

Alvaro Andres Romero Castro
Baruj Vladimir Ramirez Escalante
Brayan Alejandro Munoz Perez
Jenny Catherine Herrera Garzon
Juan David Ladino Triana

Workshop No. 1 — Requirements, User Stories, and Story Mapping

1. Requirements Documentation

1.1. Functional Requirements

The system must:

1. Allow the player to create and name their character.
2. Display the character's main statistics (health, attack, defense, experience, and level).
3. Enable player movement and interaction within the game map.
4. Generate enemies randomly in specific zones of the map.
5. Start a battle automatically when the player collides with an enemy.
6. Calculate combat outcomes based on the player's and enemy's statistics.
7. Allow the player to collect items after defeating enemies or exploring.
8. Display an inventory showing collected items.
9. Allow the player to use items to restore health or enhance attributes.
10. Update the player's experience and level after each battle.
11. Save and load player progress using local storage in the browser.
12. Record and store the total playtime in an online database.
13. Display appropriate end-game screens for victory or defeat.
14. The system must provide an internal testing mode that allows authorized users to manually spawn enemies or items for validation and balancing purposes.

1.2. Non-Functional Requirements

The system should:

1. Usability: Provide an intuitive interface for easy navigation and interaction, accessible for both beginners and experienced players.

2. Performance: Maintain a minimum frame rate of 30 FPS when executed in the browser using Unity WebGL.
3. Compatibility: Be fully functional on major web browsers (Chrome, Firefox, Edge) without additional installations.
4. Scalability: Support future expansion, allowing new enemies, items, or maps to be added without altering the core system.
5. Reliability: Ensure data persistence even if the player refreshes or closes the browser tab.
6. Security: Use secure communication (HTTPS or Firebase authentication) to prevent unauthorized data access.
7. Maintainability: Keep the code modular and documented for future updates by different team members.
8. Portability: Allow easy deployment to web platforms such as Itch.io or institutional servers.
9. Availability: Be accessible online 24/7 with minimal downtime once deployed.
10. Aesthetics: Use consistent visuals, readable fonts, and clear HUD elements to enhance the player's experience.

2. User Stories

2.1. Set Character Name

As a player, I want to name my character so I can have a personalized experience.

Acceptance Criteria:

Given the player is on the start screen,

When they select "New Game" and enter a name,

Then the system creates a character profile and starts the adventure.

2.2. Check Stats

As a player, I want to see my character's stats so that I can monitor health, attack, defense, and experience.

Acceptance Criteria:

Given the game is running,

When I open the stats panel or HUD,

Then my current values are displayed accurately.

2.3. Move

As a player, I want to move my character freely around the map so that I can explore the environment.

Acceptance Criteria:

Given the player is in the main game area,

When movement keys or buttons are pressed,

Then the character moves smoothly in the selected direction.

2.4. Encounter Enemies

As a player, I want to encounter enemies while exploring so that I can engage in combat and earn experience.

Acceptance Criteria:

Given enemies exist in the area,

When I approach them,

Then combat begins automatically or through interactions.

2.5. Combat

As a player, I want to attack, defend, and take turns during combat so that I can defeat enemies strategically:

Given I'm in combat mode,,

When I choose an action (attack, defend, wait),

Then the system executes it, updates stats, and applies enemy responses.

2.6. Win or Lose

As a player, I want the game to show a victory or defeat screen so that I can understand the outcome of the battle.

Acceptance Criteria:

Given combat ends,

When either the player or the enemy's HP reaches zero,

Then the corresponding win or lose screen appears.

2.7. Pick Up Objects

As a player, I want to collect items I find so that I can use them later or improve my character.

Acceptance Criteria:

Given I'm near an item,

When I press the interaction key,

Then the item is added to my inventory.

2.8. Equip Armor / Tools

As a player, I want to equip weapons or armor so that I can enhance my attack and defense stats.

Given I have an equipable item in my inventory,

When I choose to equip it,

Then my stats update accordingly.

2.9. Use Consumables

As a player, I want to use items from my inventory so that I can restore health or improve stats.

Acceptance Criteria:

Given I have a consumable in inventory,

When I use it,

Then the effect is applied and the item is removed.

2.10. Level Up

As a player, I want to gain experience and level up so that my character becomes stronger.

Acceptance Criteria:

Given I have earned enough experience points,

When I reach the threshold,

Then the system increases my level and updates my stats.

2.11. Create Account

As a player, I want to create an account to have my characters associated with it, for ease of access to them:

When I choose to register with my email and password,

The system creates a new account and allows me access to manage my characters.

2.12. Sign Into Account

As a player, I want to be able to access my account so that I may play the game with my stored characters.

Given I have registered previously,

When I sign in with my email and password,

The system grants me access to my characters and progress.

2.13. Save Character

As a player, I want to save my character's progress so that I can continue playing later from the same point:

Given I have completed a session,

When I choose to save,

Then the system stores stats and inventory in the database.

2.14. Load Character

As a player, I want to load a previously saved character so that I can continue my adventure:

Given saved data exists,

When I choose to load a character,

Then the system retrieves and restores my previous state.

2.15. Check Time

As a player, I want to see my total playtime so that I can track how long I've been playing.

Acceptance Criteria:

Given a session is active,

When I play,

Then the system records playtime and uploads it to the database.

2.16. Reach Goal

As a player, I want to reach specific objectives or checkpoints so that I can progress in the game.

Acceptance Criteria:

Given a goal is defined,

When I reach its location,

Then the system triggers a success message.

2.17. Spawn Enemy

As a tester, I want to spawn enemies manually so that I can verify that combat and detection systems work correctly.

Acceptance Criteria:

Given the game is running in test mode,

When I press the “Spawn Enemy” button or shortcut,

Then a new enemy should appear near the player and behave according to its AI rules.

2.18. Spawn Object

As a tester, I want to spawn items or objects manually so that I can validate pickup and inventory functionalities.

Acceptance Criteria:

Given the game is running in test mode,

When I press the “Spawn Object” button or shortcut,

Then the object should appear and be collectible by the player.

2.19. Change Player Values

As a tester, I want to modify player statistics such as health, damage, or experience so that I can test different gameplay scenarios and give feedback.

Acceptance Criteria:

Given the game is running in test mode,

When I adjust a stat value,

Then the new value should update immediately in the game.

3. User Story Mapping

link: https://miro.com/app/board/uXjVJE-rrk0=