

Project report

Kronos Agon^{tn}

By Atlas Studio

Ugo PARODAT (*Leader*) Lucien TEILLET Nathan BABEC Paul GUIHENEUC
Jeremie THEVENIN



Contents

1	Introduction	4
1.1	Reminder	4
1.2	Table	5
1.3	Methods in place	6
1.4	Adjustments	6
2	Project progress	7
2.1	Lore	7
2.2	Gameplay mechanics	8
2.2.1	Player Movements	8
2.2.2	Turns	9
2.2.3	Indicator	9
2.2.4	Multiplayer	11
2.2.5	LAN	12
2.3	Interface and Functionalities	14
2.3.1	Game Menu	14
2.3.2	HUD and UI	16
2.3.3	Player indicators	16
2.4	IA's	17
2.4.1	Enemies	17
2.4.2	Player NPC	17
2.5	Graphics	18
2.5.1	Maps	18
2.5.2	Assets	19
2.5.3	Characters	22
2.5.4	Animation's	23

2.6	Website	25
2.6.1	Theme	25
2.6.2	Characteristics and functionality	26
2.6.3	Responsive	28
2.7	Music and SFX	29
2.7.1	Music	29
3	Progress review	31
3.1	Future development	32

1 Introduction

1.1 Reminder

Our project is a 2D game set in the universe of Greek gods, featuring frenetic and rhythmic combat where tempo is key. Entitled Kronos Agon, which means "battle of time," it re imagines the concept of a party game through intense solo matches against non-player characters (NPCs) or multiplayer sessions via machines connected to the same network (LAN).

The gameplay is based on a unique concept: rhythm mastery. Each Greek god represented in the game has a distinct identity and a set of powers inspired by mythology, but the effectiveness of these powers depends entirely on the player's ability to follow rhythmic sequences. The battles, therefore, become a frenzied dance where rhythmic precision and reactivity are the keys to victory.

In terms of level design, the arenas will be dynamic, adapting to the players' actions and the progression of the match. Each environment will influence the rhythm of the battles: a fight in the Underworld will be slow and heavy, while a duel on Mount Olympus will be faster and unpredictable, reflecting the divine energy of the location.

The objective of the Kronos Agon project is to allow the members of Atlas Studio to diversify their technical knowledge, particularly in C sharp, 2D graphics, game design, and network communication (LAN).

Ultimately, our quest is not just about creating a video game. It is also about our individual and collective growth, our ability to collaborate, overcome challenges, and never lose sight of our common goal: successfully completing our project.

1.2 Table

Task	In charge	Alternate
Base code	Lucien	Ugo
IA	Lucien	Ugo
Story	Lucien	Ugo
Web Site	Ugo	Nathan
LAN	Ugo	Jeremie
Animations	Ugo	Lucien
Music	Nathan	Lucien
Characters	Nathan	Jeremie
Publicity	Nathan	Paul
Maps	Paul	Jeremie
Menus	Paul	Nathan
Game cover	Paul	Nathan
Sound Effects	Jeremie	Paul
Testing and launch	Jeremie	Paul

Table 1: Planning of each task and who is in charge

1.3 Methods in place

For the successful execution of our project, we established several tools and strategies to ensure effective collaboration and continuous tracking of progress.

First and foremost, we utilized GitHub as our primary platform for collaborative work. By using GitHub, we were able to work simultaneously on the same files, making it easier to manage contributions from multiple team members. The platform’s built-in tools allowed us to track changes, review code, and address potential conflicts in a structured and efficient manner. This ensured that all team members could contribute without the risk of overwriting each other’s work, fostering a streamlined and productive workflow.

In addition to GitHub, we relied heavily on Discord for communication and real-time collaboration. Discord served as our central hub for exchanging ideas, brainstorming solutions, and discussing the project’s ongoing progress. The platform’s versatility allowed us to organize our discussions into specific channels, making it easier to separate technical conversations, general updates, and task assignments.

Moreover, Discord became an essential space for documenting and sharing information related to the project. It was on this platform that we kept track of all updates, including progress milestones, challenges encountered, and tasks that remained to be completed. By using Discord as a repository for this information, we ensured that every team member had access to the latest developments and could contribute to problem-solving and decision-making effectively.

In summary, the combination of GitHub for technical collaboration and Discord for communication and information sharing created a robust framework for our team to work cohesively. These tools not only helped us stay organized but also promoted transparency, accountability, and teamwork throughout the entire project.

1.4 Adjustments

Numerous adjustments had to be put in place due to overconfidence in the time we had to put together the project. Hence we had to modify some of general characteristics of the game all of the changes will be specified in the document parts of the corresponding tasks.

2 Project progress

2.1 Lore

The story is an important part of our game, even though understanding it is not necessary for the player to grasp the gameplay. We have already outlined the main thread of the story to establish the atmosphere of the various game levels that the player will explore. The following is the main lore that will be updated for each map and character we create :

Every 500 years, an event of extraordinary cosmic significance shakes the very foundations of the heavens: the Olympian Ascension. This grand millennial trial is a divine tournament where new gods, born from the collective hopes, fears, and dreams of mortals, fight for the honor of joining the ranks of Olympus and securing their place among the immortals.

The tournament is a spectacle unlike any other, consisting of a series of divine trials crafted by the Olympian god entrusted with overseeing the event. For this edition, the responsibility falls upon none other than Apollo, the radiant god of music, poetry, and harmony. Each presiding god leaves their unique imprint on the tournament, and Apollo is no exception. Staying true to his essence, he has chosen a theme that embodies his artistic soul: rhythmic and frenetic battles that test not only strength and strategy but also the ability to resonate with the celestial melodies of the cosmos.

Contestants must prove themselves worthy by aligning their divine essence with the pulse of the universe, moving in perfect harmony with its grand symphony. Only those who can attune their hearts to the cosmic rhythm will stand a chance of claiming victory.

Who among these aspiring deities will rise to the occasion, conquer the trials, and ascend to the divine glory of Olympus? The answer lies in the music of the universe and the resolve of those bold enough to dance to its tune.

2.2 Gameplay mechanics

2.2.1 Player Movements

In Kronos Agon, player movement is rhythm-based, requiring precise inputs in time with the beat of the game's music, where missing the rhythm can leave the player vulnerable. Players move one square at a time in four cardinal directions—up, down, left, and right.

Here is part of the player movement code that takes care of the rhythmic movement:

```
globalFrameCounter++;
if (globalFrameCounter >= 60)
{
    >| globalFrameCounter = 0;
    >| hasMoveSaved = false; // Reset saved move every interval
}

else if (!hasMoveSaved && globalFrameCounter >= 25 && globalFrameCounter <= 35)
{
    >| Vector2 direction = Vector2.Zero;
    >|
    >| if (Input.IsActionJustPressed("ui_left"))
    >| >| direction = new Vector2(-1, 0);
    >| else if (Input.IsActionJustPressed("ui_right"))
    >| >| direction = new Vector2(1, 0);
    >| else if (Input.IsActionJustPressed("ui_up"))
    >| >| direction = new Vector2(0, -1);
    >| else if (Input.IsActionJustPressed("ui_down"))
    >| >| direction = new Vector2(0, 1);

    >| else if (globalFrameCounter == 36 && hasMoveSaved)
    >| {
    >| Vector2 newPosition = GlobalPosition + savedMoveDirection * GridSize;
```

The frame-by-frame movement system is designed to ensure controlled and predictable player movement while enforcing timing restrictions. At its core, the system operates using a global frame counter (`globalFrameCounter`), which increments every frame within the `PhysicsProcess()` function. This counter resets every 60 frames, corresponding to one second in a standard 60 FPS game loop. The purpose of this counter is to define specific time windows during which player input is processed, preventing spammed or uncontrolled movement.

The player's movement is checked only between frames 25 and 35. During this period, if the player presses a movement key, the desired direction is stored in `savedMoveDirection`, but the player does not immediately move. Instead, the input is saved, ensuring that movement only occurs at a designated time. This approach helps to synchronize movements with a fixed interval, giving the game a structured, grid-like feel.

At exactly frame 36, the saved movement direction is executed. The player's `targetPosition` is updated based on the stored direction, moving exactly `GridSize` pixels in that direction. The movement is then performed smoothly using `MoveToward()`, ensuring that the transition to the new position is fluid rather than instantaneous. Once movement completes, the `hasMoveSaved` flag is reset, preventing multiple movements in a single cycle. This ensures that each player action follows a precise timing structure, making movements deliberate and eliminating the possibility of rapid, unintended inputs.

Finally, movement boundaries are enforced by restricting `targetPosition` updates to stay within the defined arena dimensions (`ArenaWidth` and `ArenaHeight`). This prevents the player from leaving the playable area, ensuring logical constraints on movement. Overall, this structured approach creates a frame-perfect, controlled movement system that balances input responsiveness with timing constraints, making it ideal for grid-based or turn-like mechanics.

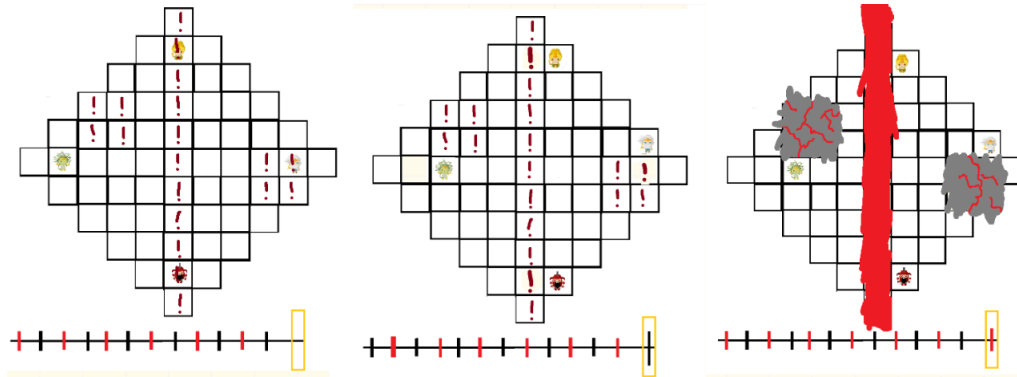
2.2.2 Turns

Core Concept

Beat Cycle: The game operates on a sequence of "beats" (or turns), each representing a distinct game phase. On odd-numbered beats: The player takes an action (move, attack, use an item). On even-numbered beats: The map processes events (enemy movement, environmental hazards, spawns, traps).

Alternating Actions: Players and map elements never act simultaneously, ensuring a clear cause-and-effect relationship between the player's choices and map changes. Every action feels deliberate, as the player must plan for potential map events occurring after their movement.

Here is how a 3 beat interval a round could look like:

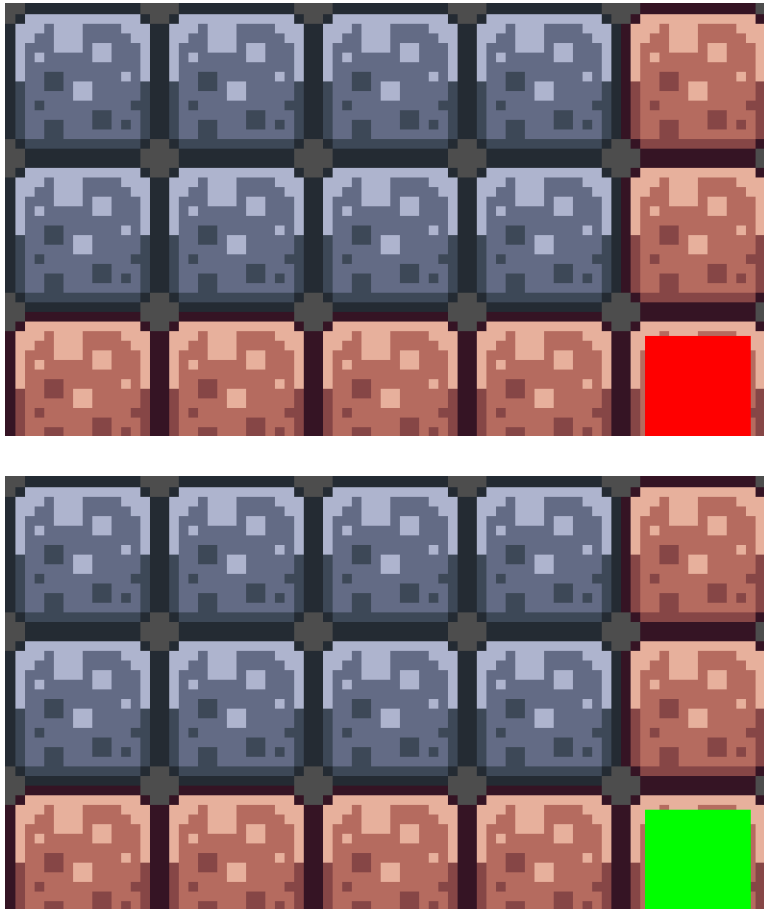


2.2.3 Indicator

The indicator is a basic prototype designed to help players understand when movement input will be processed. Since movement is restricted to a specific frame window (frames 25-35), the indicator visually communicates when the player should press a movement key. When this time frame is active, the indicator turns green, signaling that input will be saved. At all other times, it remains red, indicating that movement keys will be ignored.

This is the first iteration of the system and is far from the final version. In future iterations, the indicator may be redesigned with smoother animations, different visual cues, or even integrated into a more dynamic UI system. The goal is to ensure better player feedback and a more polished visual experience, but for now, this prototype serves as a functional way to test the timing mechanics.

Here is what it looks like:



And here is the code of it changes the color with the timing:

```
private static int globalFrameCounter = 0; // Global frame counter

public override void _Ready()
{
    >| globalFrameCounter = 0; // Initialize frame counter
}

public override void _Process(double delta)
{
    >| globalFrameCounter++;
    >| if (globalFrameCounter >= 60)
    >| {
    >| >| globalFrameCounter = 0; // Reset frame counter every 60 frames
    >| }
    >|
    >| if (globalFrameCounter >= 25 && globalFrameCounter <= 35)
    >| {
    >| >| SelfModulate = new Color(0, 1, 0); // Green during the interval
    >| }
    >| else
    >| {
    >| >| SelfModulate = new Color(1, 0, 0); // Red otherwise
    >| }
}
```

2.2.4 Multiplayer

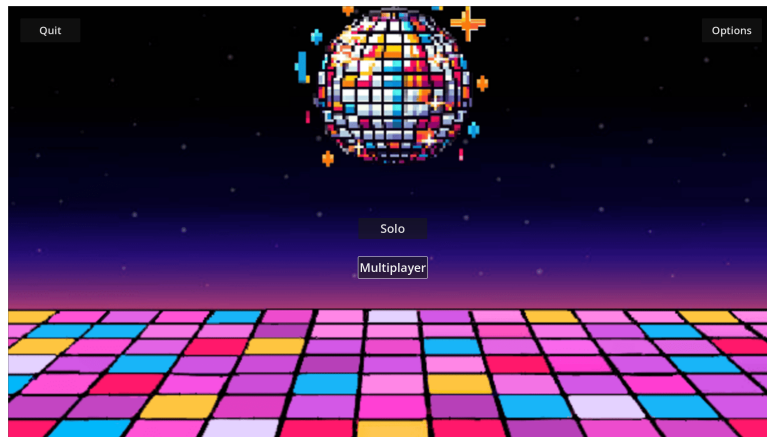
The multiplayer system is designed to prioritize couch gaming while offering LAN support for added flexibility. Up to four players can join a session, with the option for two players to share a single keyboard and the other players to use external controllers. This hybrid setup encourages a classic local multiplayer feel, where friends gather around a shared screen, but it also allows players on different devices to connect through a local network.

In a local setup, players on the same machine can configure their inputs to ensure seamless gameplay. For example, two players sharing a keyboard can have distinct key mappings, such as one using WASD for movement and the other using arrow keys. Controllers are also fully supported, with plug-and-play functionality and customizable input options. In a LAN environment, players on separate devices can join the same lobby, enabling mixed setups where some players share inputs on one machine while others connect remotely.

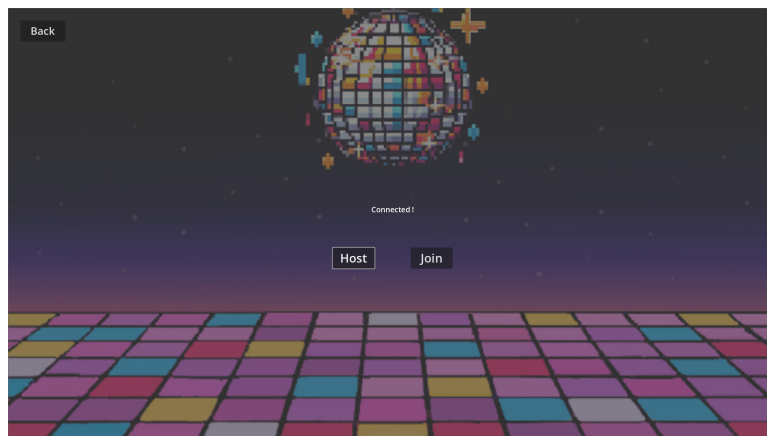
The lobby system dynamically assigns players to input devices, ensuring a smooth setup process regardless of the combination of keyboards and controllers. Once the game starts, the input management system handles simultaneous actions from all devices, maintaining responsiveness and fairness. The shared screen focuses on local players in couch gaming mode, while LAN games synchronize inputs and statuses across devices to keep everyone in sync.

2.2.5 LAN

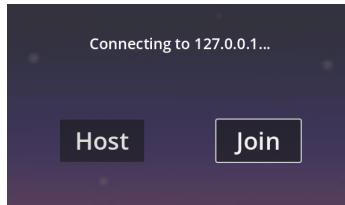
We started a LAN multiplayer system using Godot 4 and C Sharp. This allows players on the same network to host or join a game through a simple menu.



On the game menu click on the multiplayer button.



You are now in the multiplayer section, players can host a game to create a server or join one by connecting to a local IP, with the system updating the connection status on screen.



```
private void OnJoinPressed()
{
    string serverIP = "127.0.0.1";

    peer = new ENetMultiplayerPeer();
    var error = peer.CreateClient(serverIP, PORT);

    if (error != Error.Ok)
    {
        statusLabel.Text = "Connection failed";
        return;
    }

    Multiplayer.MultiplayerPeer = peer;
    statusLabel.Text = $"Connecting to {serverIP}...";

    GD.Print($"Connection attempt {serverIP}...");
}
```

When you click Join, the program creates a network client using `ENetMultiplayerPeer()` and tries to connect to the server at 127.0.0.1 on port 12345. If the connection fails, the status label displays "Connection failed"; if successful, it updates to "Connecting to 127.0.0.1..." and assigns the client to `Multiplayer.MultiplayerPeer`. The `Process()` function continuously checks the connection status, updating the label to "Connected!" once connected.

Currently, the client only connects to the same device, so to join another computer, the IP must be changed dynamically.

```
private void OnHostPressed()
{
    peer = new ENetMultiplayerPeer();
    var error = peer.CreateServer(PORT, MAX_PLAYERS);

    if (error != Error.Ok)
    {
        statusLabel.Text = "Error while creating the server.";
        return;
    }

    Multiplayer.MultiplayerPeer = peer;
    statusLabel.Text = "Server waiting for connection...";

    GD.Print("Server started !");
}
```

When hosting, the game creates a server using `ENetMultiplayerPeer()` and waits for players to join. It listens for connections on a specified port and allows up to a set number of players. If successful, the status label updates to "Server waiting for connection...", indicating that the game is ready for others to join.

For now the LAN and the game are still in development and will be our main concerns for the rest of the year, we will focus actively on it to make a good and working multiplayer game.

2.3 Interface and Functionalities

2.3.1 Game Menu

The Start Title for the game is not yet finalized but is planned to be integrated in a clear, intuitive way to ensure players can navigate effortlessly through the interface. Our aim is to create a streamlined experience, avoiding any confusion while maintaining the focus on the core gameplay. To achieve this, the Start Title will be divided into three simple and well-defined phases:

1. Settings and Start Game

The first phase will allow players to access essential game settings and start their journey. This section will feature a straightforward and user-friendly layout where players can adjust key settings such as audio preferences, graphic quality, and control configurations. Additionally, the option to start a new game or load a previous save will be prominently displayed, ensuring a smooth and efficient entry point into the game.

Here a preview of the start menu :



Figure 1: Kronos Agon Main Title by Nathan

Some weapons prompts will be displayed on the dance floor to make this battlefield-like aura and keep an eye on adding a configuration menu that fit on the style of the main title.

2. Character Selection

The second phase will focus on selecting a character, a crucial part of the player's immersion and strategy. Players will have the opportunity to choose from a variety of characters. This section will include detailed character descriptions and visual previews, enabling players to make informed decisions about which character best suits their preferred approach to the game.

(To see the character models go to section 2.5.3 Characters)

The fighter selection will use the same background as the main title (maybe blurred in the back)

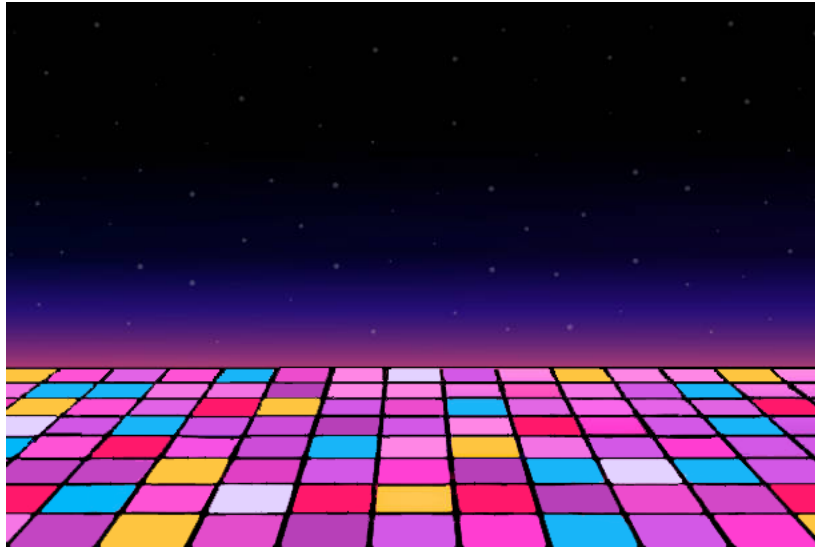


Figure 2: Kronos Agon Main Title by Nathan

3. Game Rules

In the final phase, players will be introduced to the game's rules and mechanics. This step is designed to provide a concise yet comprehensive overview of how the game works, including objectives, controls, and key strategies. The rules will be presented in a visually appealing and easily digestible format, such as icons, tool-tips, or short animations, to keep players engaged while ensuring they have a clear understanding of the game before starting.

By breaking the Start Title into these three manageable phases, we aim to create an on boarding process that feels intuitive and welcoming, allowing players to quickly dive into the gameplay without feeling overwhelmed. This structured approach ensures that every player, regardless of their experience level, can get started with confidence and enjoy the game to its fullest.



Figure 3: Kronos Agon Main Title

2.3.2 HUD and UI

The HUD (Heads-Up Display) will be designed to be simple and intuitive, ensuring that players do not feel overwhelmed or disoriented while navigating the game. This minimalistic approach will allow players to stay focused on the gameplay itself, rather than being distracted by unnecessary visual elements. One key feature of the HUD will be a tempo bar, which will play a crucial role in helping players align their moves more easily with the rhythm of the music. By providing a clear, visual representation of the beat, the tempo bar will enhance the player's ability to stay in sync with the music, adding a dynamic and engaging layer to the gameplay.

In addition to the HUD, the game's User Interface (UI) will incorporate essential menus and overlays to facilitate navigation and provide critical information. The pause menu will grant players access to options such as adjusting settings, returning to the main menu, or resuming the game seamlessly. Another important aspect of the UI will be the interface that appears between rounds. This interface will display key details, such as the scores of the players participating in the game, offering a clear overview of performance and encouraging friendly competition.

The goal is to create an interface that is not only functional but also complements the overall aesthetic and flow of the game. By striking a balance between simplicity and utility, the HUD and UI aim to enhance the player's experience without detracting from the core elements of the gameplay.

2.3.3 Player indicators

To provide a balanced and comprehensive gameplay experience, the game will include additional visual indicators on the game tiles that work in harmony with the tempo bar displayed in the UI. These tile-based hazard indicators are designed to alert players to any imminent dangers or obstacles present on the map, giving them a clear understanding of potential threats. By doing so, players will have the opportunity to anticipate and respond to hazards in advance, reducing the likelihood of dying unexpectedly or feeling frustrated by unforeseen challenges.

These indicators will be seamlessly integrated into the game's visual design, ensuring that they are noticeable without being overly intrusive or detracting from the immersive atmosphere of the map. Whether it's a visual cue such as a glowing effect, a change in color, or an animation that highlights an approaching hazard, the goal is to communicate critical information to the player in a way that feels natural and intuitive.

This system not only enhances the overall fairness of the game but also empowers players to develop strategies and improve their reaction times, as they learn to interpret and act upon the visual signals provided by the hazard indicators. By combining these tile-based warnings with the tempo bar, the game strikes a balance between rhythm-based gameplay and situational awareness, ensuring that players remain engaged and challenged without feeling overwhelmed or confused.

Ultimately, these features work together to create a more enjoyable and rewarding experience, allowing players to focus on honing their skills and navigating the game world with confidence and precision.

2.4 IA's

Artificial Intelligence (AI) is responsible for managing the behavior of the monsters. It plays a crucial role in the gameplay of our game. Indeed, the monsters' AI is programmed to follow the player, creating an immersive experience.

Additionally, a feature has been added to the AI that allows it to follow the player, which can be useful for luring monsters into traps and kill them in intense rush moments.

Currently AI has not yet been implemented in our project. According to our detailed planning and timeline, work on this aspect is scheduled to begin later in the year, allowing us to ensure proper development and integration at the appropriate stage.

2.4.1 Enemies

A wide variety of enemies could be implemented efficiently with the assistance of a specialized library, such as Navmesh or an equivalent designed for 2D environments. These enemies could feature diverse and unique patterns, attacks, and weaknesses, adding depth and strategic complexity to the gameplay.

Additionally, the monsters' AI is programmed to actively attack the player, contributing to an intense sense of fear and heightened tension, particularly during high-pressure rush moments. This design ensures that every encounter feels dynamic, challenging, and immersive for the players.

2.4.2 Player NPC

When a player is absent from the party or gets disconnected during gameplay, an AI will seamlessly take their place to ensure the room can still be completed without interruption. This AI will be designed with varying levels of difficulty, allowing it to adapt to different scenarios and skill levels. By offering customizable challenges, this feature aims to make the game more dynamic, competitive, and entertaining for all players involved, even when unexpected disruptions occur.

2.5 Graphics

The graphical aspect of the project plays a vital role in bringing the game world to life and immersing players in its atmosphere. From detailed tile designs to dynamic environmental assets, every element has been carefully crafted to reflect the unique themes and settings of the game. By focusing on a cohesive visual style, the graphics aim to enhance the storytelling, gameplay experience, and overall appeal of the game.

This section of the report will cover the key graphical components, including tile designs, character visuals, environmental assets, and plans for future visual enhancements. Together, these elements contribute to creating a captivating and engaging world for players to explore.

There is obviously still work to be done on the project, and while progress is ongoing, it is important to note that no delays have been officially declared at this time.

2.5.1 Maps

First and foremost, the game "Kronos Agon" is planned to feature at least two distinct maps, each offering a unique and immersive experience. One map will be inspired by Mount Olympus, the legendary throne and dwelling place of Zeus, while the other will be based on Tartarus, the dark and ominous underworld from Greek mythology.

For the Mount Olympus map, we envision creating a simple yet visually captivating design characterized by light and clear colors, along with a stunning and picturesque landscape that reflects the grandeur of the gods' domain. The layout will be straightforward, taking the form of a rectangular grid with dimensions of 12 by 8 tiles, providing a balanced and approachable space for gameplay.

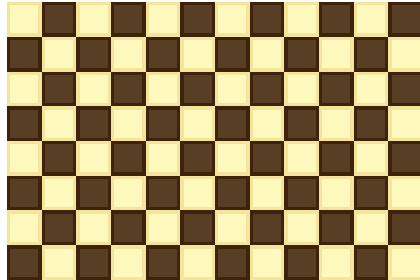


Figure 4: Board of the God's houses

For the Tartarus, we envisioned creating a map that exudes a sense of heaviness and despair, utilizing a palette of dark, muted colors to effectively convey the difficulty and oppressive atmosphere of the environment. This map is designed not only to challenge players but to immerse them in a foreboding and tense experience, making every step feel like a struggle.

Additionally, we plan to introduce new elements that will diversify and enrich the gameplay experience, offering players a refreshing change in the game's dynamics. One of the exciting ideas we're considering is the inclusion of a lava river that flows across the map, adding an extra layer of danger and complexity. As we continue to develop the game, there are other potential features and environmental changes that may be integrated to further enhance the challenge and keep the gameplay feeling fresh and engaging for players in the future.

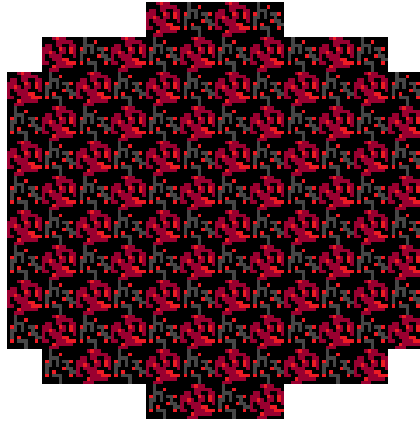


Figure 5: Lava board

2.5.2 Assets

I have designed a wide variety of tiles, some of which are currently being used in the Mount Olympus map, while others are specifically crafted for the Tartarus map. In addition to these, I've also developed a number of other tile designs that are intended for future use, either to be incorporated into new maps or to replace older tiles that may no longer fit the evolving aesthetic and gameplay needs. These new tile designs offer greater flexibility, enabling us to seamlessly introduce fresh environments, enhance the visual appeal, and improve the overall gaming experience as we continue to expand and update the game.

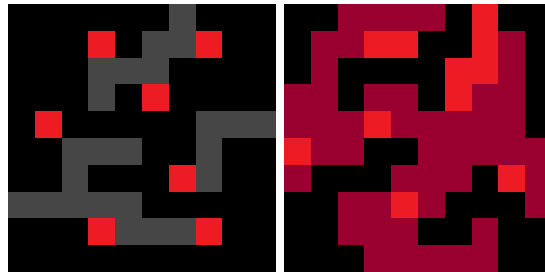


Figure 6: Tiles for Tartarus Map

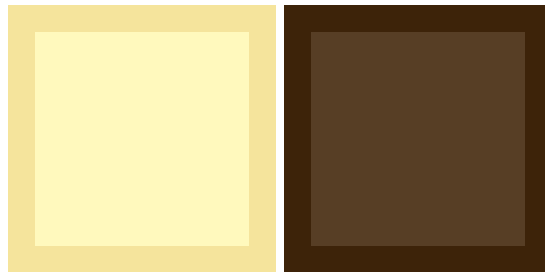


Figure 7: Tile white and brown

I have designed tiles in white, brown, and yellow (including broken versions) to emphasize the battlefield aesthetic of the Mount Olympus map, reflecting the intense and dynamic nature of the environment. These color and texture choices help convey the grandeur and chaos of Mount Olympus as a legendary battleground.

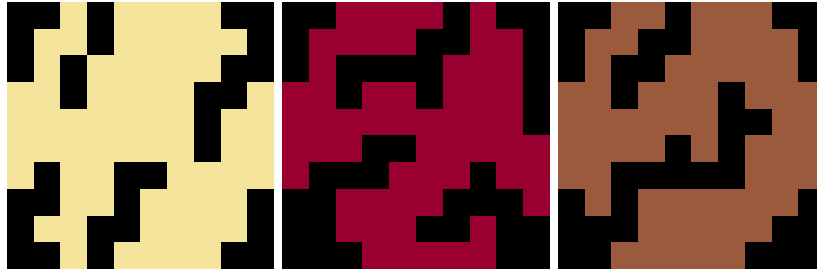


Figure 8: Tile white, brown and yellow (broken version)

In addition to the tiles, I have also worked on creating various assets such as obstacles and decorative objects. These elements not only enhance the visual appeal of the maps but also contribute to the overall gameplay experience by adding depth, interactivity, and storytelling to the environments.

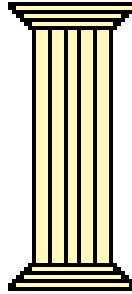


Figure 9: Light pillar

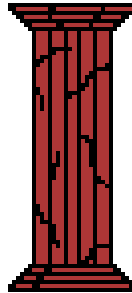


Figure 10: Red broken pillar

One set of these assets is tailored specifically for the Mount Olympus map, while the other is designed for the Tartarus map, ensuring that each environment has its own unique identity and style. For Mount Olympus, we plan to enhance the map's diversity by potentially designing a broken marble pillar in the future. This addition would introduce variety to the existing pillars, further emphasizing the map's ancient, war-torn atmosphere and adding to the visual storytelling of the legendary battleground.

As we carried on with the game's development, we redesigned our tiles to be more clear and visible for the player. We rethought the way to mark each separation between the tiles. We made a 4-pixel-wide darker line with a bigger contrast of color to answer that problem.

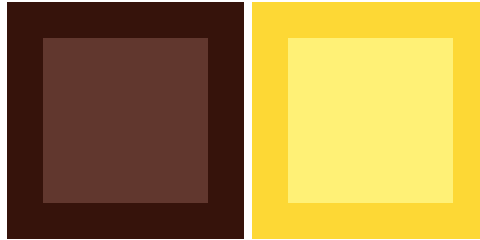


Figure 11: New Red and Grey Broken Tiles

We included more details to add some realism and make the environment more enjoyable for the players.

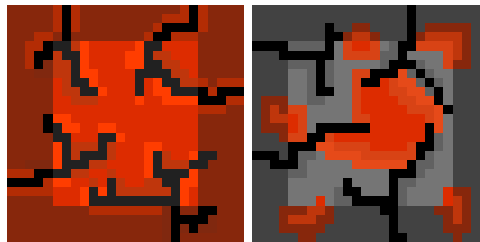


Figure 12: New Brown and White Tiles

To give players better control and feedback, we added a guiding sprite that indicates where the attack will land. This marker appears briefly before the attack is executed, helping players aim more precisely and plan their moves strategically. Whether it's a glowing reticle for a projectile or a shadow for a melee strike, this visual cue enhances clarity without disrupting the game's fast-paced action.

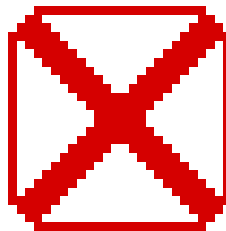


Figure 13: Attack preventing cross

2.5.3 Characters

As we continue to develop the game, introducing new characters remains one of our top priorities. These characters will bring fresh dynamics to the gameplay with their unique designs, personalities.

Two new Character Made theirs entrance to the Disco-arena !

A spartan like character an fighter in honor of the goddess Athena and a fighter of Poseidon one of the cleverest and most mischievous of Character in the roster. The two still have to find a name. Their have been made using like a base the collection of sprites of The Crypt Of The Necrodancer and shaped to our game world it was easier for Nathan to have a starting point, shapes, and measurements to not make out of bounds Characters two versions has been taken into account one for the more white maps (a black outline) and another for the more darker map (a withe outline) . Here the two fighters :



Figure 14: Spartan Fighter



Figure 15: Poseidon Fighter

2.5.4 Animation's

Alongside the new characters, we are also focusing on enhancing the game's visual and immersive aspects by introducing animations.

These animations will bring life to the characters, showcasing fluid and dynamic movements that highlight their personalities and actions.

Whether it's a character's dancing, attack sequence, or interaction with the environment, the animations are being designed to add depth and polish to the overall presentation.

These visual upgrades are intended to make the game world feel more alive and engaging, ensuring that every moment, from battles to exploration, is as captivating as possible.

The animation process as started like seen in the image above there is an idle form of the fighter (that's why there are four of each version) Nathan is currently learning of how animate in Photoshop to ease the jobs of implementing the character in the final game.

In a pixel art game, attack animations play a crucial role in making the gameplay feel responsive and satisfying, even with a limited number of frames. By focusing on key visual elements, we can create impactful effects that enhance the gameplay experience without requiring overly complex animation.

Creating attack animations for a pixel art game requires a balance between fluidity and efficiency, especially when working with limited frames. For a dynamic feel, we designed a few distinct attack effects, including a sharp thunder strike and a vibrant rainbow beam, and a small 1x1 explosion. The thunder animation consists of just a few frames, emphasizing a sudden flash followed by jagged energy dispersing outward, giving it a quick and impactful look. Meanwhile, the rainbow beam, though minimal in frames, shows the different colors of the rainbow as it shoots forward, creating a magical yet concise effect. The tiny 1x1 explosion adds an extra touch of feedback, giving attacks a subtle yet satisfying impact. By carefully timing each frame and using contrast effectively, these animations maintain a strong visual presence without overwhelming the game's pixel aesthetic.

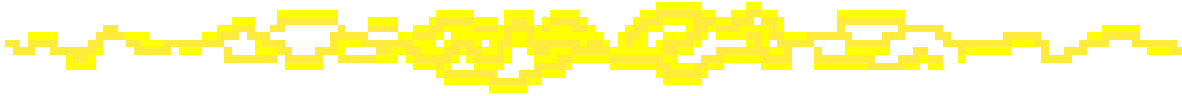


Figure 16: Different frames of the lightning attack



Figure 17: Different frames of the Rainbow beam

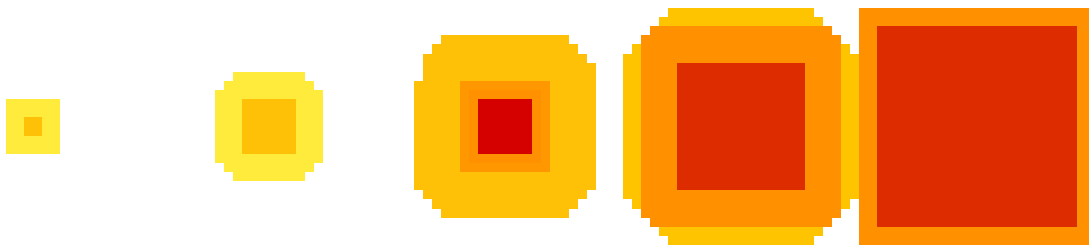
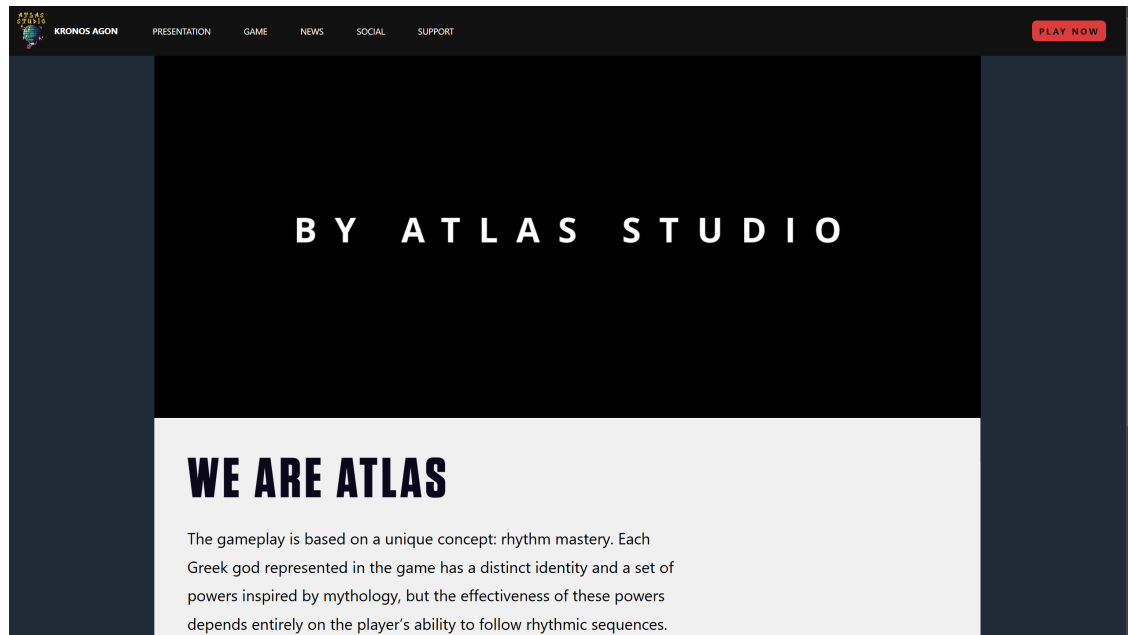


Figure 18: Different frames of the 1x1 Explosion

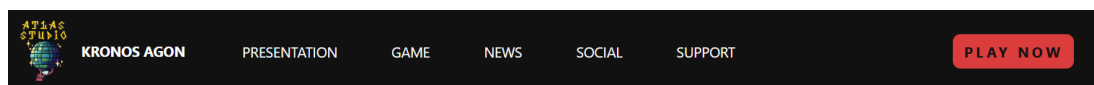
2.6 Website

2.6.1 Theme

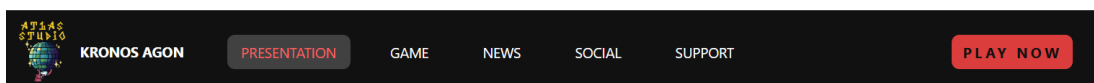
The Website is a important part of our project since it represents our company Atlas Studio and our game Kronos Agon. To do so I will be using HTML/CSS, on Visual Studio Code, which are two programming language used to create and design Website.



Here is the progress of our website. As we can see, we have started building its structure. The colors will be changed later, as this is currently just a prototype representing the architecture and layout of the different parts of our site. For example, the dark gray rectangle is intended to host an introductory video of our game to encourage users to install it, while the white rectangle below could be used for a brief presentation of our studio...



We can see the navigation menu that allows us to access the various pages of the website and the "play now" button that redirects directly to the download page.

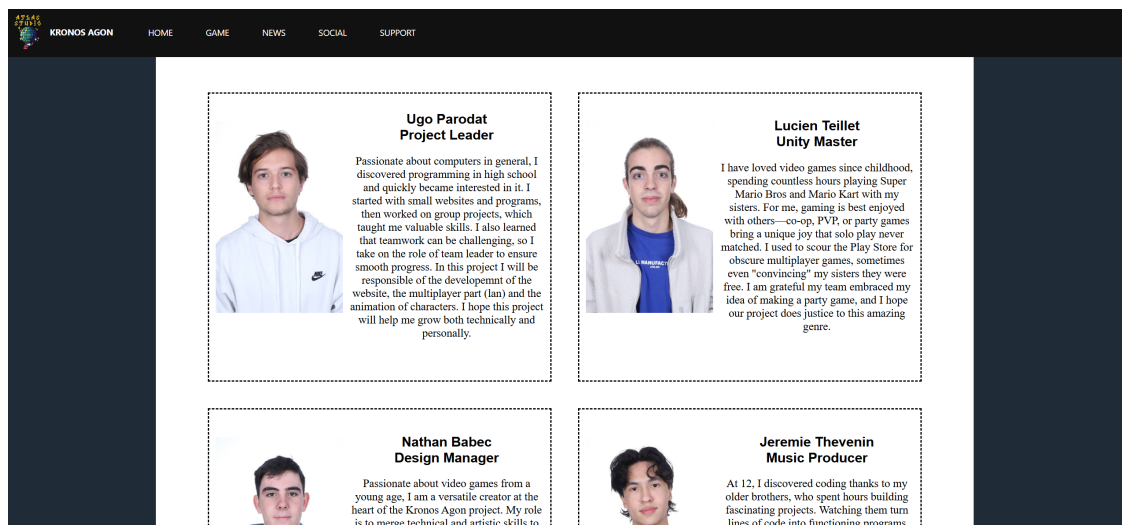


The buttons have a slight animation when hovering over them: the text turns pink and a light gray border appears around them, all with a transition of 0.3 seconds to create a smooth and pleasant effect for the eye.

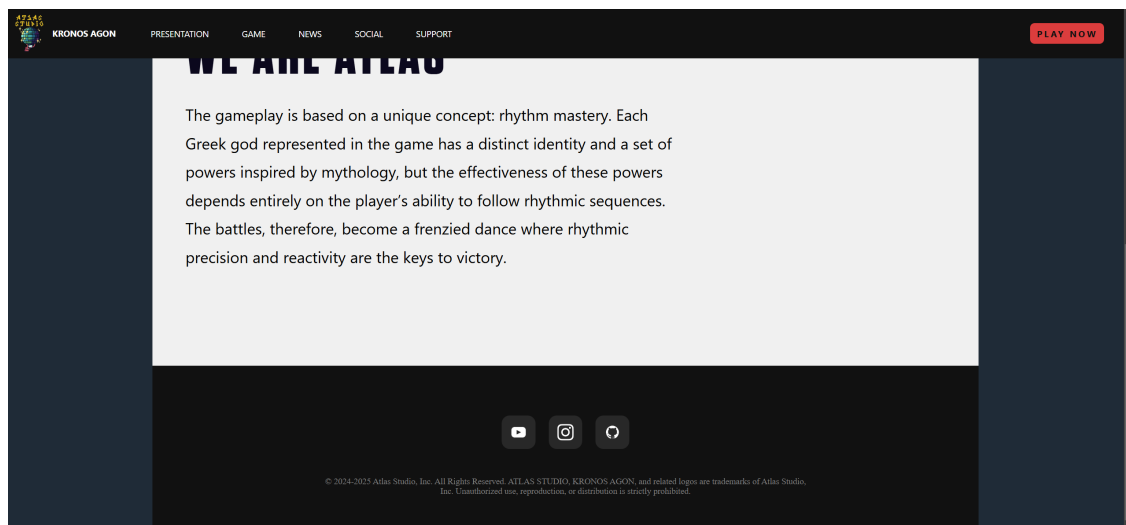
2.6.2 Characteristics and functionality



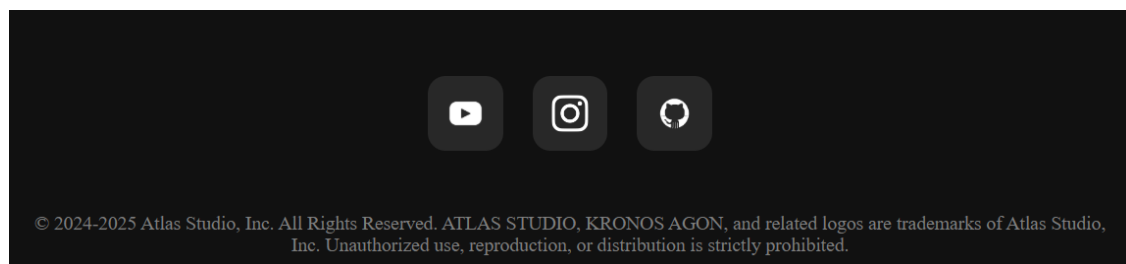
The first thing we will see on our website will be this small teaser video made by Jeremie that will give a preview of our game. For now we only have 5 seconds of video that displays "BY ATLAS STUDIO" and disappears with a smooth transition.



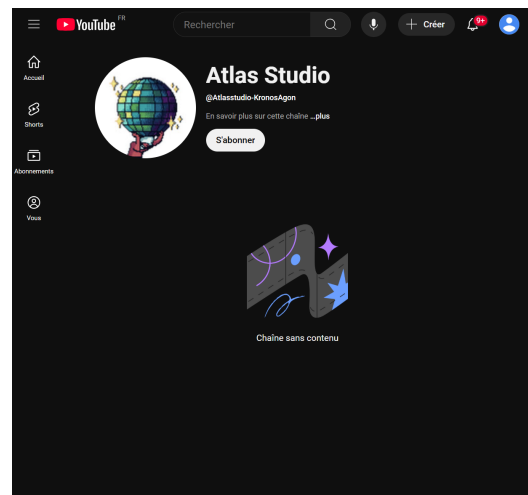
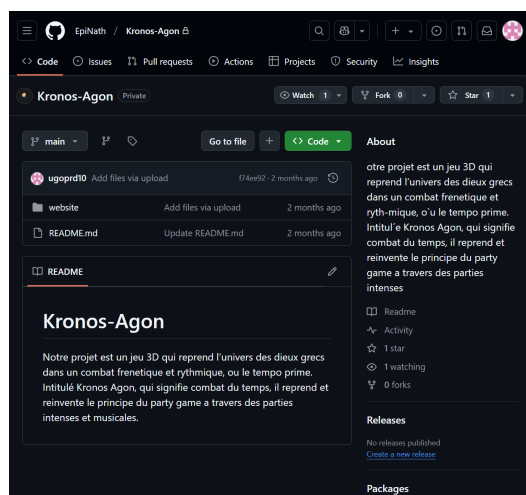
Using the navigation menu, we can go to the presentation part. This part introduces the different members of our Studio, each one responsibility in the project, our goals, our skills, etc. For now, the architecture is complete, but the colors and placement will be enhanced so that the page feels smoother.



At the end of each page, we can find the copyright section and our socials.

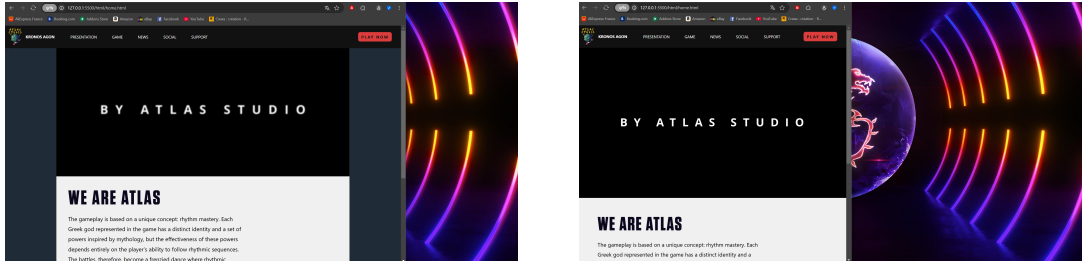


When we click on one of the three logos we are redirected to the social media. For now the only one that does not work is instagram since we have not yet started it but we created the github and the youtube channel.



2.6.3 Responsive

Atlas Studio's website will adapt to all devices, whether you are on computer, tablet, or smartphone, you will be able to access and appreciate the beauty of our website. This feature is possible thanks to responsive CSS. Responsive CSS is a method that allows one to adapt website pages and disposition depending on pixels size.



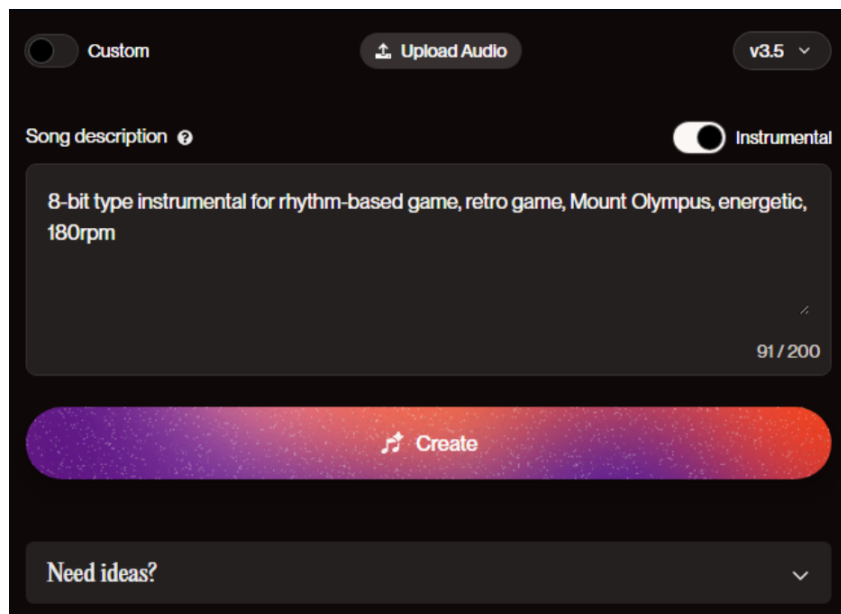
As you can see the website is different, the style is changing depending on the size of the window, this is possible with a method in CSS called @media. In CSS, @media is used to apply styles based on specific conditions, like screen size, resolution, or device type. It is part of media queries and help to make a website responsive.

```
home.html # home.css X
css > # home.css > {} @media (max-width: 950px) > .p1
243 .copyP
244 {
245     text-align: center;
246     font-size: 13px;
247     color: #7e7e7e;
248 }
249
250
251
252 @media (max-width: 950px)
253 {
254     .topbody, .midbody, .copyright
255     {
256         margin: 0;
257     }
258
259     h1.we
260     {
261         margin-left: 30px;
262         padding-top: 20px;
263         font-size: 7vw;
264     }
265
266     .p1
267     {
268         font-size: 2vw;
269         margin-left: 30px;
270     }
271 }
272
```

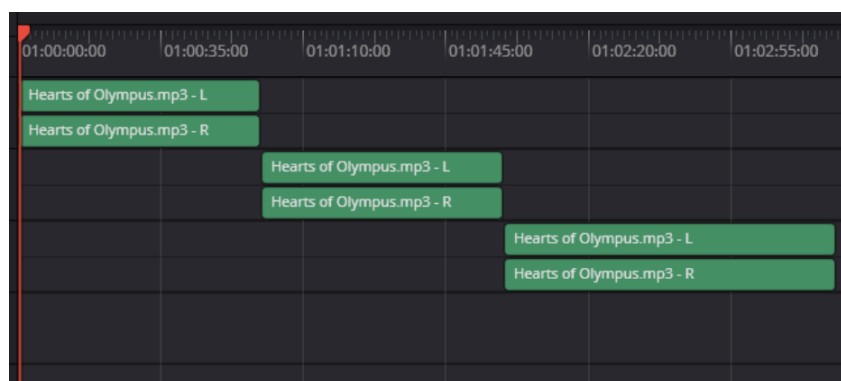
This code is the CSS of the of our first page, as we can see if the width of our screen is under 950 pixels some new CSS code replace the old one. Same thing happens the other way around, if the screen width is greater than 950 pixels, the @media is no longer effective and the CSS goes back to its original code.

2.7 Music and SFX

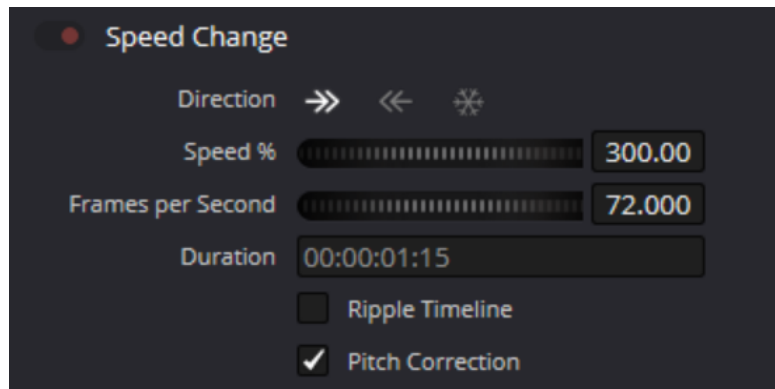
2.7.1 Music



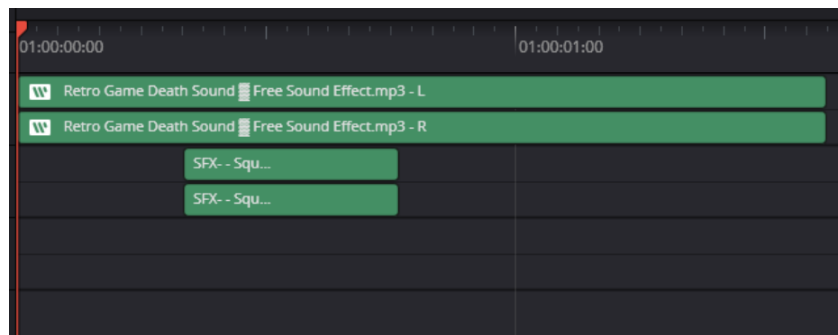
The music and sound effects are crucial elements of our game, as they enhance immersion and bring life to the different maps and environments. To create the music, I use an AI composition tool called Suno, which allows me to generate dynamic soundtracks tailored to the themes of each map.



For example, Mount Olympus features a majestic and uplifting orchestral tune, while the Underworld has a darker, ominous melody with deep bass and eerie undertones. Each map has a rising tempo that increases the difficulty of the gameplay throughout the rounds. The music is divided in 3 distinct parts each following different BPMs.

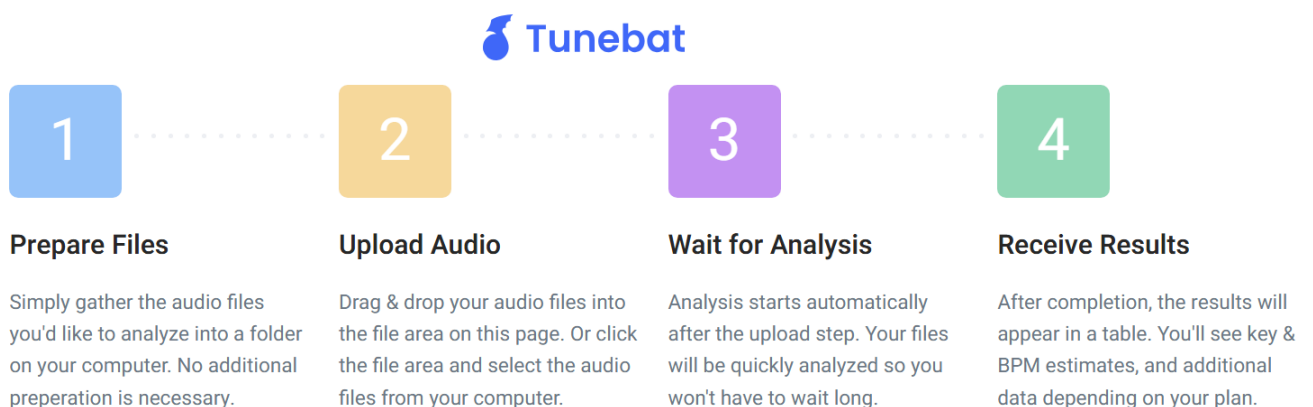


Additionally, I gathered various sound effects (SFX) from online libraries and modified them using DaVinci Resolve. For instance, I combined different sounds to finalise the sound of a character dying or I sped up or slowed down certain sounds to fit the dynamic of the game.



The editing process involved layering, equalizing, and applying filters to make the SFX feel authentic and seamless within the game world.

The final touches, like fade-ins and spatial effects, ensure an immersive audio experience that complements the gameplay perfectly.



I used an app called Tunebat to find the different BPMs of the songs and adjust them accordingly. This helped me match the tempos for a smoother mix.

3 Progress review

Gameplay Mechanics

Core gameplay focuses on rhythm-based movement and combat, where players sync actions with music beats. A grid system enables precise directional moves, while missing beats leaves players vulnerable. Turns alternate between player actions and environmental responses, encouraging strategic planning. Multiplayer supports up to four players via couch gaming or LAN, with dynamic input assignment and synchronization. Movement and combat are still in progress close to be finished meanwhile multiplayer is not yet integrated.

Interface and Functionalities

A planned three-phase game menu allows intuitive navigation through settings, character selection, and rules. The HUD features a tempo bar to align actions with beats and visual indicators for hazards, improving accessibility and game-play awareness. Nathan has finished the Main menu through it misses a configuration menu and more buttons assets had been created with the by handmade pixel art fonts for the navigation.

AI's

AI for enemies is planned to include path finding, attack patterns, and interaction with traps. Player NPCs will take over in multiplayer if a player disconnects. However, AI development has not started yet.

Graphics

Two maps are under development by Paul: Mount Olympus (bright, majestic) and Tartarus (dark, ominous), each with unique tiles and environmental elements. Assets like decorative objects and obstacles have been designed, and character and animations are being developed to add life to the game. Poseidon fighter and the Athena spartan has been added to the roster with their two idle animations by Nathan.

Website

A responsive prototype includes navigation menus, placeholder sections for videos, and hover animations. The layout adapts to different screen sizes, ensuring usability across devices. Still thing to be added but overall a good advance in the website.

Music and SFX

AI-generated dynamic soundtracks align with map themes, adjusting tempo to gameplay difficulty. Custom SFX, designed using DaVinci Resolve, enhance immersion with layered and spatial effects. Music is a central part of the project and can be changed during development depending on feedback. VFX founs are great and needs to be implemented.

3.1 Future development

Gameplay Mechanics

The rhythm system needs to be fine-tuned to make sure it works accurately and fairly for all players. Multiplayer, especially LAN, should be tested to fix any lag or syncing problems. The turn system needs balancing to keep it challenging but not too difficult, and a system should be added to handle player disconnections smoothly.

Interface and Functionalities

The game menu needs to be finished, with improvements to visuals and accessibility. The HUD and hazard indicators should be polished to make them clear, easy to use, and match the game's style. Usability testing is important to spot any issues, especially for new players.

AI's

Enemy AI needs to be built with movement, attack patterns, and the ability to interact with traps to make the gameplay more strategic. The AI should also adjust to different player skill levels. A system for replacing disconnected players with AI in multiplayer needs to be created and tested for smooth transitions.

Graphics

The work left includes finishing character animations, improving map details, and adding elements that tell a story through the environment. More assets and visual effects are needed to make the game world feel richer and more varied. All graphics should be tested to ensure they fit the game's style and run smoothly. Nathan still leaning pixel art animation through there is a start with the idle base animations of the spartan and Poseidon fighter at least two or three fighters would be great to have in the final game.

Website

The website design still needs to be completed, including adding game content like trailers, updates from the developers, and download options. The visuals, such as colors and animations, should be improved to match the game's style. It also needs to be optimized to work well on all devices, making sure it's easy to access for everyone.

Music and SFX

AI-created dynamic soundtracks match the map themes and change speed based on the difficulty of the gameplay. Custom sound effects, made with DaVinci Resolve, add to the experience with layered and maybe 3D effects.