Ailouros - Specifications

Team Purr-fect (https://github.com/Purr-fect)

January 14, 2022

Contents

Figure 1: Team Purr-fect logo

Abstract

Members (EPITA 2026):

- Maïeul Boyer
- Clovis Lefebvre
- Bamlak Tadele GURARA
- Maxence Bornecque

Chapter 1

Introduction

This book of specifications will provide you an overview of our second-semester project, Ailouros. The team Purr-fect will create this game in C# on Unity, with the support of additional software such as Blender for 3D assets, and LMMS and/or DarkWave Studio for audio assets.

Ailouros is a MOBA (Multiplayer Online Battle Arena) game, whereby more than one person can play in the same game environment at the same time.

The players are organized into two teams of one to five cats.

While fighting the enemies, each team will be tasked with hunting preys on the map, and getting them back to the team's base. When a player dies, all of their collected preys are dropped onto the ground. Anyone can then steal these preys. The contest is won by the first team to gather the target amount of preys.

Players will also be the preys to one or a few foxes on the map. They will thus have to pay attention while wandering around.

While the AI will be, at first, fairly basic, we would like to explore an eventual stealth mode, where preys might not detect a player if they're slow and hidden by plants. Players may also want to sneak into their opponents' base using this stealth mode. Moreover, making too much noise while hunting might attract the foxes.

Finally, you'll discover a thorough explanation, strategy, and timetable for all of these tasks in this book of specifications, along with the deadlines for each of the presentations.

Chapter 2

Outline

2.1 Origin and type of project

Ailouros is a game where we play cats who battle for food and survival. The idea came from Warriors, a novel series where clans of cats battle for their territory. The game mechanics, however, were greatly inspired by Brawl Stars. Ailouros means cat in Greek.

2.2 Object of study

The goal of the project is to discover the creation of a game as a group. We'll have to adapt to the others and distribute the work according to everyone's pace and facilities. This will allow us to develop skills specific to collaborative efforts, allowing us to achieve the maximum potential of the team.

Maxence Bornecque: I started programming with Python around 12 years old. I learned a lot more languages since, and eventually discovered Rust. This language is well-designed and encouraged me to study algorithms. From 12 to today, I never really finished any big project, and stuck to more simple utilities in the terminal, or extensions for my browser. I also fixed issues in some open source software I use. This project will force me to achieve a big project, as I usually stop working on them when the hype goes down. On the technical side, I never used Unity for any real project beyond little tests, before EPITA. Thus, I will learn to use Unity for a project, and in a team. I'm used to git for big projects, and know how to use the pull requests approach, as well as the mailing-list approach. While the latter is really efficient, it is more adapted to open-source projects, accepting a lot of code from outside developers. We are clearly not in this context. We will mostly be using branches for different features, and open a pull request when each feature is ready. I'm probably going to be the merger, pulling the commits from the others, and testing them before pushing them to the

master.

Maïeul Boyer: I learned to program when I was around 14 years old, I loved it and continue to work in different languages, including Python, JavaScript (both in node.js and in browser), Typescript, and Rust which I mainly use today. This project will be the first real game I will make, as I have made some pet project using my custom game engine. This will also force me to use git with collaboration in mind, as I use it personally. Because I am lazy and tend to do something 1 day before the deadline, I will have to force myself so that my teammates aren't limited by my progress. This isn't the first big project I will have, I started making a 2d game engine in Rust around 3 years ago and still make an update to it from time to time. The multiplayer aspect of games always interested me, I have done some networking in some of my projects, but I never tried doing a multiplayer game.

Clovis Lefebvre: I have always been passionate about programming. I started it during the secondary school. During that time, I programmed my first video game on Scratch. In High school, I learned to work in Python. I still mainly use it now, especially for hobbies. Indeed, in my free time, I make some 3D animations with Blender and Python. I think that thanks to the knowledge and the experience that I acquired while making my video game and doing my 3D animations, I can use them to help my team making our game with the usage of the Blender software as I'm now use to it. Thanks to this project, I expect to improve my skills in Unity because I only did a few little projects on it thanks to the NTS. By doing this project, I hope also to gain experiences in programming in a team, as I have always programmed alone until today.

Bamlak Tadele GURARA: I have always been passionate about overcoming complex issues. I recently discovered that I had passion to coding, since it has a lot to do with solving problems. Even though I had little to no experience of coding before joining EPITA, I believe that I have learned quite a lot throughout my first semester in EPITA. I now know how to develop a game in C# using Unity, and also improved my problem-solving skills. Though code, I am now able to design and ideate possible solutions, and then execute them. Even though I have done many group works throughout my high school, this is by far the biggest! This project will not only enhance my coding skills, but also foster my creativity. I am eager to gain new experiences, enhance my skills, and utilize them to give a great contribution to the group.

2.3 State of the art

Ailouros is a MOBA (Multiplayer Online Battle Arena), as it brings team fights and strategy on a map.

The first MOBA game was **Future Cop L.A.P.D.** (1998).
The most famous games of this type are the following:

- Brawl Stars: It features a "gem grab" mode, from which Ailouros was inspired. The game features a 3v3 gaming experience, which can help build useful skills such as teamwork and analytical thinking so that the players can collectively win against their opponents. Solo gaming is also available as an option.
- **League of Legends** (LoL): LoL involves minions, and thus feature very different game mechanics from what Ailouros will provide. It features an ultra-competitive and balanced game experience. The game is set in a fantasy world, and the players interact with this world through their champions. These champions have magical abilities. In each game, you have to destroy the opponent's base.
- **Dota 2**: Dota 2 and LoL are both inspired from DotA, and thus have very similar game mechanics and features. However, Dota 2's mechanics are more stable, while LoL evolves a lot over time.

2.4 Parts of the project

Table 2.1: Parts of the project

Task	**Main**	**Substitute**
UI	Maxence	Bamlak
Character movements	Maxence	Clovis
Combat system	Maxence	Clovis
Multiplayer	Maïeul	Maxence
AI (preys and foxes)	Maïeul	Maxence
Level design	Clovis	Bamlak
Particle Effects	Bamlak	Clovis
3D modeling	Clovis	Maïeul
3D animation	Clovis	Maïeul
Music	Bamlak	Clovis
Sound Effects	Bamlak	Clovis
Website	Maxence	Maïeul

2.5 Description of the tasks

- UI: UI or User Interface makes a game more comfortable and usable for the user. The UI for this game consists of a life bar, the number

of remaining respawns, the score of the team, the time remaining and finally the information about the game if it is your first time playing this game.

- Character movements: They include basic controls of your character, and can actually be one of the most important parts of making a fun and engaging game. These include being able to move to the left/right/forward/backward, and also to walk and run.
- Combat system: Combat is essentially an interaction between the player and the enemies. To make the game even more intriguing, the player can not only attack, but also dodge hits aimed at the player.
- Multiplayer: This is one of the most critical parts of the game, as it will be multiplayer-only. Players will be able to create a room/lobby and will be given a code. Other players will be able to join the room using this code.
- AI: preys' movements, foxes' movements and combat
- Level design: It is a crucial component as it sets the atmosphere of the game and other entertainment projects. It includes trees, nature, plants and player bases.
- Particle Effects: To spice it up a little, we have implemented a particle effect that interacts with the game environment, such as rewarding feedbacks when hunting and also footprint particles while running.
- 3D modeling: 3D modeling is used to create unique life-like characters and to instill vibrant effects in a game. This includes all the characters in the game such as preys, cats, foxes, and the overall environment.
- 3D animation: 3D animation catches audience attention by giving life to the characters(cats, foxes, preys).
- Music: ambient music, menu music, battle music
- Sound Effects: To make the game more dynamic, sound effects will be associated with the movement of cats/foxes/preys
- Website: presentation of the project, rules of the game, credits, link to the sources of the game

Chapter 3

Structure

3.1 Functional

The game will be a MOBA style game where two teams of wildcats will fight for **food supplies**. The game will focus on stealth and hunting.

There will be foxes that will act as a predator, where if they sense your presence, they will start to hunt you. These foxes will be fitted with a custom AI to sense you as if they were real.

The cats will be able to go into a stealth mode, where they will move slowly, but will grant them better hiding.

We plan to make a server that will register statistics and metadata about the users. This will allow us to make a level system, and maybe add some cosmetics like different appearances.

3.2 Technological and methodological

To make the game, we are going to use Unity to make the game itself, thanks to the C# language and Rider as integrated development environment (IDE) for C# programming. Moreover, we are going also to use Blender to make the design of all characters that are involved in our game (the foxes, the cats and the objects to collect) and the map. We are also going to use, Discord and WhatsApp to communicate and share documents. Finally, we are also going to use LMMS or DarkWave Studio to make all the sounds and musics of the game. To create the website, we are going to use a simple code editor and host it on GitHub Pages.

3.3 Operational

To make the game work, we will need a game server that will handle the connection with the game clients, maybe one server that will handle everything around persistent storage and metadata, like player level or time

played. This could cost us around €10 per month for a basic 2-core machine at OVH (https://www.ovhcloud.com/fr/vps/). But as some of us own a real server hosted by ourselves, we won't have to consider this cost. We also need to think about buying software to make our assets, this will probably cost around €20 maximum. Fortunately, GitHub Pages is free, thus hosting the website will be free. For time cost, some features will take more time than other features.

The most time-consuming features are listed here with a time estimate:

Table 3.1: Parts of the project

Features	**Time estimate**
Designing combat system	4 weeks
Making 3D assets	3 weeks
Making AI	4 weeks
Stance Management	3 weeks

The balance between hunting, stealing preys, protecting your base, and simply killing enemies will also be time-consuming and hard to get right, but is not a feature by itself.

3.4 Progression

Table 3.2: Parts of the project

Task	**1st Defence**	**2nd Defense**
UI	30%	60%
Character movements	50%	75%
Combat system	50%	75%
Multiplayer	60%	90%
AI	20%	60%
Level design	40%	60%
Particle Effects	5%	40%
3D modeling	15%	55%
3D animation	10%	45%
Music	15%	65%
Sound Effects	5%	65%
Website	60%	90%

We didn't think having a 3rd column for the last Defense is useful, as it is obvious that everything should be done at 100

Chapter 4

Conclusion

To conclude, Ailouros is going to be a MOBA game, walking in the steps of giants like **League of Legends** and **Brawl Stars**. This project led us to many ideas which may be fun but time-consuming. We eventually achieved to strip the game down to an achievable goal. In order to complete Ailouros, we scheduled the task distribution and work completion for each defense. The work was divided between the team members according to their facilities. Each one of us will learn from the others as, currently, no member of the team would be able to manage all the aspects of the game by themselves. Finally, we intend to fulfill the scheduled set of features and maybe add on top of it.