GoGirls – Sarp Çağan Kelleci - Kemal Serbay Uğur - Emsalinur Arıca - Berke Karamanlı - Muhammet Üçler Çetin - Eray Önder

Use Cases for Rokue-Like

Use Case 1: Move Hero

Scope: Go Girls

Level: Subfunction

Primary Actor: Player

Stakeholders and Interests: Player, Developer Team, System

Preconditions:
- Play Mode is selected from Modes.
- The hero is controlled by the player.

Main Success Scenario
- The player uses arrow keys to move the hero.
- The system validates the move:
  - If the square is empty, the hero moves successfully.
  - If the square has a wall/object, the move is blocked by the system.
- Hero's position is updated on the grid.

Postconditions
- Hero's position is updated, or the move is blocked.

Extensions
- Move outside grid boundary is blocked
  - **Condition:**
    - The player attempts to move the hero beyond the defined grid boundaries.
  - **System Response:**
    - The move is blocked by the system.
    - Optional: A message is displayed to inform the player, (Example: "You cannot move outside the grid.").
  - **Outcome:** The hero remains in the current position within the grid.

- Interaction check if the hero is adjacent to a monster.
  - **Condition:**
    - Following an attempt at movement, the system determines that a monster is present in a nearby square (either can be north, south, east, or west).
  - **System Response:**

        - System Validation: Confirms the hero is adjacent to a monster and triggers an interaction sequence.
        - Interaction Sequence Options:

          a. Attack:

            - The hero's battle readiness is verified by the system.
            - If equipped, the system initiates an attack sequence:
                - Hero attacks the monster.
            - Damage is calculated based on:
                - Hero's stats (Example: strength, level, weapon damage).
                - Any equipped gear or enchantments.
                - Random factors (Example: critical hits).
            - The monster retaliates if still alive, or its health is updated.

          b. Evade:

            - If the hero cannot attack:
                - The system displays a warning message, e.g., "A monster is nearby—stay alert!"
            - The player may choose to avoid further interaction with the monster.
    - **Outcome:**
        - If the monster is defeated, it is removed from the grid.
        - If the hero takes damage, the hero's health is updated.
        - If neither the hero nor the monster is defeated:
        - The hero remains in the adjacent square, preparing for further action.

Special Requirements

- **Real-time Feedback**: Movement and interactions must respond within 100ms.
- **User Feedback**: Clear messages for blocked moves or monster alerts.

Technology and Data Variations List

- **Input Devices**: Keyboard (Arrow/WASD), Touchscreen (swipe).
- **Grid Representation**: 2D array or hash map.

Frequency of Occurrence

- **Hero Movement**: 5-10 times/second during gameplay.
- **Monster Interaction**: 5-10% of movements lead to encounters.

Miscellaneous (Open Issues):

None

GoGirls – Sarp Çağan Kelleci - Kemal Serbay Uğur - Emsalinur Arıca - Berke Karamanlı - Muhammet Üçler Çetin - Eray Önder

Use Case 2: Search For Rune

Scope: Go Girls

Level: Subfunction

Primary Actor: Player

Stakeholders and Interests: Player, Developer Team, System

Preconditions

- The hero is adjacent to the object being searched.
- The game timer is active to permit searches.

Main Success Scenario

- Hero's Positioning:
    - The hero moves adjacent to an object in the hall.
- Player Action:
    - The Player makes a click on the object for rune searching.
- System Validation:
    - The system validates that the hero is adjacent to selected object by the player.
- Rune Search Logic:
    - The system checks whether the rune is hidden under the clicked object by player:
    - If the rune is present:
        - The rune is revealed on the grid visually.
        - A sound effect plays to confirm the discovery.
        - The hall state is updated to "completed."
        - The timer for the current hall stops.
        - The exit door to the next hall is unlocked.
    - If the rune is not present:
        - The object remains unchanged.
        - The player can continue searching for other objects in the hall.

Postconditions

- **If the Rune is Found:**
    - The rune is revealed on the grid.
    - The exit door to the next hall is unlocked.
    - The hall is marked as "completed" in the game state.
    - The timer for the current hall is stopped by the system.
- **If the Rune is Not Found:**
    - The search action completes with no result for finding the rune.
    - The player can attempt to search other objects as the rune is not found.
    - In order to maintain the urgency, the timer keeps counting down.

Extensions

- **Invalid Search:**
    - Condition: The player clicks on object which is not adjacent to the hero.
    - System Response:
        - An error message is displayed by the system, such as "You must be adjacent to the object to search it."
        - The search action is not executed by the system.
- **Timer Runs Out:**
    - Condition: The game timer expires before the player finds the rune.
    - System Response:
        - The system displays a “Game Over” message.
        - The game ends, and the player is returned to the main menu or a retry screen.

Special Requirements

- The system must validate player adjacency and manage rune state updates in real time.
- Include sound and visual feedback when the rune is discovered.

Technology and Data Variations List

- Randomized rune placement system to ensure replayability.
- Timer system to enforce time limits for each hall.
- Logic to unlock the hall exit upon successful rune discovery.

Frequency of Occurrence

- Occurs once in each hall as the player searches for the rune.

Miscellaneous (Open Issues)

None

Use Case 3: Collect Enchantment

Scope: Go Girls

Level: Subfunction

Stakeholders and Interests: Player, Developer Team, System

Primary Actor: Player

GoGirls – Sarp Çağan Kelleci - Kemal Serbay Uğur - Emsalinur Arıca - Berke Karamanlı - Muhammet Üçler Çetin - Eray Önder

Preconditions

- An enchantment must have spawned on the grid in a valid, unoccupied location.
- The hero must be adjacent to the enchantment to perform a valid collection.

Main Success Scenario

- Enchantment Spawns:
  - During gameplay, an enchantment will sporadically show up in an empty spot on the grid.
- Player Navigation:
  - Using the arrow keys, the player maneuvers the hero next to the enchantment.
- Collection Action:
  - The player clicks on enchantment to collect it.
- System Validation:
  - The system verifies that the hero is adjacent to the enchantment:
  - If adjacent:
    - The enchantment is collected, and the following actions occur:
    - Immediate Effects (Example extra time, extra life):
      - The system applies the effect instantly.
    - Effects Requiring Later Activation (Example: Cloak, Reveal, Luring Gem):
      - The system adds the enchantment to the hero's bag for future use.
- Gameplay Continued:
  - The player keeps playing while reaping the benefits of the enchantment they have accumulated, which can be used immediately or saved for later.

Postconditions

- If the Enchantment is Collected:
  - The enchantment is removed from the grid.
  - The system updates the game state:
    - For Instant Effects: The effect is applied immediately.
    - For Stored Effects: The enchantment is added to the hero's inventory.
  - If the Enchantment Disappears:
    - The enchantment is removed from the game session.
    - The player loses the opportunity to benefit from the enchantment.

Extensions

- Enchantment Expires:
  - Condition: After 6 seconds from the presence, the enchantment is not gathered.
  - System Response:
    - The enchantment disappears from the grid by the system.
    - The enchantment is no longer available for the player to collect.
  - Unauthorized Collection Attempt:
    - Condition: The hero is not next to enchantment when the Player clicks on it.
    - System Response:

- An error notice (Example: "You must be adjacent to the enchantment to collect it") is displayed by the system.
- The enchantment can still be collected by the hero if the player positions them adjacent to it before the 6-second expiration.

Special Requirements

- Visual feedback mechanism to confirm moves
- Real-time processing system for player input

Technology and Data Variations List

- Grid base movement system for positioning the hero
- Boundary checking algorithm
- Collision detection to preventing overlapping objects

Frequency of Occurrence

- Continuously during gameplay as the player navigates the hall

Miscellaneous (Open Issues):

- None

Use Case 4: Timer Countdown

Scope: Go Girls

Level: User-goal

Primary Actor: Hero (Player)

Stakeholders and Interests:

Player: Wants to locate the rune before the time runs out.
System: Controls the timer to function correctly and be visible for player.

Preconditions:

- The player have logged into the game.

Success Guarantee(Post Condition):

- Player find the rune before time reaches zero and time resets for the next hall.
- If the timer reaches zero before player finds the rune, the game ends.

Main Success Scenerio:

1. Timer countdown starts when player starts the game.
2. The system displays the remaining time on the screen.
3. The player finds the rune before the remaining time reaches 0.
4. When the rune is found:
    4a. System resets the timer and players proceeds with the next hall.
    4b. If not Timer reaches 0 and system displays “Game Over” message.

Extensions:

*a If time runs out:

1. The system ends the game and displays “Game Over” message to user.

*b If an extra time enchantment is used:

1. The system adds 5 seconds to the timer and updates the time counter.

Special Requirements: None

Technology and Data Variations List:

*Time Display Area on Screen

Frequency of occurence:

- Time counter runs continuously during gameplay. It resets at the beginning of each hall.

Misceooaheous(Open Issues):
None

<u>**Use Case 5:** Unlock Door</u>

Scope: Go Girls

Level: User-goal

Primary Actor: Hero (Player)

Stakeholders and Interests:
Player: Wants to unlock doors by finding the rune to continue with the next hall.

GoGirls – Sarp Çağan Kelleci - Kemal Serbay Uğur - Emsalinur Arıca - Berke Karamanlı - Muhammet Üçler Çetin - Eray Önder

System: Ensures the correct functionality of unlocking doors after the rune is found.

Preconditions:

- The player must have logged into the game.
- The hidden rune must have found by the player.
- The door must exist and be interactable.

Success Guarantee (Post Condition):

- If the player succeeds, the timer resets or updates for the next hall.
- If the timer reaches zero, the game ends.

Main Success Scenerio:

1 -The player searches for the rune.

2 -Once the rune is found, the system activates the door interaction.The player interacts with objects to search for the rune.

3- Player goes to the door.

4- The hero interacts with the door.

5- System checks whether the rune is found.

5.a If found the door is opened.

5.b If not the interaction is blocked.

6- Door opens and the hero enters the next hall

Extensions:

*a If hero couldn't finds the rune:

1-The system continues to locks the door.

2-Player continues to search for the rune.

Special Requirements:

- Once the door is unlocked, a distinct sound effect is made by the system.

Technology and Data Variations List:

*2D Door object on the screen.

Frequency of occurence:

- Occurs in every hall, when the player finds the rune.

Misceooaheous(Open Issues):

None

GoGirls – Sarp Çağan Kelleci - Kemal Serbay Uğur - Emsalinur Arıca - Berke Karamanlı - Muhammet Üçler Çetin - Eray Önder

Use Case 6: Use Enchantment

Scope: Go Girls

Level: User-goal

Primary Actor: Hero (Player)

Stakeholders and Interests: Player, Developer Team, System

Player: Wants to use enchantments to gain advantages in the game.
System: Ensures that selected enchantment applies and functions correctly.

Preconditions:

- The player must have logged into the game.
- The player must have collected the specific enchantment.
- Enchantment must be available in player's inventory.

Success Guarantee (Post Condition):

- The chosen enchantment is applied.
- The effect of the enchantment is applied. (e.g., protection from monsters, revealing the rune's location, or luring a monster).

Main Success Scenario:

1 -The player opens inventory to view available enchantments.
2 -Player selects a enchantment to use by pressing specific key on keyboard.
2a- Reveal: Presses R button.
2b- Cloak of Protection: Presses P button.
2c- Luring gem: Player clicks the B button and then one of A , D W or S to throw desired way.
3- System checks whether the chosen enchantment is available on the person.
3a- If enchantment is available, system activates the enchantment.
3b- If not system displays an error message.
4- The effect of the enchantment is applied:
4a- Reveal: Specifies a 4x4 location where the rune can be found.
4b- Cloak of Protection: Hides the hero from archer monster for 20 seconds.
4c- Luring Gem: Fools the fighter monster.

4d- Extra life: Increases the hero's life by one.
4e- Extra time: 5 seconds are added to hero's timer.
5- System updates the inventory by decreasing the used enchantment count by one.

Extensions:
*a If player selects and enchantment that player doesn't have :
1- System outputs an error message.
*b If player collects Extra Life or Extra time enchantment:
1- The effect of these enchantments are applied instantly without player's need to press a key.

Special Requirements:

- System should display the effect of enchantment by UI.

Technology and Data Variations List:
*Enchantment activation are done by their desired keys on keyboard
*Inventory : System should update the inventory correctly according to usage.

Frequency of occurrence:

- Whenever the player collects enchantments and uses them based on their strategy.

Misceooaheous(Open Issues):
None

Use Case 7: Interact with Monster

Scope: Go Girls

Level: User-goal

Primary Actor: Hero (Player)

Stakeholders and Interests: Player, Developer Team, System

Player: Wants to survive incoming attacks from monster.
System: Ensures that monster actions act correctly.

Preconditions:

- The player must have logged into the game.
- The player and the monster should be on the same hall
- Distance between monster and player should be on the ciritic range. (Eg. Archer can attack on a 4x4 distance away)

Success Guarantee (Post Condition):

- The player protects himself/herself from the monster attacks.
- If the player takes damage player's health point decrease by one or the special enchantment is loosen

Main Success Scenario:

1- The player encounters a monster in the hall.

2- The system identifies the monster and displays its abilities.

- **Archer**: Shoots an arrow if the player is within a 4x4 range.
- **Fighter**: Stabs the player when they are nearby.
- **Wizard**: Randomly relocates the lure.

3- The player takes action:

- **3a. Escape**: The player moves to avoid the monster's attack range.
- **3b. Counter or Evade**: The player uses an enchantment (e.g., a *Cloak of Protection* or a *Luring Gem*) to neutralize or evade the monster's attack.

4- System Updates:

- **Health Tracking**: If the player takes damage, their health is reduced accordingly.
- **Enchantment Management**: Any enchantments used are consumed or applied.

5- Encounter Resolution: The encounter concludes if any of the following occur:

- The player successfully escapes the monster's range.
- The monster is neutralized using an enchantment.
- The player advances to the next hall.

Extensions:

1. **Failure to Respond**:
   If the player does not move or use an enchantment in time, the monster successfully attacks, reducing the player's health by 1 point.

2. **Health Depletion**:
   The game ends if the player's health reaches zero, and a "Game Over" screen is displayed.
3. **Multiple Monsters**:
   - Each monster operates independently based on its abilities.
   - The player must strategically prioritize threats to evade or counter them.
4. **Successful Enchantment Usage**:
   - **Cloak of Protection**: Temporarily makes the player invisible to Archer monsters.
   - **Luring Gem**: Distracts Fighter monsters, causing them to move toward the gem's location, giving the player a chance to escape.

Special Requirements

1. **Real-Time Systems**:
   - Track the positions of the player and monsters accurately.
   - Monsters react immediately based on proximity.
2. **Seamless Enchantment Integration**:
   - Enchantments like invisibility or distractions must be applied dynamically during gameplay.

Technology and Data Variations

- **Proximity Sensors**:
  Used to measure the distance between the player and monsters.
- **Enchantment Management**:
  Systems for activating effects, tracking duration, and managing consumption

Frequency of Occurrence

- Encounters are triggered if a monster spawns in the player's hall.

Miscellaneous (Open Issues)

- None at this time.