Code PFE : 334

Ministry of Higher Education and Scientific Research
University of Manouba
Higher Institute of Multimedia Arts

Graduation Project Report

Prepared for the purpose of obtaining the degree of Engineer in Computer Science and Multimedia

---

Virtual Festival VR experience

Produced by

Sarah KHAMASSI

supervised by

Mr. Mohamed Karim AZIZI– ISAMM

Mr. Oussama BEN MARIEM- Lanterns Studios

Academic Year: 2021/2022

Dedications

I would like to dedicate this work to my marvelous mother who always believed in me and has always been the source of my never-ending strength, and always will be.

To my brother who always encouraged me to follow through with anything and never give up.

To my friends and classmates who I shared with many memories through those years and went with through the toughest hardships.

To anyone who crossed my path, believed in my capabilities and supported me by any means, and wanted me to succeed.

Thank you.

Acknowledgment

I would like to express my sincere gratitude to :

Mr.Oussama BEN MARIEM for his guidance and mentorship, I learned a lot by his side in this internship from his extensive experience and helped me through the completion of this ambitious project.

I would like to also thank my academic supervisor **Mr.Mohamed Karim AZIZI** for his advice, patience during my internship and for his contribution to the success of this project.

A special acknowledgment goes to each member of the splendid **team of Lanterns Studios** for their immense support and encouragement through my internship, working with them and learning from them was an considerable privilege for me.

Contents

List of Figures

List of Tables

Glossary

VR *Virtual Reality*

XR *Extended Reality*

BP *Blueprint*

VIP *Very Important Person*

SCRUM *Systematic Customer Resolution Unraveling Meeting*

UMG *Unreal Motion Graphics*

MRC *Mixed Reality Capture*

VRC *Virtual Reality Checks*

FPS *Frame Per Second*

3D *Three Dimensional*

General Introduction

During difficult times like the pandemic most artists and performers encountered difficult times to meet their public and continue to lead their careers as wished as most performance venues closed their doors to the public.

Like everyone else, artists and companies have turned to other means to keep their business going despite difficulties.

This is where modern technology comes in: now most professionals works more than ever on their social presence and earning money through virtual and digital representations.

And some were more innovative than others by turning everything related to a concert experience into a full digital experience by benefiting of the surge of the metaverse and XR development.

To introduce this innovation in the Tunisian market, the company "Lanterns Studios" decides to develop a Meta Quest application for dynamic virtual festival where users watch virtually artists and performers in a unique experience from the comfort of their homes.

With this digitalization of the entertainment and musical industry and the revenues generated from it, we are watching a new era of entertainment come to life where everything happens behind a screen or a headset.

Chapter 1

General Context

In this first chapter, the project will be put in its broader context. It includes four parts :in the first part, the context of the project will be presented. The second part will be dedicated to the introduction of the host organization.
The third part will be reserved for the presentation of the project. Finally, in the fourth part, the methodology adopted will be presented and the method applied in this project will be explained.

1.1 Project Context

The project is called "Virtual Festival VR experience". It was proposed by the company Lanterns Studios, as part of the development of an end-of-study internship for the purpose of obtaining the degree of national engineering diploma in computer science and multimedia, specializing in digital image and virtual reality, at the Higher Institute of Multimedia Arts of La Manouba (ISAMM) for the year university 2021/2022.

1.1.1 Company overview

The project took place in Lanterns Studios. In this section is listed the company's primary activities.

1.1.1.1 Lanterns Studios company

Lanterns studios is an independent video game studio based in Tunis, Tunisia focusing mainly on narrative video games.

Lanterns studios aim to deliver compelling and responsible game-play experiences and stories to the world. The company also provides extended reality apps and motion capture services ranging from creating an AR, MR, or VR application to high-quality motion capture or custom animations.

It is composed by a highly skilled team of various domains related to IT, CG, branding and media content: from mobile apps development to AR/VR experiences, PC to console video games. [1].

Figure 1.1: Lanterns Studios Logo

1.1.1.2 Company leading sector

Lanterns Studios is the first studio in Tunisia that offers high-quality full-performance motion capture solutions using the latest technologies. "Lanterns Mocap" is the performance capture studio and it is unique of its kind, it offers clients a new horizon of possibilities to experiment and do things using the latest mo-cap technologies.

It aims to be the pillar of the regional XR market and a trustworthy partner in this fast-evolving industry. The company strives to stay at the forefront of XR solutions and guarantees customers highly professional and enriched services.

1.1.1.3 Company mission

In order to stay the market leader, Lanterns Studios is keeping its teams up to date with the latest technologies and media trends who are keen to prove themselves in this industry and invest in training new profiles and talents within the company.

1.1.2 Project overview

1.1.2.1 Problem Description

Numerous science fiction books from the past century have envisioned a lot for the future with the surge of technology and how it's taking a big part of our lives. With the arrival of the Metaverse, some of those fictional writings may come true.

Plenty of people are investing in this new world and buying appropriate devices to assist to the Metaverse birth in real life.

In the virtual world, people will be able to connect, work, play, learn and shop among other activities unlocked in the Metaverse, and therefore the conception and creation of virtual spaces to welcome them are highly requested.

Adding to this, The epidemic has accelerated the adoption of digital experiences and drawn in more individuals than ever before to the Metaverse expecting much

more from it and looking to benefit from all its capabilities to the fullest.

As a result, the entertainment industry was among the first industries to see its virtual counterpart come to life. Now users are keener to seek entertainment and relaxation through a device and live new experiences only possible virtually and thus opening a brand new market to expand on.

1.1.2.2 Suggested Solution

"Virtual festival VR experience" is a module of a bigger project part of the Metaverse. The scope of this project encompasses the creation of an online multiplayer application accessible through virtual reality.

This application allows users to assist a music concert or an artist performance by visiting the virtual spaces available in the application, engaging in conversations, following live broadcasts, and interacting with the representatives. The application offers players to assist in performances as if they were in fact there physically and give to the performers the chance to display their skills and talents to a much broader audience.

1.1.3 Development Methodology

In order to deliver high-quality software in a timely and cost-effective manner that meets the objectives, deciding the best development approach for a project is a crucial step in the project's life cycle. After assessing the existing factors, we divided the project into multiple iterations. Each iteration of the project will focus on a different component. At the end of each iteration, an increment will be provided. As a result, we've opted to employ the Agile SCRUM development process to accomplish this.

1.1.3.1 Agile methodology

Due to the high level of uncertainty in XR projects, it is impossible to acquire all needs in advance, and customers' expectations can evolve or be misread, resulting in a significant cost for consumers.

The choice of this methodology is justified by taking into account its relevance and convenience. Indeed: this choice depends on several factors: customer needs, frequency of requests for change, improvements, cost of delay and team experience, etc. And this methodology is an approach that responds to these requirements and allows the construction of applications on the functional and technical levels.

1.1.3.2 Scrum framework

Scrum is a fast process framework simple to understand for managing complex adaptive challenges while developing and delivering products .

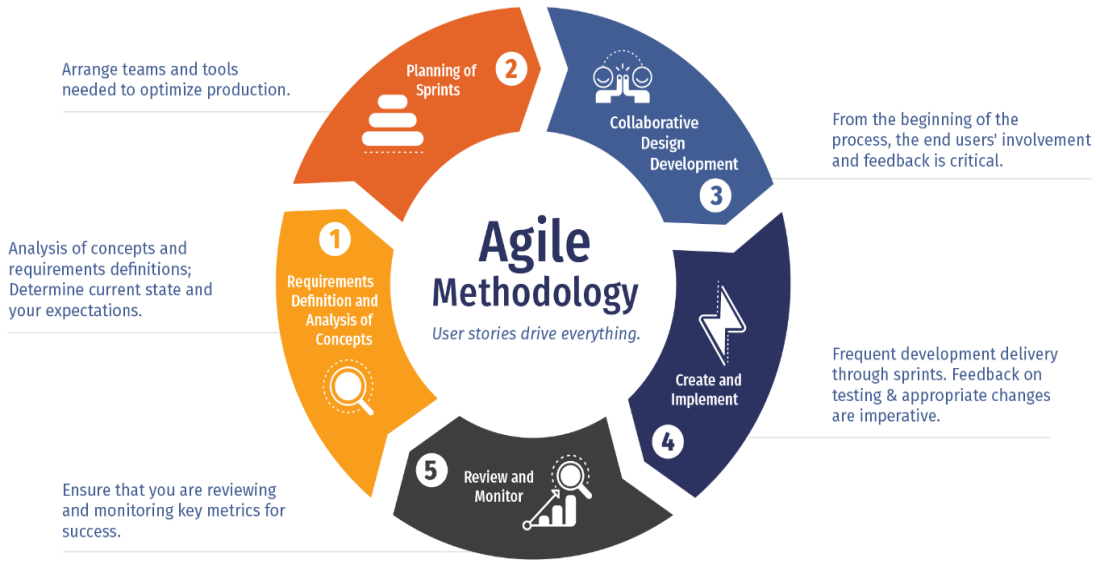

Figure 1.2: Agile methodology cycle

Because scrum is exceptionally compatible with agile principles, it is regarded as a set of Agile and has been widely used in a wide range of industries for over twenty years [6].
It's a suitable method for members adhesion when working in groups to complete a shared project.

Scrum is established on dividing the project into iterations known as "SPRINT." A Sprint can last generally between two weeks and one month. Ahead of each Sprint, tasks are estimated in terms of time and complexity.

1.1.3.3 Scrum terminology

- **User Story :** This is a backlog item consisting of a brief description of a require feature as seen by the user.
- **Vision :** It describes the main objectives, and what is foreseen for the product in the short, medium and long term.
- **Product Backlog :** It holds a list of the customer's needed features into User Stories format. All stakeholders are allowed to add elements, have the visibility of the product backlog. The Product Owner is constantly updating the priorities of the items.
- **Sprint Backlog :** It outlines the requirements that must be agreed upon by the product manager and the project team prior to the Sprint being scheduled. Everyone on the team can see it, and it's used as a reference in daily scrums.

- **Daily Scrum :** It is a meeting held every day with the team members and the product owner lasting on average 15 minutes to learn about the progress of the project, determine the tasks of the day and identify the obstacles preventing the team from reaching the sprint goal.
- **Sprint planning meeting :** It is a vital and important meeting in the project's life cycle that takes place at the start of each Sprint. In this meeting, the project team and product manager define all Sprint goals based on the prioritized "Product Backlog" and estimate the production capacity required to achieve these goals.
- **Sprint Retrospective Meeting :** At the end of every Sprint, another meeting moderated by the SCRUM Master is held where the project team examines the team's behavior and performance throughout the development process To handle the forthcoming Sprints more efficiently.
- **Sprint Review Meeting :** The project team delivers the functionality developed during the last Sprint and collects feedback from users and the Product Owner at the end of it.

Agile methods constitute an innovative movement that aims to add more value to clients and users.

1.2 Study case

Working on this ambitious project requires understanding its needs and challenges. So, after setting the global context of the project, this section is dedicated to studying other projects working on similar visions.

It will start with thorough research on virtual concerts and festivals, what are they, in what way they are different from the classic ones, what do they offer, and review similar applications that provide similar services.

1.2.1 Virtual Concert definition

A virtual concert, also called V-concert or virtual live, refers to a performance in which the performers are represented by virtual avatars. Virtual concerts can take place in real life, where digital representations of the performers are projected in on stage, or within fully digital virtual worlds. Performers in virtual concerts may represent real individuals, but can also be entirely fictitious characters.[2]

1.2.2 Metaverse definition

The Metaverse is the post-reality universe, a perpetual and persistent multiuser environment merging physical reality with digital virtuality. It is based on the convergence of technologies that enable multisensory interactions with virtual environments, digital objects and people such as virtual reality (VR) and augmented reality (AR).

Hence, the Metaverse is an interconnected web of social, networked immersive environments in persistent multiuser platforms. It enables seamless embodied user communication in real-time and dynamic interactions with digital artifacts.[3]

In fact, it is an inter-connected virtual spaces merging different platforms and real-life activities into one place where users can explore and interact with each others as well as with the environment surrounding them.

1.2.3 Project study

This step is essential for the proper functioning of any IT or other project and allows to address the various shortcomings in the current project.
So we chose these systems mainly based on their popularity.

1.2.3.1 WaveXR

Developer of virtual reality-based music platform designed to empower artists and music lovers by transforming the connection through music. The company's platform allows viewing, hosting, and socializing in live and recorded shows worldwide, while also letting artists perform live music by importing their tracks, customizing the visuals of the venue, and sharing virtual shows, enabling artists and music fans to express themselves and connect in new ways.[4]

Figure 1.3: John Legend live concert

In this part we will mention the main features offered by this platform.
Wavexr offers to :

- Make live music into a whole experience.
- Interactivity during performance.
- Globally streamed events.
- make intimate environments.
- Watch the performances across popular social and gaming platforms.
- Transforms participating artists into their own digital avatar.

1.2.3.2 Epic Games

Epic Games, Inc. is an American video game and software developer and publisher based in Cary, North Carolina.Epic Games develops Unreal Engine, a commercially available game engine which also powers their internally developed video games, such as Fortnite[5] where one of the most successful attempts of virtual concerts happened for the first time in a video game featuring the artist Ariana Grande durig the game's interactive Rift Tour.

The outstanding performance where gaming meets music was the first of its kind taking place in a video game and offering an interactive experience including a concert and mini-games in performance to enhance the user experience and the idea of a virtual concert. The show ran five times over three days and accumulated millions of viewers all around the world.

In this part we will mention the main features offered by this platform.
Epic games offers to :

- In-game unique digital experience.
- Offer quests related to the performance.
- Offer experience-related rewards during and after attending the experience.
- Offer exclusive outfits related to the experience.

Figure 1.4: Ariana Grande In-game concert

1.2.3.3 Summary table

Company	Capabilities	Criticism
Wavexr	- user-friendly. - The performances are live. - Using motion capture for the most real and more accurate performance possible. Streams on the most popular platforms.	- Despite having xr on their company name,they don't take AR/VR into consideration in their streaming platforms. - Streams on accessible platforms.
Epic Games	- Uses tested and optimized video game platform as a base to deliver smooth experience. -The performance is playable multiple times. offers multiple exclusive content besides the performance.	- The performances are recorded. - More oriented gaming audiences. - User must have epic game account and Fortnite installed in order to assist to the performance.

Table 1.1: Existing projects capabilities and criticism

Conclusion

Throughout this chapter, the framework of the project is defined by presenting the host organism, describing the problem and the methodology adopted to arrive at the results desired then proceeding to the presentation of a variety of research on the trend of virtual concerts and festivals. Based on this research, features will be identified for the developed solution taking into consideration the weaknesses of other platforms.

Chapter 2

Needs analysis and specification

Introduction

This chapter presents the different actors of the application and a detailed description of the different functional and non-functional needs and ends with the planning of sprints and users stories as a product backlog.

2.1 Needs analysis

Virtual festival is an online multiplayer application dedicated to the wide user. The project's mission is to simplify access to festival performances for virtual reality headset users.

2.1.1 Actor Identification

In this section, the determination of the actors followed by their needs is an important part to able to lead to the next sections:

- **Users:** Users are entities that will connect to benefit from the services offered by the application.
- **VIP Users:** Users are entities that will connect to benefit from the services offered by the application with additional features only available to this actor.

2.1.2 Needs Identification

2.1.2.1 Specification of functional needs

- **For the user :**
  - See other players in the virtual world.
  -See other players interaction
  In the lobby:
  - Navigating in the virtual world: movement in the lobby in an autonomous and natural way with the controllers of the Meta Quest.
  - Consult available places:consult big screen for the available places in the performance stage.
  - Select chair.

- Interact with the lobby.
- Join the performance stage.

In the performance stage:
- Rotate freely in the world with the controllers of the Meta Quest.
- Interact with the performance with a like reaction.
- Interact with the performance with a heart reaction. - Interact with the performance with foam noisemakers.
- Spawn selfie stick.
- Take selfie.
- Preview the selfie taken.

- **For the VIP user :**
  In the lobby:
  - Navigating in the virtual world:movement in the lobby in an autonomous and natural way with the controllers of the Meta Quest.
  - Join lounge.
  - Interact with the lobby.
  - Join the performance stage.

  In the performance stage:
  - Navigating in the virtual world:movement in the lounge in an autonomous and natural way with the controllers of the Meta Quest.
  - Interact with the lounge.
  - Interact with the performance with a VIP like reaction.
  - Interact with the performance with a VIP heart reaction. - Interact with the performance with foam noisemakers.
  - Spawn selfie stick.
  - Take selfie.
  - Preview the selfie taken.

- **Use case Diagram :**
  The figure 2.1 represents the main case diagram of the system containing the main features that will be implemented in the application. As there will be VIP users who will have additional features of their own, they will also be able to benefit from all the features a normal user can do, thus the inheritance relationship between the two actors.
  Before using any application on the meta quest and using it, it is required to proceed to authentication by creating a meta account or by logging in through the Facebook account.

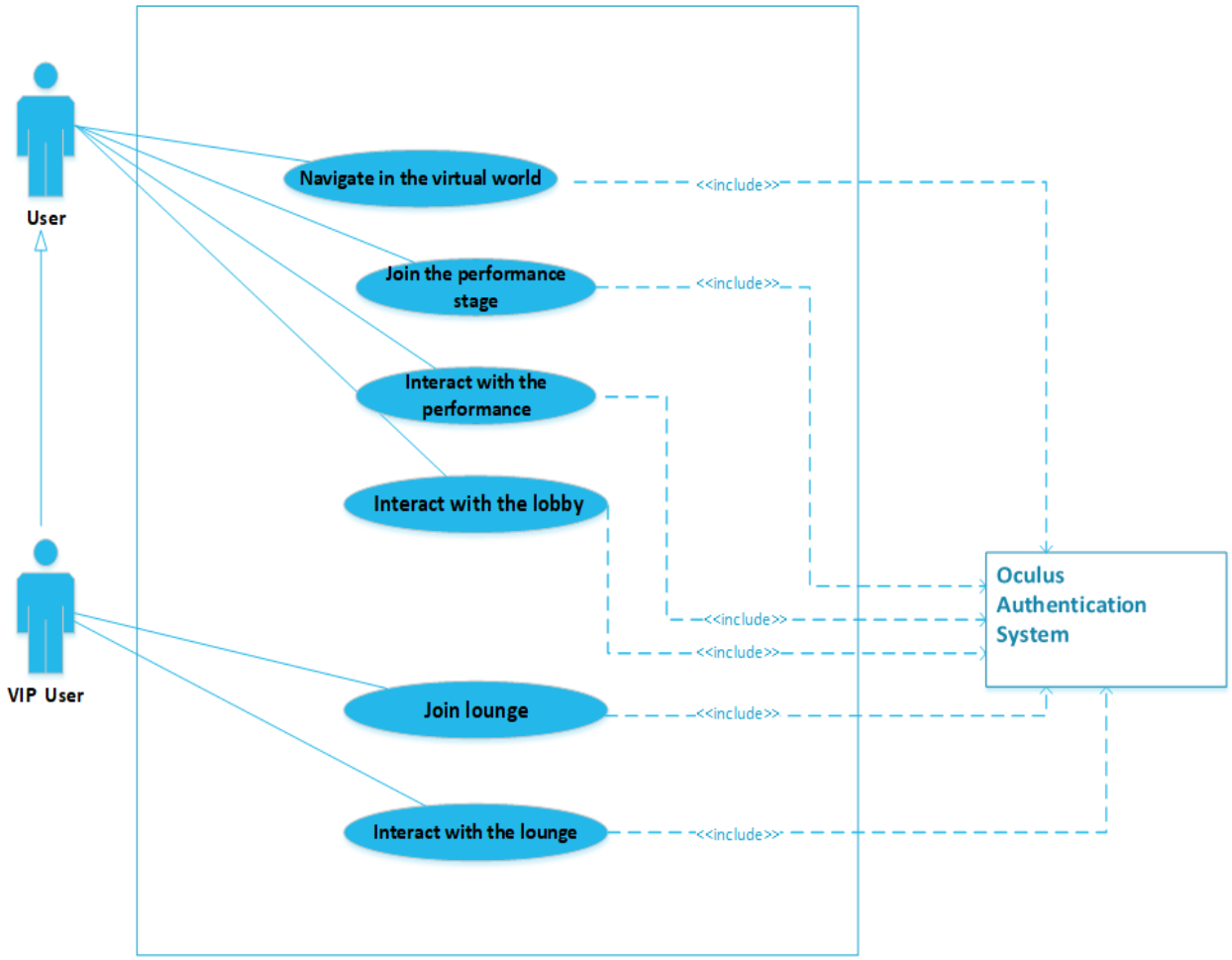

Figure 2.1: main use case diagram

2.1.2.2 Specification of non-functional needs

Non-functional requirements define the overall qualities or attributes of a system. A functional and operational application is useless if it has problems in terms of performance. Thus, the solution must meet the following standards:

- **Immersive:** The application is designed to be the most immersive and smooth possible by profiling and optimizing the application to the fullest to prevent any lag and/or FPS drop.
- **Reduced motion sickness:** The application is designed with the most motion sick-sensitive users in mind by using techniques in production to prevent most of motion sickness possible.
- **Attractive:** It is the ability of an application to be attractive to the user by the choice of graphic style chose for the 3D environment.
- **Modular implementation :** The application is divided into independent modules for a more simplified editing.

2.1.2.3 Planning of a Scrum project

The Scrum roles are divided as following:

- **Product Owner:** Mr. Ben Mariem OUSSAMA responsible for managing the Product Backlog, which includes developing and presenting the Product Goal in a clear and concise manner, creating product backlog and guarantee that the product backlog is clear and understood.
- **Scrum Master:** Mrs. Kafsi INES guarantee the method's implementation as well as ensuring that its objectives are realized and is in charge of the communication. During the several Sprints, she is in charge of eradicating any obstacles that may prevent the team and project from progressing.
- **Development team:** The team is composed of Sarah Khamassi responsible for the programming aspect of the project and Omar Fadhleoui responsible for the level art and VFX.

2.2 Product Backlog

In this part the product backlog is defined in the form of a table that carries a list of tasks to be performed to create the expected application. Please note that this backlog is evolving throughout the life of the product.This order can be changed during the project and add, modify, and even delete elements.

Feature id	Feature	User story id	User story	Priority	Estimation (days)
1	Main areas of the application	1.1	As a user , I can access and interact with the main level containing the performance stage	High	15
		1.2	As a user , I can access and interact with the lobby level.	Meduim	10
2	Online multi-player mode	2.1	As a user , I can access application online with other users	High	20
		2.1	As a user, I can see other users and their interactions.	Meduim	20

Table 2.1: Product Backlog

2.3 Sprint planning

Release	Label	ID Feature	Expected Start	Expected End
Release 1	Implementation of the main features of the application	1.1	04/04/2022	29/04/2022
Release 2	Implementation of multiplayer	1.2	02/05/2022	03/06/2022

Table 2.2: Sprint Planning table

Conclusion

In this chapter, the functional specification is modeled which has been illustrated by a use case diagram. Then the non-functional requirements have been specified. Then, ending with the description of the piloting phase of the project with SCRUM in representing the backlog and sprint planning.

Chapter 3

conception and technical choice

Introduction

After identifying the needs of the project, this chapter will be dedicated to the study conceptualization of the use cases established in the previous chapter. The graphic design will be presented first with the navigation diagram. Then, a description of the architectures on which the application is based.

3.1 Design Conception

In this section, a few sketches of the application will be displayed to get an overview of the potential application's insides, this step will give a minimalist visual overview of how the application has been imagined in the first steps of development.

The figure 3.1 shows a sketch of what the performance stage is imagined to look like. Since it is the main area where the users are supposed to spend the most time in, it is imagined with rich lights, decors, and screens to please users' eyes.

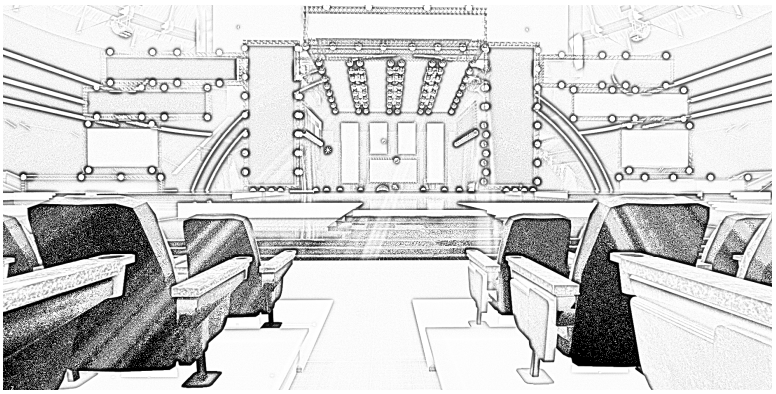

Figure 3.1: Performance Stage Sketch

The figure 3.2 shows an initial sketch of how the interaction system is imagined. Interacting with the performers is an important part of the experience and adds considerably to the user experience, thus the choice of three initial interactions to choose from: like, heart and applause.

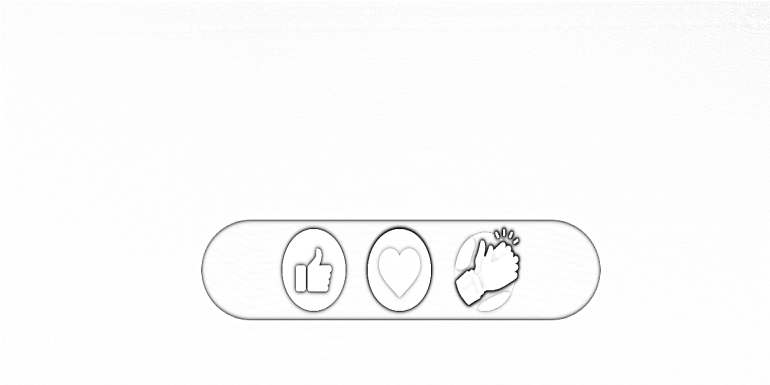

Figure 3.2: Interaction system Sketch

The figure 3.3 shows the sketch of another adequate addition to the user experience proposed in the project: the creation of virtual foam noisemakers like the existing ones in stadiums.

Figure 3.3: Noisemakers Sketch

The figure 3.4 shows the sketch of the selfie mode system proposed in the project, a feature specially created to make the experience in the application the most realistic possible.

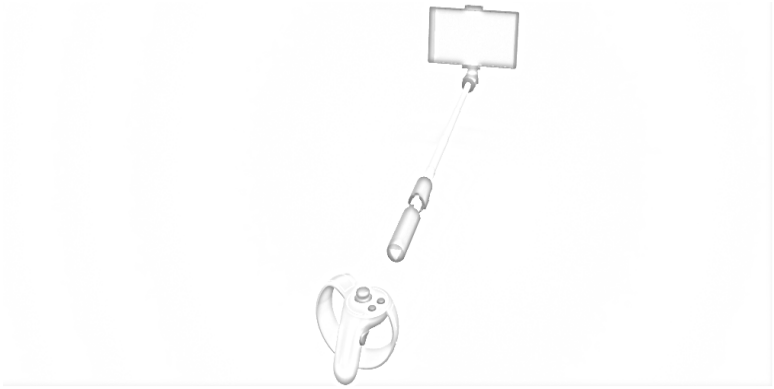

Figure 3.4: Selfie mode system Sketch

The figure 3.5 shows the sketch of how the lobby is imagined.

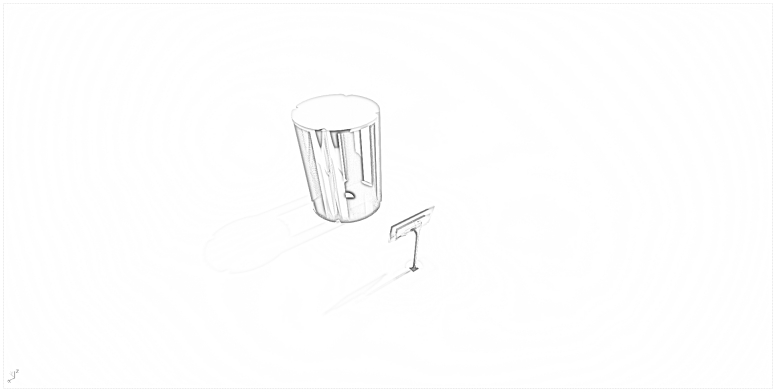

Figure 3.5: Lobby Sketch

3.2 Technical conception

One of the most crucial components of the conception process is the decision of the logical architecture. It does not describe how an application should function, but rather how it should be built to achieve these goals.

3.2.1 Physical architecture

In this part, the architecture of the system is presented :

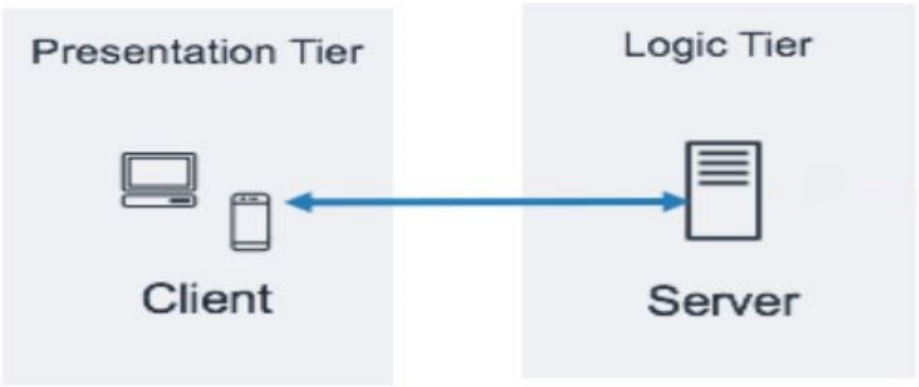

Figure 3.6: Physical architecture.

- Presentation: Its purpose is to present data and interaction to the user.
- Logic: Is responsible of providing the resources required by the user.

3.2.2 Global class diagram

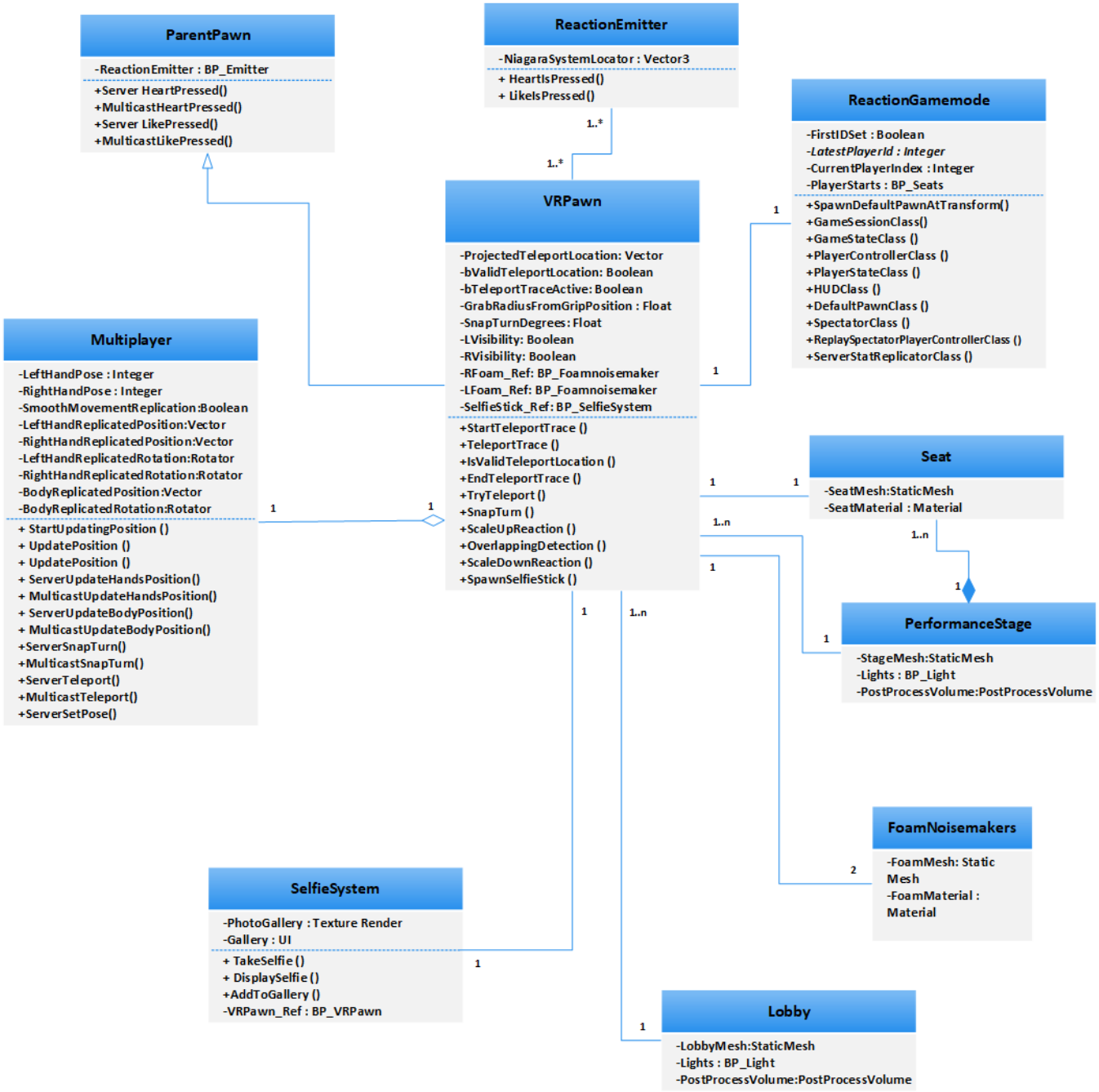

Figure 3.7: Global class diagram.

- ParentPawn: A parent class to VRPawn containing the replication logic passing it to its children.
- ReactionEmitter: A class responsible for emitting the final part of an interaction.

- ReactionGamemode: A class responsible for setting the rules of the game instance specifying the defaults.
- VRPawn: A class responsible for creating an avatar and a vr camera. it's the virtual representation of any user in the application containing all the logic needed for a smooth experience.
- Multiplayer: A class responsible for the multiplayer logic in the application.
- Seat: A class representing the user seats in the performance stage.
- Performance stage: A class representing all what the performance stage contains of components.
- Lobby: A class representing all what the lobby section contains of components.
- SelfieSystem: A class responsible for the selfie system logic.
- FoamNoisemakers: A class representing the foams used by the user.

3.3 Technological choices

In this section, the technology used is presented as well as the technological choices made during the project.

3.3.1 Technologies

- **Unreal Engine:** Unreal Engine (UE) is a 3D computer graphics game engine developed by Epic Games, first showcased in the 1998 first-person shooter game Unreal. Initially developed for PC first-person shooters, it has since been used in a variety of genres of games and has seen adoption by other industries, most notably the film and television industry. Written in C++, the Unreal Engine features a high degree of portability, supporting a wide range of desktop, mobile, console and virtual reality platforms.[5]
- **Epic Games:** Epic Games, Inc. is an American video game and software developer and publisher based in Cary, North Carolina. The company was founded by Tim Sweeney as Potomac Computer Systems in 1991, originally located in his parents' house in Potomac, Maryland.[6]
- **Oculus App:** A software desktop application is created to set up the headset, access and edit the settings and information related to the headset and serves as middle-ware between the headset and the game engine.

3.3.2 Tools

- **Visual Studio Code:** It is an extensible code editor developed by Microsoft. It allows the debugging support, syntax highlighting, smart code completion and code refactoring. Users can change the theme, hotkeys,preferences and install extensions that add additional functionality.

- **Git Hub:** It is a cloud service that helps developers store files online and track changes to the code of a software project, using the control system of open-source Git release.
- **ClickUp:** ClickUp allows team project management through a single interface that centralizes all collaborative functions.

  The tool allows creating simple task lists and more complex lists in the form of Kanban-like columns for more complex project management.
- **Latex:** It is a document composition system, it includes features for production of technical and scientific documentation. It is available as software for free.
- **Overleaf:** It is an online Latex editor, real-time collaborative.
- **Microsoft Visio:** Microsoft Visio (formerly Microsoft Office Visio) is a diagramming and vector graphics application and is part of the Microsoft Office family. [8]

3.3.3 Hardware

To achieve the application's implementation, the following hardware has been used :

3.3.3.1 Computer

- **Screens:**
  DELL 1920x1080
- **Workstation:**
  processor :12th Gen Intel(R) Core(TM) i5-12400F 2.50 GHz
  RAM: 32.0 GB
  Graphics : NVIDIA Geforece RTX2060

3.3.3.2 Headset:

- **Meta Quest 2:**
  1832x1920 per eye, 6GB of RAM and 265 GB of storage.

Figure 3.8: Meta Quest 2 picture

Conclusion

In this chapter, the graphic and technical aspect of the solution has been discussed as well ad the technological choices made for this project.

Chapter 4

Learning Unreal Engine

Introduction

In this chapter "Learning Unreal Engine" the main focus is put on the training phase consisting of learning the required video game engine software used by the host company.

The training phase will be divided on two sprints: the first sprint focus on learning the basics of the engine, and the second sprint for learning advanced knowledge.

4.1 Definitions

Before moving on to the next sections, crucial terms must be clarified:

- **Blueprint Visual Scripting:** The Blueprint Visual Scripting system in Unreal Engine is a complete gameplay scripting system based on the concept of using a node-based interface to create gameplay elements from within Unreal Editor. As with many common scripting languages, it is used to define object-oriented (OO) classes or objects in the engine. This system is extremely flexible and powerful as it provides the ability for designers to use virtually the full range of concepts and tools generally only available to programmers.[9]
- **Blueprint:** Generic term that can refer to the visual scripting system as a whole or a standard class Blueprint that defines a new class or type of Actor.[10]
- **Event:** Executable method defined through code that has no return value and do not have any output parameters. Events are the starting points for execution and, thus, have only an output exec pin. These are executed from native code.[11]
- **Custom Event:** Event defined through the Blueprint Editor that can be executed via nodes in an EventGraph.[12]
- **Pawn:** The Pawn class is the base class of all Actors that can be controlled by players or AI. A Pawn is the physical representation of a player or AI entity within the world. This not only means that the Pawn determines what the player or AI entity looks like visually, but also how it interacts with the world in terms of collisions and other physical interactions.[13]

4.2 Sprint 1

4.2.1 Sprint backlog

In this section, all the details related to the realization of the sprint 1.

- **Sprint start date:** 31/01/2022.
- **Sprint end date:** 25/02/2022.
- **Estimation:** 112 hours.
- **Measuring scale:** A day is equal to 8 hours of work from 8:30 a.m. to 12:30 p.m. and from 2:00 p.m. to 6 p.m.
- **Sprint objective:** Learn the basics of unreal engine to ensure working with ease during the programming phase.

The table 4.1 presents the list of tasks that will be carried out during this sprint and the load of work for each in number of days.

ID	Task name	Task id	Task	Estimation
3.1	Welcome to Game Development.	3.1.1	First Hour in Unreal Engine 4.	1
		3.1.2	Introducing Unreal Engine .	1
		3.1.3	Comprehending Projects and File Structure.	1
		3.1.4	Blueprints and Gameplay for Game Designers.	1
		3.1.5	Becoming an Environment Artist in Unreal.	1
3.2	Unreal Engine Kickstart for Developers.	3.2.1	World Building Kickstart.	1
		3.2.2	Rendering Kickstart.	1
		3.2.3	Animation Kickstart.	1
		3.2.4	Engine Structure Kickstart.	1
		3.2.5	Programming Kickstart.	1
		3.2.6	Blueprint Kickstart.	1
		3.2.7	Character Kickstart.	1
		3.2.8	Materials Kickstart.	1
		3.2.8	Materials Kickstart.	1
		3.2.8	Materials Kickstart.	1
		3.2.8	Materials Kickstart.	1
		3.2.8	Materials Kickstart.	1

3.3	Exploring Blueprints.	3.1.1	Blueprints - Essential Concepts.	1
		3.1.2	Interactive Material Swaps Using Blueprints.	1
		3.1.3	Making a Blueprint Product Configurator.	3
3.4	Independent Film-making with Unreal Engine.	3.1.1	Pre-Production Breakdown.	1
		3.1.2	Optimizing the Project for Real-time Performance.	1
		3.1.3	Utilizing Blueprints During Production.	1
		3.1.4	Color Correction and Post-Process Effects.	1

Table 4.1: Sprint 1 Backlog

4.2.2 Gantt diagram

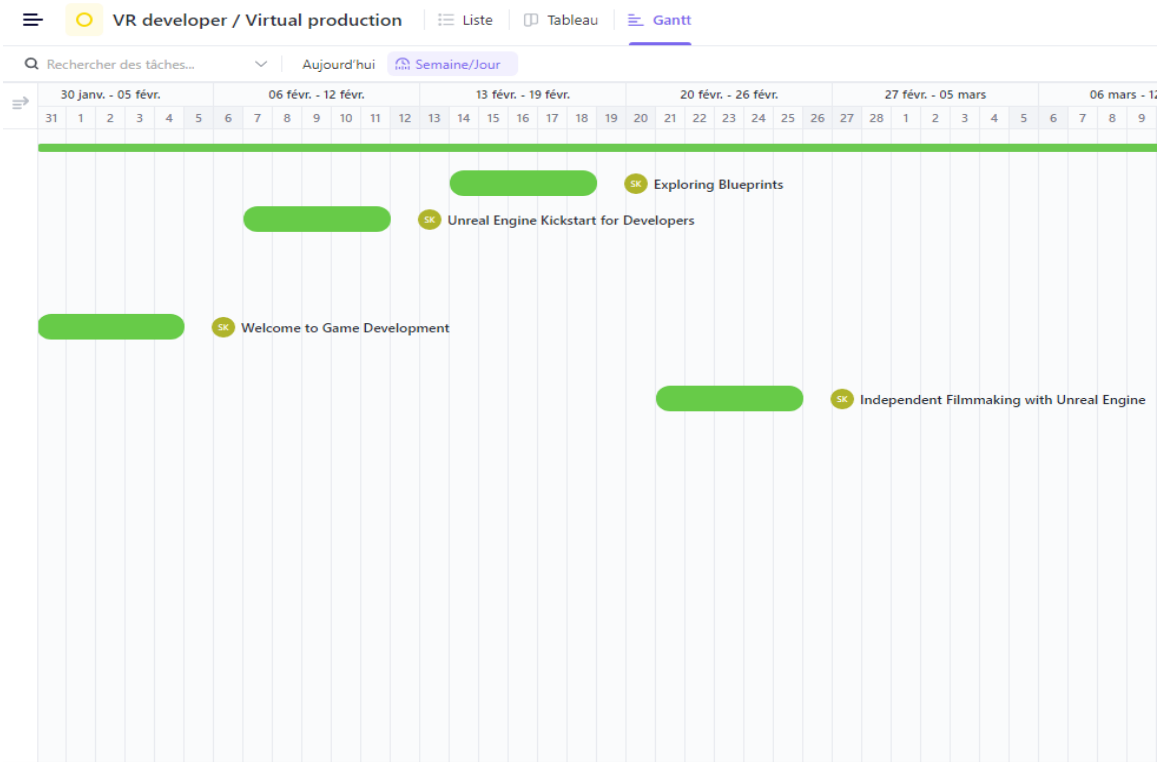

Figure 4.1: Gantt diagram of sprint 1

The figure 4.1 gives an overview of gantt diagram related to the first phase of the training.

4.2.3 Realisation

In this section, we present few screenshots of the training.

4.2.3.1 Welcome to Game Development

Give a consistent summary of the engine for newcomers who have never discovered the engine on their own: what features and opportunities it has to offer.

A robust training that paves the path and emphasizes to beginners the importance of developing with optimization in mind and understanding how the engine works.

The figure 4.2 shows how the project must be set for desired outcome from default maps and modes to input etc.

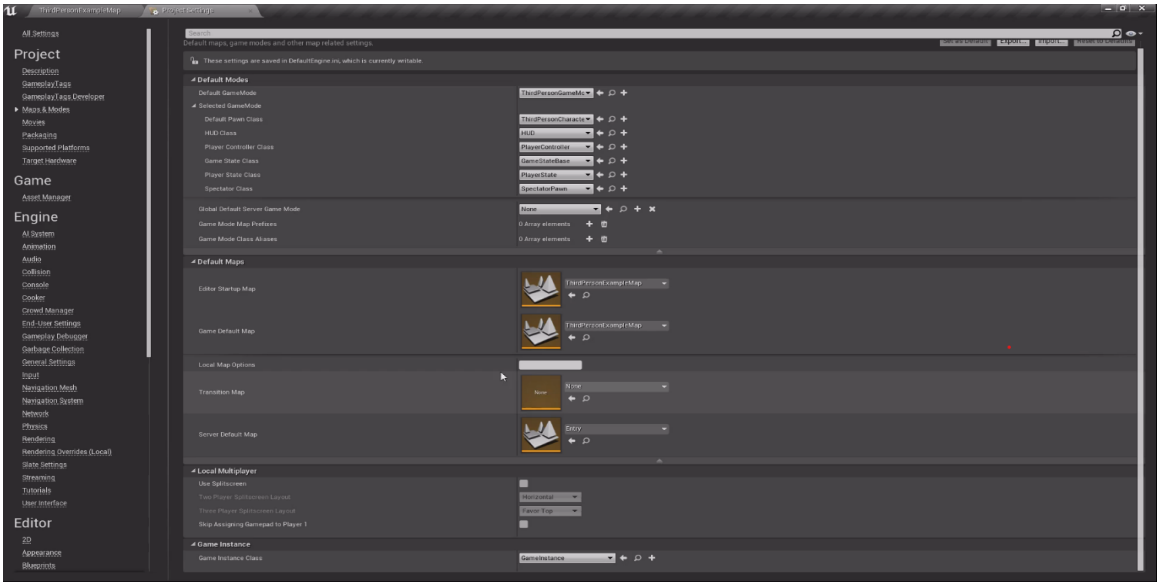

Figure 4.2: Setting the project settings image

The figure 4.3 shows the standardized project folder structure for every Unreal engine project.

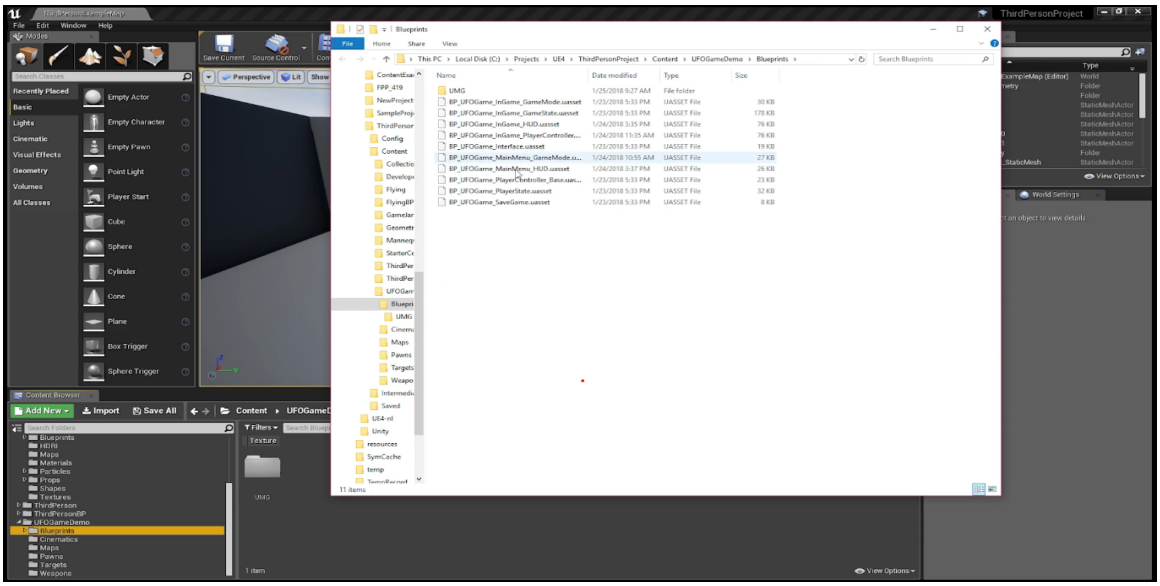

Figure 4.3: Understanding project structure image

The figure 4.4 shows some of the standard optimisation techniques used in video games : Setting up LOD.

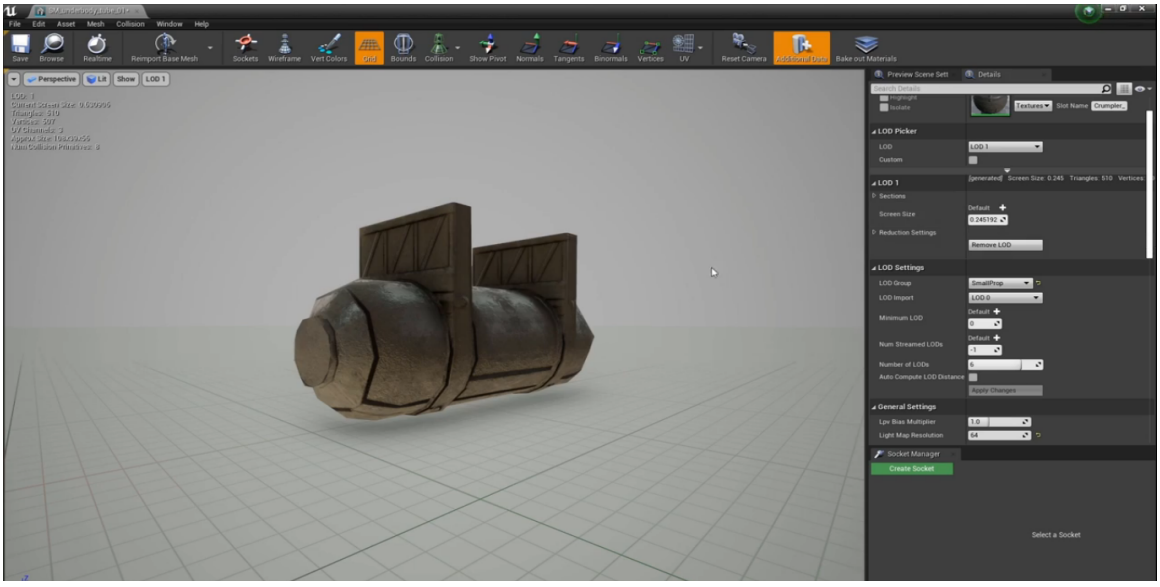

Figure 4.4: Optimisation image

4.2.3.2 Unreal Engine Kickstart for Developers

Give a more detailed description of the engine and how it operates. Learn essential theoretical knowledge from professionals is the goal.

The figure 4.5 shows the engine structure : where the assets are stored, how engine and projects folders are organised in the engine, the tools.

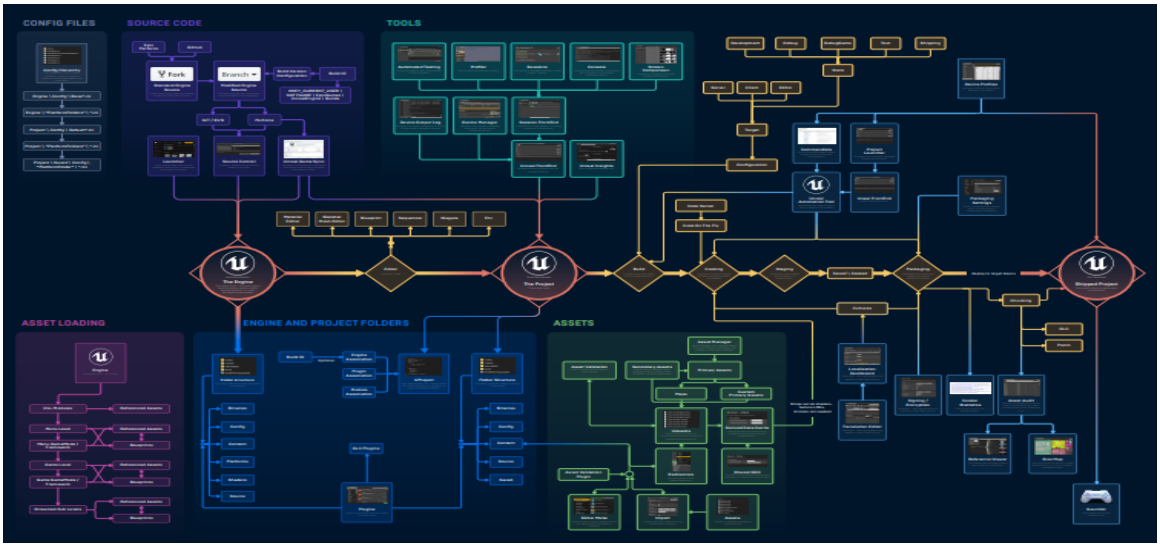

Figure 4.5: Engine structure schematic image

The figure 4.6 shows the programming schematic : the base types in the engine, tools to profile and debug, memory allocation,replication.

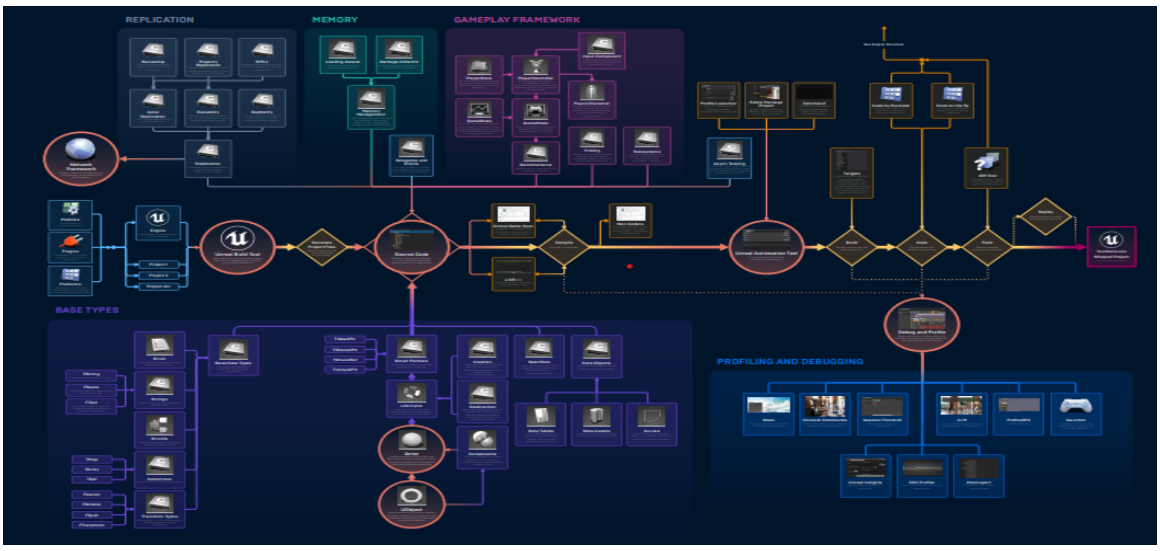

Figure 4.6: Programming schematic image

4.3 Sprint 2

4.3.1 Sprint backlog

In this section, all the details related to the realization of the sprint 2.

- **Sprint start date:** 28/02/2022.
- **Sprint end date:** 01/04/2022.
- **Estimation:** 112 hours.
- **Measuring scale:** A day is equal to 8 hours of work from 8:30 a.m. to 12:30 p.m. and from 2:00 p.m. to 6 p.m.
- **Sprint objective:** Learn more advanced skills on unreal engine to ensure the least blockage possible in future programming phase.

The table 4.2 presents the list of tasks that will be carried out during this sprint and the load of work for each in number of days.

ID	Task name	Task id	Task	Estimation
3.1	Getting Started in VR with Unreal Engine.	3.1.1	Establishing a Consistent Folder Structure.	1
		3.1.2	Supporting Various VR Platforms .	1
		3.1.3	Adding Motion Controllers.	1
		3.1.4	Getting Ready To Publish To the Oculus Store.	2
3.2	Creating Virtual Reality Walkthroughs.	3.2.1	VR Theory and Best Practices.	1
		3.2.2	Adding Locomotion.	1
		3.2.3	Adding a User Interface.	1
		3.2.4	Testing and Debugging the Scene.	1

ID	Task name	Task id	Task	Estimation
3.3	Oculus VR Production for Unreal Engine.	3.2.1	Prepare Unreal for Virtual Reality Development.	1
		3.2.2	Identify Graphics and Rendering Considerations for VR.	1
		3.2.3	Understand and Implement Locomotion and Ergonomics .	1
		3.2.4	Incorporate Handpresence and Interaction.	1
		3.2.5	Implement UI with Unreal Motion Graphics.	1
		3.2.6	Demonstrate Level Load Optimization.	1
		3.2.7	Sound Design for Virtual Reality.	1
		3.2.8	Create a Social VR Experience.	1
		3.2.9	Oculus Mixed Reality Capture.	1
		3.2.10	Tools and Principles for Project Optimization.	1
		3.2.11	Analyze Performance and Apply Common Optimizations.	1
		3.2.12	Prepare Your Oculus VR App for Submission.	1
3.4	Live Link for Performance Capture.	3.4.1	Blueprints and Live Link.	2
		3.4.2	Setting up Live Link.	2
3.5	Multiplayer Development.	3.5.1	Stealth game mini-project.	5
		3.5.2	Cooperation game mini-project.	5

Table 4.2: Sprint 2 Backlog

4.3.2 Gantt diagram

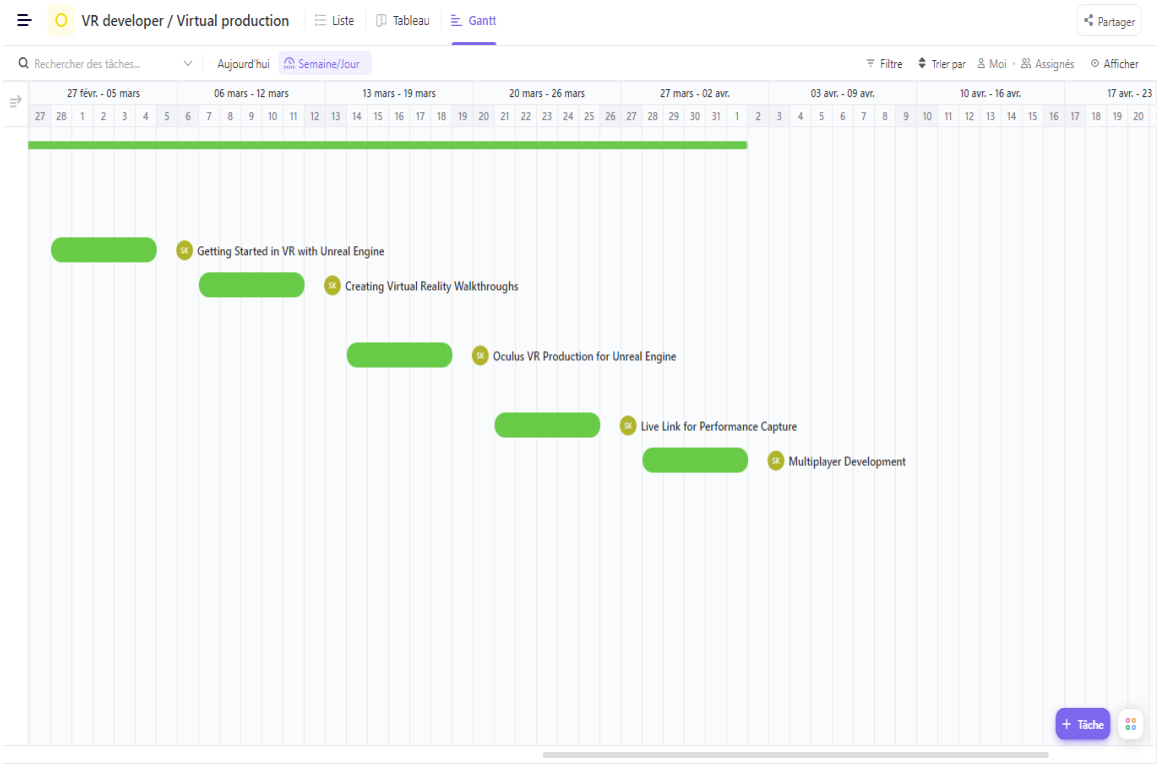

Figure 4.7: Gantt diagram of sprint 2

The figure 4.7 gives an overview of the second phase of the training.

4.3.3 Realisation

In this section, we present the different screenshots of what was learnt during the training phase.

4.3.3.1 Getting Started in VR with Unreal Engine

A good introduction to VR development in Unreal is presented and Emphasize the need of good folder organization, source control, and work sharing.

The figure 4.8 shows how to edit the document "defaultEngine.ini" in order to enter the right setting for various VR platforms support.

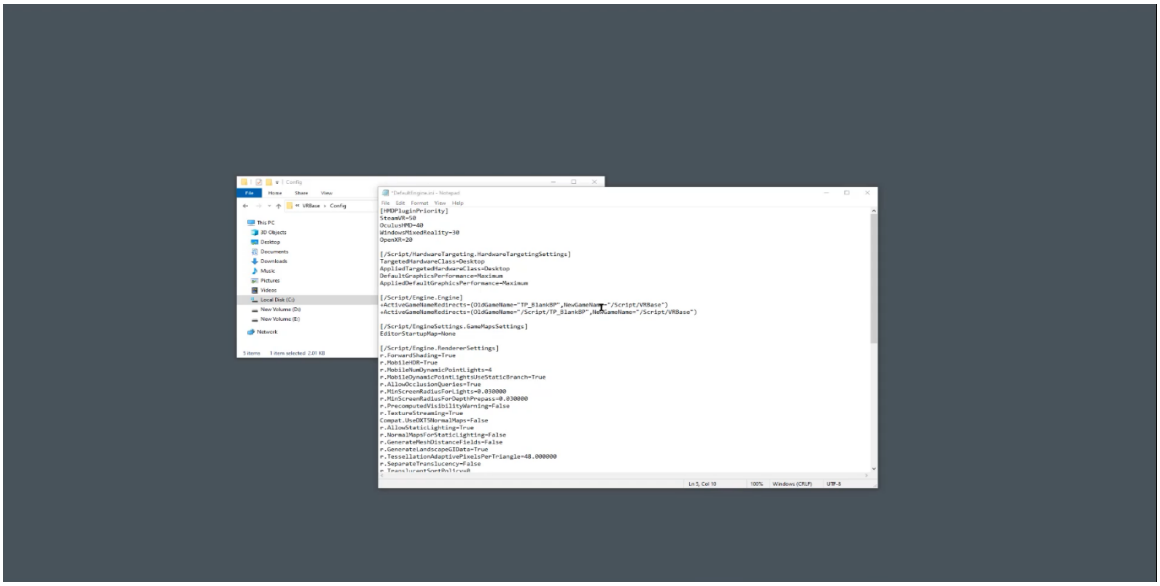

Figure 4.8: Supporting Various VR Platforms image

The figure 4.9 shows how to configure the developer oculus account online and submit an application for publication.

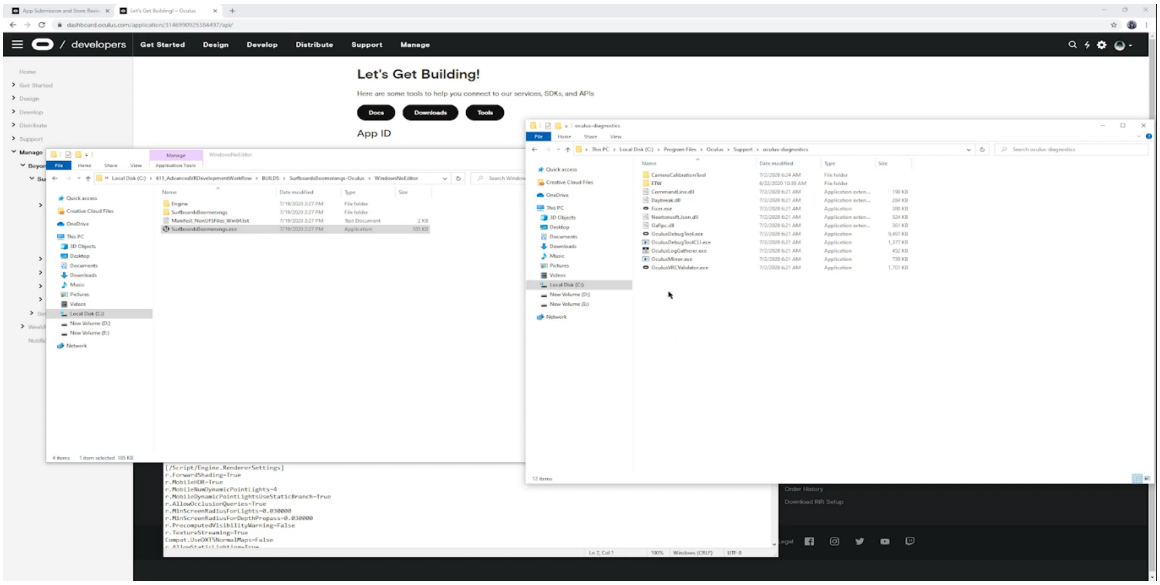

Figure 4.9: Publish To the Oculus Store image

4.3.3.2 Creating Virtual Reality Walkthroughs

The figure 4.10 presents the best practices performance wise to take into consideration when developing for virtual reality.

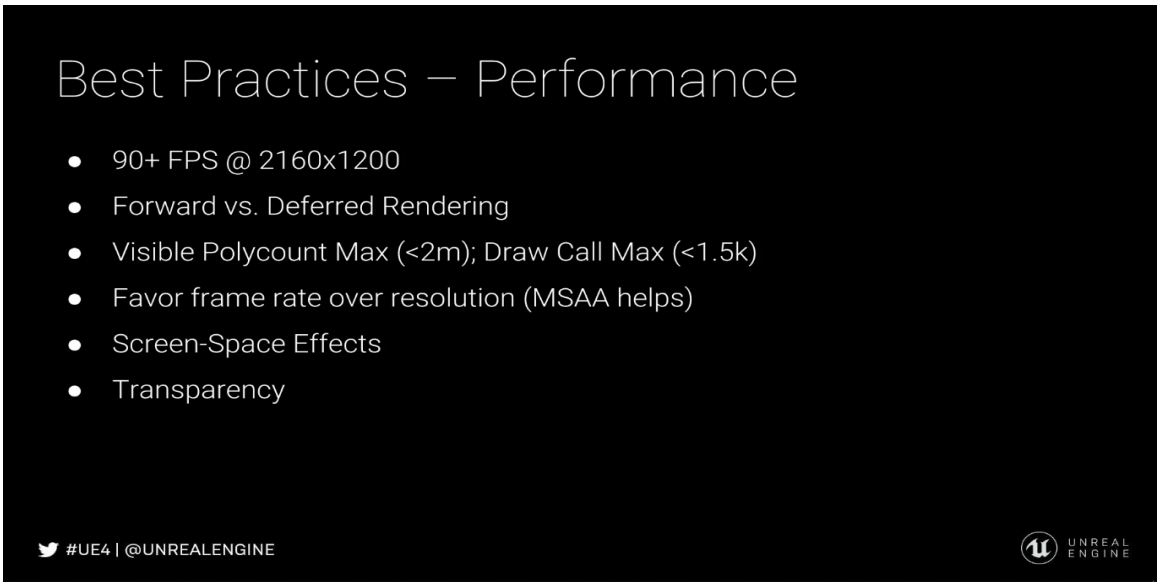

Figure 4.10: VR Theory and Best Practices image

4.3.3.3 Oculus VR Production for Unreal Engine

Give an in-depth overview on VR development and Approaches essential VR-specific knowledge.

The figure 4.11 shows the creation of the VR pawn, an essential actor when programming in virtual reality. It contains the Camera, the motions controllers, the avatar and all the logic needed to represent the player in the virtual world.

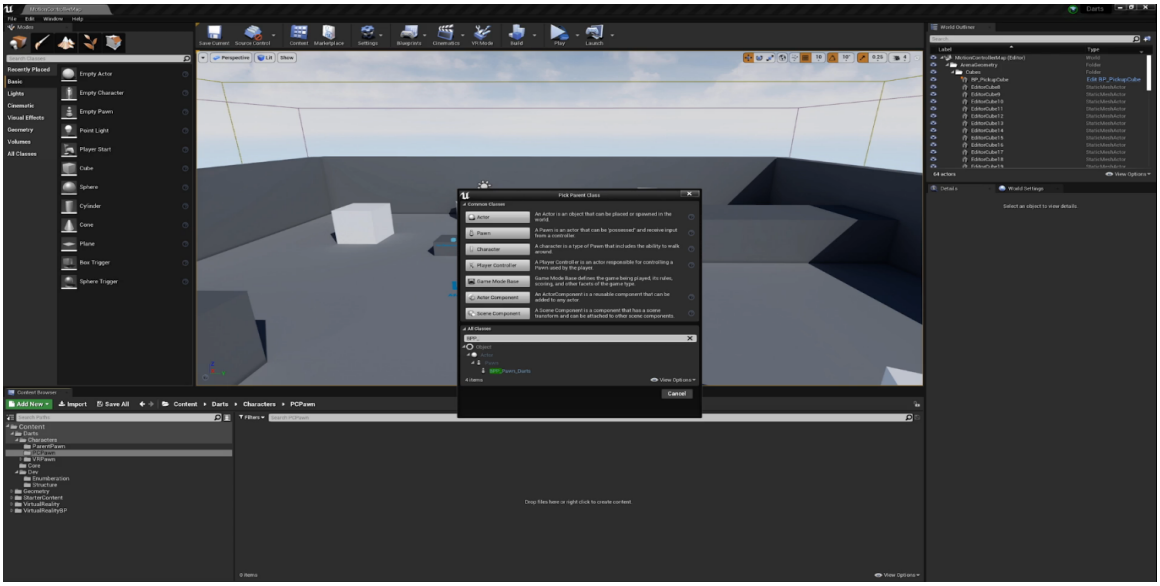

Figure 4.11: Creation VR Pawn image

The figure 4.13 shows the implementation of teleport locomotion within the VR pawn.

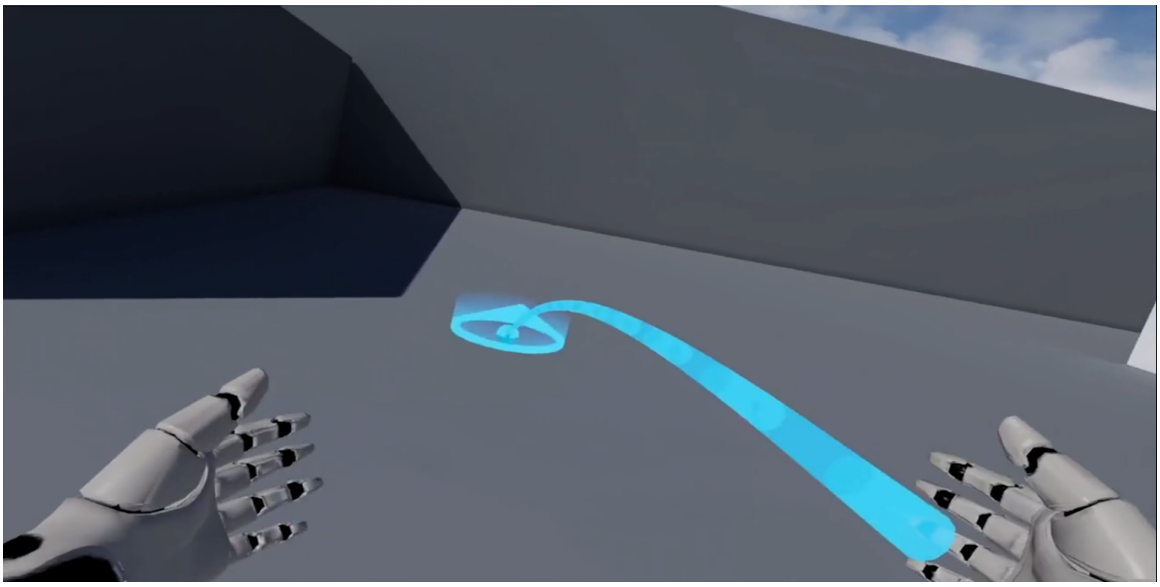

Figure 4.12: Implement Locomotion image

The figure 4.13 shows where the VR project stands in terms of its performance by considering framerate using command lines.

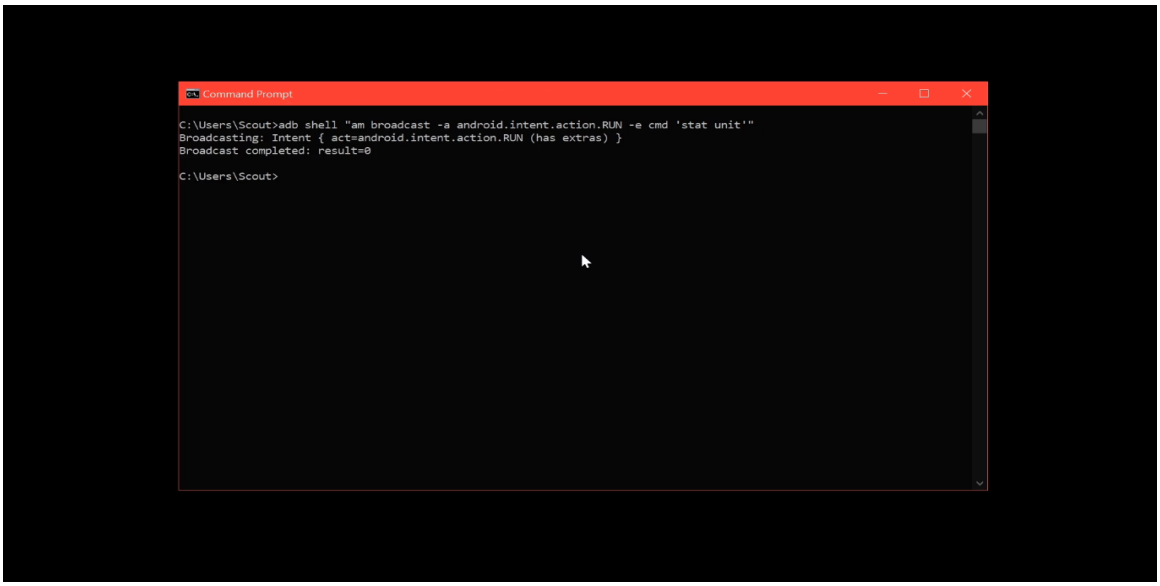

Figure 4.13: commands image

The figure 4.14 shows how to use optimization tools in editor to track perfomance issues.

Figure 4.14: Addressing performance issues image

4.3.3.4 Live Link for performance capture

The figure 4.15 shows the blueprint set up needed to link a motion capture software capturing a live performance to a character inside Unreal.

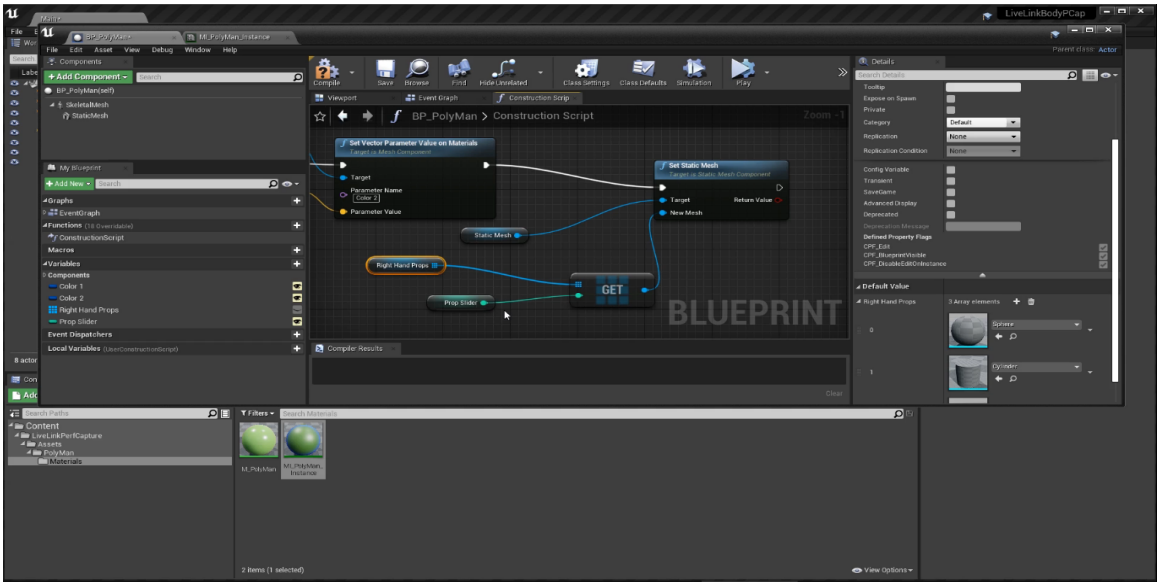

Figure 4.15: BP livelink image

4.3.3.5 Multiplayer Development

The figure 4.16 represent the final result of programming a local multiplayer stealth game with a quest to accomplish using C++ scripting.

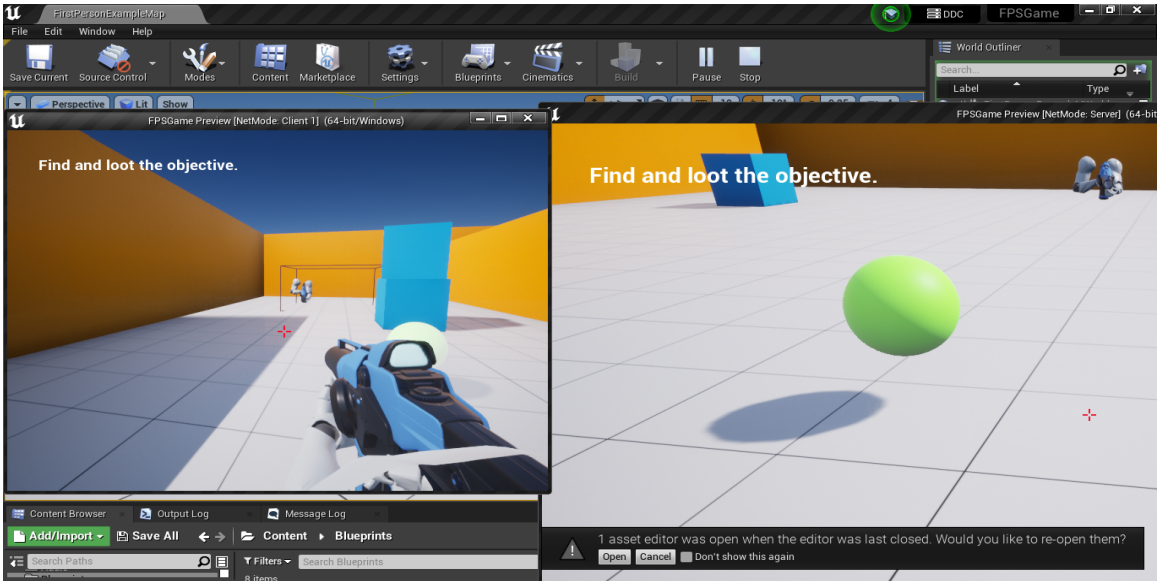

Figure 4.16: Stealth multiplayer mini-game image

The figure 4.17 represent the final result of programming a local multiplayer cooperation game with replication of the firing gun process using C++ scripting.

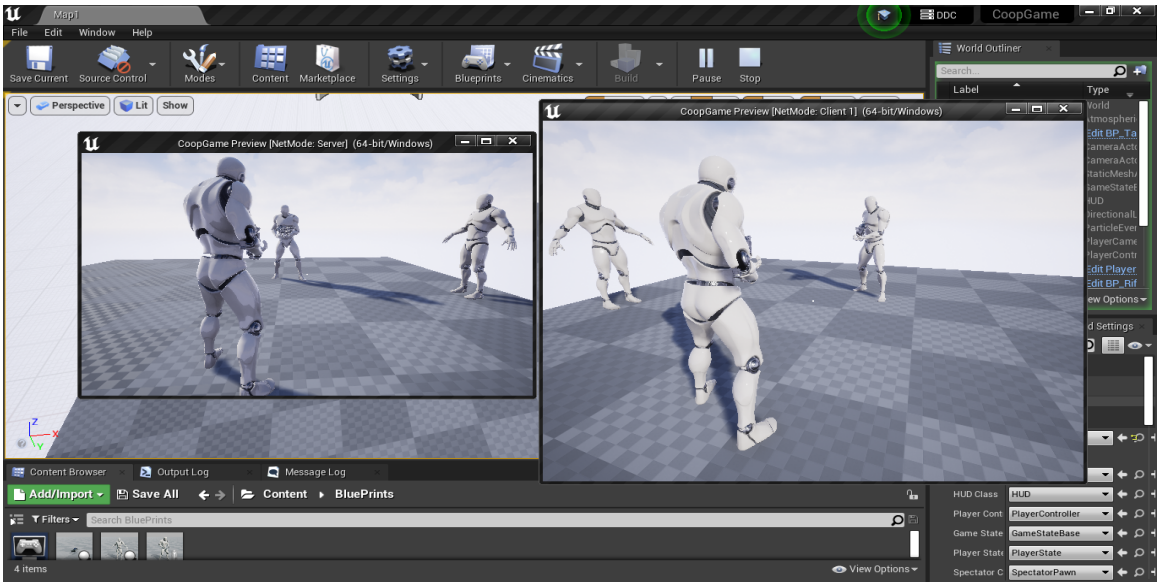

Figure 4.17: cooperation multiplayer mini-game image

Conclusion

In this chapter, the training phase was presented. This chapter is essential to the implementation of the future application. In the next chapter the training will be applied to create the wanted application.

Chapter 5

Implementation

Introduction

After specifying the functional needs and the architecture of the application, in this chapter a description of the progress of the first release entitled "Implement the main level containing the performance stage" will be presented as well as the description of the progress of the second release entitled " : Implementation multiplayer".

Each release is made of one sprint with a duration of one month.

5.1 Release 1 : Implementation of the main features of the application

5.1.1 Sprint backlog

At the beginning of each sprint, its backlog is presented to get an overview of what is included in this sprint.

User story id	User story	Task id	Task	Story point
1.1	Implement the main level containing the performance stage.	1.1.1	Implement free rotation in the world.	2
		1.1.2	Implement Interaction system .	5
		1.1.3	Implement Selfie system .	5
		1.1.4	Implement chat system.	5
		1.1.5	Integrate the performance stage environment.	2

User story id	User story	Task id	Task	Story point
1.2	Implement a lobby level.	1.2.1	Implement navigation in the virtual world.	1
		1.2.2	Implement available place consultation.	3
		1.2.3	Implement chair selection.	3
		1.2.4	Implement Join performance.	5
		1.2.5	Implement lobby interaction.	2

Table 5.1: Sprint 1 backlog

5.2 Requirements specification

This section is devoted to offer an overview of the features implemented in this sprint, as well as a textual descriptions for the use case "Interact with the performance".

5.2.1 Use case diagrams

The figure5.1 shows the use case related to the first release of the project.

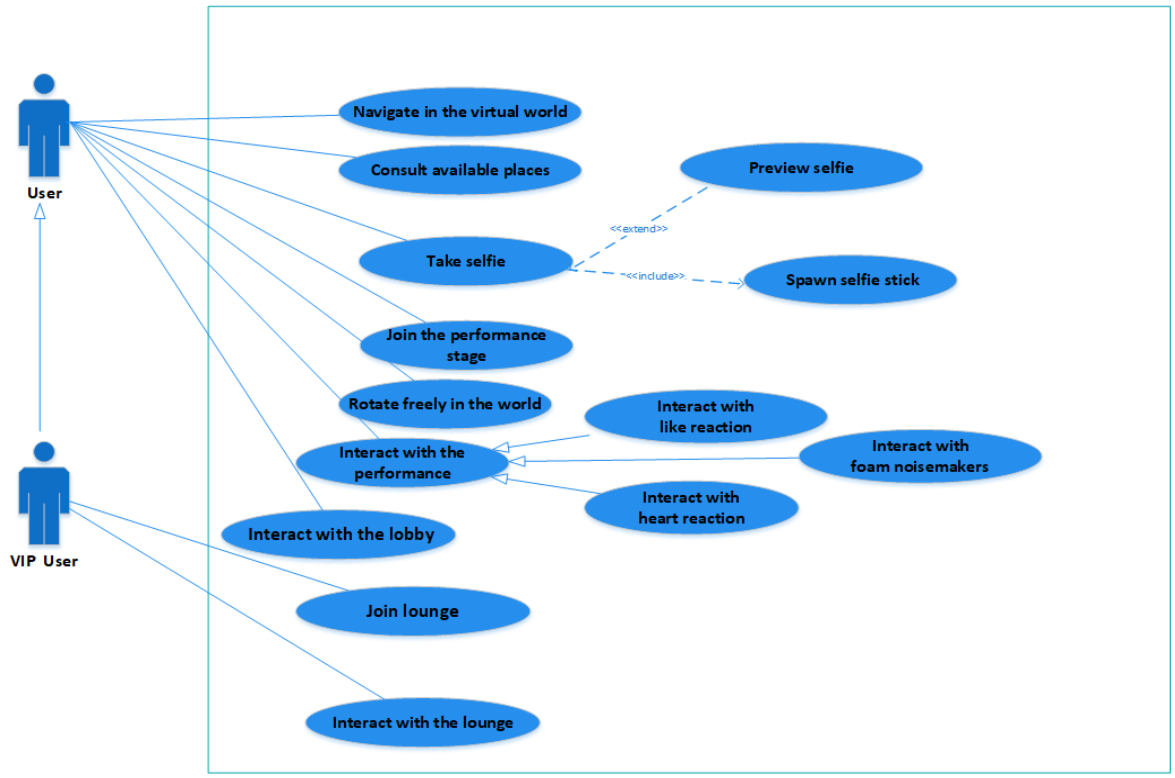

Figure 5.1: Use case of the first sprint

The figure 5.2 gives the refinement of the use case "Interact with the performance" where the user can use the listed interactions.
The list includes three types types of interactions: Interact with heart reaction, interact with like reaction and interact with foam noisemakers. To do those actions, he have to enable them with the dedicated buttons to each one of them.

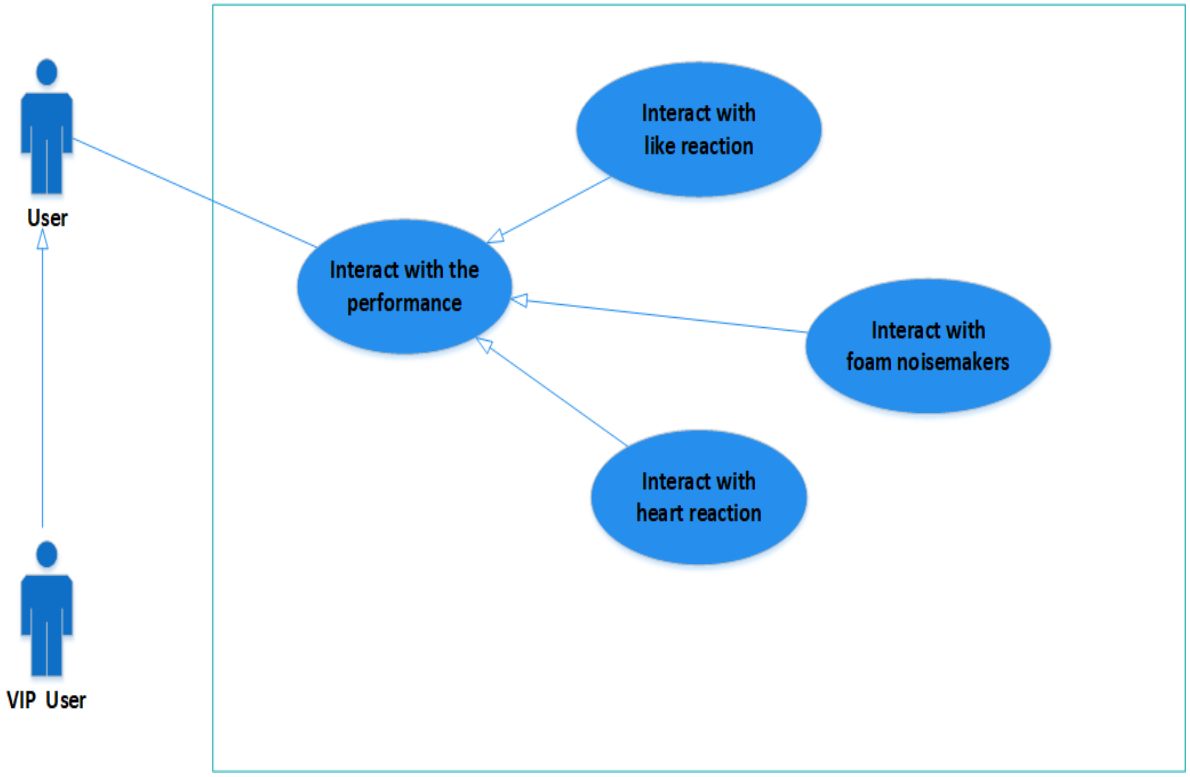

Figure 5.2: Refinement of the use case "Interact with the performance"

Since the VIP users are able to do those interactions, VIP users inherit from users.

5.2.2 Textual description

The following table give the textual description of "Interact with heart reaction"use case:

Use case	Interact with heart reaction
Description	This use case explains how a user can interact with heart reaction.
Pre-conditions	- User must be authenticated -User must be in the performance stage level.
Actor	User
Scenario	1) The user starts pressing the dedicated button. 2) The system starts generating a bubble reaction. 3) The user hold presses the button for two seconds. 4) The system increases the bubble reaction. 5) The system sends the bubble reaction with effects towards the stage. 6) The system enables heart emitters placed around the stage.
Exceptions	- The user releases the button before finishing the reaction process : in this case the bubble generated decreases and finish by disappearing.
Post-conditions	The desired reaction is executed and displayed in the player's headset.

Table 5.2: Interact with heart reaction textual description

The following table give the textual description of "Interact with foam noisemakers " use case:

Use case	Interact with foam noisemakers
Description	This use case explains how a user can interact with foam noisemakers.
Pre-conditions	- User must be authenticated -User must be in the performance stage level.
Actor	User
Scenario	1) The user hold presses the dedicated buttons on each controller . 2) The system instantiate a noisemakers on top of each controller . 3) The user can tap the noisemakers on each other. 4) The system enables a noisemakers sound.
Exceptions	- The user releases the dedicated buttons before using the noisemakers resulting in them to disappear.
Post-conditions	The desired reaction is executed.

Table 5.3: Interact with foam noisemakers textual description

5.2.3 Sequence diagram

The figure 5.3 is a detailed sequence diagram for the use case " Interact with heart reaction": The user interacts with the system through the Meta Quest motion controller by pressing a button.

The system get the signal that a specific button got pressed and responds the user with the corresponding interaction.Then , two situations are possible :
1- The user keeps pressing the button and as a results sees the remaining of the interaction displayed to him.
2- The user releases the button and puts an end to the remaining interaction process.

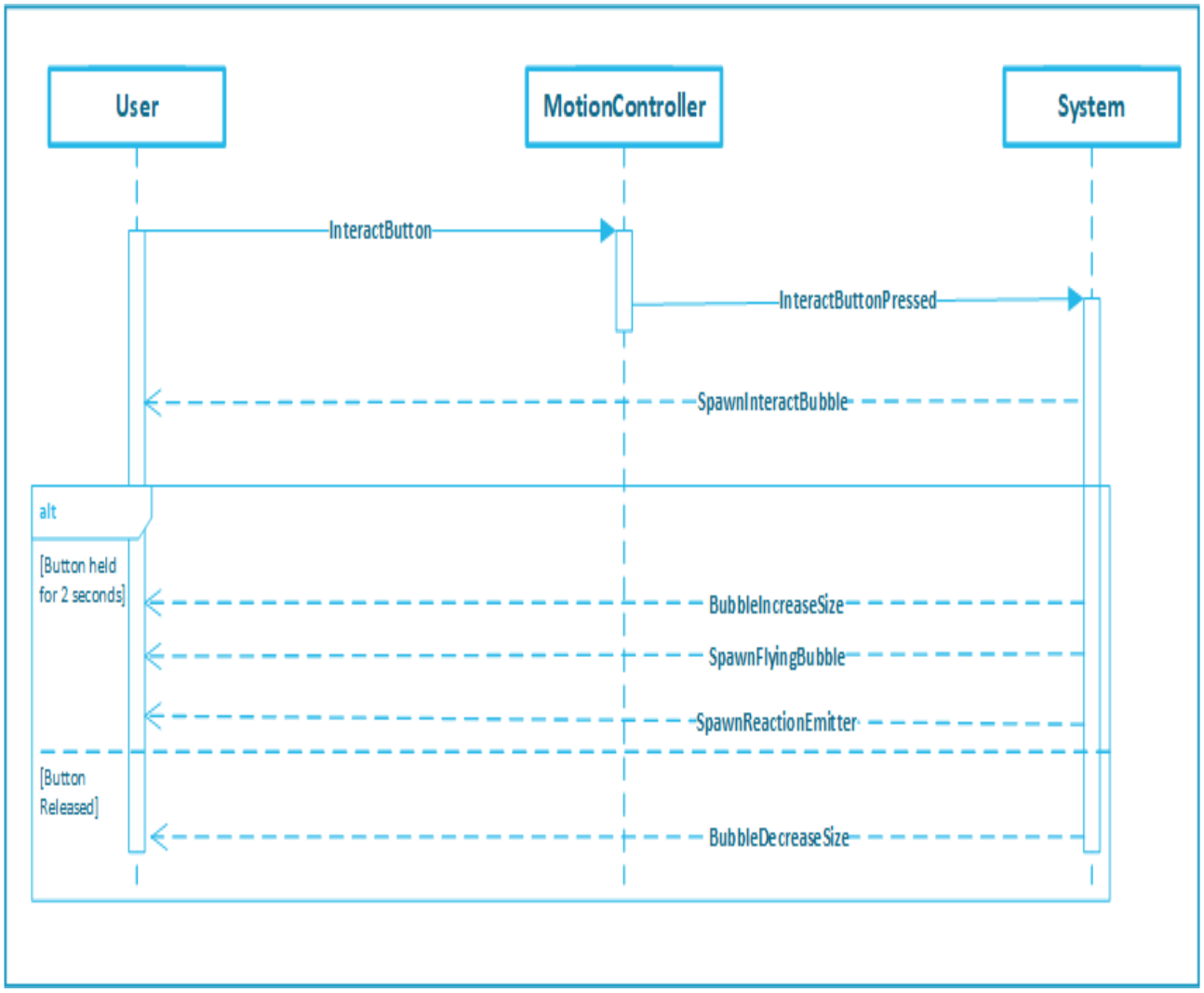

Figure 5.3: Interact with heart reaction sequence diagram

5.3 Realisation

The following figure 5.4 shows the distribution of tasks and the progress during the first sprint.

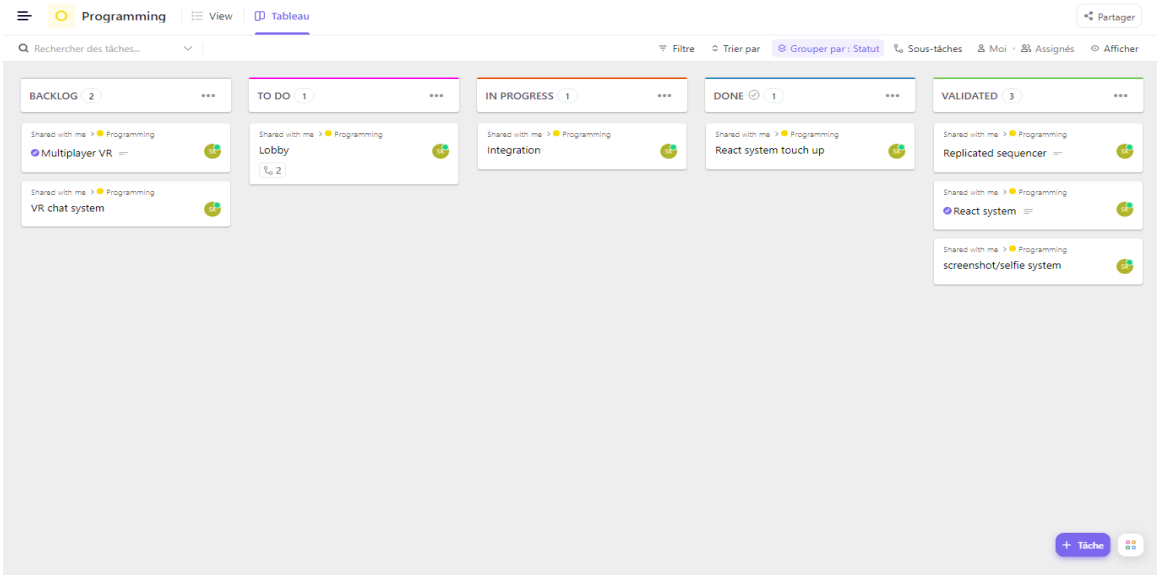

Figure 5.4: Scrum table of the project

1. **Implement free rotation in the world:**
   The figure 5.5 shows the VR avatar of a user rotating freely on the performance stage seat.

Figure 5.5: VR Avatar

2. **Implement Interaction system:**
   The figure 5.6 and figure 5.7 shows the user interface displayed to the user. From this interface the user can choose from three interactions to interact with by pressing on it.

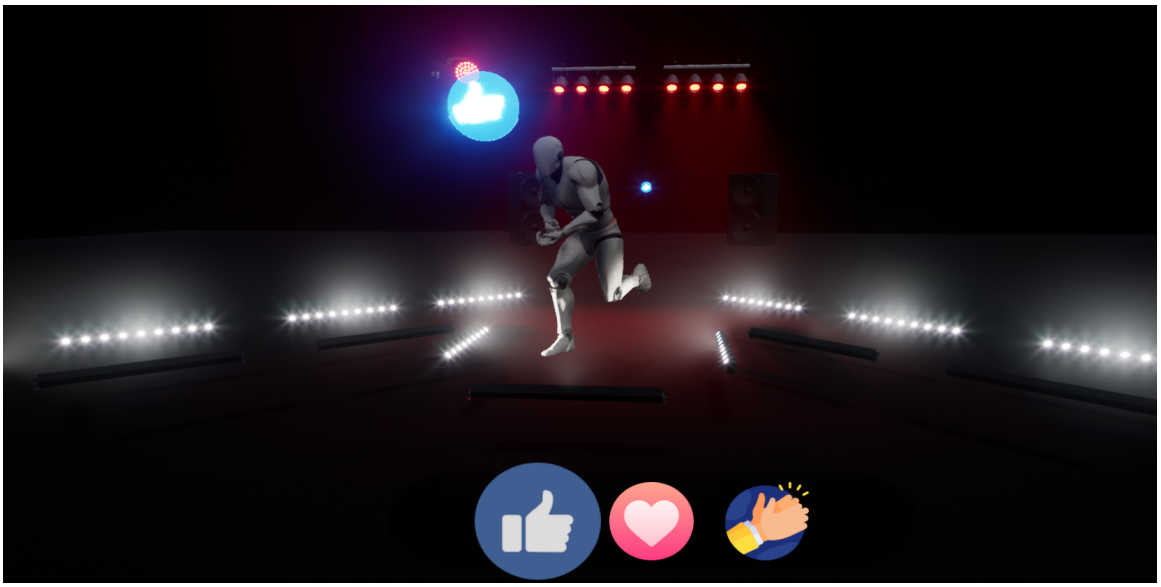

Figure 5.6: Like reaction with user interface

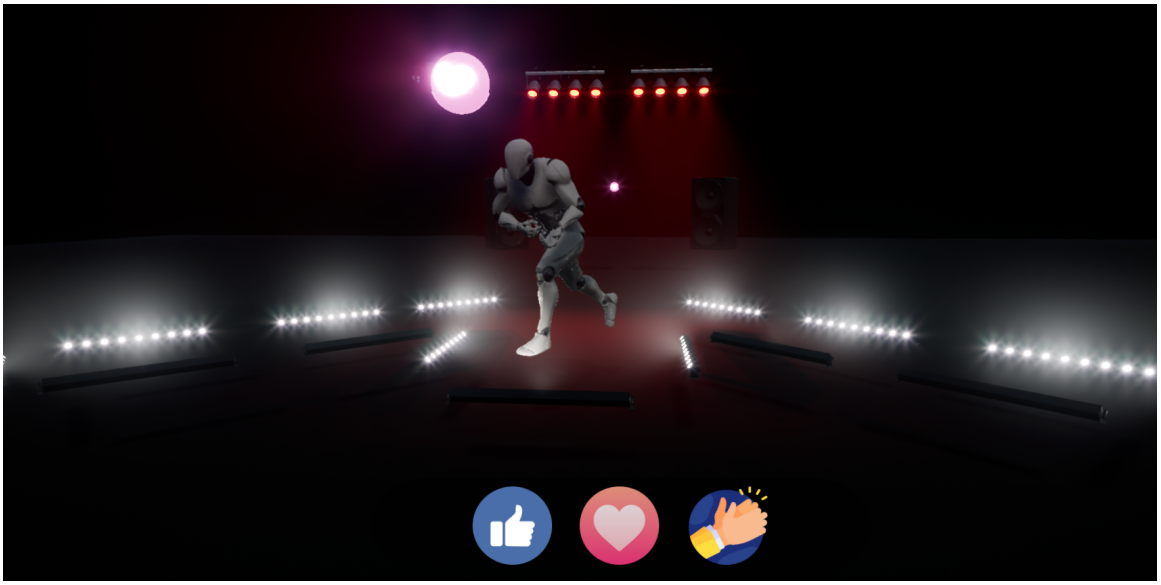

Figure 5.7: Heart reaction with user interface

With intuitiveness in mind, the interaction has been reviewed and reinvented in a more intuitive and easier way for the user and the interactions been attributed a dedicated button for fast access.

Figure 5.8: Heart reaction with motion controllers

Figure 5.9: Like reaction with motion controllers

3. **Implement Selfie system:**
   The figure 5.10 presents the virtual selfie stick that give the user the possibility to see the selfie overview, take the selfie, and display the final selfie to the user.

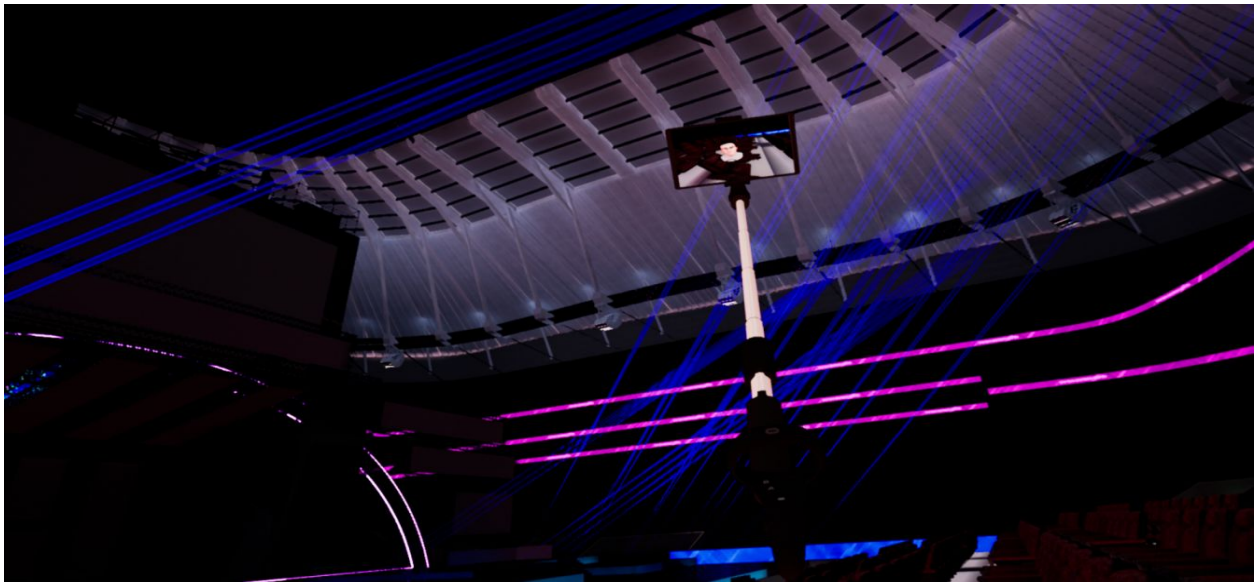

Figure 5.10: The virtual selfie stick

4. **Integrate the performance stage environment:**
   The following figures shows the cohesive environment integration in the project.

Figure 5.11: Performance stage

Figure 5.12: The lounge

Figure 5.13: Stage view

Figure 5.14: The lobby

Figure 5.15: available place consultation

5.4 Release 2 : Implementation multiplayer

5.4.1 Sprint backlog

At the beginning of each sprint, its backlog is presented to get an overview of what is included in this sprint.

User story id	User story	Task id	Task	Story point
2.1	Implement multiplayer.	2.2.1	Implement multiplayer system.	14
		2.2.2	Implement replication .	14

5.5 Requirements specification

This section is dedicated to provide a basic overview of the functions that have been built during this sprint.

5.5.1 Use case diagram

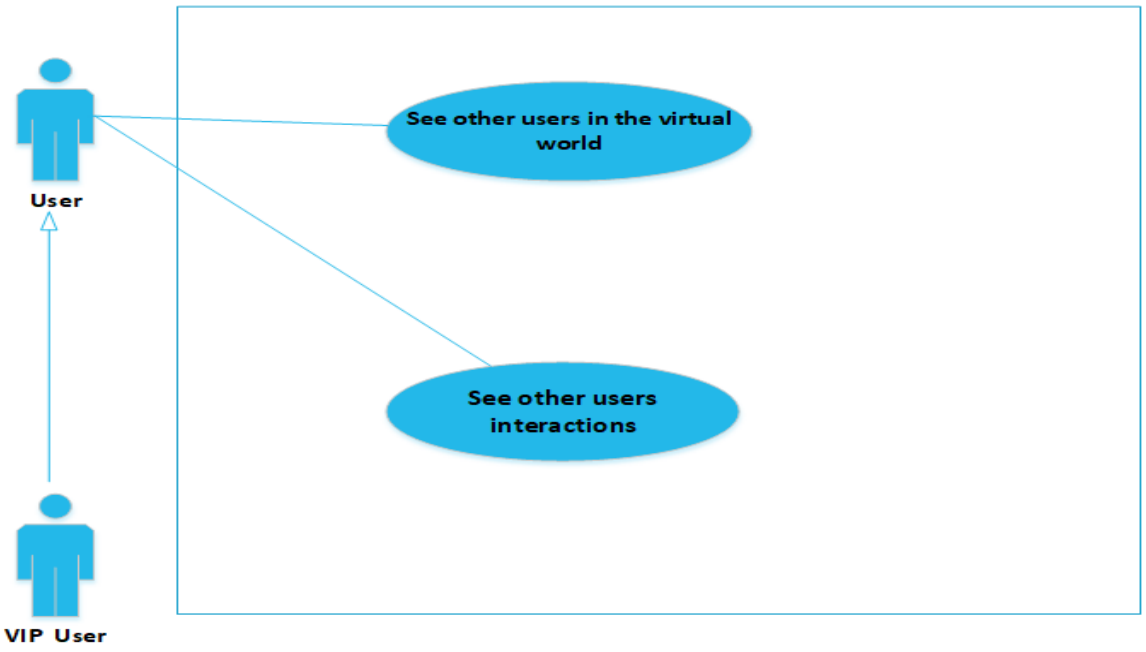

Figure 5.16: Use case of the second sprint.

The figure 5.16 gives the use case of the second sprint. This sprint focus on building one of the pillars of the application : the ability to access the virtual world in multi-player and being able to see other users' interactions.
Since any user should be able to perform those user cases, the VIP users naturally inherit them for normal users.

5.5.2 Textual description

The following tables give the textual description of "See other users interactions" use case:

Use case	See other users interactions
Description	This use case explains how a user can see other users' interactions to the performance
Pre-conditions	- user must be authenticated - User must be in the performance stage level.
Actor	User
Scenario	1) The user enables an interaction . 2) The system plays the interaction locally. 3) The system checks if the user is a server host or a client. 4) The system replicates the interaction to other players if the user is also the server. 5) The system calls the server and checks if the client is allowed to make this interaction. 7) If allowed, the server calls the multicast function. 8) The system multicast function is allowed to replicate its interaction to other users.
Exceptions	- The client is not allowed to make this action.
Post-conditions	The interaction is multicasted.

Table 5.4: See other users interactions textual description

5.5.3 Sequence object diagram

The figure 5.17 is a detailed sequence diagram for the use case " See other users interactions":

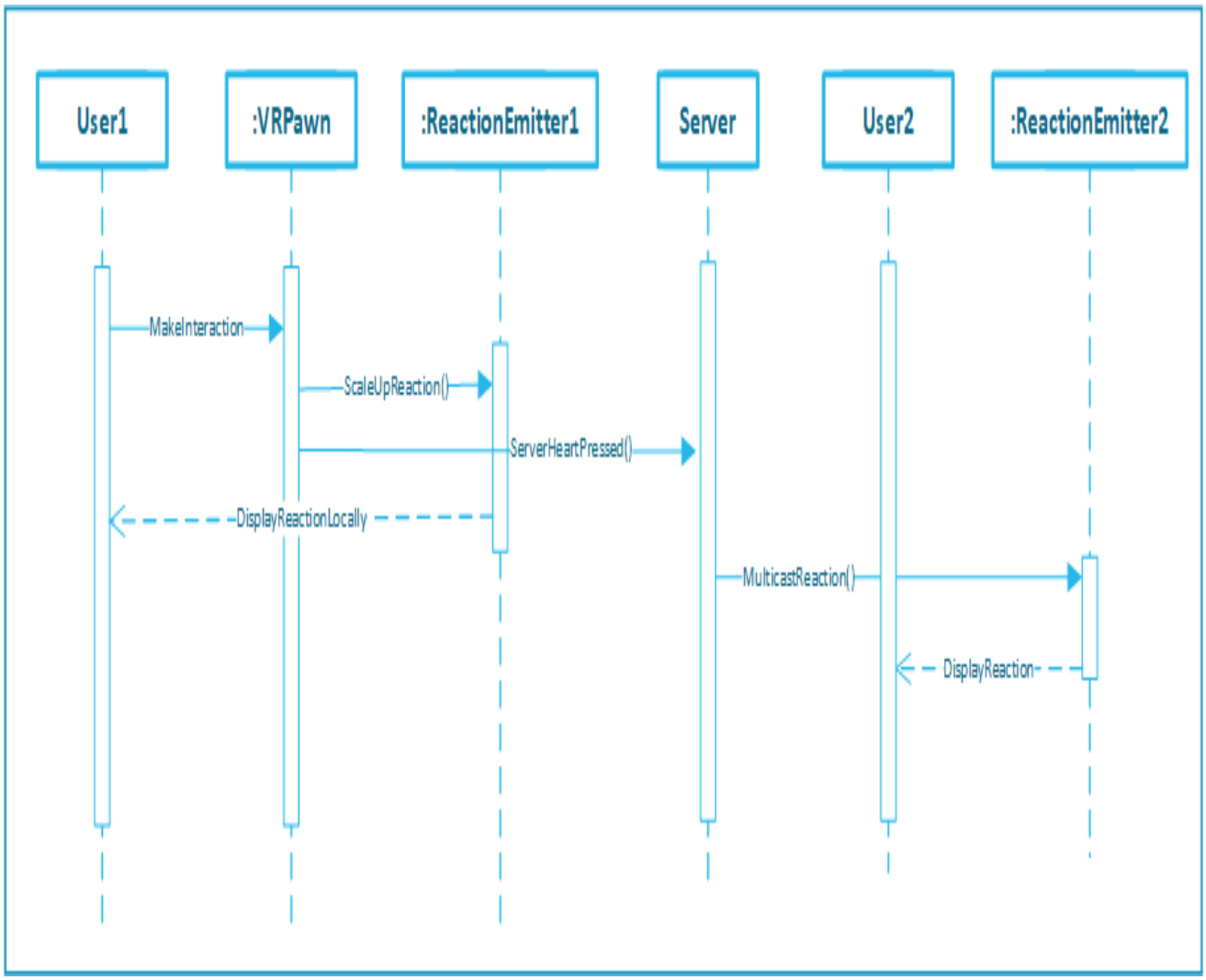

Figure 5.17: See other users interactions sequence object diagram

When user enables the interaction, VRPawn fires the ScaleUpReaction() and by results enables the Reaction emitter.As a response the Reaction Emitter display it effects locally. The VR Pawn also send to the server that the Heart Reaction has been fired.

The server then Handles the multicast logic and make sure every connected client can see that the reaction has been fired by that user.

5.6 Realisation

Multiplayer games on Unreal Engine demand the synchronization of massive volumes of data among clients.

When it comes to providing a captivating experience for users, recognizing what information to transmit and how to provide it is critical since it may have a significant impact on how the project behaves and looks. Replication is the name given to the process of syncing data and procedure calls between clients and servers in Unreal Engine.

The Replication system gives a higher level of abstraction along with low-level modification to make it more manageable to handle the numerous situations that arise while building a project for multiple simultaneous users.

5.6.1 Server client model

Unreal Engine allows developers to create a client-server network multiplayer game. One computer on the network serves as a server and hosts a multiplayer game session, as all of the other users connect as clients.

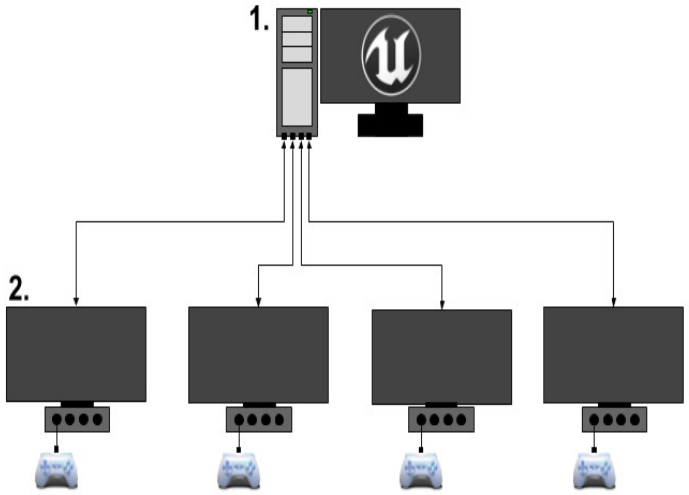

Figure 5.18: Network multiplayer schematic

The designed server then communicates with each connected client, sharing game state information and allowing them to communicate.

As the game's host, the server maintains the only real, authoritative game state: the host is where the multiplayer game takes place.

The clients each have their own Pawns on the server that they control remotely by issuing procedure calls to them to make them execute in-game behaviors.

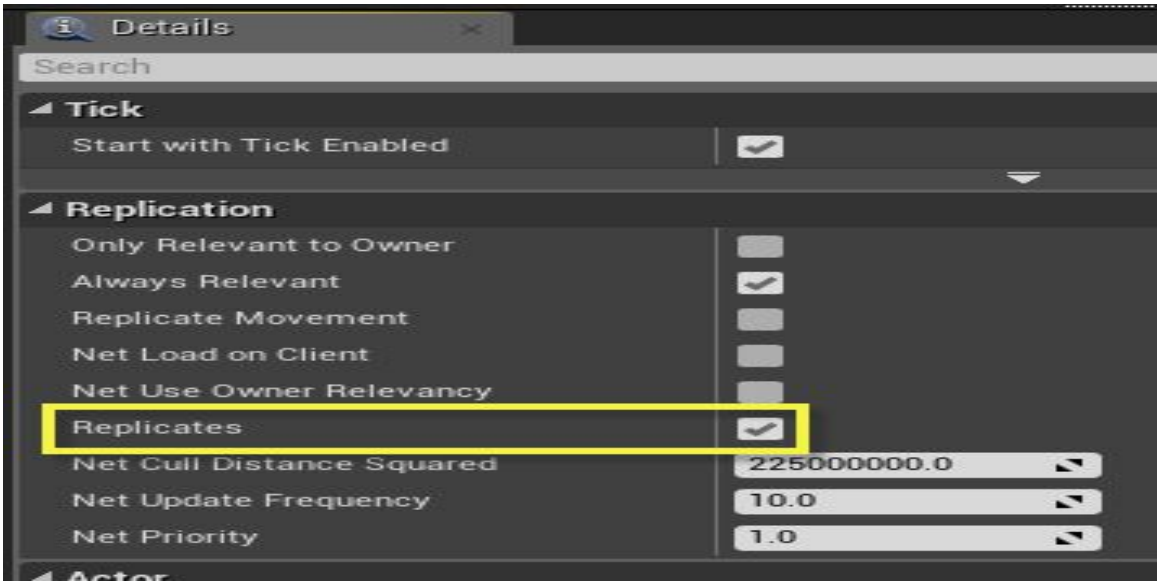

Figure 5.19: Replicates flag screenshot

The figure 5.19 shows When the "Replicates" flag on an actor is set to true, it will be instantly synchronized from the server to clients connected to that server.

It's worth noting that some functionalities aren't replicated and instead run independently on each client. The variables that control these actors and components are replicated to ensures that all clients have the same information and so replicates them in a similar manner.

1. **Implement multiplayer system:**
   The figure 5.20 represents the server client logic by showing how the server updates the body position for other users in real time.

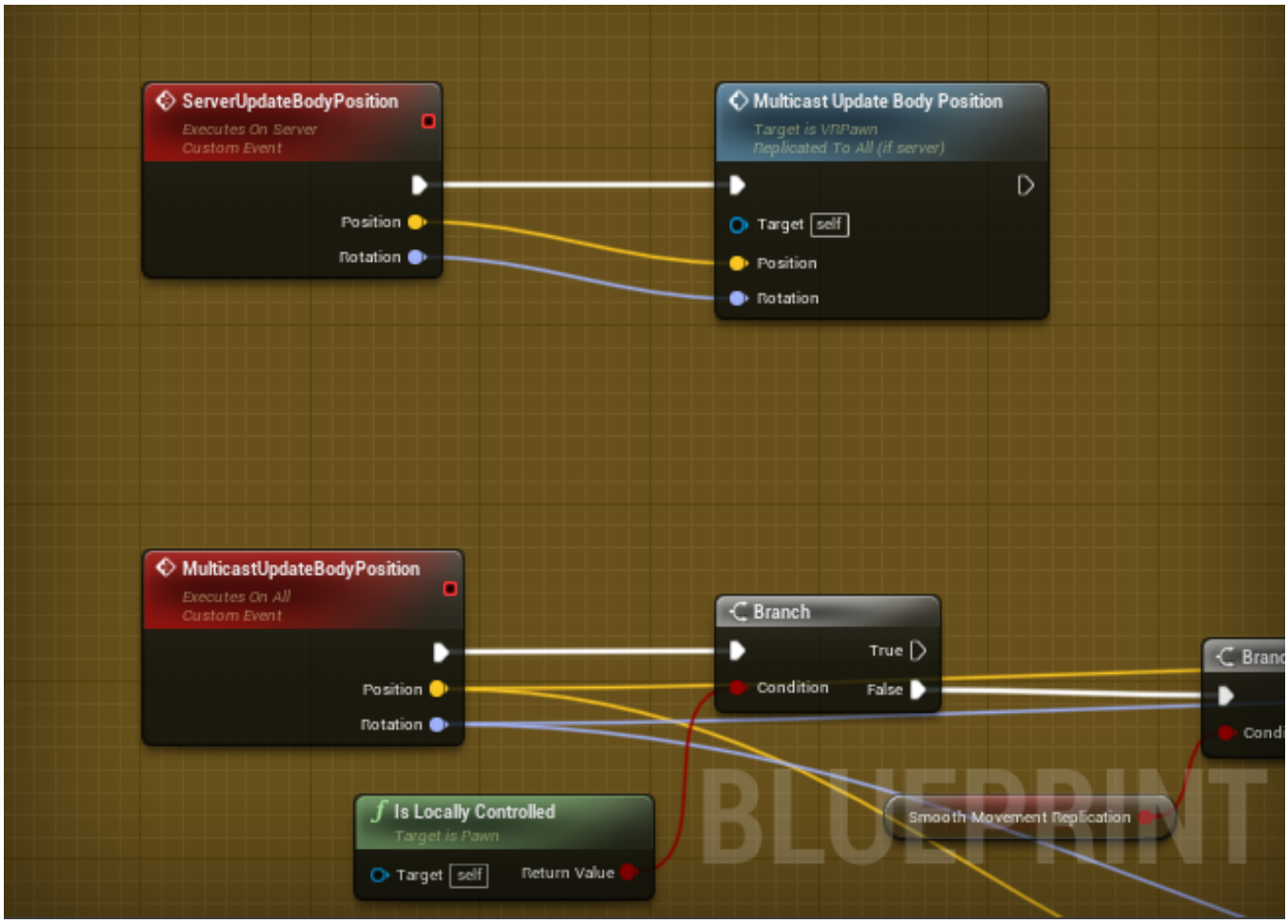

Figure 5.20: Implement multiplayer system

5.6.2 Multicast

In Unreal Engine the execution of events across the clients and the server with Blueprints is possible in addition to replicating actors and their variables.

Multicast, Run on Server, and Run on Owning Client are the three main types of replicated functions.

Multicast functions are called on the server, executed there, and then transmitted to clients automatically. Clients call server functions, which are subsequently exclusively executed on the server. The server calls client functions, which are subsequently only executed on the client that owns it.

2. **Implement replication:**
   The figure 5.21 shows how the server is multicasting the heart reaction.

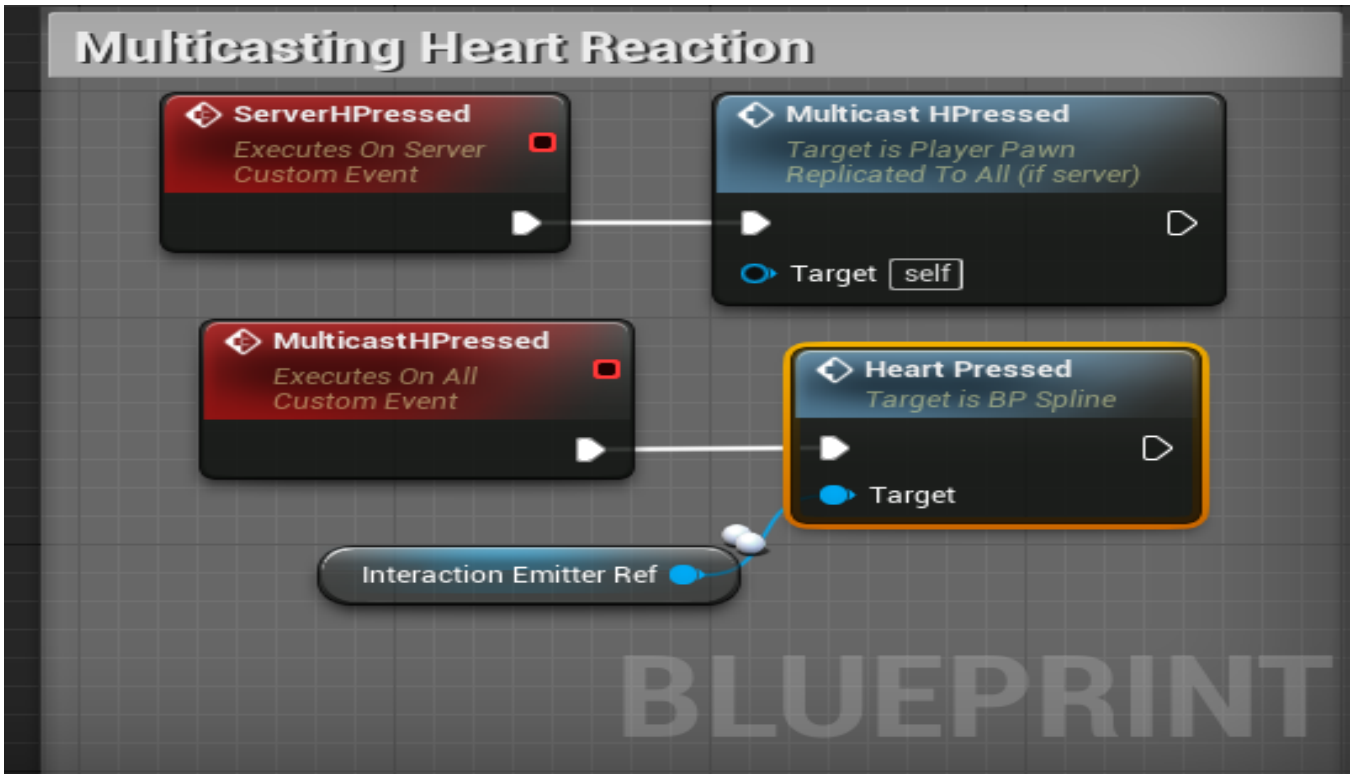

Figure 5.21: Implement replication

A client in Unreal's client-server approach isn't directly connected to any other clients; instead, it's only connected to the server. As a result, a client cannot deliver a multicast event to other clients and must instead speak with the server. imitating this behavior is however possible by using the right combination of settings of variable and

behaviour set up.

Therefore, the multicast event's implementation would run the desired logic for all connected players.

Conclusion

In this chapter, we have managed to present the of two sprints worth of development, presenting their use cases, sequence diagrams and their realisations.

General Conclusion

THis document is a detailed presentation of the work carried out over 4 months within the company "LanternsSutdios". It is part of the end-of-study project for obtaining the computer engineering degree at the Higher Institute of Arts and Multimedia of Manouba.
The main objective of this project is to develop a virtual concert application. The interest of this application is to transform the traditional world of concerts and performances into an unforgettable digital experience that can only be experienced in virtual reality.
To complete this project, it went through a succession of well-studied steps.

The internship began by placing the project in its general context, in order to identify the various useful points throughout the working process. Then, a period of time was set aside to do an in-depth study. This project has required a major effort for this phase, in order to study all the aspects around the integration of new virtual reality technologies in this sector.

This part was followed by an in-depth study of the functional and non-functional needs of the solution. In the light of this study, the conceptual study allowed to guarantee the passage from the specification stage to the realization of the application. Based on the Scrum methodology, the project was divided into tasks that were prioritized and scheduled. This step then led to planning the start and end dates of each task.

To conclude, this end-of-studies project made it possible to acquire skills in project design and planning. Similarly, it presented an ideal opportunity to learn how to develop with Unreal.

The developed product currently is only a first version testing and certainly the work does not stop there level, but it will undergo changes in the future and additional features according to the user needs and requirements.

Netography

[1] **Lanterns Studios**,
https://lanterns-studios.com

[2] **Virtual concert**,
https://www.virtualhumans.org/term/virtual-concert

[3] **The Metaverse**,
https://www.mdpi.com/2673-8392/2/1/31

[4] **Wavexr**,
https://pitchbook.com/profiles/company/163685-98

[5] **Unreal Engine**,
https://www.unrealengine.com/en-US

[6] **Epic Games**, *Epic Games* ,
https://www.epicgames.com/site/en-US/about?lang=en-US

[7] **Physical architecture**,
https://d1.awsstatic.com/whitepapers/AWS_Serverless_
Multi-Tier_Architectures.pdf/

[8] **Microsoft visio**,
https://www.microsoft.com/fr-ww/microsoft-365/visio/
flowchart-software

[9] **Blueprint Visual Scripting**,
https://docs.unrealengine.com/4.27/en-US/
ProgrammingAndScripting/Blueprints/

[10] **Blueprint**,
https://docs.unrealengine.com/4.27/en-US/
ProgrammingAndScripting/Blueprints/Glossary/

[11] **Event**,
https://docs.unrealengine.com/4.27/en-US/
ProgrammingAndScripting/Blueprints/Glossary/

[12] **Custom Event**,
https://docs.unrealengine.com/4.27/en-US/
ProgrammingAndScripting/Blueprints/Glossary/