Dev Log

Artiom Vostrenkov

512880

Table of contents

Introduction

Hello, my name is Artiom, and I am one of the engineers for our Point and Click project, working together with another engineer, two artists, and two designers. My main responsibilities are creating and hosting the website where our game will be accessible, and develop main Unity project. I am responsible for designing the overall web experience in collaboration with our designers and artists to ensure it is engaging and appropriate for our target audience, and implementing character customization and animations. During this Dev Log, it will guide you through all the processes.

Client

Our client - The Wilmink Theater in Enschede, is a cultural hub hosting diverse performances, including theater, music, and dance, enriching the city's artistic scene. The project aims to make the current teaching materials used to introduce school children to the Wilmink Theater more engaging and interactive, to help children retain theatre rules and information more effectively. Traditional materials often result in low student engagement, leading to chaos during theatre visits. To address this, the Wilmink Theater seeks to develop a playful, flexible game that can be used with or without a teacher.

Figure 1 Wilmink Theater

Project Goal

The goal is to create a fun and educational tool that keeps children engaged, ensures they remember theatre rules, and reduces disorder during school performances. The current approach employed by the teachers and theater staff is not effective for kids to retain the information. Teachers and staff take the time before and during the school visit to explain the code of conduct to the children but due to the low engagement and the stimulation of a new environment, the information doesn't stick.

This project was first initiated by Bo Hamer, during her graduation internship at the Wilminktheater. She developed a prototype for a 2D Point & Click game as a medium to show the theatre locations and introduce the children to the code of conduct at the theatre. Based on user tests she conducted, the prototype showed promising results and was passed on to our team for improving and developing the concept further.

Our Team

The team working on this project is made up of six, 4th year students from the Creative Media & Game Technologies Bachelor at Saxion University of Applied Sciences in Enschede:

- Alexis de Cazenove: Designer, Team Leader
- Jose Peiro: Designer
- Amber Kortier: Engineer
- Artiom Vostrenkov: Engineer
- Thomas Reijmerink: Artist
- Jekaterina Markova: Artist

Design Model

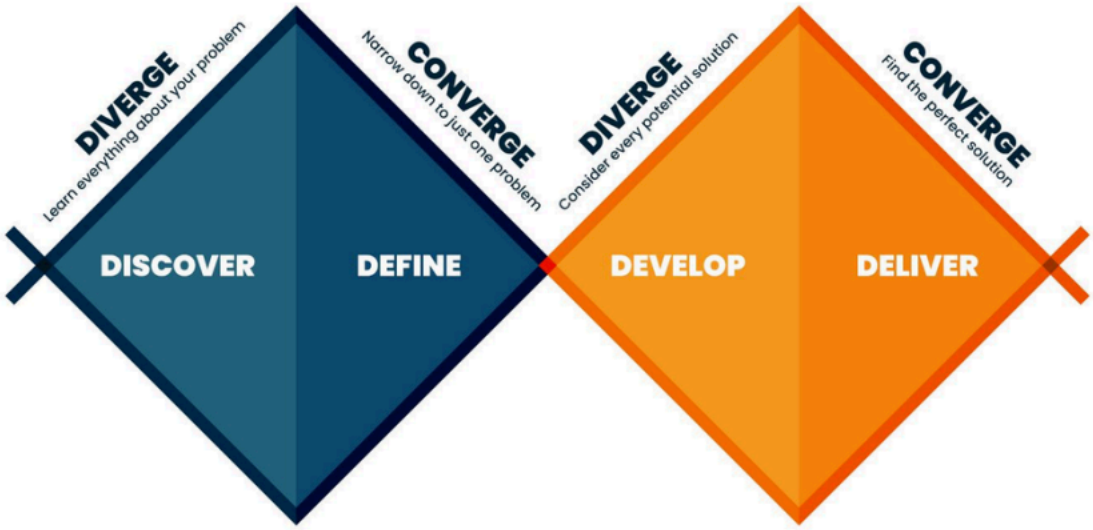


Figure 2 Double Diamond design process

To help during each phase, the Double Diamond Design Thinking Model was used as a framework for delivering the best results possible. The Double Diamond Design Model is a framework for creative problem-solving, divided into two phases: Discover and Define (divergent and convergent thinking for understanding and framing the problem) and Develop and Deliver (divergent and convergent thinking for ideating and implementing solutions). It helps teams explore a wide range of ideas before narrowing down to the best solutions. Applications include product design, UX/UI, and game development, where clear problem definition and iterative solution testing are crucial.

Concept phase

Our team chose the Double Diamond model of design thinking to guide our development process. This methodology emphasizes exploring a wide range of options before converging on the best solutions, allowing us to thoroughly research our target audience and iteratively develop our concepts. This approach is particularly crucial during the concept phase as we aim to establish not only the visual style and core gameplay mechanics of our 2D point-and-click game but also the digital platform that hosts it. The game is designed to teach children aged 5–10 the rules of the Wilminktheater.

Empathize

During the empathize phase, I focused on understanding the digital habits, preferences, and needs of children aged 5–10 to create an engaging website experience. Recognizing that simplicity and visual appeal are important for this age group, I aimed to design a website that is easy to navigate and visually consistent with the game's art style.

I considered how children interact with websites, noting that they prefer straightforward interfaces with large buttons and minimal text. Bright colors and interactive elements can capture their attention and make the experience more enjoyable.

Target audience research

During that period, Thomas (Artist) and Jose (Designer) made visits to schools to gather feedback and conduct tests with our target audience of children aged 5–10. These insights helped refine our understanding of their preferences, confirming that children are naturally drawn to colorful and visually stimulating environments. Research supports this observation, showing that bright colors capture children’s attention more effectively, while dynamic visuals and animations make the experience more engaging and enjoyable.

By focusing on these aspects, the aim was to create a digital platform that not only effectively hosts the game but also enhances the overall experience for children. This approach ensures that the website is accessible on any platform and engaging without overwhelming young users with complex navigation or unnecessary features. The findings highlight the importance of using vibrant colors and intuitive design to support cognitive and emotional development while keeping the user experience enjoyable and easy to navigate.

Simultaneously, our team explored various point-and-click game approaches and engines to find the best fit for our project, considering both what appeals to our audience and what is feasible within our time constraints. The team decided against using a custom engine due to the significant time investment required. Fortunately, our

client specifically requested a 2D game, simplifying the development process since Unity provides enough tools for building browser-compatible games.

Outcome of the target audience research

Through school visits and additional research, we established that vibrant colors and dynamic visuals are critical for maintaining young players' attention and enhancing their gaming experience. A colorful design approach not only entertains but also supports cognitive and emotional development, making it a critical consideration for creating engaging and educational games.

Defining the user's needs and pain points

Based on the research and empathy mapping, I identified the following key needs and pain points:

- **Needs:**
    - Simple and intuitive navigation.
    - Quick access to the game without unnecessary steps.
    - Visual consistency with the game's art style to maintain immersion.
    - Interactive elements that are responsive and engaging.
- **Pain points:**
    - Confusing interfaces with too many options.
    - Slow-loading websites due to heavy frameworks or unoptimized code.
    - Inconsistent visual experiences that can distract or confuse young users.

Gameplay Concept

Our research into educational games highlights their ability to engage children while promoting learning in fun and interactive ways. Games like “DragonBox” and “Prodigy Math” demonstrate the following benefits:

- **Increased Engagement**: Vibrant visuals, interactive elements, and gamified mechanics capture children’s attention, enhancing focus and motivation.

- **Educational Theories**: Games utilize behaviorist (reward systems), cognitive (problem-solving), and constructivist (open-ended exploration) approaches to make learning effective and enjoyable.

The insights presented in the image below align with our gameplay design approach. This illustration (Figure 3) from van Staalduinen and de Freitas emphasizes the connection between learning, instruction, and assessment in educational games. Key elements include:

1. **Learning Objectives**: Establishing clear player goals and relevant learning content to drive engagement.
2. **Instructional Design**: Ensuring gameplay mechanics and content are tailored to the user's learning needs and behavior.
3. **Assessment**: Providing immediate feedback and debriefing to reinforce learning and ensure the player progresses effectively.

The image (Figure 3) below visually represents how game-based learning frameworks intertwine these three key areas to create impactful educational games.

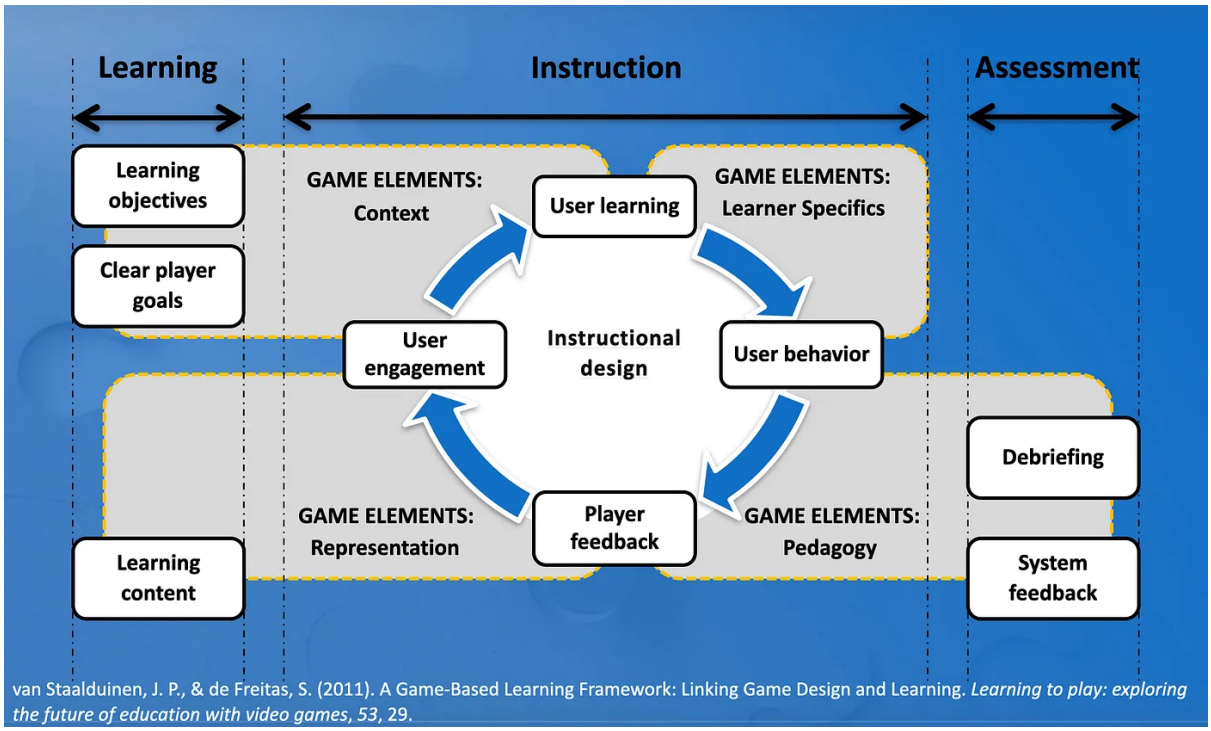


Figure 3 van Staalduinen, J. P., & de Freitas, S. (2011). A Game-Based Learning Framework: Linking Game Design and Learning. Learning to play: exploring the future of education with video games, 53, 29.

To understand what our final goal is, our team conducted existing games research with similar age groups. For inspiration we used games like "Monkey Island 2" (Figure 4) and "Toca Boca World" (Figure 5) which had things for our inspiration like:

- The games are essentially one big sequence of interactions done in the right order
- “2.5D” game perspective. The game is 2D but feels like 3D
- Most areas of the game are available to the player from the start and require the player to come back to places they already went to
- The dialogue system with other characters
- Main way to know something is interactive is by hovering the mouse over it and seeing if the cursor changes
- Interactions are generally very simple and just make the world react in some way through animations and sounds
- To help the user differentiate from the static environment, the interactive props are outlined in black to indicate that they can be interacted with

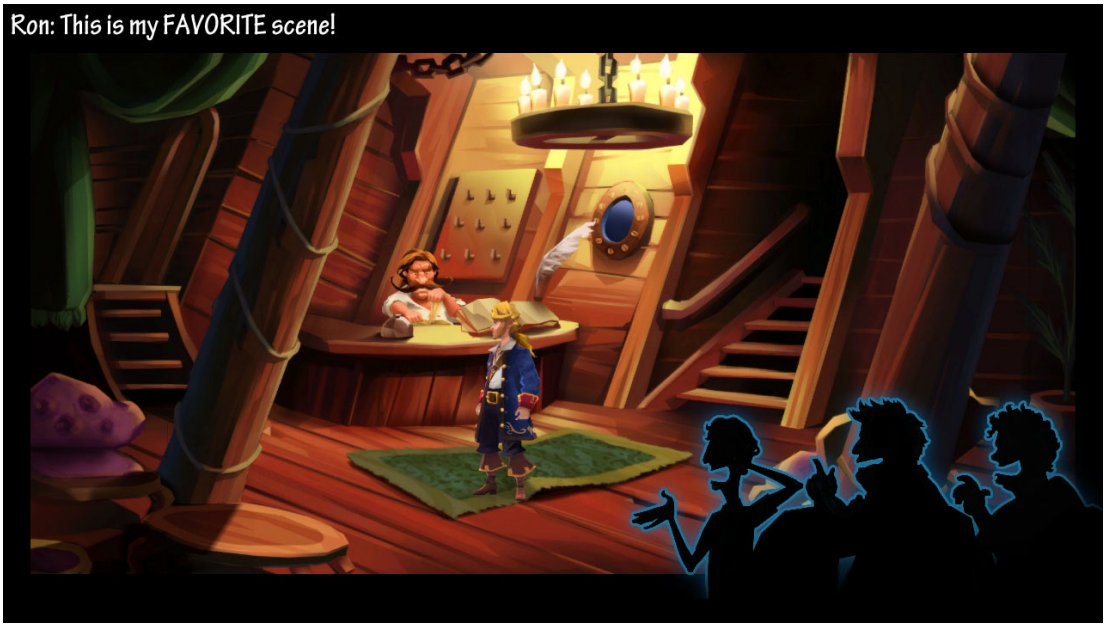


Figure 4 Monkey Island 2 gameplay screenshot

Figure 5 Toca Boca World gameplay screenshot

Concept summary

The game is 2D, character can move around according to mouse clicks through all theater locations to introduce player to locations they will visit later in real life. Some locations can be visited at any time, some of them only after completing specific tasks.

The goal is to complete all tasks from task list on the corner of the screen, meanwhile completing minigames which are based on gamified rules of the theater.

Player will have an Inventory, where will be stored items required for some tasks.

Researching Web Development Frameworks

To address these needs, I researched familiar to me website creation frameworks like Angular and React. While these frameworks offer robust features, I found that they can introduce unnecessary complexity. Additionally, implementing the game within these frameworks caused problems.

Considering these factors, I decided to proceed without using a complex framework. This approach allows for:

- **Improved Performance:** Faster loading times by reducing overhead.
- **Greater Control:** The ability to optimize code specifically for our game's needs.
- **Seamless Integration:** Easier alignment with the game's visual style without being constrained by framework-specific limitations.

Outcome of the Web Development Frameworks Research

Through this research, we established the foundation for creating a user-friendly, visually appealing, and engaging website tailored to children aged 5–10. This research also informed our decision to use Unity for game development, ensuring seamless integration with the website and addressing our client's requirements. The findings from the empathize phase are guiding our design and development process, ensuring the project stays aligned with the needs of our young audience and the client's goals.

Problem Statement

Based on the insights gathered, I refined my problem statement:

"How can I, as an engineer, develop a simple yet engaging website to host our 2D point-and-click educational game for children aged 5–10, ensuring that the digital experience is accessible, aligns seamlessly with the game's visual style, and maintains optimal performance across platforms without using complex frameworks like Angular or React?"

This problem statement is based on the research, and directly addresses the needs of our client. Our next task is to finish the design of the website and start implementing prototype into the actual website.

Conclusion

By adopting the Double Diamond model, we ensured a structured approach that allowed us to explore a wide range of possibilities before converging on the best solutions. Through the empathize phase and target audience research, we identified critical needs, such as vibrant visuals, simple navigation, and engaging interactive elements, which are essential for capturing and retaining the attention of our young audience.

The decision to dismiss complex frameworks like Angular or React in favor of a more tailored approach was driven by the need for improved performance and greater control. This choice has laid the foundation for a user-friendly and visually consistent digital platform that aligns with both the game's goals and the needs of its users.

As we progress to the implementation phase, the insights gathered throughout the research and development process will continue to guide our design and technical decisions. Our goal remains clear: to deliver an engaging, accessible, and high-performing platform that not only meets the expectations of our client but also provides an enjoyable learning experience for children. By staying focused on user needs and leveraging our research-backed approach, we are confident in the success of our project.

Sources:

- Wilmink Theater: https://www.wilminktheater.nl/
- Design solutions to problems with the Double Diamond process: https://www.bitesizelearning.co.uk/resources/double-diamond-design-process-explained
- Children's Emotional Associations with Colors by ResearchGate: https://www.researchgate.net/publication/15176872_Children's_Emotional_Associations_with_Colors
- Top 10 UI/UX Design Principles for Creating Child-Friendly Interfaces by aufaitUX: https://www.aufaitux.com/blog/ui-ux-designing-for-children/
- Designing website for kids by Canva: https://www.canva.com/learn/kids-websites/
- Pedagogy in Games: Exploring the Intersection of Learning Theories and Game Theories by Medium: https://nicoking.medium.com/pedagogy-in-games-exploring-the-intersection-of-learning-theories-and-game-theories-25c21dfb0bf5.
- The Benefits and Limitations of Software Development Frameworks by TechAffinity: https://techaffinity.com/blog/the-benefits-and-limitations-of-software-development-frameworks/
- What are the Pros and Cons of Using Web Development Frameworks? by Land Of Coder: https://blog.landofcoder.com/web-development-frameworks-pros-and-cons/
- The benefits and limitations of software development frameworks by TechAffinity: https://techaffinity.com/blog/the-benefits-and-limitations-of-software-development-frameworks/

Design phase

During the design phase, my primary focus as an engineer was to translate our initial concepts into functional features, including a rhythm-based minigame, a character customization system, and the initial website design. Guided by the Double Diamond model, I aimed to ensure these features aligned with the game's goals and provided an engaging experience for our target audience of children aged 5–10.

Minigame Development

Rhythm-based minigame

For the rhythm-based minigame, inspired by games like *Guitar Hero*, our goal was to create an interactive activity that was simple yet engaging for the target age group, with adjustable difficulty levels for broader accessibility. During the concept phase, we evaluated various mechanics and ultimately decided on a tap-to-hit format, synchronized with visual and sound cues to enhance immersion.

Designers developed a low-fidelity prototype to test core gameplay elements like timing, rhythm, and feedback. From there, I implemented coroutines in Unity to control timing and utilized Unity's timing functions to ensure smooth and responsive gameplay. These efforts established the foundation for a polished and age-appropriate minigame. (Figure 6)

To synchronize the falling circles with the song beats, I used the Serato DJ Pro tool, which allowed me to see the beats inside the song, then I gathered all the timings and put them inside the game, while adding a small delay which was equal to time between circle spawns and when it reaches the button on the bottom of the screen.

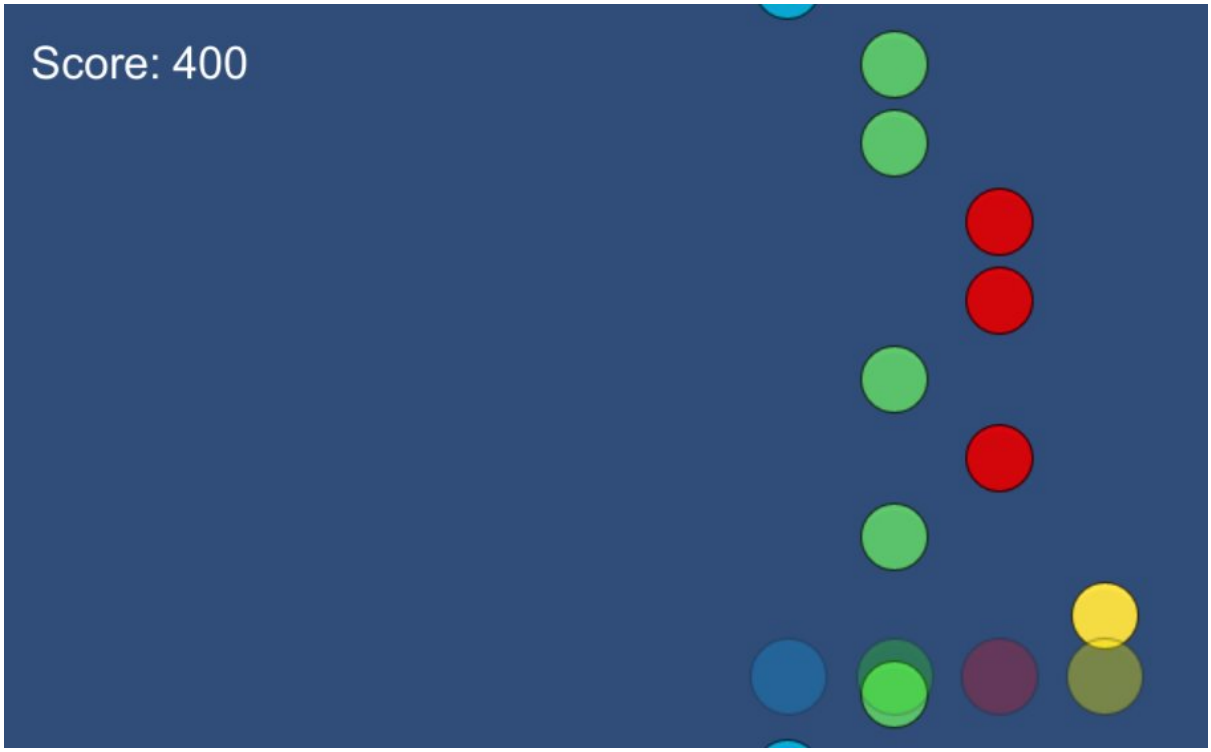


Figure 6 Rhytm based minigame Unity prototype

Puzzle Minigame

One of the minigame prototypes developed during the project was the “Puzzle Minigame”. The main concept behind this game was for players to fit all the figures of different shapes into a grid. According to our idea, the movable shapes symbolized musical instruments, aligning with the theater theme. The goal was to combine a fun, engaging activity with subtle educational value, such as problem-solving and spatial awareness. (Figure 7)

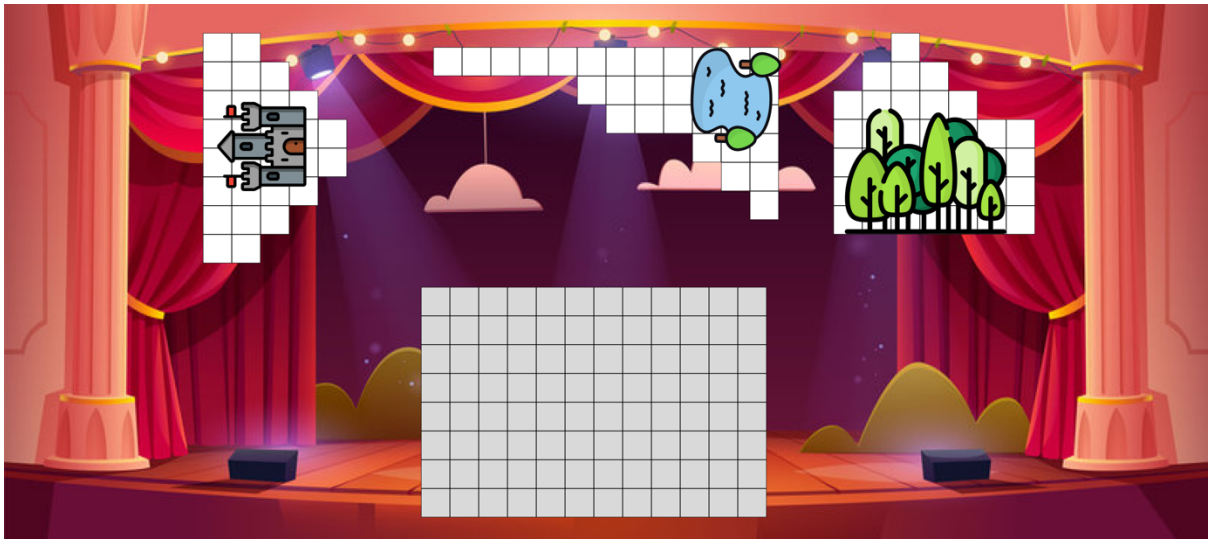


Figure 7 Puzzle Minigame Figma prototype

To design this minigame, our team researched similar puzzle mechanics used in educational games like *Tetris* and *Tangram-style puzzles*. These games demonstrated how simple mechanics combined with colorful visuals can engage younger audiences while developing cognitive skills like spatial reasoning and logical thinking.

Based on this research, I began developing the core mechanics in Unity (Figure 8). The process involved:

- **Grid System**: I created a grid-based system where pieces could snap into place when correctly aligned. This was achieved using Unity's RectTransform for precise placement and collision detection to determine if pieces fit correctly.
- **Piece Movement**: The draggable shapes were implemented using Unity's OnMouseDrag function, allowing players to pick up and move pieces with ease.

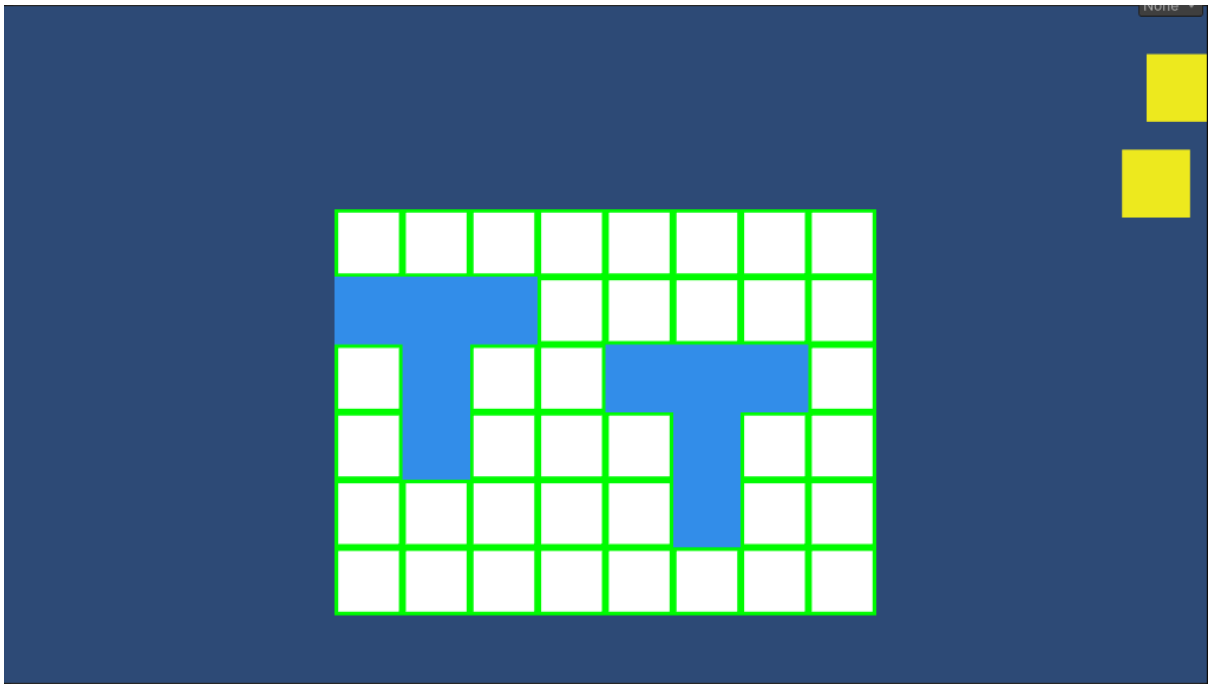


Figure 8 Puzzle Minigame Unity prototype

While the codebase for the minigame was completed and functional, the lack of time meant we could not finalize the minigame with the necessary art assets and designer input. Which means this game will not be finished.

Although the Puzzle Minigame was not fully realized, the development process provided several valuable outcomes:

- **Feasibility**: The core mechanics worked smoothly, proving that the concept was technically finished and engaging for the target audience.
- **Learning Opportunities**: Through this minigame, I gained hands-on experience with Unity's grid and collision systems, as well as implementing feedback mechanisms tailored to younger players.

- **Prototype Potential**: The groundwork laid during this project could be expanded in the future, especially with additional time and artistic contributions. The concept of combining puzzles with thematic elements like musical instruments remains a strong idea for enhancing the educational value of the game.

Even though the minigame was eventually shelved, I managed to work on my learning goals and create a code system that allowed other team members to work on the grid mechanics without needing to modify the code themselves. This system was designed to make the grid fully configurable via Unity's Inspector. The screenshot below shows the Grid Manager script, where parameters such as grid size, cell size, offsets, and cell visualization colors can be adjusted directly. This approach ensured flexibility and accessibility, allowing the team to tweak the grid easily without relying on me for further adjustments. (Figure 9)

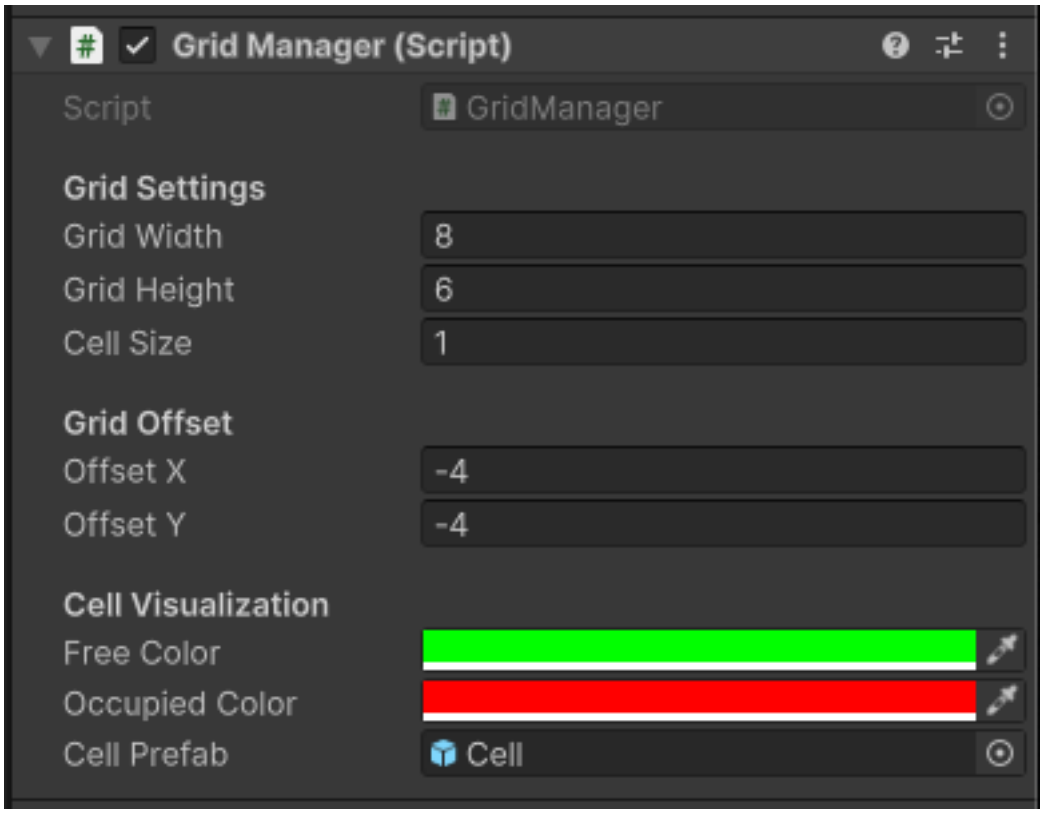


Figure 9 Grid Manager Code Inspector

Voice Pitch Minigame

Alexis (designer) came up with idea to create a minigame, where player would have to follow the line using the mouse - "Voice Pitch Minigame". The first concept of it was developed by Alexis (designer) in Unity, later on, I improved it by adding difficulty levels according to players age, so if player younger, the line would get more straight, and added the bonus points counter, if the time between entering the line and exiting is more that 2 seconds, the overall score is being multiplied.

Figure 10 Voice Pitch Minigame Figma concept

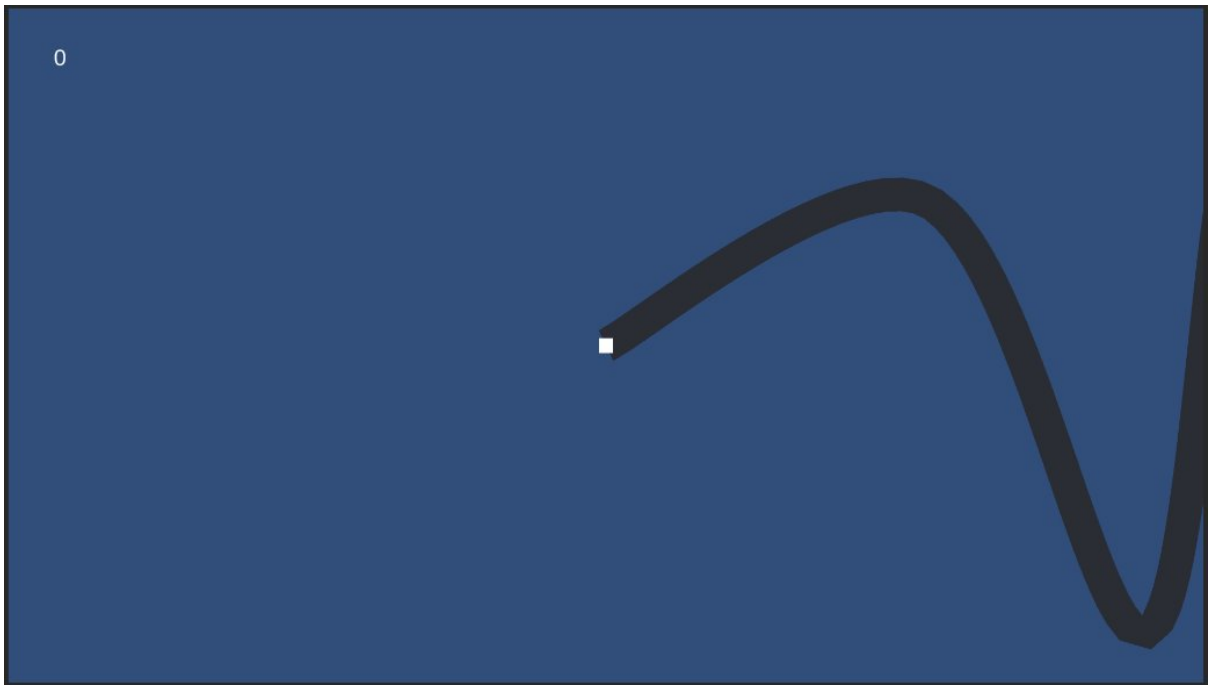


Figure 11 Voice Pitch Minigame Unity prototype

Unfortunately, this minigame was not included in the final project. However, working on it provided a valuable learning experience. While the initial groundwork for the minigame was created by Alexis (designer), I expanded on it and gained a deeper understanding of Unity's Line Renderer, a tool I had not used extensively before. This process allowed me to enhance my technical skills and contribute meaningful improvements, such as difficulty levels and bonus scoring mechanics, even though the minigame remained a prototype.

Character

According to our planned game concept (2.5D game perspective) we needed a 2D character which will be made by Kit (artist) using Spine2D tool.

The character customization feature required building a customization system that allowed players to select outfits and later a skin color, with their choices persisting across different game scenes. Since the character assets were not yet finalized, I worked with placeholder templates while collaborating closely with the artists and designers. Together, we defined technical requirements such as bone hierarchy, pivot placements, and template layouts to ensure seamless integration once the final assets became available.

Using PlayerPrefs in Unity was the best solution for saving character customization because it is simple, efficient, and built for small data storage. It allows seamless data persistence between scenes with minimal setup, making it ideal for saving sprite and body part preferences. As highlighted in the Unity tutorial, it's lightweight, reliable, and perfect for projects like yours where quick implementation and cross-scene data consistency are key.

Character customization (Wardrobe Scene)

To support customization, I developed a user interface (UI) where players could browse outfit options and preview their selections (Figure 12). I used Unity's PlayerPrefs, allowing outfit selections to carry over into other game scenes. This feature enhances the sense of continuity and personalization for players.

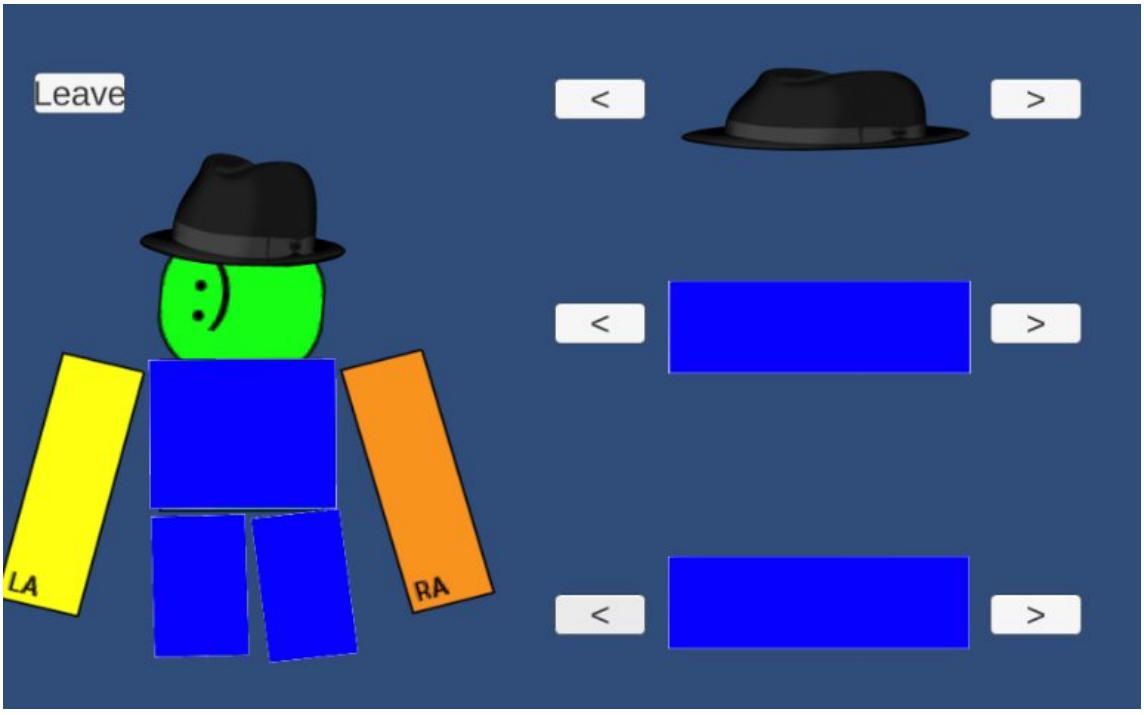


Figure 12 Character customization scene prototype

Collaboration with Artists and Designers

Although the character assets were still in development, I worked closely with artists and designers to define the functionality and technical requirements for the customization feature. We discussed details like bone hierarchy, pivot placements, and template layouts to make sure that the character's outfits could be easily adapted to the final character model. This collaboration was essential to prepare a system that would be compatible with the finished art assets, minimizing the need for adjustments later.

I also focused on achieving one of my learning goals: making my code more user-friendly for other team members. To do this, I improved the Unity Inspector for greater accessibility. Every scene I created included a GameManager GameObject in the hierarchy, which allowed other team members to adjust key settings and components of the game without needing to modify the underlying code. This included features like character customization, animation controls, and other configurable elements. The screenshots below showcase how the Character Customization and Player Animation Controller scripts were designed to be intuitive and easily adjustable, ensuring a smoother workflow for the entire team. (Figure 13 & 14)

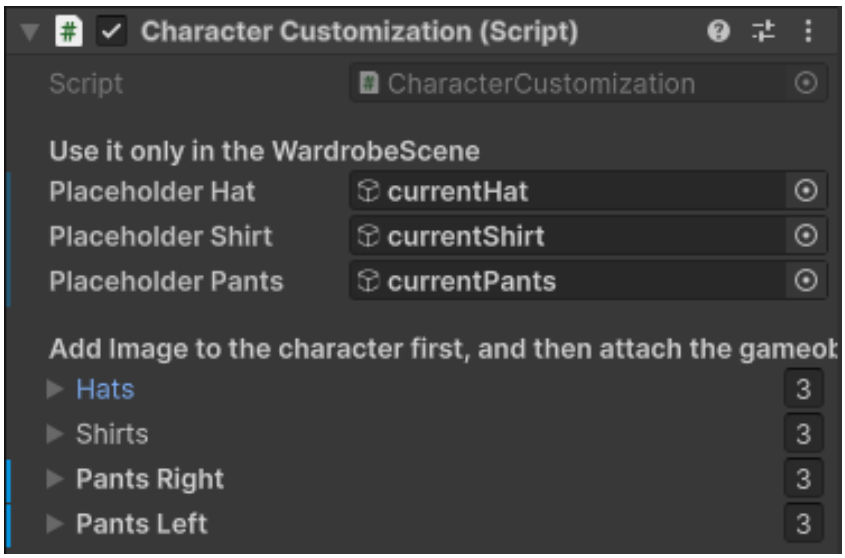


Figure 13 Character customization script inspector

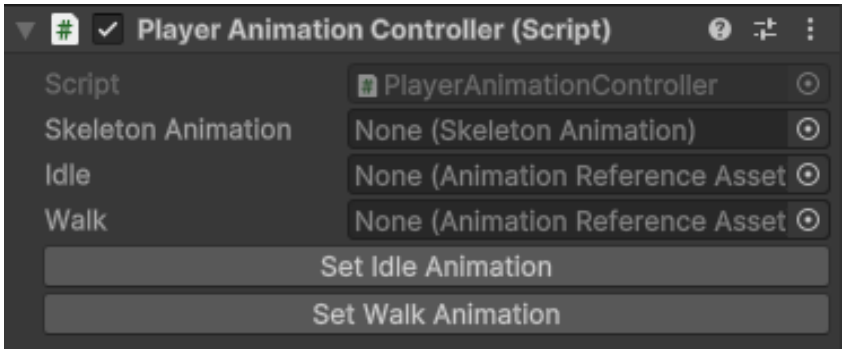


Figure 14 Player Animation Controller script inspector

Challenges

The biggest challenge I faced in this phase was achieving one my learning goals: designing a customizable and easily modifiable code structure that would be accessible for all team members through Unity Inspector. My goal was not only to build a flexible system but also to document it thoroughly, so that any team member—whether a developer, artist, or designer—could understand and modify it without extensive technical guidance.

I focused on building a modular system that could handle different character assets and configurations. Working with templates and placeholders, I established a solid foundation, ensuring that the system would seamlessly integrate with final character assets once they were ready. This approach allowed us to proceed with development without waiting for completed assets, providing flexibility in the project timeline.

Website Design

As part of the design phase, I also took responsibility for creating the initial website design to host the game. Leveraging my prior experience in web design (Figure 15), I used Figma to draft a rough concept based on the research conducted during the empathize phase. The design was colorful, playful, and child-friendly, with custom background art and thematic icons symbolizing elements of the Wilmink Theater, such as masks and emotions.

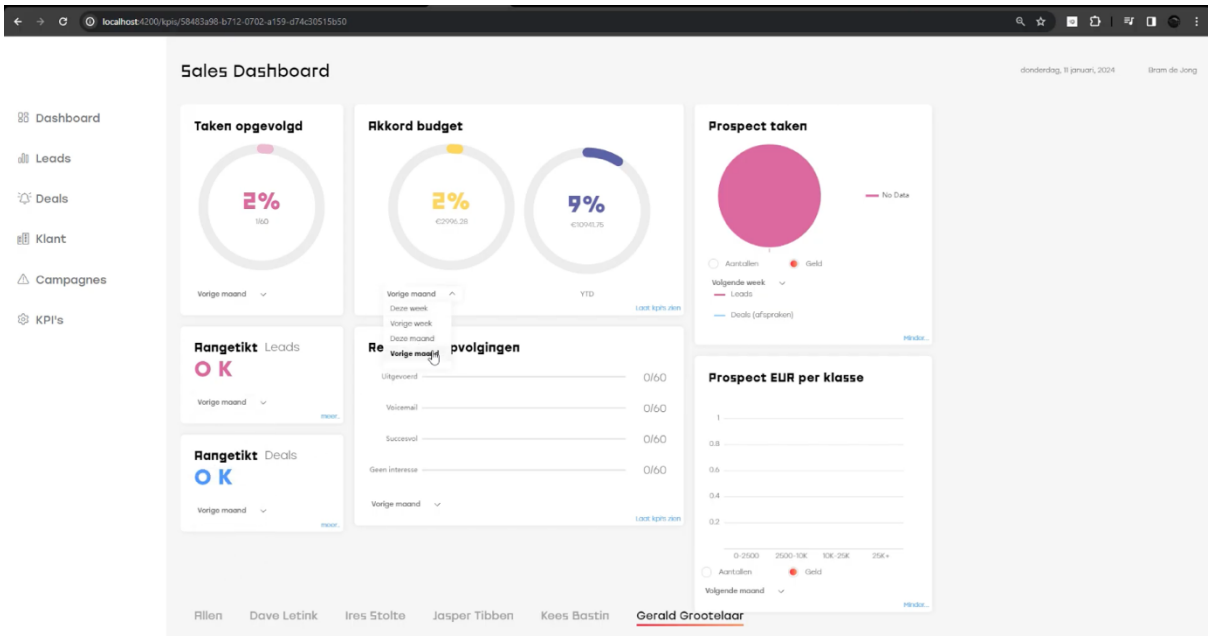


Figure 15 Website from my portfolio

While initially considering using the existing design style of the Wilmink Theater, I opted for a more playful and vibrant look to align with the target audience of children. The goal was to make the website modern, engaging, and consistent with the game's visual style.

According to our research I've made a rough design of the website in Figma using integrated tools (Figure 16). It was simple, colorful and informative. It looked modern, the background was made by me too. There are some Icons which symbolised aspects of the theater like (masks, emotions, etc). Firstly i was considering using a default Wilmink Theater design style, but since we are making a game for kids a decided to more with playful and childish design, according to my research.

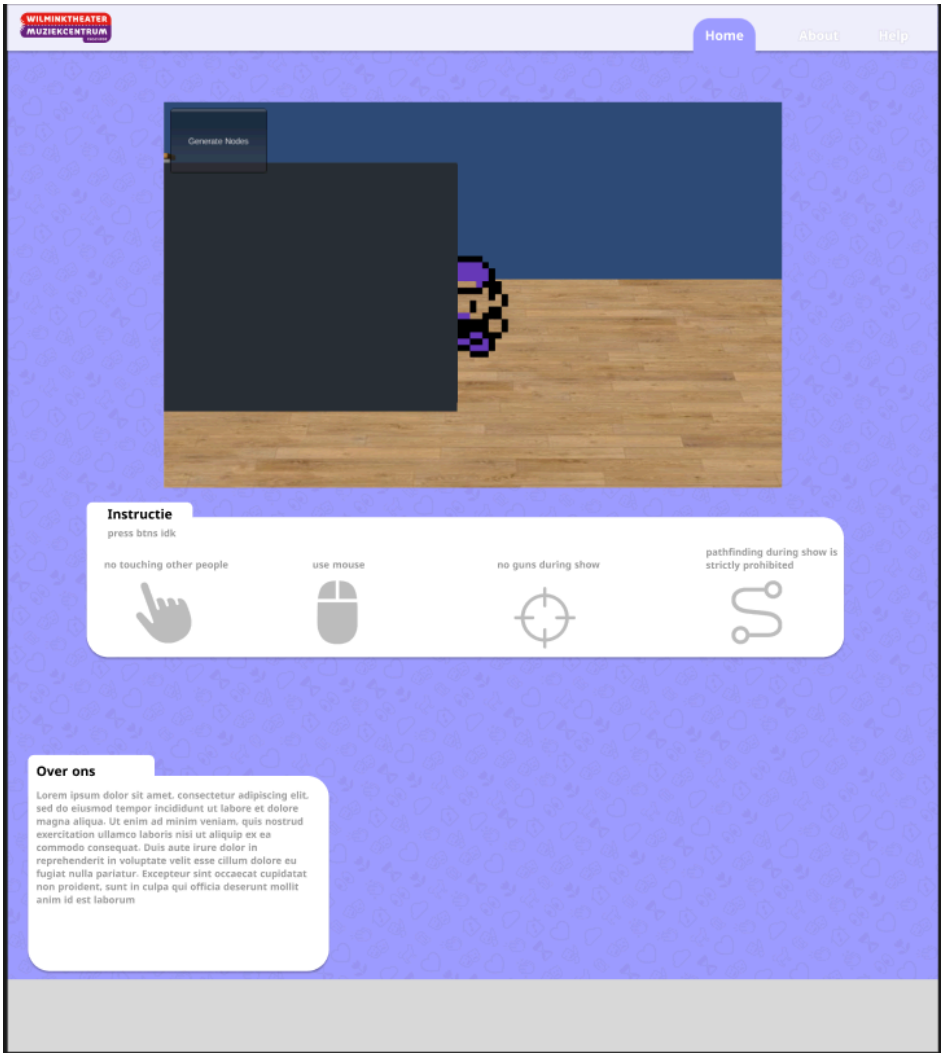


Figure 16 Website prototype in Figma

Documentation

To achieve my learning goal - learn how to make Unity project more tweakable for the designers and artists in our team, I created the code and features documentation using Jira Atlassin platform. I wrote clear, step-by-step instructions detailing the structure of

the customization system and troubleshooting tips. The goal was to make sure that anyone on the team could make adjustments independently. (Figures 17 & 18)

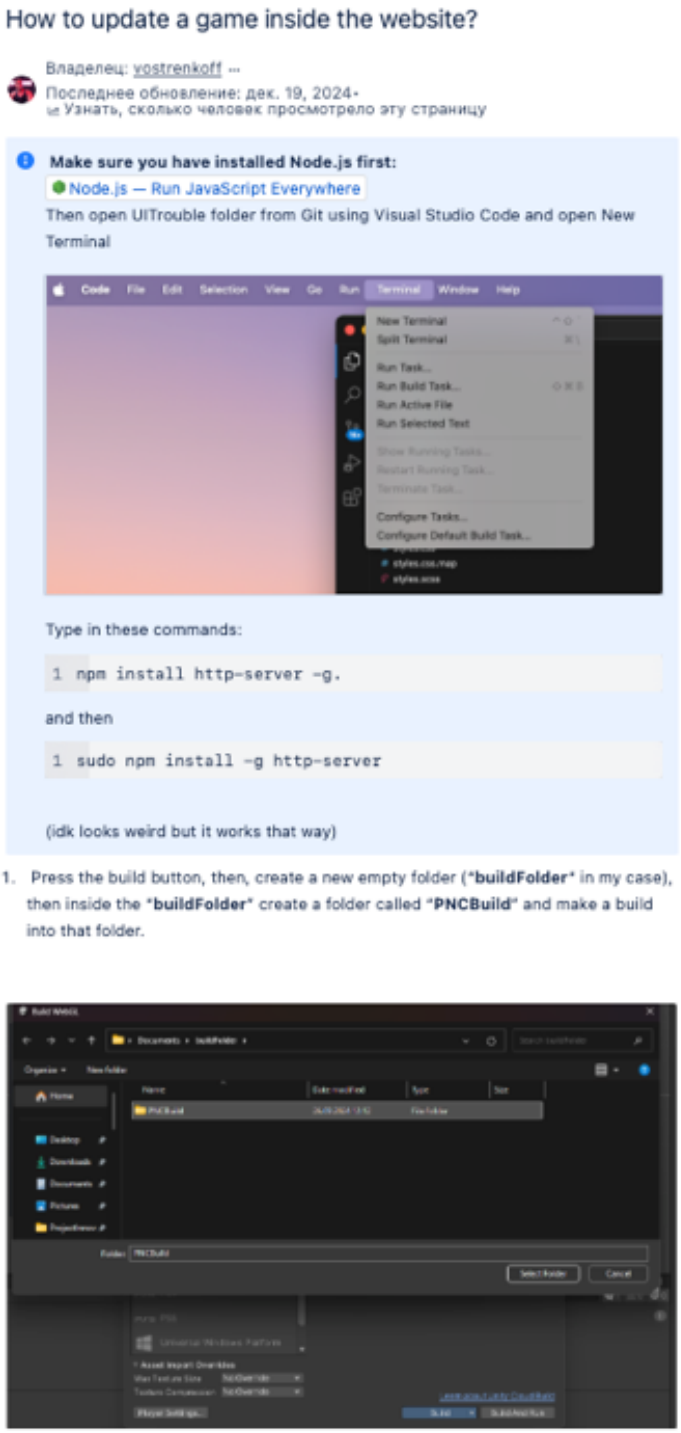


Figure 17 How to update a game inside the website? Documentation

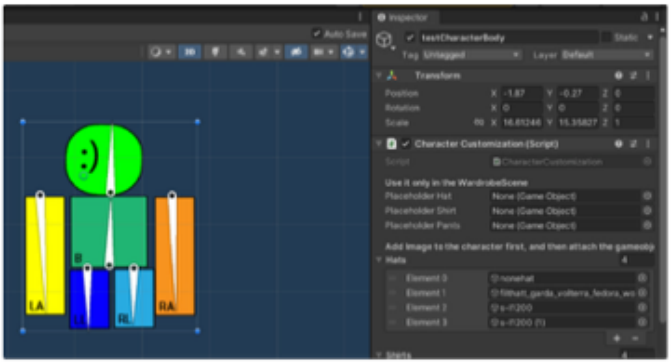
How to change customization elements inside the character?

Владелец: vostrenkoff ...
Окт. 24, 2024 · Узнать, сколько человек просмотрело эту страницу

Version 1.0 (in this version it is possible to only change Head, Torso, Left Leg, Right Leg)

Step 1:

Open Character Prefab. Currently its in the .../Assets folder.



Step 2:

For example we want to add a new hat (fedora), to do that we need to:

1. Drag a hat Image to the project assets
2. Drag the Image from the project assets to the Head gameobject, so when the head moves, our hat will move as well
3. Make sure that new hat has the Sprite Renderer component.

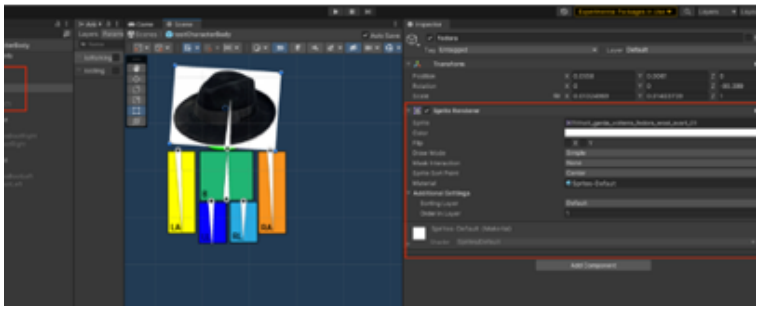


Figure 18 How to change customization elements inside the character? Documentation

Conclusion

The design phase resulted in a functional rhythm-based minigame and a flexible wardrobe customization system, both ready for further development as final assets become available. Through close collaboration with the art and design teams, I was able to establish a solid foundation for character customization, ensuring that future assets can be integrated smoothly. This phase allowed me to bridge initial concepts with technical implementation, setting up these features to support an engaging and personalized gameplay experience for players.

Sources:

- Tangram Puzzles: https://www.mathplayground.com/tangram_puzzles.html
- Line Renderer Unity Docs: https://docs.unity3d.com/Manual/class-LineRenderer.html
- Implement data persistence between scenes by Unity Learn: https://learn.unity.com/tutorial/implement-data-persistence-between-scenes

Production & Testing

Website Development

Website development process was postponed to the end of the project since our team decided that we should create a game first, and then think about appropriate design of the website, to make it fit into the game design code.

According to one of my learning goals - create a website, host it on my PC online, and integrate our project game into the website, the plan was clear. However, I encountered two major difficulties. Initially, I chose the Angular framework by Google for website creation because it offers a well-defined structure for building applications, with features like modules, components, services, and dependency injection. Its consistent architecture makes code easier to scale and maintain.

Unfortunately, the Unity game we developed could not integrate properly with the Angular framework. After conducting research, I was unable to find a viable solution to resolve this issue but identified the primary reasons for the incompatibility:

- Asset Loading Issues
- Frameworks may re-render or modify elements interfering with the Unity
- Different script execution timing and mechanisms

After many hours of problem solving, the decision was made – create a website without any framework. Using only HTML and SCSS I created the website based on Jose's (designer) made in Figma (Figure 19).

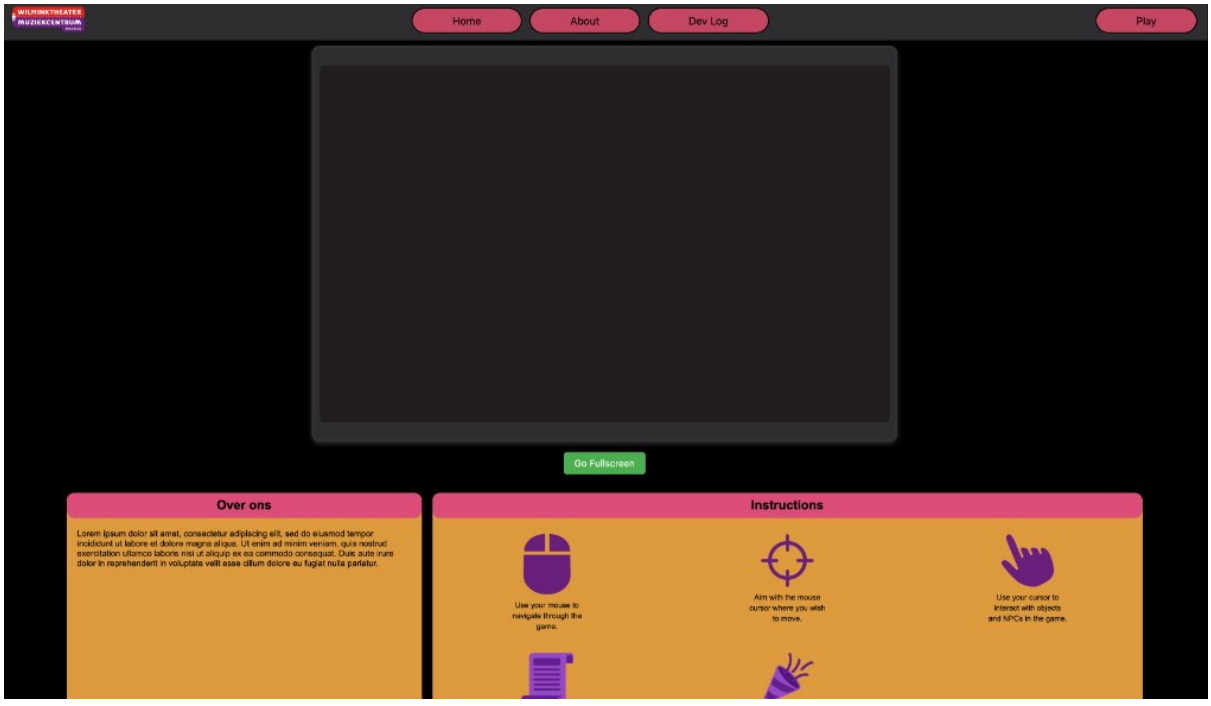


Figure 19 Website made according designers design from Figma

Game worked smoothly, but after gathering feedback, we agreed that we don't like the overall look and design of the website.

Feedback received:

- Color palette is inappropriate
- Text can't be readed comfortably
- Game should have a possibility to run fullscreen

Due to the limited time left in our project, we were unable to fully redesign and prototype the website's layout and design. As a result, I made direct improvements to enhance its appearance and usability. I adjusted the color scheme, increased font size, added bold styling for readability, and improved contrast between text and background. Additionally, I implemented a white border highlight for selected buttons to improve interactivity. While the team agreed that these changes made the design look better, we acknowledged that the final result was not as polished as we had hoped. Unfortunately, some planning mistakes earlier in the project prevented us from achieving a design we could be fully satisfied with. (Figure 20)

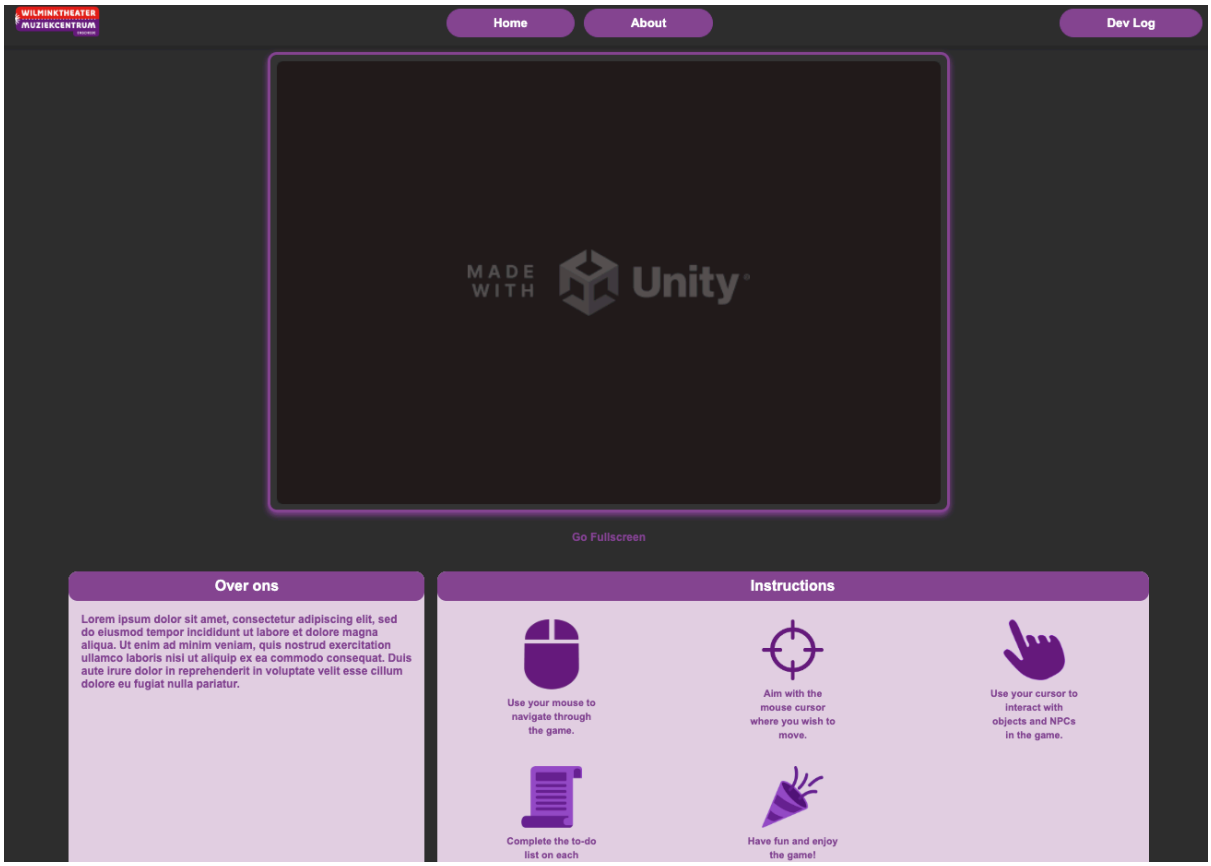


Figure 20 Final website version

Website hosting

According my learning goal, I was planning to host a website on my own, using my PC at home. Envisioning it as a personal project to explore web hosting and server management.

To set up the necessary software for hosting the website, I began by installing a local web server. I chose Apache HTTP Server as my server software because of its reliability and previous experience with it during my internship in the past. I downloaded and configured the server to run on my PC, ensuring it was properly installed and operational. After setting up the server, I hosted our website with HTML and SCSS files, placing them in the designated root directory of the server to ensure they would be accessible when hosted. I also tested the site locally by accessing http://localhost in my web browser, confirming that the server could serve the files as intended on my PC.

Challenges

However, I ran into an issue because my internet connection is provided through eduroam, the university's network. Eduroam is designed primarily for secure access to the internet, not for hosting services. It uses network policies and firewalls that block incoming connections to prevent unauthorized access and maintain security across the network. Despite trying various troubleshooting steps, including configuring port forwarding and testing different ports, the eduroam restricted any attempts to make my PC accessible from the outside. This made it impossible to serve the website publicly, as other users could not connect to my server through the eduroam network. Although it was a great experience, and a good lesson for the future.

The best way around was to use GitHub server hosting. It’s free, but only if your website’s source code is set to public in GitHub repository. And after adjusting some settings, our website is now publicly visible to anyone on the internet.

Character customization and movement

During Production and Testing phase most of the time i worked on implementing character into the game, however it appeared to be challenging enough. The character that was made in Spine2D, required me to install the Unity Package which would allow to import the character and control its animations, however after importing the character into the game, we realised that Spine2D doesn’t allow us to modify anything related to the character, so we couldn’t just replace the sprites in Wardrobe Scene.

Figure 21 Character structure made by Kit

Although by getting advices from teachers, we were considering a way around by applying the shaders with different colors on different sprite parts, by making initial character color fully gray. I didn't worked, since we wanted player to have abiility to change hair type, and after digging into some code, i found out that it is possible to export character from Spine2D with inactive sprites, and using code, since the sprites were attached to bones directly, we could replace the sprites. Later this task was handed to Amber (engineer) and together with Jose (designer) they implemented the UI and finished the Wardrobe scene (Figure 22).

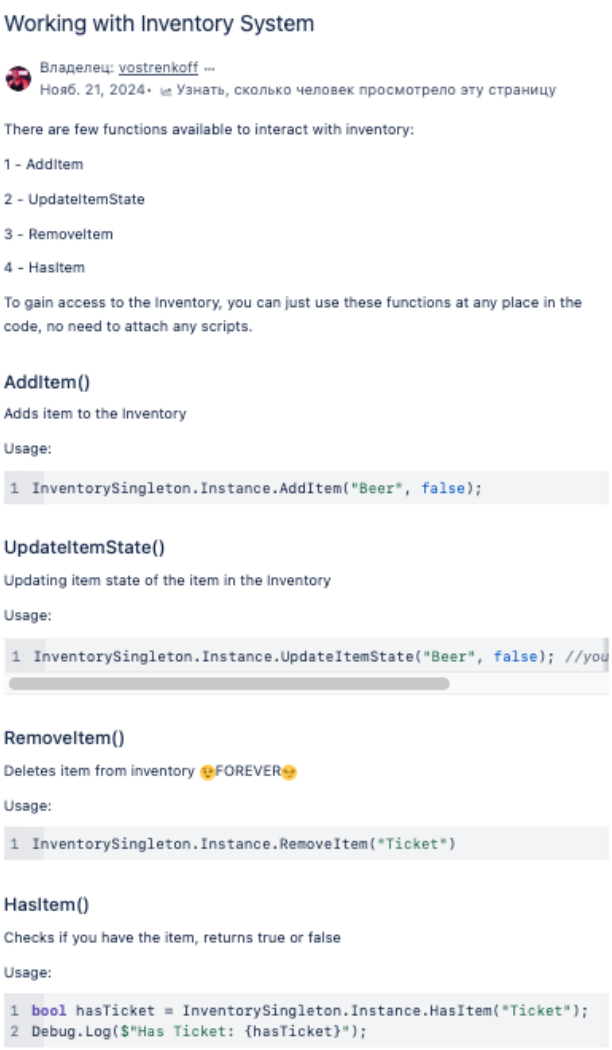
Figure 22 Final version of Character Customization scene

Inventory system

As planned, I was responsible for creating the inventory system. To implement it, I evaluated several approaches, including Array-Based, List-Based, Dictionary-Based, and ScriptableObject-Based systems. Ultimately, I chose a Dictionary-Based system because it was efficient, met all our project requirements, and was easy for other team members to use in the future.

The system was designed with simplicity in mind, allowing team members to interact with it through straightforward commands such as AddItem(), RemoveItem(), UpdateItemState(), and HasItem() to check if a specific item exists. This approach ensured the inventory system was both functional and user-friendly.

To support my learning goal of improving team collaboration, I also created comprehensive documentation for the system, which is accessible to the team via Atlassian. This documentation provides clear instructions on how to use and expand the inventory system, ensuring a smoother workflow for everyone involved. (Figure 23)



Working with Inventory System

Владелец: vostrenkoff ···
Нояб. 21, 2024 · Узнать, сколько человек просмотрело эту страницу

There are few functions available to interact with inventory:

1 - AddItem

2 - UpdateItemState

3 - RemoveItem

4 - HasItem

To gain access to the Inventory, you can just use these functions at any place in the code, no need to attach any scripts.

AddItem()

Adds item to the Inventory

Usage:

```
1 InventorySingleton.Instance.AddItem("Beer", false);
```

UpdateItemState()

Updating item state of the item in the Inventory

Usage:

```
1 InventorySingleton.Instance.UpdateItemState("Beer", false); //you
```

RemoveItem()

Deletes item from inventory FOREVER

Usage:

```
1 InventorySingleton.Instance.RemoveItem("Ticket")
```

HasItem()

Checks if you have the item, returns true or false

Usage:

```
1 bool hasTicket = InventorySingleton.Instance.HasItem("Ticket");
2 Debug.Log($"Has Ticket: {hasTicket}");
```

Figure 23 Part of "Working with Inventory Systems" documentation

Falling lights minigame polishing

During the Prototype phase, it was time to complete the Falling Lights Minigame. Designers created a prototype in Figma, where I took all the assets and integrated them into the Unity scene. Building upon the initial concept, I added several enhancements to improve the gameplay experience, such as particle effects for the falling circles and a score-scaling animation that visually emphasizes points being earned.

To achieve my learning goal of creating team-friendly, customizable code, I made the minigame fully adjustable via the GameManager in Unity. The GameManager now includes a variety of configurable settings, such as game over text, score thresholds for star achievements, progress bar activation, and timer controls. These options allow other team members to modify the minigame's behavior and appearance without altering the underlying code. (Figure 24)

This approach not only streamlined the development process but also ensured flexibility, making the system adaptable for future adjustments or expansions.

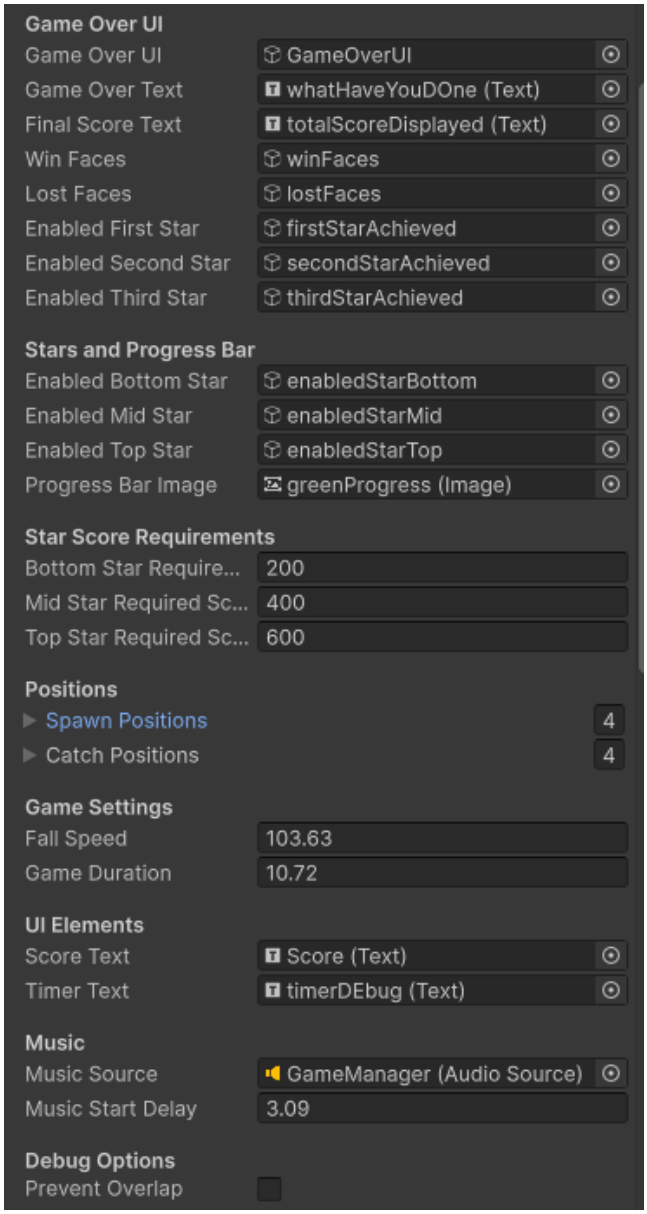


Figure 24 Part of GameManager code inspector in Unity

Also, internal testing was made and we received some feedback.

Feedback:

- he spotlights should flash when clicking at the right time, not when a circle appears
- the buttons look a bit out of place and need to reflect the perfect position better
- the colours of the circles are too saturated
- Needs more visual feedback based on how well the player clicked
- VFX of the circles is too much and out of place

- The spacing between the beats can be a bit too challenging and out of place (the challenge shouldn't come from having good snappy aim)

After analyzing the feedback, UI related issues were solved by Alexis (designer), and code related issues were solved by me.

Figure 25 Final version of the Falling Lights minigame

Publishing

During the publishing phase, our primary focus was on finalizing the game for submission. This involved fixing minor issues, optimizing gameplay elements, and seamlessly integrating the game into the website. These steps ensured the project was polished, functional, and ready for presentation.