

## **Use Case 1: Wizard Monster's Responsive Behavior**

Use Case: Wizard Monster Adapts Strategy Based on time

Primary Actor: Wizard Monster

Stakeholders and Interests:

Game System: Needs to maintain game balance

Wizard Monster: Needs to execute appropriate strategy

Player: Tries to win the game.

Precondition:

- Wizard Monster is on the game

Postconditions:

Wizard Monster executes the respective strategy based on time

Main Success Scenario:

- System spawns Wizard Monster at a random location

- System calculates percentage of remaining time and Wizard Monster evaluates game situation based on time percentage:

  - If remaining time < 30%:

    - System identifies random empty location and teleports the player

    - Wizard Monster disappears

  - If time remaining > 70%:

    - System moves rune every 3 seconds

    - System checks if time percentage is still > 70%

  - If time remaining between 30-70%:

    - Wizard Monster remains stationary for 2 seconds and disappears

Special Requirement:

Strategy patterns must be implemented for time percentages.

## **Use Case 2: Save and Load Game State**

Use Case: Save and Load Game Progress

Primary Actor: Player

Stakeholders and Interests:

Player: Wants to preserve game progress and resume later

Game System: Must keep the data of the gamer

Preconditions:

Game is in progress and player is able to save/load the game

Postconditions:

Game state is completely saved or loaded

All elements of the game are in the same condition before saving the game after loading the game

Main Success Scenario:

Player starts save game functionality

System saves game state such as object locations, enchantments, position of hero and lives

System generates new save file and game state is saved into the file

System confirms the save to the player

Special Requirements:

Loaded game must be same with saved game

Save and load operations must pause the game

### **Use Case 3: Change Rune Location**

Use Case: Change Location of the Rune

Primary Actor: Wizard Monster

Stakeholders and Interests:

Player: Must reach to new location of the rune

System: Game rules must be same

Wizard Monster: Must change location of the rune

Preconditions:

Game is in progress

Rune exists on the game

Wizard Monster is active with > 70% time remaining

Valid empty spaces exist on the board

Postconditions:

Rune is relocated to a new position

Game rules are not broken

Player can access the new rune location

Main Success Scenario:

System identifies all valid empty locations

System selects new location from valid positions

System removes rune from current location and places rune at new location

System updates game state with new rune position

System waits 3 seconds before next relocation

#### Extensions:

##### 1) No valid empty locations available:

System maintains rune in current position

System waits for next relocation

##### 2) Time percentage drops below 70%:

System completes current relocation

System terminates relocation cycle

#### Special Requirements:

Rune must always be accessible to player

Rune Relocation must occur every 3 seconds

Relocation must not break other game mechanics

System must validate new location of the rune