Tecnológico de Monterrey

Break Into Valhalla

Software Requirements

Specification Document

Version 1.0

May 14th, 2023

The Einherjar Team

Joaquín Badillo, Pablo Bolio, Shaul Zayat

Table of contents

User Stories and Derived Software Requirements

Organizational User Stories

User Story	Requirements Functional Non Functional	Related Requirements	Priority (highest/medium/ lowest)
1.1. As a client I want to see a dynamic presentation where the project is pitched.	Create final presentation slides respecting our aesthetic.		Medium
1.2. As a client I want formality.	Practice the presentation Find a set of beta testers and test the game ourselves to detect any unexpected behaviors		Highest
1.3. As a client I would like for the game to be hosted locally and a scalable game.	Separate game behaviors with classes		Lowest
1.4. As a client I want to be informed on the development efforts (programming time) needed to complete the project.	Create a trello board to track the development efforts of each sprint. Create the software requirements specification document.		Highest

Table 1. Organizational Requirements

Database User Stories

User Story	Requirements Functional Non Functional	Related Requirements	Priority
2.1.As a user I want my database to capture my play data.	Create an ER model of the data. Detect what are the different relationships between entities. Add test data that help test CRUD operations for future API implementation.	2.4. - Implement a schema in MySQL following the third normal form. - Create all the possible restrictions for the tables like PK, FK, Not Null, Index, Unique, Check. - Implement the possible relationships among tables, following the one to one, one to many and many to many cardinalities.	Highest
2.2. As a user I want to have an account to store my game information.	Users can create accounts to store their information. Implement database views to create a security barrier.	2.1. - Create an ER model of the data. 2.3. - Implement CRUD operations over the database. 2.4. - Implement a schema in MySQL following the third normal form. - Create all the possible restrictions for the tables like PK, FK, Not Null, Index, Unique, Check.	Medium
2.3. As a user I want to have control over the elements stored in the database,	Implement CRUD operations over the database.		Highest
2.4. As a client I want a relational	Implement a schema in MySQL following		Highest

database in SQL.	the third normal form. Create all the possible restrictions for the tables like PK, FK, Not Null, Index, Unique, Check. Implement the possible relationships among tables, following the one to one, one to many and many to many cardinalities.	

Table 2. Database Requirements

Web Dev User Stories

User Story	Requirements Functional Non Functional	Related Requirements	Priority
3.1. As a user, I want to have a website that is inspired by the style of my video game.	Create the website's homepage Create an about section Create the info page Create the game page Create a statistics view (with at least 3 visualizations)	3.4 - Embed the video game in the website.	Medium
3.2. As a user I want my website to have a section for the video game manual.	Create a manual section on the website.		
3.3. As a user, I want to have a section on my website that	Read/consult database stored data through the API	3.1 - Create a statistics view (with at least 3	

displays player statistics.	Design and implement plots on the website.	visualizations)
3.4. As a user I want to access the game through a webpage.	Embed the video game in the website. Deploy the website using a hosting service (Vercel).	
3.6 As an administrator I would like to have a website view (administrator page) to manage CRUD operations	Create an API endpoint for administrators. Create the webpage for the administrators to manage database operations.	

Table 3. Web Dev Requirements

Game Dev User Stories

User Story	Requirements Functional Non Functional	Related Requirements	Priority
4.1. As a user I want a 2D RPG game.	Create the game design document Design and implement the dungeon layout as a graph. Implement the procedural generation algorithm that uses a graph to create the dungeon with static rooms. Create at least 3 variations of each type of room		

	(treasure room, battle room, key room…) Add trigger events that load rooms when the player advances (similar to doors). Create at least one cutscene Design and implement the UI elements for the title screen Create a pause menu that allows the player to exit the game. Create the title screen Add a loading screen Design and implement the pause menu Create the ending screen and show credits	
4.2. As a user I want to see an attractive UI that displays relevant information for the gameplay	Create a health bar display Add a cooldown display Create an ammo display Create a glow particle effect for cooldown Create a fire particle effect for upgraded weapons (main focus on arrows)	

4.3. As a user I want the game to contain some sort of tutorial.	Create a game tutorial. Create a small dungeon to serve as the tutorial map. Create panels to show the resultant behaviors from keyboard inputs.	
4.4. As a user I want enemies to do more than just approach players.	Implement A* algorithm for enemy movement Implement functionalities for enemies. Create attack animations and scripts Create ranged characters	Lowest
4.5. As a user I want to know the risks going into the game.	Create a starting cutscene to inform the player about the roguelite genre without it being boring.	
4.6. As a user I want the game to be fun and something I would play again.	Create unique experiences through procedural dungeon generation.	
4.7. As a user I want an exhilarating boss fight.	Create an enemy with increased stats and difficulty (boss) Create starting boss stage (Normal attacking phase) Create second boss	

	stage and transition: attacking and summoning phase	
4.8. As a user I want to know when my character interacts with any game object, enemy, etc.	Add audio resources related to interaction: hurt sound, chest opening sound, etc.	
4.9. As a user I want the gameplay to be unique and change based on each decision you make.	Create mechanics for the player to be able to upgrade their weapon stats. Create both melee and ranged combat mechanics.	4.6
4.10. I want the game to be balanced but also with different types of gameplay.	Pending, waiting for Gil's approval	
4.11. As a user I want to create an account in the game and save my progress.	Create the login screen Create an API endpoint to write to the database. Connect Unity with the API to authenticate users in plain text.	
4.12. As a user I want the game to inform me about the story and role of my character.	Create a cutscene explaining the story and specific details of games and characters	
4.13. As a user I want the game to feel as though it has a beginning,	Incorporate all scenes in unity with a consistent timeline.	

ending or to be continued.	Players must be able to finish the game in “equal conditions”, fair enemy stats compared to player’s

Table 4. Game Dev Requirements

Restrictions

Use of the UML modeling language.
The project development should be completed by June 2023.
The website must be developed with HTML5 CSS3 and JavaScript technologies.
The video game must be developed with Unity technologies.
The project will be managed using the SCRUM agile methodology.
The video game must be embedded in a website.
The database must be developed in MySQL.
Complete use of github workflow.

Table 5. Software Development Restrictions

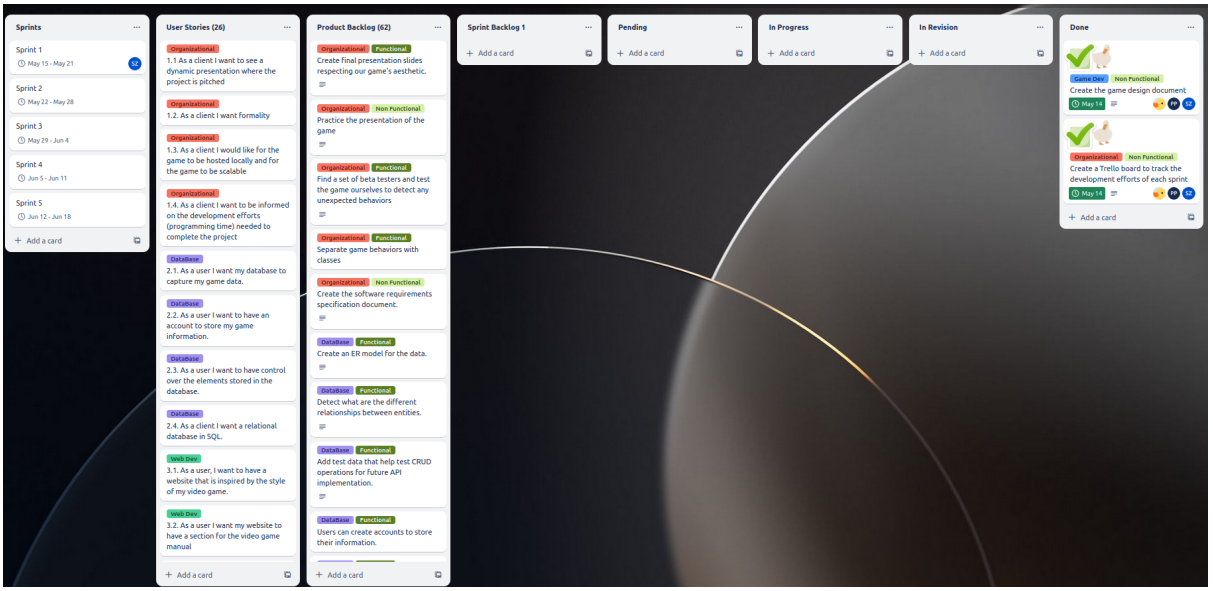


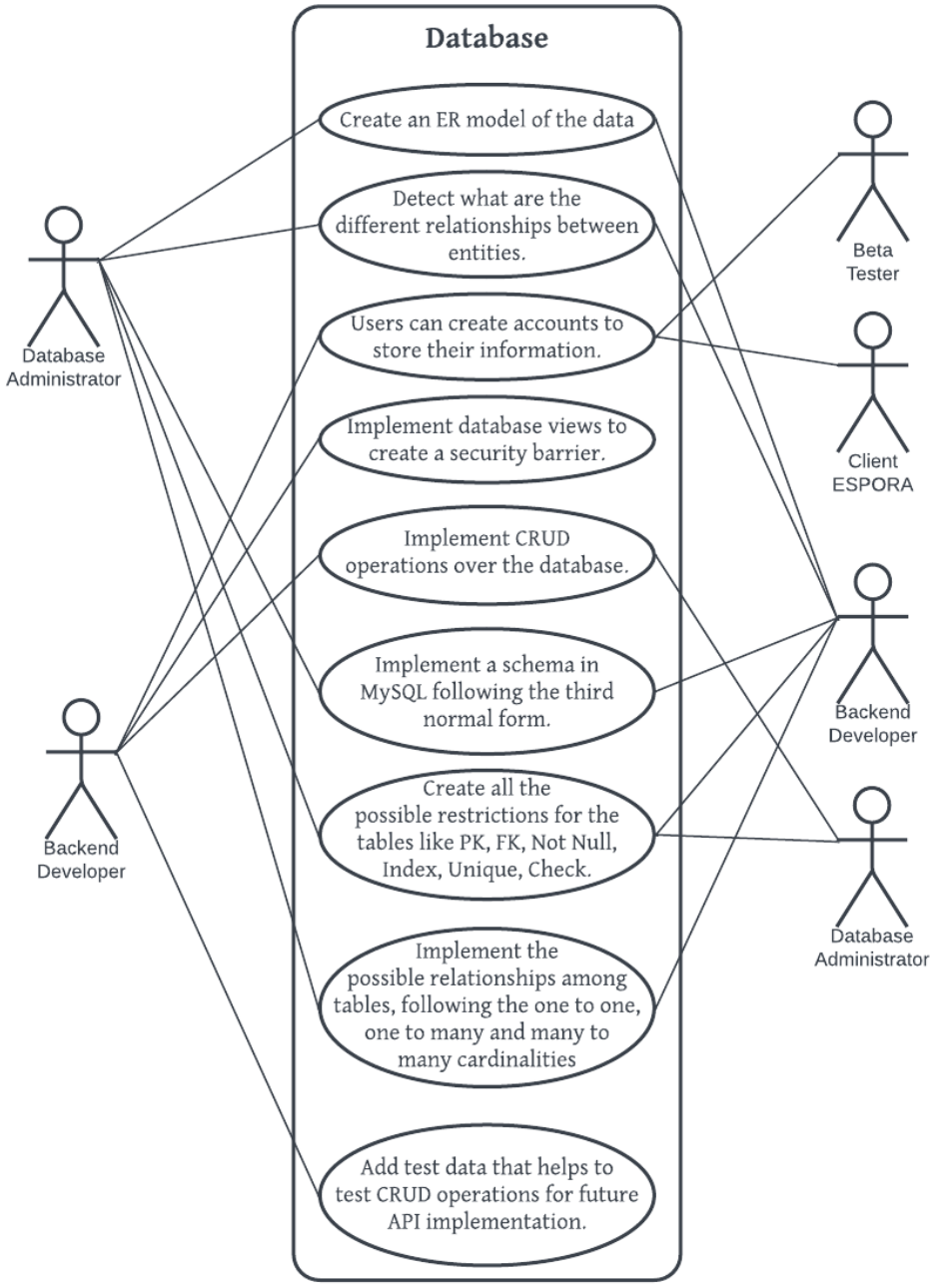
Figure 1. Trello Board

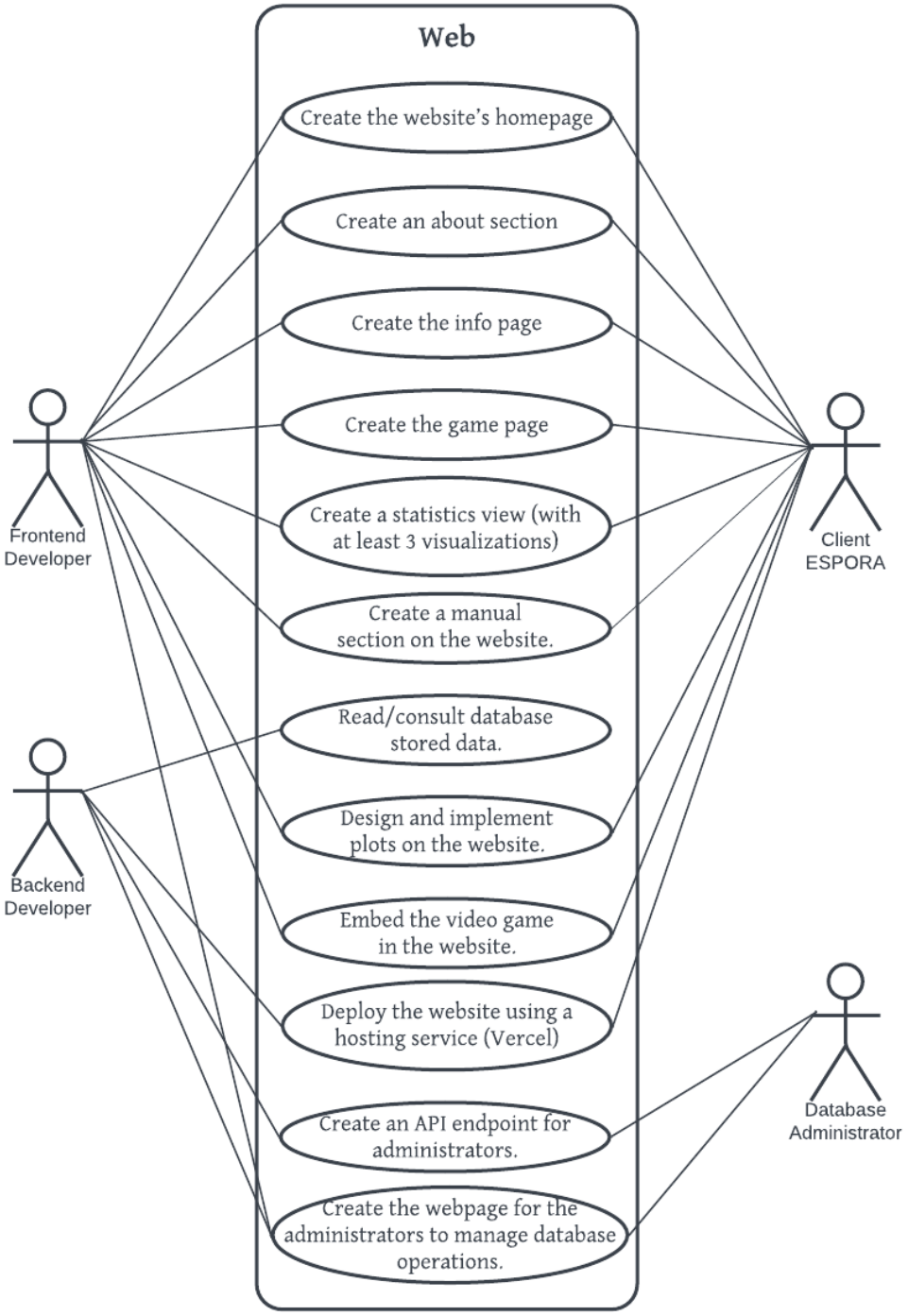
List of Requirements

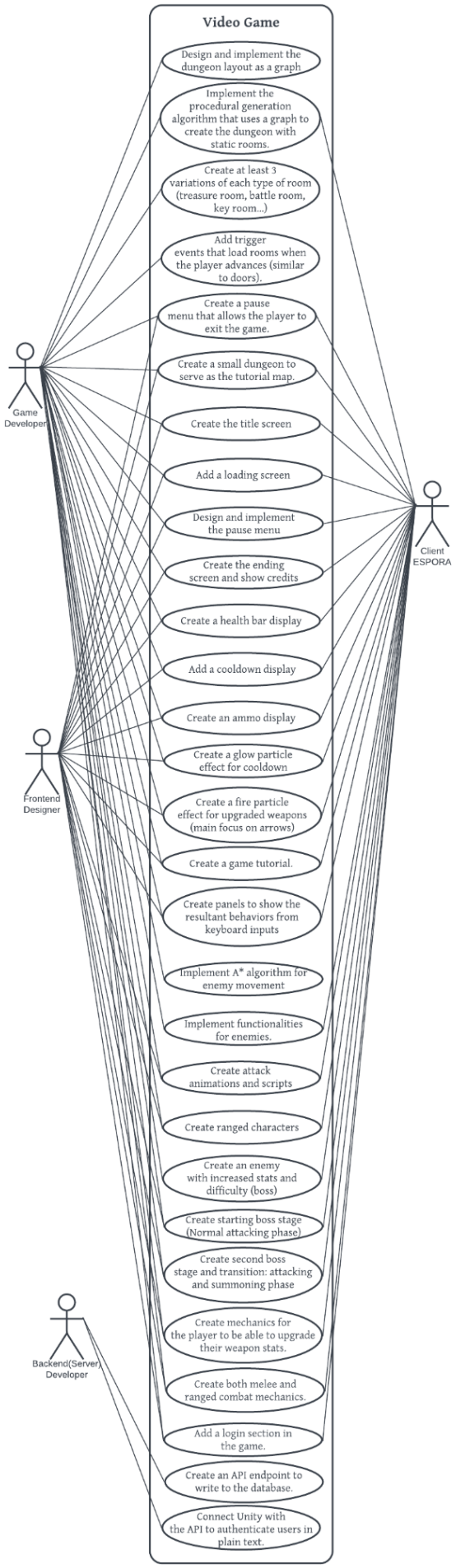
1. Create final presentation slides respecting our aesthetic.
2. Find a set of beta testers and test the game ourselves to detect any unexpected behaviors.
3. Separate game behaviors with classes.
4. Create an ER model of the data.
5. Detect what are the different relationships between entities.
6. Add test data that helps to test CRUD operations for future API implementation.
7. Users can create accounts to store their information.
8. Implement database views to create a security barrier.
9. Implement CRUD operations over the database.
10. Implement a schema in MySQL following the third normal form.
11. Create all the possible restrictions for the tables like PK, FK, Not Null, Index, Unique, Check.
12. Implement the possible relationships among tables, following the one to one, one to many and many to many cardinalities.
13. Create the website's homepage.
14. Create an about section.
15. Create the info page.
16. Create the game page.
17. Create a statistics view (with at least 3 visualizations).
18. Create a manual section on the website.
19. Read/consult database stored data through the API.
20. Design and implement plots on the website.
21. Embed the video game in the website.
22. Deploy the website using a hosting service (Vercel).
23. Create an API endpoint for administrators.
24. Create the webpage for the administrators to manage database operations.
25. Design and implement the dungeon layout as a graph.
26. Implement the procedural generation algorithm that uses a graph to create the dungeon with static rooms.
27. Create at least 3 variations of each type of room (treasure room, battle room, key room…).
28. Add trigger events that load rooms when the player advances (similar to doors).
29. Create a pause menu that allows the player to exit the game.
30. Create the title screen.
31. Add a loading screen.
32. Create a health bar display.
33. Add a cooldown display.
34. Create an ammo display.
35. Create a glow particle effect for cooldown.
36. Create a fire particle effect for upgraded weapons (main focus on arrows).
37. Create a game tutorial.
38. Create a small dungeon to serve as the tutorial map.
39. Create panels to show the resultant behaviors from keyboard inputs.
40. Implement A* algorithm for enemy movement.
41. Implement functionalities for enemies.
42. Create attack animations and scripts.
43. Create ranged characters.

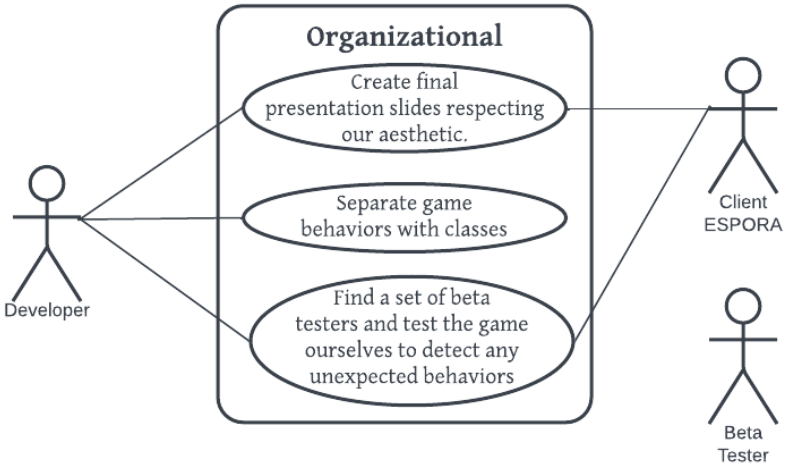
44. Create an enemy with increased stats and difficulty (boss).
45. Create a starting boss stage (Normal attacking phase).
46. Create a second boss stage and transition: attacking and summoning phase.
47. Create mechanics for the player to be able to upgrade their weapon stats.
48. Create both melee and ranged combat mechanics.
49. Create the login screen.
50. Create an API endpoint to write to the database.
51. Connect Unity with the API to authenticate users in plain text.

UML Use Case Diagrams




Web
Create the website's homepage
Create an about section
Create the info page
Create the game page
Create a statistics view (with at least 3 visualizations)
Create a manual section on the website.
Read/consult database stored data.
Design and implement plots on the website.
Embed the video game in the website.
Deploy the website using a hosting service (Vercel)
Create an API endpoint for administrators.
Create the webpage for the administrators to manage database operations.
Frontend Developer
Backend Developer
Client ESPORA
Database Administrator


Video Game
Design and implement the dungeon layout as a graph.
Implement the procedural generation algorithm that uses a graph to create the dungeon with static rooms.
Create at least 3 variations of each type of room (treasure room, battle room, key room...)
Add trigger events that load rooms when the player advances (similar to doors).
Create a pause menu that allows the player to exit the game.
Create a small dungeon to serve as the tutorial map.
Create the title screen
Add a loading screen
Design and implement the pause menu
Create the ending screen and show credits
Create a health bar display
Add a cooldown display
Create an ammo display
Create a glow particle effect for cooldown
Create a fire particle effect for upgraded weapons (main focus on arrows)
Create a game tutorial.
Create panels to show the resultant behaviors from keyboard inputs
Implement A* algorithm for enemy movement
Implement functionalities for enemies.
Create attack animations and scripts
Create ranged characters
Create an enemy with increased stats and difficulty (boss)
Create starting boss stage (Normal attacking phase)
Create second boss stage and transition: attacking and summoning phase
Create mechanics for the player to be able to upgrade their weapon stats.
Create both melee and ranged combat mechanics.
Add a login section in the game.
Create an API endpoint to write to the database.
Connect Unity with the API to authenticate users in plain text.
Game Developer
Frontend Designer
Backend(Server) Developer
Client ESPORA



UML Tables

Use Case 1	**Create final presentation slides respecting our aesthetic.**
Related Requirements	
Goal in Context	This presentation is intended for the closing of our work where the finished project is shown.
Preconditions	Define a style that will be the basis of the presentation.
Successful End Condition	The final presentation complies with the proposed style guidelines.
Failed End Condition	The final presentation is not aligned to the proposed style base.
Primary Actors	Developer.
Secondary Actors	Client.
Trigger	The project manager defines the style of the final presentation.
Main Flow	• Choosing the style of our project. • Build presentation slides.
Extensions	

Use Case 2	**Find a set of beta testers and test the game ourselves to detect any unexpected behaviors.**
Related Requirements	21, 22
Goal in Context	Inviting people to test the game during development to patch any bugs.
Preconditions	Have game scenes built.

Successful End Condition	All bugs (if they exist) were discovered and fixed.
Failed End Condition	Bugs were found that were not corrected or bugs were found that were not discovered during testing.
Primary Actors	Developers.
Secondary Actors	Beta Testers.
Trigger	Beta testers need access to the game to try it out.
Main Flow	• Build playable scenes. • Invite people to try these parts of the game. • If errors are found, they are reported to fix them.
Extensions	

Use Case 3	**Separate game behaviors with classes**
Related Requirements	
Goal in Context	Each class in the game is separated by its characteristics and abilities with its own name.
Preconditions	Have our characters with defined characteristics and skills.
Successful End Condition	The classes are easily distinguishable from each other.
Failed End Condition	It is not possible to differentiate between one class and another since they do not have unique characteristics.
Primary Actors	Developers.
Secondary Actors	Client.
Trigger	Propose gameplay diversity.
Main Flow	• Analyze in-game world scalability. • According to the context of the world, plan future and actual implementations of classes with unique characteristics.
Extensions	

Use Case 4	**Create an ER model of the data.**
Related Requirements	5
Goal in Context	Create an ER model that allows clarity and easy data management.
Preconditions	Define which data is to be stored.

Successful End Condition	The defined entities of the model are appropriate.
Failed End Condition	The defined entities of the model are not adequate.
Primary Actors	Database Administrator.
Secondary Actors	Backend Developer.
Trigger	Important information to be stored is known.
Main Flow	• Know the information to be stored. • Set up the ER model according to the data.
Extensions	

Use Case 5	**Detect what are the different relationships between entities.**
Related Requirements	4
Goal in Context	The database schema is normalized and the use of database design tools to preserve the accuracy and integrity of queries.
Preconditions	Have the entities defined in the relational database schema.
Successful End Condition	Relationships between entities have a logical meaning and preserve the rules of normalization.
Failed End Condition	Relationships between entities are not specific and there is redundancy between the entities.
Primary Actors	Database Administrator.
Secondary Actors	Backend Developer.
Trigger	ER diagram is proposed.
Main Flow	• Identify all entities that describe our model. • Identify how the entities interact with each other. • Specify the relationships between entities. • Add the relationships to the ER diagram. • Verify that the model satisfies the 3rd normal form
Extensions	

Use Case 6	**Add test data that helps to test CRUD operations for future API implementation.**
Related Requirements	9
Goal in Context	Populate the database tables temporarily to verify that it is possible to Create, Read, Update and Delete elements from the schema.

Preconditions	ER model of the data.
Successful End Condition	Tables are filled with test data and all CRUD operations are executed correctly.
Failed End Condition	At least one of the CRUD operations is not executed as expected.
Primary Actors	Backend Developer.
Secondary Actors	
Trigger	Create the tables in MySQL.
Main Flow	• Try Create operation by filling tables with test data. • Create SQL queries to test Reading, Updating and Deleting.
Extensions	

Use Case 7	**Users can create accounts to store their information.**
Related Requirements	49
Goal in Context	Users have a database where information about their progress in the game is stored.
Preconditions	Having a functional database.
Successful End Condition	The user can create an account and store his information.
Failed End Condition	The user cannot create an account.
Primary Actors	Backend Developer.
Secondary Actors	Client. Beta Tester.
Trigger	There is a login or account creation screen.
Main Flow	• Create a username field. • Create a password field. • User creates a username and password.
Extensions	

Use Case 8	**Implement database views to create a security barrier.**
Related Requirements	17, 19, 20
Goal in Context	Reserved queries that convert into tables that derive from the entire database to get specific and complex results faster.

Preconditions	Have functional relational database schema.
Successful End Condition	Tables obtained from specific queries work accurately and quickly.
Failed End Condition	Tables obtained from specific queries do not yield accurate and efficient results.
Primary Actors	Backend Developer.
Secondary Actors	
Trigger	Need to generate a table with specific data.
Main Flow	• Define query. • Write the query in MySQL (Create View). • Check the view.
Extensions	

Use Case 9	**Implement CRUD operations over the database.**
Related Requirements	6
Goal in Context	The database must be capable of creating, updating, reading and deleting records.
Preconditions	Having a defined relational database schema.
Successful End Condition	The operations performed on the tables work in the correct way. No records are broken and no ambiguities are created.
Failed End Condition	Tables are not modified or operations corrupt records.
Primary Actors	Backend Developer.
Secondary Actors	Database Administrator.
Trigger	You need to modify a table.
Main Flow	• Create tables in the relational database schema. • Define the appropriate restrictions in the fields of the tables so that CRUD operations work correctly.
Extensions	

Use Case 10	**Implement a schema in MySQL following the third normal form.**
Related Requirements	4
Goal in Context	The MySQL database schema is standardized with the third normal form

	to reduce duplicate data, anomalies and simplify data management.
Preconditions	Have a relational database model.
Successful End Condition	No redundancy or anomalies between tables.
Failed End Condition	Duplicate data, ambiguities and complicated data management are encountered.
Primary Actors	Database Administrator.
Secondary Actors	Backend Developer.
Trigger	Tables are defined within the relational database schema.
Main Flow	• Have tables in the database schema. • Apply the third normal form between tables. • Verify that there is no ambiguity or duplicate data.
Extensions	

Use Case 11	**Create all the possible restrictions for the tables like PK, FK, Not Null, Index, Unique, Check.**
Related Requirements	
Goal in Context	Create constraints to ensure data accuracy and integrity.
Preconditions	Have the table fields and their context defined.
Successful End Condition	The integrity of the data is preserved.
Failed End Condition	Data is corrupted or queries have low accuracy.
Primary Actors	Database Administrator
Secondary Actors	Backend Developer. Database Administrator.
Trigger	Create a table.
Main Flow	• Define the fields for the table. • Identify the context of the data in each field. • Implement the necessary restrictions according to the context of the data.
Extensions	

Use Case 12	**Implement the possible relationships among tables, following the one to one, one to many and many to many cardinalities.**

Related Requirements	4, 5
Goal in Context	Determine the cardinality of a relationship between entities.
Preconditions	Have a defined ER diagram.
Successful End Condition	The relationships between each entity are visible and their cardinality is logical.
Failed End Condition	The cardinality defined in the relationships between entities is not consistent.
Primary Actors	Database Administrator.
Secondary Actors	Backend Developer.
Trigger	Have the relationships between entities defined.
Main Flow	• Propose the ER model. • Define the relationship between entities. • According to the context of the relationship between entities. • Define cardinality.
Extensions	

Use Case 13	**Create the website's homepage.**
Related Requirements	14, 15, 16, 17, 18
Goal in Context	Have a homepage on the website where the project will be presented.
Preconditions	Design a homepage UI in *Figma*.
Successful End Condition	The homepage is visible on the website.
Failed End Condition	The home page does not exist on the website.
Primary Actors	Frontend Developer.
Secondary Actors	Client.
Trigger	Create the HTML document for development.
Main Flow	• Propose the elements of the homepage. • Propose an idea of the final result using graphic design tools. • Create the HTML for the structure. • Create the CSS for the styles.
Extensions	

Use Case 14	**Create an about section.**
Related Requirements	13, 15, 16, 17, 18
Goal in Context	Display information about the creators of the game in a section of the webpage.
Preconditions	Design an about section UI in *Figma*.
Successful End Condition	The about section is rendered correctly on the devices where the game should be supported.
Failed End Condition	The about section is not displayed as expected on a target device.
Primary Actors	Frontend Developer.
Secondary Actors	Client.
Trigger	Create the static files (HTML, CSS) for the webpage.
Main Flow	• Propose the elements of the about section. • Propose an idea of the final result using graphic design tools. • Create the HTML for the structure. • Create the CSS for the styles.
Extensions	

Use Case 15	**Create the info page.**
Related Requirements	13, 14, 16, 17, 18
Goal in Context	Display the relevant information concerning the game Break Into Valhalla.
Preconditions	Design an info page UI in *Figma*.
Successful End Condition	Any user can access the information of our game in the webpage.
Failed End Condition	No information is displayed about the game.
Primary Actors	Frontend Developer.
Secondary Actors	Client.
Trigger	Create the HTML document for development.
Main Flow	• Propose the elements of the info page. • Propose an idea of the final result using graphic design tools. • Create the HTML for the structure. • Create the CSS for the styles.
Extensions	

Use Case 16	**Create the game page.**
Related Requirements	13, 14, 15, 17, 18
Goal in Context	Display a space reserved for the video game on the web page.
Preconditions	Design a game page UI in *Figma*.
Successful End Condition	The game is shown in its defined space and is playable.
Failed End Condition	The game isn't shown in its defined space and isn't playable
Primary Actors	Frontend Developer.
Secondary Actors	Client.
Trigger	Create the HTML document for development.
Main Flow	• Propose the elements of the game page. • Propose an idea of the final result using graphic design tools. • Create the HTML for the structure. • Create the CSS for the styles. • Embed the game in it.
Extensions	

Use Case 17	**Create a statistics view (with at least 3 visualizations).**
Related Requirements	13, 14, 15, 16, 18
Goal in Context	Display relevant metrics for the players in a webpage.
Preconditions	Determine the statistics to show. Design the plots. Create the statistics section UI in *Figma*.
Successful End Condition	The statistics section is rendered correctly in all target devices, the plots show data that corresponds to the one stored in the database and plots are displayed correctly.
Failed End Condition	If the plots are incorrectly displayed, the data is not consistent with the stored results or the webpage is not rendered correctly.
Primary Actors	Frontend Developer.
Secondary Actors	Client.
Trigger	Create the HTML document for development.
Main Flow	• Propose the elements of the Statistics page. • Propose an idea of the final result using graphic design tools. • Create the HTML for the structure. • Create the CSS for the styles.

	• Access the database through API endpoint. • Display them in a way they are clear and easily readable.
Extensions	

Use Case 18	**Create a manual section on the website.**
Related Requirements	13, 14, 15, 16, 17
Goal in Context	Display the video game manual on the website.
Preconditions	Have a game manual and an HTML document.
Successful End Condition	The game manual is displayed in its defined space and loads correctly.
Failed End Condition	The game manual is displayed in an unwanted space or does not load correctly.
Primary Actors	Frontend Developer.
Secondary Actors	Client.
Trigger	Create the HTML for development and have detailed instructions on how to play the game.
Main Flow	• Propose the elements of the manual section. • Propose an idea of the final result using graphic design tools. • Create the HTML for the structure. • Create the CSS for the styles. • Redact a game manual.
Extensions	

Use Case 19	**Read/consult database stored data through the API**
Related Requirements	6, 9
Goal in Context	Create API endpoints that allow developers and administrators to obtain data from the database.
Preconditions	Have a database with uncorrupted data that is properly stored.
Successful End Condition	Database records are accessed.
Failed End Condition	Database records are inaccessible.
Primary Actors	Backend Developer.
Secondary Actors	

Trigger	Create an Express.js API.
Main Flow	• Have an API endpoint implemented. • Make simple queries to test if the read function is working properly. • Make simple queries to test if the consult function is working properly.
Extensions	

Use Case 20	**Design and implement plots on the website.**
Related Requirements	17
Goal in Context	Display statistical graphs on the website.
Preconditions	Connect the website to the database. Have an HTML document.
Successful End Condition	Website displays plots in the sections they should be for users to read.
Failed End Condition	No plots are displayed.
Primary Actors	Frontend Developer.
Secondary Actors	Client.
Trigger	Have a HTML document and API endpoint connection.
Main Flow	• Have an API endpoint implemented. • Find a plotting library to create the visualizations. • Add the HTML elements, CSS styles and JS behaviors to make the plots dynamic with the data read from the database.
Extensions	

Use Case 21	**Embed the video game in the website.**
Related Requirements	16
Goal in Context	Make the game accessible through the website
Preconditions	Game section in the website is designed.
Successful End Condition	The game can be played in any (recently new) web browser.
Failed End Condition	The game cannot be played in a desired web browser.
Primary Actors	Frontend Developer.

Secondary Actors	Client.
Trigger	Game is ready to be built.
Main Flow	• Build the game with WebGL in Unity. • Create a JS script to create the container and canvas element for the game.
Extensions	

Use Case 22	**Deploy the website using a hosting service (Vercel).**
Related Requirements	13, 14, 15, 16, 17, 18, 21
Goal in Context	Make the website accessible with an Internet connection using a hosting service.
Preconditions	The website elements are implemented.
Successful End Condition	The website is stored at least locally.
Failed End Condition	The website isn't even hosted locally.
Primary Actors	Backend Developer.
Secondary Actors	Client.
Trigger	Have a HTML document and cloud services.
Main Flow	• Create an account in Vercel. • Add the source code repository to Vercel. • Create GitHub actions to deploy the website. • Check that the website is accessible through the link.
Extensions	

Use Case 23	**Create an API endpoint for administrators.**
Related Requirements	9, 10
Goal in Context	Separate administrator's functionality (e.g. accessing the database directly) and make it accessible through an API endpoint.
Preconditions	
Successful End Condition	Administrators can access specific functionality through the API.
Failed End Condition	Administrators are unable to access the database through the API.
Primary Actors	Backend Developer.

Secondary Actors	Database Administrator.
Trigger	Create an Express.js API.
Main Flow	• CRUD operations are implemented in the database. • API endpoint is added to express.
Extensions	

Use Case 24	**Create the webpage for the administrators to manage database operations.**
Related Requirements	9, 19, 23
Goal in Context	Secure and isolated management of database operations.
Preconditions	Implement CRUD operations. Populate tables with data (dummy data or real data).
Successful End Condition	Administrators can perform database operations securely in a private environment.
Failed End Condition	Administrators do not have the privacy to perform database operations in a secure manner.
Primary Actors	Frontend Developer. Backend Developer.
Secondary Actors	Database Administrator.
Trigger	Implement the API endpoint for administrators.
Main Flow	• Create database schema. • Create an isolated space for administrators. • Grant the administrator a special url so that they can access their environment.
Extensions	

Use Case 25	**Design and implement the dungeon layout as a graph.**
Related Requirements	26, 27, 28
Goal in Context	Abstract the creation of the dungeon to ensure that rooms are accessible in a specific order, yet it can be procedurally generated.
Preconditions	Knowledge of graph functionality. Procedural generation algorithm knowledge.
Successful End Condition	The levels generate at random, multiplying the ways levels are configured and impeding the player from memorizing the levels.

Failed End Condition	The levels are loaded and made like a predetermined map.
Primary Actors	Game Developer.
Secondary Actors	
Trigger	Separate the dungeon design into rooms and the rooms into categories (key room, chest room, etc.)
Main Flow	• Design the abstract layout of the dungeon graphically. • Create a graph representation in computer memory using C# that is optimized for search operations
Extensions	

Use Case 26	**Implement the procedural generation algorithm that uses a graph to create the dungeon with static rooms.**
Related Requirements	25, 27, 28
Goal in Context	Procedural generation is implemented and levels are generated at random using prefabricated rooms.
Preconditions	Create the graph representation of the dungeon.
Successful End Condition	Random dungeons are created according to the abstract graph representation.
Failed End Condition	Dungeons are not created, they are always the same or some rooms are not accessible.
Primary Actors	Game Developer.
Secondary Actors	Client.
Trigger	Have prefabricated rooms.
Main Flow	• Create a graph representation in computer memory using C# that is optimized for search operations. • Create a game object that stores room prefabs • Create a script that selects rooms using random indices
Extensions	

Use Case 27	**Create at least 3 variations of each type of room (treasure room, battle room, key room…).**
Related Requirements	25, 26, 28
Goal in Context	Create sufficient content for the procedural generation to create unique levels.

Preconditions	Create the graph representation of the dungeon. Create or find a tileset to use.
Successful End Condition	At least 3 prefabricated rooms are created for each of the room categories.
Failed End Condition	There is at least one room category that is missing variations.
Primary Actors	Game Developer.
Secondary Actors	
Trigger	Dungeon is represented as a graph and rooms are divided into categories.
Main Flow	• Create a room template to create rooms with the same size (29 by 29 tiles). • Create different tilemaps to distinguish between different “materials” and walls. • Design level prefabs.
Extensions	Design or find tilesets.

Use Case 28	**Add trigger events that load rooms when the player advances (similar to doors).**
Related Requirements	25, 26, 27
Goal in Context	Add game components that allow the player to traverse the dungeon.
Preconditions	Dungeon is represented as a graph. Prefabricated rooms are stored.
Successful End Condition	When reaching the trigger the player can move to a new room.
Failed End Condition	Trigger event is not present or it does not load rooms correctly.
Primary Actors	Game Developer.
Secondary Actors	
Trigger	Procedurally generated rooms are stored in memory and the player reaches the end of a room.
Main Flow	• Create game objects with colliders set as triggers. • Create a script to instantiate rooms on trigger enter.
Extensions	

Use Case 29	**Create a pause menu that allows the player to exit the game.**
Related Requirements	
Goal in Context	Give the player the ability to pause the game at any given moment.
Preconditions	Have a game to pause.
Successful End Condition	The player can decide at any moment to quit the game or pause it for a while.
Failed End Condition	The game is one that is constantly playing and can't stop.
Primary Actors	Game Developer. Frontend designer.
Secondary Actors	Client.
Trigger	A new scene that is empty.
Main Flow	• Create a new empty scene. • Make it stop time. • Create buttons for quitting, options and resume.
Extensions	

Use Case 30	**Create the title screen.**
Related Requirements	28, 50, 51
Goal in Context	Have some sort of landing page in the game where players can see a leaderboard.
Preconditions	Both having a database to store statistics saved by different player runs and having beta testers to play and have save data.
Successful End Condition	The game has a title screen that shows a leaderboard and has buttons that take you to different screens or the game itself (options, leaderboard, play, and exit).
Failed End Condition	The game doesn't have a title screen and plays automatically.
Primary Actors	Game Developer. Frontend designer.
Secondary Actors	Client.
Trigger	A new scene that is empty.
Main Flow	• Have some mechanics implemented for the game. • Create a new scene before the game plays that has our designs. • Make buttons for it to work. • Implement statistics view.

	• Make it start the game if the play button is clicked.
Extensions	

Use Case 31	**Add a loading screen.**
Related Requirements	25, 26, 28
Goal in Context	Make a scene that plays a simple animation while the game is loading.
Preconditions	Map generation is working. Title screen works and play triggers an event that starts map generation.
Successful End Condition	The loading screen plays when needed.
Failed End Condition	The loading screen fails and doesn't play in any instance.
Primary Actors	Game Developer. Frontend designer.
Secondary Actors	Client.
Trigger	A new scene that is empty.
Main Flow	• Design the loading screen in an external graphics tool. • Create the animations within the scene in Unity. • Implement a scene change in Unity when loading is finished.
Extensions	

Use Case 32	**Create a health bar display.**
Related Requirements	
Goal in Context	Have a visual aid that shows hit points to the user at all times.
Preconditions	The implementation of hitpoints.
Successful End Condition	The health bar not only shows the player's vitality at all times but it also changes color depending on the percentile.
Failed End Condition	Health bar is either inconsistent or not showing at all.
Primary Actors	Game Developer. Frontend designer.
Secondary Actors	Client.
Trigger	Interactive game objects (player, enemies and boss).

Main Flow	• Acquire knowledge of health bar implementation. • Implement damage dealing functions. • Implement hit point. • Design health bar. • Implement graphics onto the scene and give them behavior.
Extensions	Implement hit points for enemies and make a health bar for Boss (Hel).

Use Case 33	**Add a cooldown display.**
Related Requirements	
Goal in Context	Make the players aware of the use of their cooldown ability.
Preconditions	Cooldown ability.
Successful End Condition	The player sees a display of the cooldowns they have (roll and blessing) which upon use start a timer. • 30 second timer after using the “Blessing” cooldown. • 6 second timer after using the “Dash” cooldown.
Failed End Condition	There is no display of cooldowns
Primary Actors	Game Developer. Frontend designer.
Secondary Actors	Client.
Trigger	Player behavior implementation.
Main Flow	• Design cooldown UI element in *Figma.* • Create cooldown logic with a script (setting timers). • Add visual elements (like a timer on top of the cooldown) to help the user understand when they are able to use the abilities again.
Extensions	Implementation for the whole weapon and cooldown display.

Use Case 34	**Create an ammo display.**
Related Requirements	
Goal in Context	Make the players aware of the use of their ammunition.
Preconditions	Projectile logic.
Successful End Condition	The player sees a display of the projectiles (mainly arrows) they can still use.
Failed End Condition	There is no display of ammo.
Primary Actors	Game Developer. Frontend designer.

Secondary Actors	Client.
Trigger	Player behavior implementation.
Main Flow	• Design ammo UI element in *Figma.* • Create ammunition logic with a script (saving projectiles). • Add visual elements for ammo to the game.
Extensions	Implementation for the whole weapon and cooldown display.

Use Case 35	**Create a glow effect for cooldown.**
Related Requirements	33
Goal in Context	Make the player aware of the cooldown being active.
Preconditions	Cooldown ability.
Successful End Condition	The player feels as though the gods have blessed them and feel more powerful while a “holy” glow is shown around the player.
Failed End Condition	There is no glow upon blessing activation.
Primary Actors	Game Developer. Frontend designer.
Secondary Actors	Client.
Trigger	Player behavior implementation.
Main Flow	• Choose the color for our glow. • Implement it through code and Unity UI.
Extensions	

Use Case 36	**Create a fire particle effect for upgraded weapons (main focus on arrows).**
Related Requirements	
Goal in Context	Make the player aware of the strength of their weapons.
Preconditions	Weapon sprites. Fire sprite (for particle effect). Weapon upgrade logic.
Successful End Condition	There is a visual indicator that tells the player their weapons are stronger.
Failed End Condition	It is not clear for the player that weapons are stronger after upgrading them.

Primary Actors	Game Developer. Frontend designer.
Secondary Actors	Client.
Trigger	Implementation of player combat mechanics.
Main Flow	• Add particle effect to the arrows. • Add fire sprite to particle effect. • Disable particle effect by default. • Enable the particle effect through a script and game logic.
Extensions	

Use Case 37	**Create a game tutorial.**
Related Requirements	
Goal in Context	Show a tutorial that allows the player to learn how to use the controls and game mechanics.
Preconditions	Have functional game mechanics and controls defined.
Successful End Condition	The tutorial is displayed on the screen and matches the internal specifications of the game.
Failed End Condition	The tutorial is not displayed or controls do not match internal game specifications.
Primary Actors	Game Developer. Frontend designer.
Secondary Actors	Client.
Trigger	Start a new game.
Main Flow	• Define the movements and mechanics of the players. • Implement the mechanics that respond to the assigned keys. • Design and implement the tutorial that explains to new players how to play the game.
Extensions	

Use Case 38	**Create a small dungeon to serve as the tutorial map.**
Related Requirements	37
Goal in Context	Design the starting level to teach the player game mechanics in a controlled environment (with little to no risk).
Preconditions	Design or find a tileset for *Valhalla.*

Successful End Condition	There is a starting level where the player can learn the game mechanics while playing.
Failed End Condition	The starting level does not allow the player to learn the game mechanics in a controller environment.
Primary Actors	Game Developer. Frontend designer.
Secondary Actors	Client.
Trigger	Design a sequence (or timeline) of mechanics to teach the player.
Main Flow	• Create a tilemap grid. • Create different tilemaps to distinguish between different "materials" and walls. • Design level prefabs
Extensions	

Use Case 39	**Create panels to show the resultant behaviors from keyboard inputs.**
Related Requirements	
Goal in Context	During the game, when interacting with objects, the key that must be pressed for a specific action is displayed.
Preconditions	Have defined types of interactions in the game.
Successful End Condition	When the player approaches an object in the game, the action that can be performed and which key activates it are displayed on the screen.
Failed End Condition	The player approaching an object in the game does not explain which actions can be executed and which keyboard input triggers it.
Primary Actors	Game Developer. Frontend designer.
Secondary Actors	Client.
Trigger	The player wishes to interact with an object in the game.
Main Flow	• Define the actions that can exist in the game with keyboard inputs. • Define the single key that will trigger each action. • Design and implement a screen that shows the player what he can do if he has an object near him.
Extensions	

Use Case 40	**Implement A* algorithm for enemy movement.**
Related Requirements	
Goal in Context	Make our enemies and boss avoid obstacles while chasing after our player.
Preconditions	Enemies can move in the game.
Successful End Condition	Enemies are able to follow the player avoiding the obstacles that appear in their path.
Failed End Condition	Enemies are unable to avoid obstacles and get stuck while chasing the player.
Primary Actors	Game Developer.
Secondary Actors	
Trigger	The enemy detects the player.
Main Flow	• Understand the A* algorithm. • Implement the behavior for path calculation and refreshing. • Make movement change animation states. • Add scripts to enemies.
Extensions	

Use Case 41	**Implement functionalities for enemies.**
Related Requirements	40, 42
Goal in Context	Implement enemy behaviors like attacking and dropping loot.
Preconditions	Enemy sprites and animations.
Successful End Condition	Enemies drop loot, attack players and spawn on rooms.
Failed End Condition	At least one of the expected behaviors for enemies (attacking, dropping loot, spawning, etc).
Primary Actors	Game Developer.
Secondary Actors	Client.
Trigger	Implementation of enemy behaviors and animations.
Main Flow	• Create enemy sprites. • Create C# script for enemy behaviors. • Create enemy animations (walking, attacking).
Extensions	

Use Case 42	**Create attack animations and scripts.**
Related Requirements	41, 43
Goal in Context	Make the player notice when their attack starts and finishes.
Preconditions	Have the player's frames for his attack movement.
Successful End Condition	The frames create a fluid movement of the character showing an attack animation that works thanks to C# programming.
Failed End Condition	The frames create a motion that is not very fluid and does not respond as expected because of the way it was programmed.
Primary Actors	Game Developer. Frontend designer.
Secondary Actors	Client.
Trigger	Press the key that was assigned to the attacks.
Main Flow	• Design the types of attacks. • Create the frames to be able to perform the attack animation. • Use the Unity animator to make the sequence of frames to simulate the movement. • Use C# to make the scripts that tell the game when these animations should be activated.
Extensions	

Use Case 43	**Create ranged characters.**
Related Requirements	42
Goal in Context	Have characters that can perform actions within a defined distance range.
Preconditions	Define the character's range of reach.
Successful End Condition	Character range actions are executed correctly and are visible.
Failed End Condition	Character range range actions do not respect the defined range or are not visible.
Primary Actors	Game Developer. Frontend designer.
Secondary Actors	Client.
Trigger	Press the key assigned to range actions.
Main Flow	• Define ranged enemy behavior and pathfinding.

	• Design projectile graphics. • Implement shooting mechanics.
Extensions	

Use Case 44	**Create an enemy with increased stats and difficulty (boss).**
Related Requirements	41
Goal in Context	Have a battle that feels completely different than the rest and make it so it feels epic and overwhelming to an extent.
Preconditions	Design or find a sprite for the boss (Hel) Create common enemy behaviors (which will be extended)
Successful End Condition	The boss fight is exhilarating.
Failed End Condition	The boss is either a pushover and is bland or isn't even implemented.
Primary Actors	Game Developer.
Secondary Actors	Client.
Trigger	Clearing other rooms.

Main Flow	• Design the spritesheet for Hel making it so her depiction is accurate.
	• Implement A* pathfinding for her.
	• Implement the two phases of the battle.
	  ○ Attacking behaviors.
	  ○ Summoning the dead while maintaining the same attack pattern.
	• Implement the second phase when Hel's health bar gets to 50%.
	• When Hel "dies", instead of going to the win screen, the scene will change and Hel stands up and hits the player so hard they die instantly.
	• Implement a winning cutscene of either Odin or Freyja welcome the player into either Valhalla or Fólkvangr.

Extensions	

Use Case 45	**Create a starting boss stage (Normal attacking phase).**
Related Requirements	44
Goal in Context	To make the final battle exciting and challenging, the final boss will have a starting stage with "normal" attacking behaviors.
Preconditions	Design or find a sprite for the boss (Hel). Create common enemy behaviors (which will be extended).

	Create an enemy with increased stats and difficulty.
Successful End Condition	The boss has stages and starts with normal attacking behaviors
Failed End Condition	The boss fight does not have different phases or does not follow the expected behaviors on each phase.
Primary Actors	Game Developer. Frontend designer.
Secondary Actors	Client.
Trigger	The player reaches the final boss.
Main Flow	• Implement melee attack mechanics with Hel's weapon being a Scythe.
Extensions	

Use Case 46	**Create a second boss stage and transition: attacking and summoning phase.**
Related Requirements	44, 45
Goal in Context	To make the final battle exciting and challenging, the final boss will have a second stage with frantic attacking behaviors and summoning the dead to attack for her.
Preconditions	Design or find a sprite for the boss (Hel). Create common enemy behaviors (which will be extended). Create an enemy with increased stats and difficulty. Create erratic attacking behaviors Create enemy spawner logic
Successful End Condition	The boss has stages and changes to an erratic phase with "minions" when dropped to 50% or below.
Failed End Condition	The boss fight does not have different phases or does not follow the expected behaviors on each phase.
Primary Actors	Game Developer. Frontend designer.
Secondary Actors	Client.
Trigger	The player reaches the final boss.
Main Flow	• Implement melee attack mechanics with Hel's weapon being a Scythe. • Implement Hel summoning animation. • Implement summoning mechanics. • Set the summoning to be melee and ranged Draugr.

Extensions	44

Use Case 47	**Create mechanics for the player to be able to upgrade their weapon stats.**
Related Requirements	
Goal in Context	Make the player feel they're sort of leveling up by giving the feel of getting stronger.
Preconditions	Weapon sprites and animations. Weapon upgrade item sprite.
Successful End Condition	Players can upgrade their weapons through game items.
Failed End Condition	The player has no weapon upgrade.
Primary Actors	Game Developer. Frontend designer.
Secondary Actors	Client.
Trigger	Store or chest drop.
Main Flow	• Design the weapon upgrades. • Implement a logic stat bonus for weapons.
Extensions	Design and implement chests.

Use Case 48	**Create both melee and ranged combat mechanics.**
Related Requirements	
Goal in Context	Have unique melee and ranged mechanics making the game more dynamic.
Preconditions	Have characters specifically designed for both melee and ranged mechanics.
Successful End Condition	Characters have melee and ranged combat mechanics.
Failed End Condition	Characters do not possess melee and ranged combat mechanics or do not function correctly.
Primary Actors	Game Developer. Frontend designer.

Secondary Actors	Client.
Trigger	Being in combat.
Main Flow	• Design of combat mechanics. • Implement the animations for each combat mechanic. • Define a cooldown for the next attack to be executed.
Extensions	

Use Case Name 49	**Add a login section in the game.**
Related Requirements	50, 51
Goal in Context	There is a visible form with fields for accounts.
Preconditions	The game must exist.
Successful End Condition	When starting the game the login screen appears.
Failed End Condition	When starting the game, the login screen does not appear.
Primary Actors	Game Developer. Frontend designer.
Secondary Actors	Client.
Trigger	The user opens the game.
Main Flow	• The user enters the page to play the video game. • When the game opens, the login screen appears.
Extensions	

Use Case 50	**Create an API endpoint to write to the database.**
Related Requirements	49, 51
Goal in Context	Allow the game to create new users in the database when they are first registered through the API.
Preconditions	Database tables for users exist and can be modified through create operations.
Successful End Condition	API endpoint is functioning properly, allowing the game to write new data in the database.
Failed End Condition	API endpoint doesn’t work and the game doesn’t communicate with the database.
Primary Actors	Backend (Server) Developer.

Secondary Actors	
Trigger	Player registers or logs into the game.
Main Flow	• Create express.js API • Create functions to write to the database • Expose functions for the game to write to the database
Extensions	

Use Case 51	**Connect Unity with the API to authenticate users in plain text.**
Related Requirements	
Goal in Context	Authenticate players with a username and password within the game.
Preconditions	Players register their accounts.
Successful End Condition	Players can only login to their accounts using the correct username/email and password combination.
Failed End Condition	Players are having problems logging into their account with their username and password, or anyone can access your account despite incorrect passwords (security breach).
Primary Actors	Backend (Server) Developer.
Secondary Actors	
Trigger	Game login page.

Main Flow	• Create login screen elements.
	  ○ Email/username field.
	  ○ Password field.
	• Create API function to authenticate email/username and password combination.
	• Connect unity to the API.

Extensions	