]. Available: <https://interestingengineering.com/how-game-engines-work>. [Accessed: 07- Jun- 2021].
- [6] "Unity (game engine) - Wikipedia", *En.wikipedia.org*, 2021. [Online]. Available: [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine)). [Accessed: 07- Jun- 2021].
- [7] "Unreal Engine - Wikipedia", *En.wikipedia.org*, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Unreal_Engine. [Accessed: 07- Jun- 2021].
- [8] "Comparison of virtual reality headsets | Wikiwand", *Wikiwand*, 2021. [Online]. Available: https://www.wikiwand.com/en/Comparison_of_virtual_reality_headsets. [Accessed: 07- Jun- 2021].
- [9] "Buy VIVE Hardware | VIVE European Union", *Vive.com*, 2021. [Online]. Available: <https://www.vive.com/eu/product/vive/>. [Accessed: 07- Jun- 2021].
- [10] "About the VIVE controllers", *Vive.com*, 2021. [Online]. Available: https://www.vive.com/hk/support/vive/category_howto/about-the-controllers.html. [Accessed: 07- Jun- 2021].