Use case name: Wizard Monster Behavior

Scope: Phase-2

Level: User goal

Primary Actor: Wizard Monster

Stakeholders and Interests:

Player: Has a more dynamic and challenging gameplay with the implementation of the new wizard mechanics.

System: Implements the behavior changes to keep the game interesting.

Preconditions:

The game is not paused

The game is in play state

Player still has hearts.

Postconditions:

The wizard monster's behavior is executed based on the percentage of remaining time, either the player is telepoted once or the rune will change location every 3 second after the wizard appears.

Main Success Scenario:

1. Wizard monster spawns in the hall.
2. The system checks the remaining time percentage.
3. If the remaining time is:

   Less than 30%: The player is teleported to a random empty location once and then the wizard disappears.

   More than 70%: The rune is teleported to a new location every 3 seconds.

   Between 30%-70%: The wizard stays in place for 2 seconds and then disappears.

4. The wizard's behavior dynamically updates based on changes in the remaining time during its lifespan.

Extensions:

1a. Remaining time percentage changes and becomes a different percentage interval that has different responsibilities for the wizard monster while the wizard monster is still present:

   1. Wizard monster re-evaluates its behavior and completes its responsibility for the requiered new percentage interval.

Frequency of Occurrence:

Almost continuously throughout the play state.

Use Case Name: Save Game

Scope: Phase-2

Level: User goal

Primary Actor: Player

Stakeholders and Interests:

Player: Wants the current game state to be stored accurately for future game sessions.

System: Puts the game state into a file that can be retrieved later.

Preconditions:

The game is paused

The game is in play state

The player clicks the "Save" button.

Postconditions:

The full game state is saved to a file.

The saved files appears in the “Load” part after clicking the load button in the main menu screen.

Main Success Scenario:

1. The player clicks the "Save" button in the pause screen.
2. The system collects the current state of the game, including:

Game Time:

MonsterFactory: Current stop time, last creation time, and passed time since last action.

EnchantmentFactory: Current stop time, last creation time, and passed time since last action.

WizardMonster: Current stop time, last teleportation time, and passed time since last teleportation.

Hall Stats:

Current hall being played.

Objects on Map:

Type, count, and position of objects, including the rune.

Hero Stats:

Hero's position, time remaining (a single countdown timer), and number of lives remaining.

Hero Bag:

Collected enchantments (type, quantity, and slot position).

Monster Stats:

Type, count, and position of all monsters currently in the game.

3. The system serializes this data into a file (e.g., JSON, XML, or binary).
4. The saved file is added to the list of available saves that can be found after clicking the load button in the main menu screen.

Extensions:

1a. Save operation fails:

1. The system displays an error message and provides an option to retry.

1b. Player creates multiple save files:

1. Each file is uniquely named and that can be found after clicking the load button in the main menu screen.

Use Case Name: Load Game

Scope: Phase-2

Level: User goal

Primary Actor: Player

Stakeholders and Interests:

Player: Resumes gameplay from an earlier point in time without any problems.

System: Restores the saved state in a correct and complete way.

Preconditions:

At least one saved file exists.

The player selects the "Load" button from the main menu.

Postconditions:

The game state is restored to match the saved file.

The game starts from the play state instead of the build mode like it normally does.

Main Success Scenario:

1. The player clicks the "Load" button from the main menu sceen.
2. The system displays a list of saved files.
3. The player selects a saved file.
4. The system reads the file and restores the saved state, including:

Game Time:

MonsterFactory: Current stop time, last creation time, and passed time since last action.

EnchantmentFactory: Current stop time, last creation time, and passed time since last action.

WizardMonster: Current stop time, last teleportation time, and passed time since last teleportation.

Hall Stats:

Current hall being played.

Objects on Map:

Type, count, and position of objects, including the rune.

Hero Stats:

Hero's position, time remaining (a single countdown timer), and number of lives remaining.

Hero Bag:

Collected enchantments (type, quantity, and slot position).

Monster Stats:

Type, count, and position of all monsters currently in the game.

5. The system resumes gameplay from the restored state.

Extensions:

1a. If no saved files exist:

1. The player cannot press to any of the save file slots so they are left with only the new game button to press.

1b. The selected file is corrupted:

1. The system displays an error message and urges the player to select another file.