Book of specifications : EpiKart

Undergraduate (1st Year) EPITA

TEAM BINBINKS

All the members are from A3

Eliott Flechtner (Project leader)

Lancelot Doucet Titouan Verhille

January - June 2022

Summary

1 Introduction

1.1 Team Members

This document constitutes the **Book of Specifications** for the second semester project of the Team BINBINKS, which runs from January to June 2022. The name of the project is *EpiKart* and is an **arcade-style racing video game** with a focus on fun and friendliness. The game will feature online multiplayer as well as local play. The Team BINBINKS is composed of the following individuals *(all in the A3 class)*:

- **Eliott Flechtner (project leader)** : basically the one with the silkiest hair in the galaxy,
- **Lancelot Doucet** : the most loyal knight of the royal court,
- **Titouan Verhille** : the biggest of the colossal titans.

1.2 File content description

This Book of Specifications will first of all deal with the **historical and creative aspect** of the project by telling its premise. Then, it will provide information about **its interests and particularities**. Furthermore, it will cover everything related to the main game **mechanics and the graphical interface** of the game. Finally, it will detail the **tools required** for its successful completion as well as its **schedule planning** and the distribution of tasks by person.

The aim of this project is to put us in the condition of an important project realization as in a company. This simulation is directed **within the framework of the creation of a software or a video game** and will allow us to familiarize ourselves with the development tools related to the latter. It also will make us learn more about the creation of a website, the respect of the dates and deadlines, the work in team and its management, the preparation of oral presentations in group and the creation of official documentations.

2 Project Presentation

2.1 Origin of subject

2.1.1 Video game VS Software

When we first started to share the ideas we had for the project, we realized that we were hesitating strongly between making a video game or some kind of software. On the one hand, we believed that designing a piece of software would be **more engaging**, as aspiring computer engineers, merely because we would be much more likely to be creating the latter than video games in our professional lives. We found it interesting that we had to **find a problem that our software should answer** while managing to construct an enough-detailed and explicit interface for the user according to the need. The only downside was that **we had never done this before** and would probably require a higher level of engineering to do it. On the other hand, working on a video game seemed **more appealing and entertaining** since we all played and had fun with them in the past! It was also something we knew very well and could talk about for hours. It thus seemed only natural that we would prefer and **choose this option over the other**. This choice can also be justified from a technical standpoint: we favored the usage of C# and the *Unity* game engine rather than OCaml or F#. Also, we felt that we wouldn't know where to start if we were to conceive software in terms of code spaces or tools we could use. Likewise, we figured out that designing and discussing features for a piece of software **could be way harder than for a game**.

2.1.2 Selecting criteria

Now that we had settled for a video game, we immediately came up with tons of ideas from various genres: a tower-defense game, platformers, puzzle games, shooters and even a musical-themed game! At first, we were going all over the place: we kept suggesting various ideas and concepts **without actually listening to each other's proposals** or fully exploring one idea after another. The experience of team members who had participated in game jams before definitely helped us **focus and sort them all out**. They pushed us to try and follow the *3-E*'s rule: **easy to understand, easy to pick up and easy to pitch**. We also wanted to avoid creating lore nor a story-line for the game: this way, anyone could be able to come up and test in just a few minutes without having to go through the entire game from the beginning. But above all, we wanted the game to **be fun to play and not too complicated to develop** since we barely knew how to use the *Unity* game engine from the AR-VR classes earlier in the semester.

2.1.3 Final decision

At this point, it was hard to find an idea that matched all the standards we had set for the project. We had explored every idea but **none of them was as striking now**. After about a week, we decided to look at the problem from another angle: maybe we could get inspired by games we knew **that matched those criteria on their own**. That is when we realized one of the **most suitable game genres was racing games**! The idea immediately seduced everyone when it came up. As for the name, we didn't think about it for too long and assumed that *EpiKart* was the best possible one out of the lot *(which it is)*.

2.2 Scope of the project

First of all, we see this project as an opportunity to learn more about **time organization, management and communication as a team and an individual**. The project immediately brings to the forefront the importance of meeting intermediate and final deadlines. It is mandatory to respect these, and even create our own, to adequately deliver the final product. We learn to project and foresee precisely when every feature needs to be finished for the correct development of the project. It is an essential quality when working in a company, whatever the sector. Moreover, respecting one's instructions and deadlines is not something that comes to everyone naturally. Thus, **the partitioning of the given time** must also be done by taking into consideration the intensity and quantity of work of the regular classes and evaluations. The project also teaches us to **plan for the economic aspect and therefore potentially allocate a budget for our needs**. Finally, it teaches us to document and record its progress as it occurs and to construct clear and professional reports accordingly **through various media** such as a Book of Specifications, an oral presentation or a website. This information must also be conveyed simplistically and **supported by visuals** such as charts, video clips or live demonstrations. **Being concise**, when necessary, is also part of the job!

Secondly, the project brings - mainly to the project leader - **social qualities of group teamwork and team cohesion**. We think it compels the group members to establish adequate communication through regular work meetings or conferences between the leader and members to allow **for better collaboration**. It will ensure a more even distribution of tasks according to the strengths and weaknesses of each member of the group. We also agreed that **the motivational factor was a consequential part** of the project. The leader will learn to be a strong motivational figure for his teammates and be very adaptive. For example, he must be able to **quickly decide to reassign a particular task** to another member of the group in the event of a technical or personal problem.

Finally, the project provides **non-negligible technical skills and qualities**. We will teach ourselves how to create a video game from scratch since barely anyone in the team have used the *Unity* software before. We will learn a lot more about the game engine and especially about C# and **object-oriented programming**. It will also be useful for us to uncover and learn more about **level design and modelling 3D objects**. In addition, the website to be created and presented during the oral presentation will **familiarize us with the web display languages** such as HTML, CSS and JavaScript. Finally, all of the returned official text documentation will be formatted using the software system for document preparation LaTex, which will be an opportunity for the team members **to use it daily**.

2.3 State of the art

2.3.1 Historical recap

The first racing video game was *Space Race* published by the former American video game company Atari in 1973 in the arcade. It is a two-player game where each player **must get their respective spaceship** to the top of the screen before the other *(See Figure 1. Below)*. It was innovative for its time as it was the first arcade game to take place in space (before Asteroids in 1979). It was also one of the first games, along with Pong, to include a score and highscore system. It is considered to be the very first game **in the spirit of a racing game** ever created, but it was not until 1982 that Namco published the console game *Pole Position*, this time **featuring real Formula 1 cars** *(See Figure 2. Below)* and played on realistic tracks. It can only be played alone, but the presence of bots running around the tracks with the player was already implemented. The controls are very fluid despite the hardware and controller available. Also, the visuals are very **vibrant and colourful**. Finally, in 1992, the renowned Japanese video game studio Nintendo released *Super Mario Kart* for the Super Nintendo *(See Figure 3. Below)*, the first in the *Mario Kart* series. The game featured a single player mode against the computer on a **total of twenty different race tracks and combined racing and combat in an amusing way**. It is this game and its descendants that are the biggest source of inspiration for us during this project.

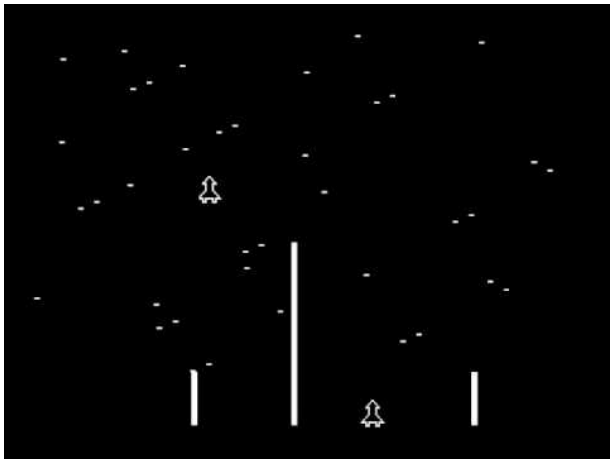


Figure 1. Atari's *Space Race*

Figure 2. *Pole Position*

Figure 3. *Super Mario Kart*

2.3.2 Other references

We find it important to distinguish between two types of racing games: those that are more competitive and realistic and those that are more fun and friendly. *EpiKart* falls into the second category, so here we will talk about the other games belonging to it. Since 1992's Super Mario Kart, Nintendo has released new versions of the same licence, the best known of which are *Mario Kart 64*, *Mario Kart DS* and *Wii* and in 2017, *Mario Kart 8 Deluxe (See Figure 4. Below)*. All of these games kept their initial objectives over the years: the desire to **make the game accessible to all ages** and to offer an experience based on speed and the **pleasure of playing rather than competitiveness**. In the same spirit, the development studio Summon Digital released in 2019 *Team Sonic Racing (See Figure 5. Below)* and proposed some changes to the gameplay by making it **more focused on cooperation** between players of the same teams while keeping the essence of a *Mario Kart*.

Finally, *Crash Team Racing: Nitro-Fueled (See Figure 6. Below)* takes the basics of *Mario Kart* and focuses **on vehicle and character customisation**.

Figure 4. *Mariokart 8 Deluxe*

Figure 5. *Team Sonic Racing*

Figure 6. *Crash Team Racing: Nitro-Fueled*

2.3.3 Strengths and attributes

The particular feature of this type of racing game is the **combat between vehicles to spice up the race** as well as the **zany and varied design of the circuits**. Players can pick up bonuses during races to slow down others for their own benefit. In addition, mechanics such as **drifting and speed boosts pads** on the ground allow a player to momentarily increase his speed. Finally, the goal of this kind of game is to get the first place in each race, thus a **ranking of the players' times** per lap and on the whole circuit is always displayed and pushes the player to go faster each time.

3 Features and content

3.1 Gameplay and mechanics

3.1.1 Movements and controls

We particularly want to make the handling of the vehicles **as fluid and intuitive as possible** in order not to lose new players and to allow a better control. It would therefore be preferable to implement the controls to be playable **with a joystick** rather than using a keyboard. However, both will still be usable in the event that a user does not have a controller. The range of movement will be improved by adding the **ability to drift to gain speed** and players will be able to collide with other players to slow them down for example.

3.1.2 Circuits and cups

Different circuits and maps will be playable **in groups of four or five in a row** in the context of a cup. The ranking will then be saved and kept for all tracks of each cup to determine **the top three winners of the said cup**. Each race will end after a certain number of laps depending on the size of the circuit. Finally, each circuit will have its own identity and theme.

3.1.3 Power-ups

In addition, it will be possible to pick up **bonuses and special powers** spread throughout the circuit. Their powers and strengths will be managed **according to the ranking of the player who acquired them** in order not to create too high a gap and thus disadvantage players with a low ranking.

3.2 Framework and functionalities

3.2.1 GUI and menus

The graphical user interface will allow each player to know at any time of the race **his ranking, his lap time and a mini-map** allowing to see the position of the other players on the circuit. He will also be able to see if he has recently picked up a bonus and which one it is.

At launch, the game will present **a main menu** allowing the user to select the desired game mode, i.e. offline or online. The user will also be able to access the control settings and change the assigned keys as desired. Furthermore, he will be able to access **the game credits** as well as a **link to the project's website** containing all the official documentation via two separate buttons. Finally, a last button will allow to **quit and close** the game window.

3.2.2 Modes, network and AI

As briefly said previously, we also are planning to have **two game modes**: first off, an online mode with a **limit of 8 players per game lobby**. We will therefore have to implement a **network game system** allowing players to connect to a game room via a randomly generated five-alphanumeric room code and launch a game with several players simultaneously!

Second, a single player mode, i.e. offline. It will need the **implementation of artificial intelligence**. This will be done through the offline mode, available from the main menu: the player will be able to play **locally on his machine against bots** on the maps of his choice. We will therefore have to develop these bots so that they are competitive enough and capable of interacting with their environments without being unbeatable.

3.3 Bonuses ?

Time permitting, we would also like to implement the following mechanics and features:

- A way for the offline player to choose the **difficulty and level of aggressiveness of the bots** one will face when playing offline. We were thinking of **three difficulty levels** that will vary the behaviour of the AIs to make them faster, smarter and more vicious.
- A different 'Battle Royale' game mode where the goal is to **be the last car standing**. Here, the goal is not to finish the race at first place but rather **to survive the longest**. Thus, there is no limit of laps in the race. At intervals of 30 to 40 seconds, a certain position in the race is designated as 'cursed'. The players have to avoid being the car in the latter position at the end of the timer **or will face elimination from the race until the end of the round**. Just like the classic mode, the races will be played according to the pattern of the cups described earlier: it will be possible to make a counter of points as well as the amount of survival time of each player in order to **establish the winner of the cup**.

- A **vehicle customization workshop** allowing users to apply different textures to their karts or characters and even import their own logos and designs! With the implementation of a **virtual currency system** that can be earned by winning races, players could also improve the capabilities of their vehicles such as their speed for example.

4 Schedule management

4.1 Technological

4.1.1 Game

We will use *Microsoft Visual Studio 2022* to program the scripts and necessary code for the game. We will use *Unity* as the game engine to create scenes, animations, effects and more! Code-wise, it'll allow us to link code editors and assign each script to its corresponding entity in order to make the game work properly. We will use *Blender* to design and model the in-game objects such as karts, tracks elements, etc. As for the O.S.T. (the music and the in-game sound) we plan on using *Pro Tools*, a professional music and production software.

4.1.2 Website

For the website development, we settled with *Node.js*, *Tailwind CSS*, and *Eleventy* in combination with HTML and CSS. For editing our content we decided to **build a server-less structure**. *Eleventy* is being used to compile the Markdown styling language and *Nunjucks* files into pure HTML and *Tailwind CSS* compiles our CSS code. We built it keeping in mind that the website **should be easily available and modifiable offline**. It updates every modification once the changes have been committed to the repository **instantly** and thus makes it easy to **show and test out for ourselves**.

4.2 Methodological

4.2.1 Communication

As in most group projects, organisation is key. To communicate our ideas an **share documents and various resources for the project**, we use the multi-platform software *Discord*. We created a server with

multiple channels to facilitate the workflow and share our assignments and links to videos or other useful piece of information *(See Figure 7. Below)*.

4.2.2 Organisation

Then, we use *Notion* **as a drive to store our official documentation** *(See Figure 8. Below)*. *Notion* is a software similar to *Google Drive*, but allows for more customization. As well as storing our work, we use it to distribute tasks to each other or to **schedule deadlines**.

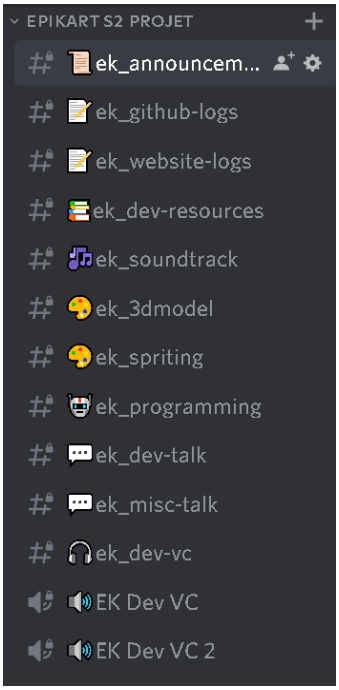


Figure 7. *Discord*

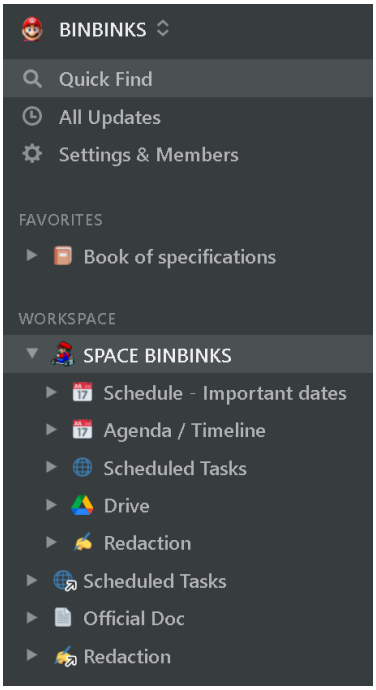


Figure 8. *Notion*

Then, we will use *GitHub* to share our code mainly through two repositories: one for the website and one for the game. *GitHub* is useful because we can commit in different branches and merge them later, thus preventing us from the use of flash drives and **allowing to work at the same time on different part of the project**. Moreover, mastering *GitHub* is important as it is a very common tool in the professional sector which we started using a lot for T.P.s and personal projects. Although, we were told that transferring our *Unity* projects via *GitHub* could be a solution, but that it was not the best or most practical one, as **the projects could sometimes be very heavy**. We will therefore use in addition the *Unity Collaboration* feature already implemented in the software in order to **allow these heavy file exchanges** between all the members of the group.

4.2.3 Handouts

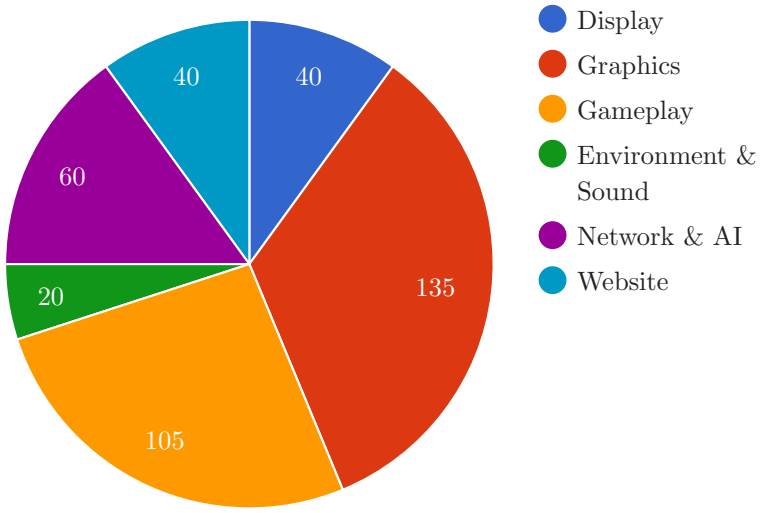
We will use *Overleaf*, an online software that uses *LaTeX* **to format the handouts PDF documents** with a proper report style. It is useful as the generated PDF can be edited and updated in real time as the *LaTeX* code is being written.

4.3 Operational

For the economic aspect, we foresee a maximum budget **of about a hundred euros** that would eventually be invested for the server costs for the online multiplayer aspect of the game or for any other online services such as the **purchase of a domain URL** for the presentation website provided. It could also be spent on multiple assets such as 3D models, more personalized textures as well as **materials or sound effects** and music to make the game more realistic and less silent by supporting the work small independent artists or by **acquiring broadcasting rights**.

We also plan to invest a total of **over one hundred hours per person** in the group. The distribution of these hours is described by sector in **the pie chart below**.

Time Repartition per Task per Hour



4.4 Parts of the project

4.4.1 Distribution

The distribution of tasks is distributed as follows:

		Eliott	Lancelot	Titouan
Display	UI / Interface	✕		✕
	Menus		✕	✕
	Key mapping	✕	✕	
	Utilities	✕	✕	
Graphics	Shaders		✕	✕
	Textures	✕	✕	
	3D Models	✕	✕	
	Animations	✕		✕
Gameplay	Movement	✕	✕	
	Animations	✕	✕	
	Power-Ups		✕	✕
	Cutscenes		✕	✕
Environment Sound	Tracks design	✕	✕	
	Soundtracks	✕	✕	
	SFX	✕		✕
Network AI	Bots		✕	✕
	Multiplayer lobbies	✕		✕
	Online servers	✕	✕	
Website	UI Design	✕		✕
	Home page		✕	✕

✕ : In Charge ✕ : Substitute

4.4.2 Progression

The completion rate of the different major parts of the project is given in the following table:

	First Presentation	Second Presentation	Final Presentation
Display	10%	90%	100%
Graphics	20%	95%	100%
Gameplay	42%	69%	100%
Environment & Sound	Tracks design 25%	Tracks design 75%	100%
	Soundtracks 0%	Soundtracks 50%	
Network & AI	50%	70%	100%
Website	25%	75%	100%

5 Conclusion

To conclude, the EpiKart project is a way for us to progress in **all aspects**: social, team management, group work and technical. Moreover, it teaches us **to create a large project over a long period of time** in an unguided way to get as close as possible to life in a company.