Use case: playing the game

Primary actor: player

Goal in context: the player navigates through the game controlling the main character, avoiding enemies, and collecting rewards at the same time. They seek to complete the game with the highest score possible.

Preconditions: computer system and game executable must both be configured correctly

Trigger: player decides to run the game executable and play

Scenario:

1. Player runs the game executable
2. Player views the menu screen
3. Player selects the "Play Game" button
4. Player character spawns and the player can take control of the main character
5. Player attempts to collect all required rewards while navigating the maze and avoiding enemies
6. Player moves to exit and wins the game

Exceptions:

1. Player decides to prematurely exit the game – see use case: "exiting the game"
2. Player decides to configure their game options – see use case: "configuring options"
3. The program encounters an error and crashes – see use case: "game error/crash"
4. The player character collides with a moving enemy – see use case: "collides with enemy"
5. The player character collides with a punishment – see use case: "collides with punishment"

Priority: high priority; all other tasks centered around the implementation of the game

==

Use case: game error/crash

Primary actor: game developer

Secondary actor: player

Goal in context: to accurately report crash diagnostics back to the developer after the game crashes

Precondition: player must be currently playing the game (the executable is launched)

Trigger: either a player action or system error causes the program to crash

Scenario:

1. Player interaction with the program or system error causes program to crash

2. Error is logged and read by the developers

Exceptions: N/A

Priority: Medium; being able to log and fix errors will help with program development

Use case: exiting the game

Primary actor: player

Goal in context: to successfully stop execution of the game

Preconditions: player must be currently playing the game (the executable is launched)

Trigger: player decides to stop playing the game

Scenario:

1. If player is on the starting menu:
    a. Player selects the “Exit” button
    b. The game successfully terminates, and the player is out of the game
2. If the player is past the starting menu and is in the game:
    a. Player presses the “Esc” key on their keyboard
    b. A menu similar in appearance to the start menu is brought up
    c. Player selects the “Exit” button
    d. The game successfully terminates, and the player is out of the game

Exceptions:

1. The program encounters an error and crashes – see use case: “game error/crash”

Priority: low; there are alternative ways to exit the application

==

Use case: configuring options

Primary actor: player

Goal in context: to configure the settings of the game

Preconditions: the player must be currently playing the game (the executable is launched)

Trigger: player wants to change the settings of the game

Scenario:

1. If player is on the starting menu:
    a. Player selects the “Options” button
    b. Player configures the game’s settings to their liking
2. If the player is past the starting menu and is in game:
    a. Player presses the “Esc” key on their keyboard
    b. A menu similar in appearance to the start menu is brought up
    c. Player selects “Options” button
    d. Player configures the game’s settings to their liking

Exceptions: The program encounters an error and crashes – see use case: “game error/crash”

Priority: low; primary function of the game should be prioritized

Use case: collides with enemy

Primary actor: player

Goal in context: subtracts score and teleports the player or ends the game

Preconditions: player must be currently playing the game (the executable is launched) and be on the game board

Trigger: Player character collides with a moving enemy

Scenario:

1. Player controls the movement of the player character
2. Player character moves into a cell occupied by a moving enemy
    a. Alternatively, a moving enemy enters a cell occupied by the player character
3. The game ends and the player's final score is shown

Exceptions:

1. The program encounters an error and crashes – see use case: "game error/crash"

Priority: high; it is a core functionality of the game

==

Use case: collides with punishment

Primary actor: player

Goal in context: subtracts score from the player; ends game if score drops below zero

Preconditions: player must be currently playing the game (the executable is launched) and be on the game board

Trigger: Player character collides with a punishment

Scenario:

1. Player controls movement of the player character
2. Player character moves into a cell occupied by a punishment
3. The punishment value is subtracted from the player's score
    a. If the player's score remains above zero
        i. Player character is removed from the board and is teleported to the starting cell
    b. If the player's score drops below zero
        i. The game immediately ends

Exceptions:

1. The program encounters an error and crashes – see use case: "game error/crash"

Priority: High; it is a core functionality of the game