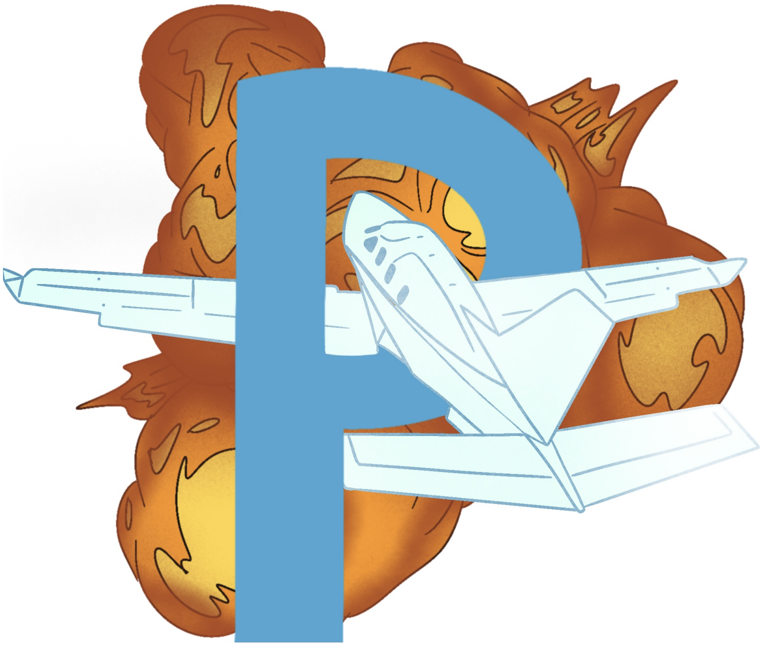
Achille Guerard
Térence miralves
Timothy Pearson
Léopold Tran

Project Pegasus



EAF

prépa sup

14th January 2022

EPITA 14th January 2022

Project Pegasus

Contents

1 Introduction

The purpose of this book of specifications is to transcribe the process of production of our IT project. It describes, in great detail, all the functionalities, and the corresponding technical specifications. It serves as a reference document, particularly during the implementation phase, to minimize the gaps between our expectations and the functionalities developed. The book of specifications must be able to be understood by every reader involved or not in the project. The focus of this book is to detail the technicalities, design, rules, production and schedule of our game.

Our game will be a multiplayer air combat game which will be precisely detailed further-long. The idea first came to us while thinking about building an ultra-realistic flight simulator, we however later realised that this had already been done and that the game would be missing some adrenaline. Our group then mind-mapped all our ideas and came up with a multiplayer game that regrouped various aspects that we like from different games including some flying, shooting, problem-solving and team cooperation.

An essential aspect of our game was indeed team cooperation, this is why we have chosen a game that will require the team to communicate but also make a decision rapidly. We were very drawn by the fact that the users would be unable to pursue the game and would even be at a loss if they were unable to organise their teamwork.

Another important aspect of our game will be the graphics, we have decided to focus a lot of time and effort on the modelling and design of our components. Despite the fact that our game may divert from reality in its spaceships or maps, we will always make sure that the graphics are as clear and coherent to our vision of the game.

Therefore, our goal is to make a fun and graphically realistic game that players will not be able to get enough of.

We also hope that building this game will teach us a lot from building an AI, multiplayer interace and modelling graphics. This project will also train us on small but essential details such as music, sound effects, animations, and teaching the user how to play the game. Most importantly, we hope that this production will strengthen our ability to work as a team and present our work not only throughout the various defences but also across our website and LaTeX reports.

Our team is composed of four members which will be sorted alphabetically:

- Achille GUERARD: I first started programming at the beginning of high school, and this was such a relief because I had found, at last, something that I liked studying. Throughout my two last years of high school, I learnt *Python* for the most part, but I also coded a website in *HTML*, *CSS* and *JavaScript*. Thus, I will be able to lend a hand to Tim on the development of the website. However, I have chosen to be in charge of 2D/3D modelling and animation because I wish to learn something new and have always been drawn to the artistic part of games. Since I already do a lot of coding in algorithmic and programming classes it is interesting for me to study other aspects of computer science projects. This will enable me to stay open to other subjects. Finally, I look forward to working in collaboration with people that I know are motivated to produce the best game possible.
- Térence MIRALVES: I started coding in the 6th grade on *Scratch* and almost immediately after went to a coding school weekly. Since then, I continued and touched a lot of different domains such as *Python*, *HTML*, *CSS* and *Arduino*. In 12th grade in *NSI* class, we were tasked with the project of building a video game which led me to be part of a programming project not only as a developer but also by managing 10 people. Thanks to this experience I learned a lot and hope to share my knowledge with the rest of the group. This project will increase my experience and enjoyment of group programming projects. Throughout this project, I hope to learn more about unity and *C#* as I haven't had the opportunity to merge them during a big project and to learn a brand-new way to work in a team.

- Tim PEARSON: My programming journey began with the *CS50 introduction to a Computer Science* course I started in 2020. I immediately concluded that this was the path I wanted to pursue. The course was in *C++* which helped a lot when it came to learning *C#*. Most of my last two years of high school consisted of applied mathematics and physics, hence my enthusiasm to be a part of this group creating a video game that involves the physics of flight. Although this project will extend my knowledge in a more specific field of physics, having some prior understanding should prove useful for the team. I have never worked with websites, *HTML* or anything of the kind. Since I am in charge of creating the website, I am looking forward to developing the skills necessary to do so. I have created a simple game using Unity to familiarize myself before starting the project. I found that I thoroughly enjoyed the process. The idea of endless possibilities in programming will always keep it exciting.

- Léopold TRAN: Ever since my youngest age, I have been passionate about computers and programming. During high school, I specialised in Computer Science and studied mostly Python. During high school, I developed many projects individually and in groups. This has taught me to be organized and to work in a team. Since I joined EPITA I have learned a lot, not only in algorithmic but also using tools that will be useful for the development of this project. I believe that my previous experience in programming will help my individual coding in *C#* but mostly be a good teammate. I believe this project will teach me more about teamwork and game development while enriching my coding skills. I also hope that this project will teach me about long term projects and particularly the organization and communication skills they require. I look forward to collaborating with this team as I believe I will learn a lot from all of them but also look forward to developing this stimulating project.

2 Presentation of the game

2.1 Concept

During this project, we will develop a multiplayer air combat game relying on the cooperation of a team, their ability to focus and react rapidly. Our project will contain aspects of a realistic flight simulator, a shooting game and also a “task completion/puzzle” game.

Once the game launches, the player will have the ability to play multiplayer or to play solo:

- The solo part will require that the player completes a mission in a plane while having AI-controlled enemy planes.
- The multiplayer will require at first for the players to choose a role. The players will pick from the following roles : pilot, gunman, repairman. Indeed, the pilot will fly the plane, the gunman will shoot and the repairman will walk inside the plane and play small puzzle games. Each puzzle game will have a link with the broken part of the plane. Once the player solves the little game, the damage will be repaired. The broken parts of the plane will have a direct impact on the player’s gaming experience, the gunman may be unable to shoot until the part is fixed, the pilot may have the controls inverted, etc. For the repairman to do his job the pilot will have to orient the plane in a certain way forcing the players to communicate.

The enemy aeroplanes with be controlled by an AI which will try to align behind the player’s aeroplane and then shoot it down. The AI will have a “human-like” behaviour to avoid the players being shot down too easily.

A part of the game will behave as a campaign, the player will complete missions and while he progresses throughout the game he will unlock new planes, skins and maps. Each mission will have a plane and map per it. For instance, if a player is playing solo in a high pursuit mission he will be able to fly a jet fighter whereas a team of four will be more likely to fly a cargo plane, a spaceship or any plane that can fit four people and may be adapted to the mission.

The goal of each mission will be explained to the players beforehand and will be displayed all along with the mission.

The game will involve different maps, planes and skins from which the character will be able to choose.

Our project will require that the players communicate with each other while rapidly making smart decisions to successfully pass the missions and unlock new elements of the game.

2.2 Progressing through the game

By progressing through the game the player will be able to unlock new routes and therefore missions. Some of these missions will require special planes or tools so the player will have to unlock all the requirements before being able to play the mission. Each mission will also cost the player a certain amount of fuel in accordance with the reward, the higher the reward the higher the price.

2.3 Modelisation and design

Our game intends to be graphically as real as possible. We will put a lot of effort into the design and modelisation of our characters, maps and planes so that they look real and of high quality.

3 Type of game

Our idea follows multiplayer air combat and cooperative role-playing type games. Air combat style video games are based around either helicopter, planes or sci-fi aircraft that attack or defend against an opposing side. Usually, they are fast-paced and require spatial awareness as well as quick decision-making skills to manoeuvre through terrains and opponents.

The first digital flight simulator of its kind was *Microsoft Flight Simulator 1.0* which was released in 1982. The player flies a *Cessna 182* around either Chicago, Los Angeles, New York or Seattle. It had a very detailed cockpit with working instruments which helped to make it as close to real-life flying as possible. To tailor the player's ability, difficulty modes helped beginners improve their skills to build up to the more realistic flying situations.

A perfect example of such is *Ace Combat 7: Skies Unknown*, released in 2019. It is an arcade-style combat flight simulation video game that includes online multiplayer as well as campaign missions. The missions start very basic and progressively become more difficult allowing the user to become familiar with controls and build skills for the multiplayer. *Ace Combat* is well known for having a large variety of different aircraft ranging from an Arsenal bird with a wingspan of 1,110m to a MiG-21bis at just over 7m. The range of planes available to the user adds to the long-lasting excitement of the game.

Air Combat, released in 1995, was one of the first combat flight simulation games. The plot involves a terrorist organization attempting to attack allied nations. There are 17 levels each with an objective to complete. The single-player campaign includes a range of different missions from protecting bases, destroying enemy battleships to taking down a squadron of enemy planes. Aside from the campaign, the game offers a split-screen multiplayer deathmatch mode where the objective is to shoot down the other player in as short a time as possible.

Lego Star Wars, first released in 2005, is a series of story mode games that offer the flexibility of playing solo or co-operative completing missions which follow the *Star Wars* story. Like *Ace Combat 7* but regarding characters rather than planes, *Lego Star Wars* has many to choose from as all characters from the films are available. Users can decide on which characters would help them to complete a mission. Additionally, users can tactically choose a character with a gun or a lightsabre which both have their own perks and roles throughout missions. During gameplay, the player/players can swap between these characters when needed.

Among us, released 2019, is an online multiplayer social deduction game which means that players aim to reveal teammates hidden roles or team alliances. When the game starts, depending on how many players there are in total, 1- 3 imposters are chosen. These deceivers try to sabotage and/or kill the rest of the team until there are an equal number of imposters and workers or the workers fail to complete the tasks in which case the imposters win. On the other hand, the players win if all imposters are sniffed out or all tasks are completed. Thus, the interaction between crew members is vital giving the players a team-building experience.

4 Production

4.1 Task allocation

Task	Main	Substitute
Multiplayer	Léopold	Tim
AI	Léopold	Térence
Gameplay	Térence	Tim
3D modelling	Achille	Tim
2D modelling	Achille	Tim
3D animation	Achille	
Physics	Tim	Térence
Particles effect	Térence	
UI	Térence	
Level design	Achille	Térence
Tutorial	Tim	Térence
Sound effect	Achille	
Music	Achille	
Website	Tim	Léopold
LaTeXreports	Léopold	

4.2 Description of tasks

4.2.1 Multiplayer

In this project, we will be using the *LAN Host* and *LAN Client* options. Our game will rely on one person which will start a game and become a host. The host, therefore, works as a client and the server at the same time. Then, the other players will link to the host (server) as clients. Our game will use *Unity's Mirror* and *NetworkManager* very rich tools. We will also use *SyncVars* in order to sync the attributes among the instances of the game. The game will also rely on *Remote Procedure Calls* in order to communicate between the clients and host.

4.2.2 AI

The artificial intelligence inside our game will control our enemy planes. This means that we will build an AI that will pilot these planes and shoot on the player while following a "human behaviour".

In order to do so the AI will have two modes :

- The first one will fly the enemy plane behind the players plane at a certain distance, angle and velocity. When the plane is position in accordance to these factors it will enter the second mode.

- The shooting mode will aim and shoot onto the player. The AI will switch between these two modes to stay behind the players plane and shoot when behind it.

The AI will obviously respect the gameplay and will try to maintain the most "human like" behaviour.

4.2.3 Gameplay

The gameplay is the way the player interacts with the game. It is the most vital part of game design. It is split into the main categories: game rules, the connection between player and the game, challenges and overcoming them and reducing anonymity in game design. The game rules are limitations to give the gameplay structure. The connection between the player and the game is the way in which the user communicates to the game through controls and how the game interprets these controls to navigate through the digital environment. The challenges need to be optimised in terms of difficulty. If the game is too easy, the player may get bored and finish too soon. On the other hand, if the game is too hard, the player may get frustrated and give up too soon. Here the aim is to keep the player's attention for as long as possible.

Reducing anonymity in the game design means making sure the player has a clear idea of the objectives to advance especially in missions. This can cause frustration and ultimately result in a bad game. The gameplay must complement the type of game and atmosphere that the game follows.

4.2.4 3D modelling/2D modelling

The goal of this section is to firstly, model different objects that will be used in the game, for example, for the game we will need different types of planes; a cargo plane, a fighter jet and others. These objects are going to be modelled using the software blender, as it's free and open source.

Secondly, we will need different characters that the players will embody. These characters will be able to move inside the plane and interact with objects. And this leads us to 3D animation.

4.2.5 3D animation

Inside the plane, players will conduct various actions, ranging from firing a canon to repairing an engine. To launch this action, we'll need short animations for smoother transitions between stations. For example, when the repairman attempts to repair the engine, he'll crouch down and the camera will move towards the engine.

To implement this part, we are going to use the animation module of *Blender*.

4.2.6 Physics

We plan to explore the physics involved in flight so that the plane movements are as realistic as possible. We will investigate how the yaw, roll and pitch angles will change with respect to the user's controls and how the forces acting on the plane will change with orientation and velocity. Furthermore, the different planes will vary in weight, drag and engine size affecting the overall max velocity, acceleration and minimum turning circle radius.

4.2.7 Particles effect

The particles are very important in a game as it can differentiate a polished game from a rushed one as it shows the attention to detail of the team behind it. In this game, particles will help the players to better understand their situation and increase the stress level on the most critical moments so that the players have an immersive time playing our game.

4.2.8 UI

The user interface, a.k.a. UI, is the way the user will interact with the game. It must match the overall "vibe" of the game while also giving the information that the player needs. It is crucial to find the right balance between too much or too little information on the screen so that the player is not overwhelmed by the content but still has the necessary UI elements. UI is not only about design it's also about functionality. This will include the pause menu, the main menu, launch screen and settings.

4.2.9 Level design

This section will focus on creating different maps where the player can evolve. For this, we are going to use Unity, since the physics (mechanics) part is developed within it. The synergies between these two parts are quite important, mainly for rendering purposes. In fact, unity handles the rendering distance quite well. However, if we import models from other software it'll need to render all the models at once, which is a problem.

4.2.10 Tutorial

The game will include small missions so that the user fully understands how to play the game and so that he can get familiar with the controls, UI and goal of each mission. These tutorials will be interactive so that we can make sure the user has fully understood the game before leaving him to wander on his own.

4.2.11 Sound effect

Whenever a plane either crashes or collides with another plane there will be an explosion that needs a sound effect to go with it. Other sounds as clicking or rattling noises will be added according to the situation. Situations like shooting, repairing, broken engines, etc while requiring sound effects. We will be using *SFX* and *Audacity* to customize and build the sound effects.

4.2.12 Music

In order to intensity the user's experience, we will create and add some music to each section of our game. This will add some cinematic to the game and fully immerse the player in the story. The music will vary according to the mission, the state of the game but will also vary during missions to be coherent to the intensity of the situation. We will also add some softened music in the pause menu and main menu. Custom music will be built for the launch of the game.

4.2.13 Website

Our website will contain all our written reports, some information about us and a download/executable version of our game. The website will be built using *Bootstrap CSS* and will be hosted by GitHub pages.

4.2.14 LaTeX

All our written submissions will be produced using LaTeX in order to look cleaner and be more convenient for our team.

4.3 Schedule

Task	First Defense	Second Defense
Multiplayer	60%	80%
AI	30%	60%
Gameplay	40%	80%
3D modelling	50%	75%
2D modelling	50%	75%
3D animation	0%	20%
Physics	70%	90%
Particles effect	20%	80%
UI	20%	60%
Level design	30%	60%
Tutorial	0%	50%
Sound effect	0%	20%
Music	0%	20%
Website	80%	90%

We do not believe that it is usefull explicit the schedule for the third defense as all will be at 100%.

5 Tools and ressources

- **C#** : *.NET 5.0.7*
- **Rider** will be our IDE of choice for *C#*. We have chosen *Rider* as we are familiar with it and is very handy for C#. We will be using the following version : 2021.2.2
- **Unity** will be our game engine development tool. We will be also be using *Unity Teams* and *Unity Collaboration.* We will work with *Unity* because it is cross-platform, enables high-graphics and because we already have experience with it.
- **Blender** will be used to create 3D and 2D models. *Blender* is a free and powerfull open source tool that enables the user to model, animate, simulate and render powerfull graphics for our game. *Blender* has also been our choice because it is cross-platform and our team runs on various OSs.
- **GitLab** for team development. We have chosen to work with *GitLab* and not *GitHub* first because the *CRI* is already set up to work with *GitLab* but also because we have more experience with it. We will be using several functionalities such as branches, version control and more. We will also try out some new functionalities such as *CI*.
- **Visual Studio** will be used with Unity but also to code our website using *Bootstap CSS*
- **Mirror/Photon** will be used to implement the multiplayer part of the game
- **LaTeX** will be for the written submittions
- **Illustrator** Logo
- **Audacity/SFX** will be used to create music and the sound effects
- **Trello** for task managment and distribution
- **Discord** for communication

6 Conclusion

In conclusion, our team will produce a multiplayer air combat game, the player's goal is to complete different missions, one after the other, while unlocking various planes and skins. Our game will rely on the communication and cooperation of the team by distributing various roles and forcing the team to work together if they want to succeed. We will draw influences from flight simulators, co-operations, puzzle-solving, and shooting games. In addition to these inspirations, we will have our own ideas to make this game our own and to realise our initial vision. We intend to build a graphical game that players will want to come back to by implementing levels and elements that the players may unlock, therefore a feeling of progression. Our team will extract each strength from its members to maximise productivity but also to share our knowledge. Therefore each member will follow the task distribution and schedule while communicating with the rest of the team so that we progress accordingly. Indeed, we believe that communication is the key element so that a team can be efficient. We all hope to learn a lot from this project, ranging from technical skills to teamwork while enjoying ourselves to the fullest.

Bibliography

[1] *A.I. learns to fly.* URL: https://www.youtube.com/watch?v=D5xX6nRWDko.

[2] *Ace Combat 7: Skies unknown.* URL: https://en.wikipedia.org/wiki/Ace_Combat_7:_Skies_Unknown.

[3] *Air combat.* URL: https://en.wikipedia.org/wiki/Air_Combat.

[4] *Aircraft principal axes.* URL: https://en.wikipedia.org/wiki/Aircraft_principal_axes.

[5] *Bartendu.* URL: https://areas0.github.io/website/index.html.

[6] *Blender 3D modeling : The ultimate collection.* URL: https://conceptartempire.com/blender-modeling-tutorials/.

[7] *Create a map in unity.* URL: https://docs.mapbox.com/help/tutorials/create-a-map-in-unity/.

[8] GameDevLessons. *Flight SIM control, Terrain Basics, Chase Cam, skybox.* URL: https://www.youtube.com/watch?v=lCulq9J0Y9E.

[9] *GitLab.* URL: https://en.wikipedia.org/wiki/GitLab.

[10] *Gitlab Documentation.* URL: https://docs.gitlab.com/.

[11] *Host (network).* URL: https://en.wikipedia.org/wiki/Host_(network).

[12] *Local area network.* URL: https://en.wikipedia.org/wiki/Local_area_network.

[13] Renan Oliveira. *How to create a multiplayer game in Unity.* URL: https://gamedevacademy.org/how-to-create-a-multiplayer-game-in-unity/.

[14] Dwight Pavlovic. *Video game genres : HP® Tech takes.* July 2020. URL: https://www.hp.com/us-en/shop/tech-takes/video-game-genres.

[15] Unity Technologies. *Setting up Unity Multiplayer.* URL: https://docs.unity3d.com/Manual/UnityMultiplayerSettingUp.html.

[16] Unity Technologies. *Unity Collaborate.* URL: https://unity.com/fr/unity/features/collaborate.

[17] Vazgriz. *Creating a Flight Simulator in Unity3D.* URL: https://vazgriz.com/503/creating-a-flight-simulator-in-unity3d-part-3/.