Ultimate Fighting Chickenship

Report 2

Tempête group

April 2019

charlie.brosse@epita.fr
philippe.lefebvre@epita.fr
david.horozian@epita.fr

Contents

1 Introduction

By the following document, the TEMPÊTE team presents to you the advancement of its project Ultimate Fighting Chickenship as of the second presentation on April 24th 2019.

This report will debrief by order of precision the advancement of each task presented in the Book of Specifications. We will start each section by general comments about how each task has been approached and realized.

2 General comments

After building a solid framework for our project, we have begun adding in features and polishing some of the foundations.

Our main focus was to make the project start looking like a real, shippable product, whilst keeping ahead of the schedule set in the Book of Specifications.

Some goals have been reached in time, some even surpassed, while other have not been reached. Below is a table comparing our expectations (as written in the the Book of Specifications) to the actual results for this presentation.

	Expectation	**Reality**
Game mechanics	70%	85%
Movement	90%	95%
Attacks and Combos	60%	60%
Mechanics	70%	80%
Sound design	50%	75%
Music	60%	75%
Sound effects	50%	75%
Voices	50%	50%
Graphics	60%	50%
Map design	75%	30%
Character design	60%	40%
3D Animations	60%	50%
User Interface	60%	60%
Other	70%	80%
Multiplayer	90%	90%
Website	60%	90%
Artificial Intelligence	25%	0%

3 Game Mechanics

3.1 Comments

Becoming familiar with Unity, its basics features and C# programming in this environment, Charlie started working for the second phase of the project with more confidence. Of course, he still had numerous features to discover and learn how to handle. Thus, he entered a more advanced phase of video game developing.

After debugging few things, he started adding complexity to the gameplay. Switching between reflexion and code, he tried to find the best way to implement each task in order to optimize the game.

We have read documentations and already have an idea of how to implement combos but it will be a future objective.

Yet, patience and perseverance were involved in this process because everything did not work on the first try. However, with much documentations the goals have been achieved.

3.2 Movement

Adding to the already available basic movements a dash movement, the ability to double jump and the rotation of the player on each turn, a player has multiple movement possibilities during a game of Ultimate Fighting Chickenship. All these changes have been implemented in the C# script Player Controller.

At this point, the game is not far from its final form in movement possibilities. These three main new movements were more complex to implement as they are not as basic and common as moving left, right or jumping.

The double jump has been implemented using a jump counter and a reset check (see figure 1 in Appendix). Every time one touches the stage ground the jump counter is reset. On the other hand, each time one jumps pressing down the Space bar, the counter is incremented by one. Finally, the counter must be below a certain limit to enable the player to jump (the limit is 2 for double jump). Another system would have been to implement a boolean array where each element checks for a specified jump. However, this system is too bulky and won't allow simply to increase the limit of jumps. An important condition is to check for the Space bar input when it is **down**. Otherwise, there is a risk that the key is pressed during more than one frame and thus the counter will reach its limit too soon.

Furthermore, the player turns towards the direction it is moving (see figure 2 in Appendix). Charlie implemented it using a boolean informing on the direction the player is going. Then on each left/right movement input the rotation is changed if needed. A problem was that it also inverted the internal axis of the character, thus the movement vector had to be inverted.

Finally, the dash movement works like a small teleportation towards any direction. The 'dash' does not change rotation and only provides an additional mechanical possibility for the player. It is quite complex to perform one, we aim to make it a move accessible to trained players. Lastly, the 'dash' demands energy called "stamina" to be used: we will get back to this new system.

As one can see, the goals established last deadline have been achieved: the gameplay is more dynamic thanks to the dash the and double-jump and more fluid thanks to the rotation on turn. For the next deadline, the aim is not to add new movement but to make the already existent ones work perfectly.

3.3 Attacks & Combos

There have been a lot of changes in the system of attacks. Keeping the BoxCollider system for the attack boxes, a new C# script Attack have been created. This script only contains a structure Attack establishing its components and its constructor. Thus, the Box array have been replaced by an attack array in the Player Controller script. On the specified input, an attack is chosen from this array and calls the LaunchHit function in the Fighter script. Attacks have multiple variables :

- *name* of the attack, at this point 'punch' and 'kick'

- *damage* it deals on hit
- *hitbox* : the attack box it uses to trigger the ennemy's body
- *knockbackX* and knockbackY: the knockback components
- *staminacost*: the cost in energy

Charlie also implemented a method GetKnocked() called with the attack's knockback components as a vector each time the player is hit. This method constitutes in famous mechanic functionality of fighting games: the Knock Back or Push Back. When one is hit, one will be pushed away in a certain direction depending on the type of attack.

One of the major novelty is the creation of a new type of attacks: the abilities. We aim to have different types of abilities. At this point, Charlie implemented the Projectile type of abilities in a new Projectile C# script. Projectiles almost have the same components as Atttacks yet it is NetworkBehaviour inherited Class. It takes into account three new variables:

- *VelocityX*: the constant velocity at which the projectile moves
- *Prefab* to be spawned in the game
- *SpawnTransform*: the position where it will be spawned

Projectiles have their own collider that will detect collision and call OnCollisionEnter() method. This method decreases the health in the HealthBar script and destroys the projectile on hit. Projectiles deal much more damage than basic attacks and offer the advantage of the distance. However, they can be easily dodged and cannot be used as often as punches and kicks. The Player Controller script has been updated to have an Projectile array and to be able to call the LaunchProjectile method in the Fighter script on the right input. This method is a Command. It means that it is a call from a client to the server. Indeed, to make sure the projectile spawns on each clients, the server must be in charge. At this point, Ultimate Fighting Chickenship has one Projectile Attack: a Fire Ball (see figure 3 in Appendix).

For the next deadline, we aim to create dizziness after each hit and attack. In addition to the stamina, we want to create a cooldown for each attack to prevent the players from spamming. Finally, we will progressively add new attacks and combos, it is not the hardest part, it just needs imagination.

3.4 Other mechanics

3.4.1 Stamina system

A whole new system has been implemented: the stamina or energy system. Some actions cost stamina and can only be performed if the current stamina points amount is sufficient. Players have start at a maximum amount and cannot go above this value.

All attacks and abilities have a stamina cost, the dash movement also costs stamina. The stamina is updated by calling a Command taking a float in arguments: the current stamina amount is decreased by this float and the stamina bar is updated (see figures 4 and 5 in Appendix).

Additionally to these punctual calls, this command is called each frame by the Update() event to increase the stamina amount: it acts like a regeneration. The amount of stamina regenerated each frame is based on the number of frames by second of the game to make sure the regeneration amount is the same over time in any environment:

regeneration = Time.deltaTime*10

3.4.2 User Interface and Experience

Charlie implemented a stamina bar that updates during time. He integrated it with the Health Bar. The objective was to make them both work on network so that both players can see the health and stamina bar of the opponent. It has been done using the SyncVar attribute of Unity enbabling the synchronization of the amount of health and stamina and of the update of the bars. Both bars were implemented manually on the player prefab to make them appear above the player. Another script has been created for both bars to make sure they do not rotate with the player prefab.

Finally, a script ExitGame has been created to come back to the Main Menu. Pressing the Escape key leaves the game.

For the next deadline, we want to create a protection move: by holding a key one will reduce the damages taken during the holding, it will cost a significant amount of stamina. One will not be able to use any other attack or to move during the protection.

4 Sound Design

4.1 Music

During the second period, Philippe has kept on crafting sounds in order to provide players a more fulfilling experience. He made the music that would be heard during the fights, the music that would best accompany most of the player's time spent in the game. The process is the same that the one for the main menu's music: Philippe made a 1min20 loop using Ableton.

The idea of the song came from listening to "Praise The Lord" by A$AP Rocky. The fascinating flute sample playing through the whole song stuck in our heads. We decided to use the very same sample and make a remix of the song. In the first half, percussions similar to the original song's allow us to make a pop-culture reference as well as extending the hip-hop elements brought by the main menu theme. The second part of the song, however, sets itself apart from the rest of the music. It contains almost exclusively electronic music elements and are a full part of the genre-bending mindset Philippe had when crafting the game's music.

4.2 Sound Effects

Another achievement Philippe is proud to have made is the making of the sound effects. With the help of his and Charlie's voices, he established a large sound bank, containing almost a hundred sounds. This number will obviously grow as we keep on adding functionalities to the game. Sometimes sprinkled with reverb or distortion to better fit the context, almost every action in the game has its own sound-producing function in its script. It uses unity's random function to pick a .mp3 clip from the array of clips each action possesses.

For the next deadline, our goal is to have a functionality that cuts the previous sound before playing the next one in order to have a better sound fluidity when doing multiple actions in a short time span.

5 Graphics

5.1 UI and Design

David revamped the main menu to feature a background and overall more stylish buttons that blend into the picture, as can be seen in fig. X. The inspiration came from a picture of an old American fast-food chain called Burger Chef. David edited it as to remove any proprietary content. This main menu is the result of a combination of work in both Adobe Photoshop and Unity.

The main menu still features a play and exit button, along with the main theme for the game. A new appearance is the Options button, that is a placeholder for the time being. Our plan is to implement a full-fledged options menu where the player can change game settings on the fly, without having to restart the game.

The game now features a logo, that comes in two variants:

- The first is a 2-line design, that is featured on the website's homepage and the game's main menu,
- The second is a 3-line design, which is meant to be used in smaller spaces as its resolution is more square-shaped.

The logos can be seen on respectively fig. X and fig. X.

5.2 3D modelling

Philippe took care of the 3D modelling of the character. He learned how to use Blender, a free 3D editing software. He first designed a basic 3D humanoid character. He wanted to use a humanoid style of character, because it would facilitate the uses of the model in Unity, because it offers presets for humanoid models. He still wanted the model to look like a chicken, so he then buffed his thigh but kept thin leg bones, replaced the human feet by a chicken foot and put somewhat of a chicken-human mix for the head. With this first character done, its really a landmark, because making the next characters will be a breeze since we will only changes its textures. (see fig. 6)

5.3 Animations

As for the animations, Philippe first constructed and placed the bones of the character. He then proceeded to "weight paint" the 3D model to make the animations deform the character in a natural fashion. He made two basic animations: a standing pose where the chicken's arms move slightly, and a side-walking animation, where the chickens make side-steps in a bit of an on-purpose ridiculous fashion to enhance to funny side of the game. (see fig. 7 to 9)

6 Other

6.1 Multiplayer

Philippe has done little work on the overall structure of the multiplayer functionality of the game. With little adjustments, features such as the health bar have been fixed from the last presentation. The game still features the unity original network's GUI that we plan to change for a custom one by the time of the third presentation.

6.2 Website

David continued working on the website, whose completion is nearing 100%. The website now features the four pages that were specified in the previous report (Home, Project, Download, Links), including the last two that were not yet completed. The only thing there is left to do now is to populate these pages with content, a process that has already started. With input from every member of the team and following the progress that was made in each field of development, David has written newsletters that are destined to inform the community of the progress that the team is making in advancing the project.

As a reminder of what was stated in the previous report, the structure of the website is as follows:

- *Home*, a few paragraphs to introduce the website, the project and the team,
- *Project*, a page where we post advancements, screenshots and other various news for our community to see,
- *Download*, a page that will host download links to the game and various reports as they are completed and ready to be published,
- *Links*, a page that will host all information and media that was used to create the game, such as sources for images, designs, models and so on.

Screenshots of each page can be found in the appendix (fig. X to fig. X).

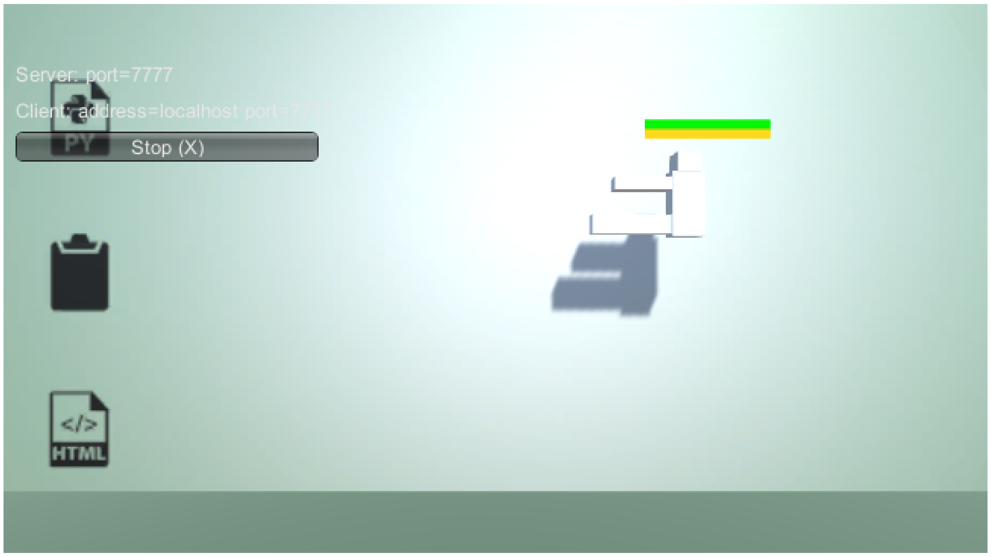
7 Conclusion

In conclusion, the project has kept on its pace following the first presentation. The team has made great progress in almost every department, whether it be gameplay mechanics, sound design or graphical design. Though, we are regretful about the domains that we are late on, being Artificial Intelligence and Map Design (as well as Character design and 3D animations, where progress has been made but not yet added into the game).

On the one hand, many parts of the schedule have been respected. Game mechanics have, once again, advanced at a rate that respects our previous schedule. In the sound design department, thanks to the team already having a head start after the first presentation, we managed to stay ahead of schedule with the implementation of sound effects, some voices and a new music track. The website is structurally complete and only waits for content to be uploaded to it. Lastly, the early foundation of good bases for the networking system in the game has allowed us to start adding in gameplay features without any worries.

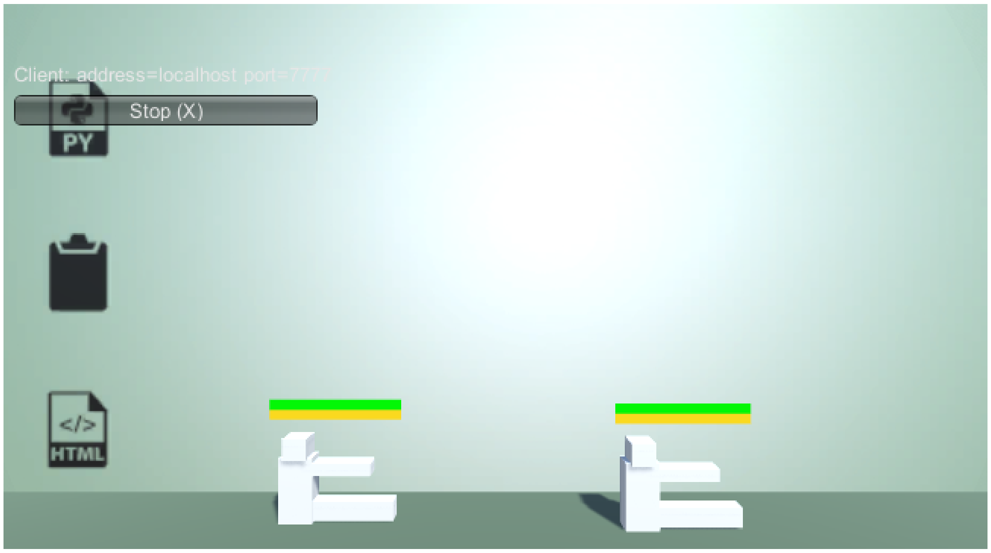
On the other hand, while character design and 3D animations have been nearly completed, we have had difficulties implementing them into the game. Other aspects such as Artificial Intelligence and Map Design have seen little advancement, even though we are still confident about advancing in those domains for the next presentation.

8 Appendix


Server: port=7777

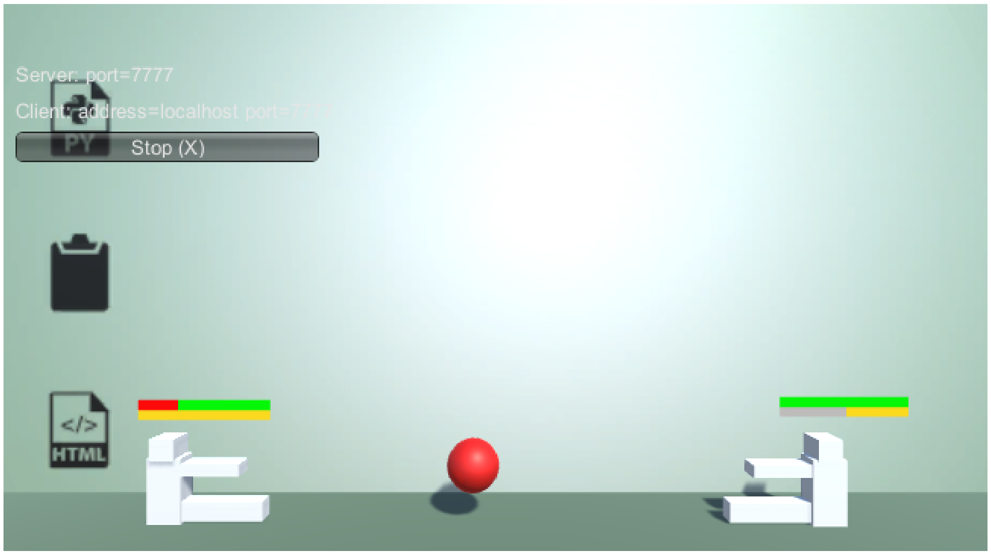
Client: address=localhost port=7777

Figure 1: New double jump height



Client: address=localhost port=7777

Figure 2: Both players face the same direction after rotation



Server: port=7777
Client: address=localhost port=7777

Figure 3: A moving fireball that will damage player

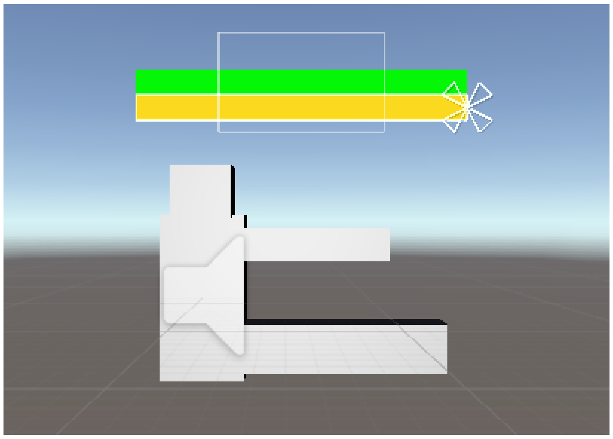


Figure 4: A full stamina bar in player prefab

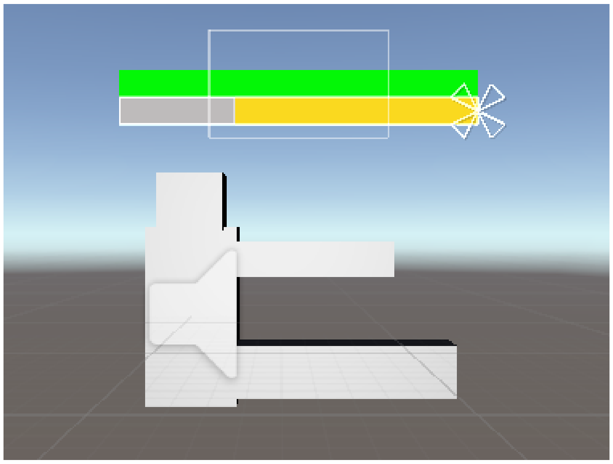


Figure 5: A filling stamina bar

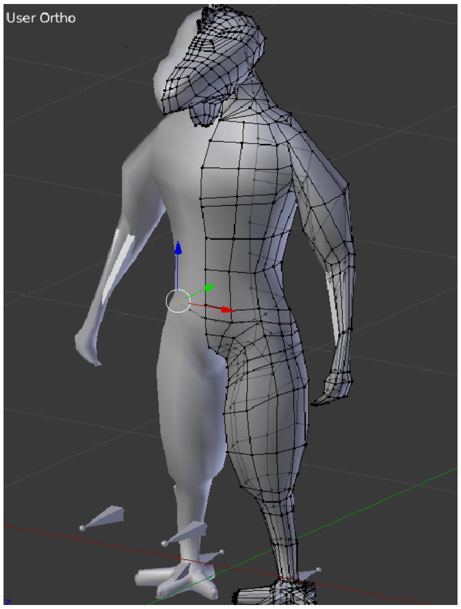


Figure 6: The creation of the model itself in Blender

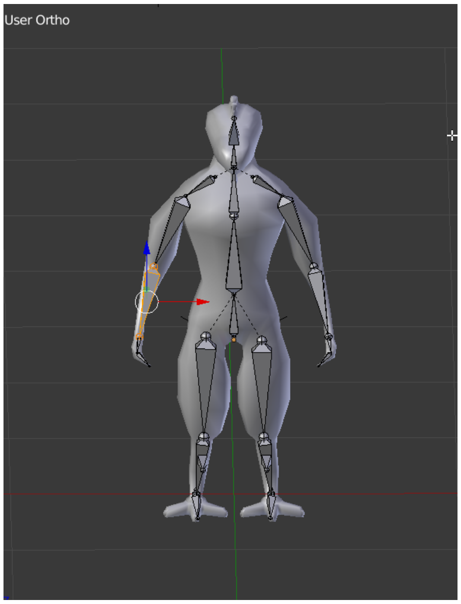


Figure 7: The placement of the bones

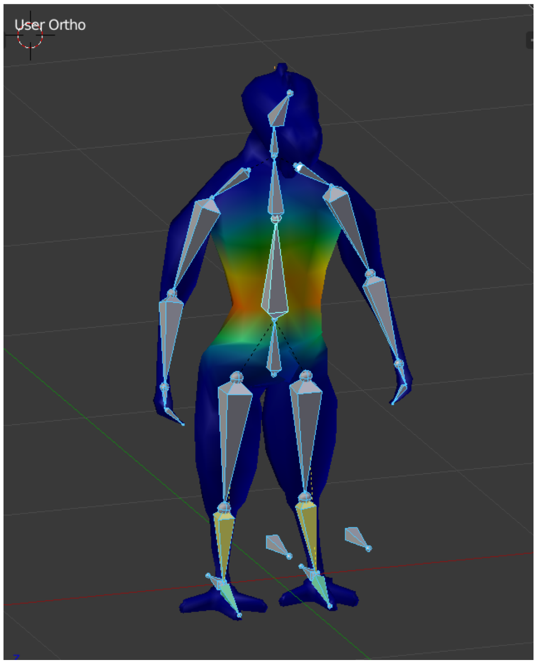


Figure 8: The weight painting

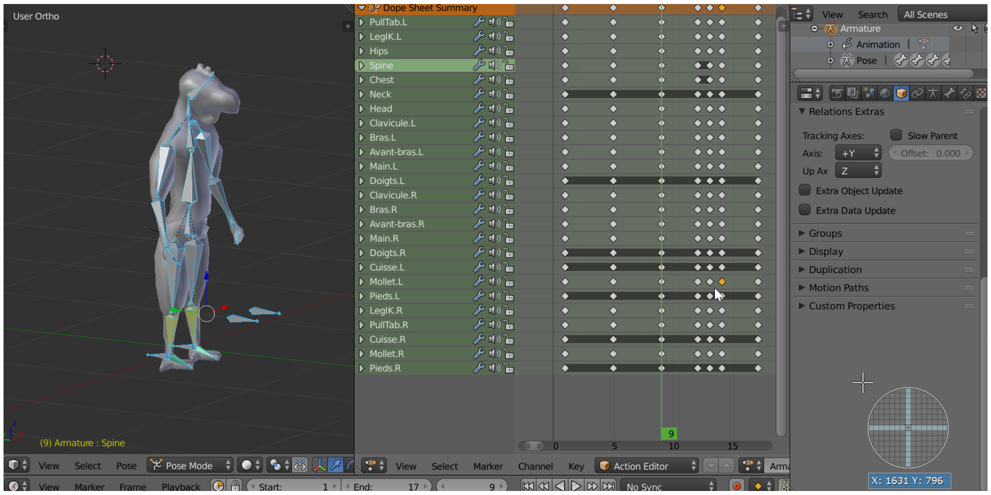


Figure 9: The making of the animations in Blender

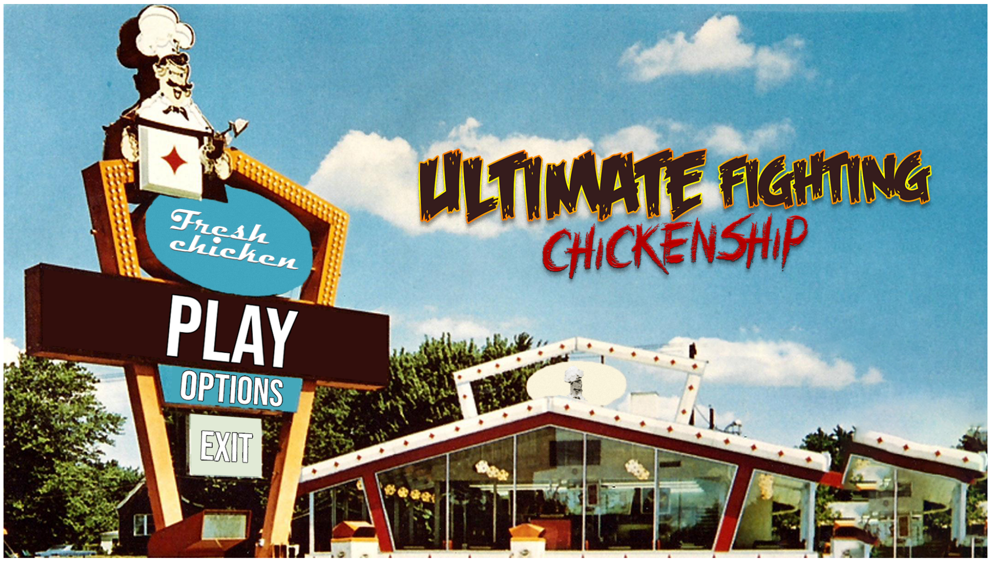


Figure 10: A screenshot of the new main menu

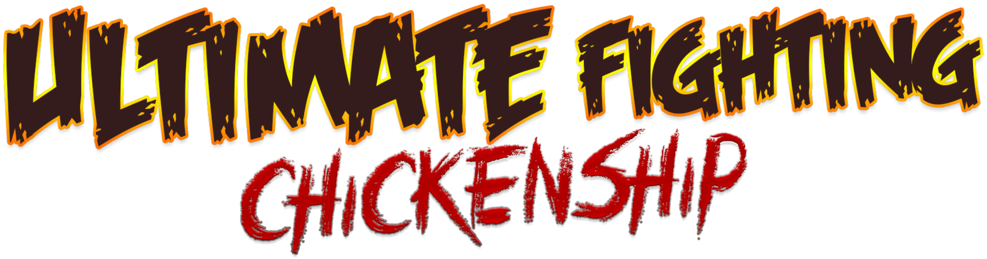


Figure 11: The two-lined logo

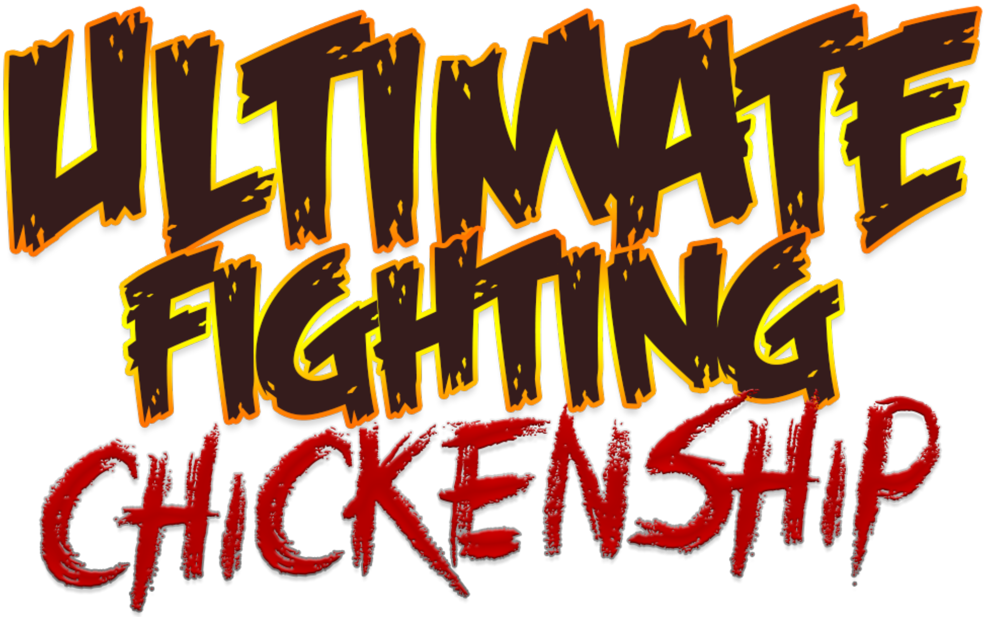
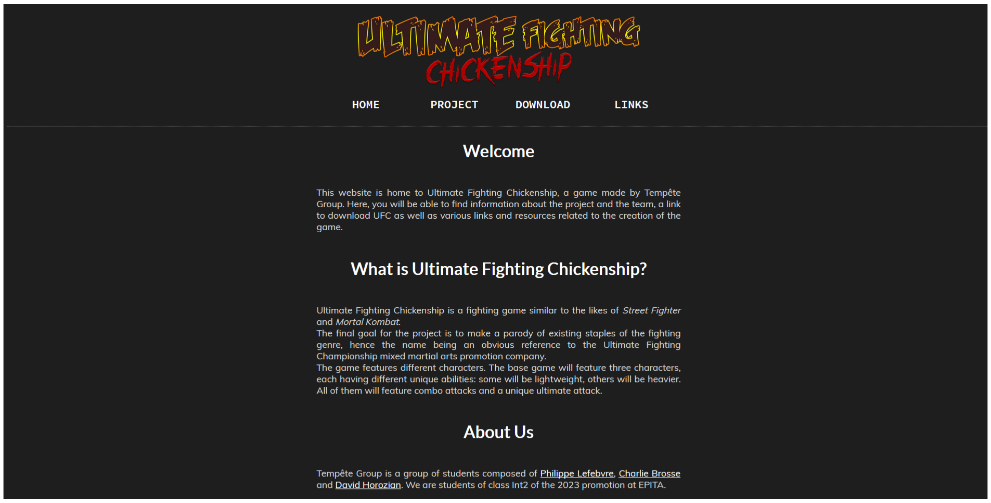


Figure 12: The three-lined logo



Welcome

This website is home to Ultimate Fighting Chickenship, a game made by Tempête Group. Here, you will be able to find information about the project and the team, a link to download UFC as well as various links and resources related to the creation of the game.

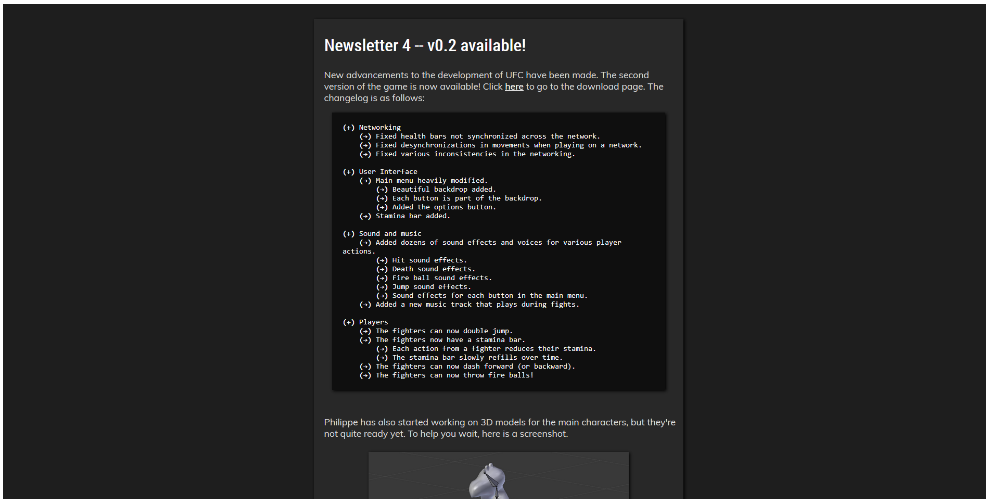
What is Ultimate Fighting Chickenship?

Ultimate Fighting Chickenship is a fighting game similar to the likes of *Street Fighter* and *Mortal Kombat*.
The final goal for the project is to make a parody of existing staples of the fighting genre, hence the name being an obvious reference to the Ultimate Fighting Championship mixed martial arts promotion company.
The game features different characters. The base game will feature three characters, each having different unique abilities: some will be lightweight, others will be heavier. All of them will feature combo attacks and a unique ultimate attack.

About Us

Tempête Group is a group of students composed of Philippe Lefebvre, Charlie Brosse and David Horozian. We are students of class Int2 of the 2023 promotion at EPITA.

Figure 13: A screenshot of the home page as of April 2019



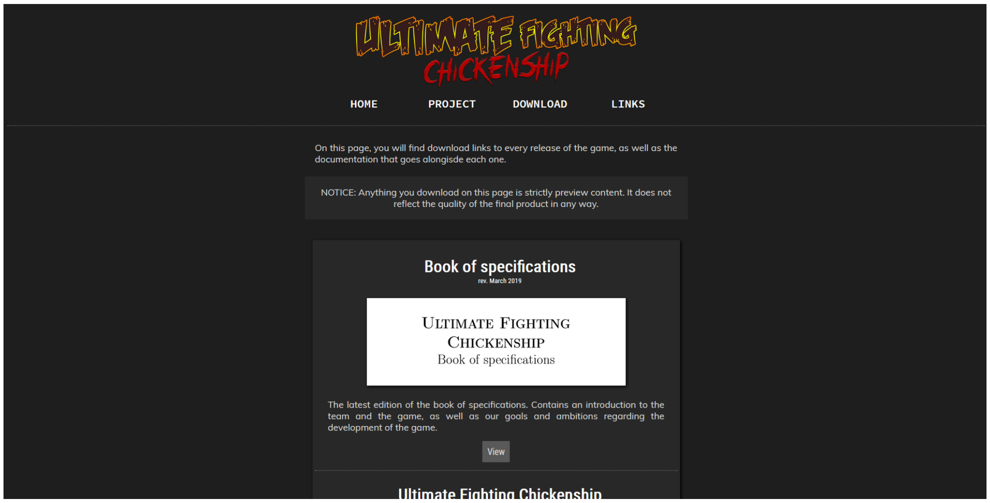
Newsletter 4 -- v0.2 available!

New advancements to the development of UFC have been made. The second version of the game is now available! Click here to go to the download page. The changelog is as follows:

```
(+) Networking
    (→) Fixed health bars not synchronized across the network.
    (→) Fixed desynchronizations in movements when playing on a network.
    (→) Fixed various inconsistencies in the networking.

(+) User Interface
    (→) Main menu heavily modified.
        (→) Beautiful backdrop added.
        (→) Each button is part of the backdrop.
        (→) Added the options button.
    (→) Stamina bar added.

(+) Sound and music
    (→) Added dozens of sound effects and voices for various player
actions.
        (→) Hit sound effects.
        (→) Death sound effects.
        (→) Fire ball sound effects.
        (→) Jump sound effects.
        (→) Sound effects for each button in the main menu.
    (→) Added a new music track that plays during fights.

(+) Players
    (→) The fighters can now double jump.
    (→) The fighters now have a stamina bar.
        (→) Each action from a fighter reduces their stamina.
        (→) The stamina bar slowly refills over time.
    (→) The fighters can now dash forward (or backward).
    (→) The fighters can now throw fire balls!
```

Philippe has also started working on 3D models for the main characters, but they're not quite ready yet. To help you wait, here is a screenshot.

Figure 14: A screenshot of the project page as of April 2019



On this page, you will find download links to every release of the game, as well as the documentation that goes alongisde each one.

NOTICE: Anything you download on this page is strictly preview content. It does not reflect the quality of the final product in any way.

Book of specifications

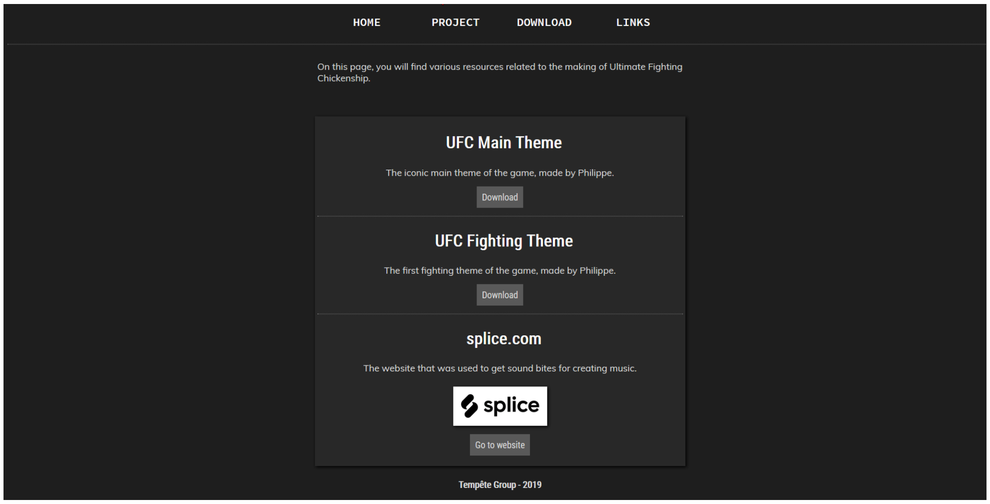
rev. March 2019

Ultimate Fighting
Chickenship
Book of specifications

The latest edition of the book of specifications. Contains an introduction to the team and the game, as well as our goals and ambitions regarding the development of the game.

Ultimate Fighting Chickenship

Figure 15: A screenshot of the download page as of April 2019



On this page, you will find various resources related to the making of Ultimate Fighting Chickenship.

UFC Main Theme

The iconic main theme of the game, made by Philippe.

UFC Fighting Theme

The first fighting theme of the game, made by Philippe.

splice.com

The website that was used to get sound bites for creating music.

Tempête Group - 2019

Figure 16: A screenshot of the links page as of April 2019