Use Cases of the “Emoji Maze”

Use case #1: Completion of the game by gathering all the regular rewards and leaving the map through the exit
Primary actor: Player
Goal in context: To complete the game by collecting the required regular rewards
Preconditions: Game must be completable; access to exit and all regular rewards are required
Trigger: The player successfully attempts to finish the game by getting all the regular rewards

Scenario:

1. The player runs the game’s desktop application.
2. The game displays a start screen displaying the title and an “Enter” button.
3. The player navigates their mouse to the “Enter” button and clicks it to start the game.
4. The player spawns on the starting cell of the maze.
5. The player controls their sprite with the arrow or ‘WASD’ keys to collect all the regular rewards while not encountering any of the moving enemies.
6. The game displays a positive score that increases with each reward gathered.
7. The player moves back to the starting point (which functions as the exit to the maze) after collecting all the regular rewards.
8. The game displays an end screen saying the player passed the game.

Exceptions:

1. The player encounters an enemy - see use case **Failure of the game due to enemy run-in**
2. The player obtains enough punishments to accumulate a negative score - see use case **Failure of the game due to negative score**
3. The player does not collect all regular rewards - see use case **Warning requiring the remaining rewards before finishing**

Priority: High priority, the player should be able to complete the game

Open issues:

1. How difficult is it to complete the game?
2. Is it possible for the game to render itself unwinnable? How can that scenario be prevented?
3. How many points should a regular reward give?

Use case #2: Completion of the game with a higher score by collecting any number of bonus rewards
Primary actor: Player
Goal in context: To complete the game by collecting the required regular rewards as well as some bonus rewards
Preconditions: Game must be completable; access to exit, regular rewards, and bonus rewards are required. Game must also be properly loaded into a ready-to-play state (after steps one to four in use case #1).
Trigger: The player decides to finish the game as well as achieving a higher score.

Scenario:

- The player controls their character with the arrow or 'WASD' keys to collect all the regular rewards and as many bonus rewards as possible while not encountering any of the moving enemies.
- The game displays a positive score that increases with each regular and bonus reward gathered.
- The player goes back to the starting point after collecting all the regular rewards.
- Game displays an end screen saying the player passed the game.

Exceptions:

1. The player encounters an enemy - see use case **Failure of the game due to enemy run-in**
2. The player obtains enough punishments to accumulate a negative score - see use case **Failure of the game due to negative score**
3. The player does not collect all regular rewards - see use case **Warning requiring the remaining rewards before finishing**

Priority: High priority, the player should be able to complete the game and collect some bonus rewards to increase their score

Open issues:

1. How difficult is it to collect the bonus rewards?
2. How many points should the bonus rewards be worth? Should they be higher or lower than the point loss from a punishment?

Use case #3: Failure of the game due to enemy run-in by player
Primary actor: Player
Goal in context: To fail the game by colliding with an enemy
Preconditions: The game must allow for enemies to move towards the player and the game is properly loaded into a ready-to-play state (after steps one to four in use case #1).
Trigger: Not applicable unless the player purposely wishes to end the game by running into an enemy. This use case is typically not intentional behaviour.

Scenario:

1. The player controls their character with the arrow or 'WASD' keys to try to collect either regular or bonus rewards.
2. The player bumps into a moving enemy.
3. The game displays an end screen saying the player failed.

Exceptions:

1. The player obtains enough punishments to accumulate a negative score - see use case **Failure of the game due to negative score**

Priority: High priority, the player should face consequences for colliding with an enemy

Open issues:

1. Does the enemy have any path-finding algorithm to navigate around corners and walls towards the players?
2. How many and how fast should these enemies be?

Use case #4: Failure of the game due to negative score
Primary actor: Player
Goal in context: To fail the game by obtaining a negative score
Preconditions: The game must allow for punishments to be picked up and the game is properly loaded into a ready-to-play state (after steps one to four in use case #1).
Trigger: Not applicable unless the player purposely wishes to end the game by picking up punishments. This use case is typically not intentional behaviour.

Scenario:

1. The player controls their character with the arrow or 'WASD' keys to try to collect either regular or bonus rewards.
2. The player steps into a punishment cell.
3. The game decreases their score.
4. The player continues to enter into punishment cells multiple times.
5. Their score keeps decreasing with every punishment cell entered until it reaches below zero.
6. The game displays an end screen saying the player failed.

Exceptions:

1. The player encounters an enemy - see use case **Failure of the game due to enemy run-in**

Priority: High priority, the player should face consequences for continually picking up punishments.

Open issues:

1. How many punishment cells should spawn at the start?
2. Should they be allowed to spawn in 1-cell wide hallways on the way to the regular rewards? If so, how many are allowed to spawn in those areas?
3. How many points should they decrease?

Use case #5: Warning displayed requiring the remaining rewards before exiting the map
Primary actor: Player
Goal in context: To exit the maze without all rewards and triggers a warning
Preconditions: The game must allow for all regular rewards to potentially be picked up and the game is properly loaded into a ready-to-play state (after steps one to four in use case #1).
Trigger: Not applicable unless the player purposely wishes for this warning to show. This use case is typically not intentional behaviour.

Scenario:

1. The player controls their character with the arrow or 'WASD' keys to try to collect either regular or bonus rewards.
2. Before collecting less than the number of required regular rewards, the player tries to exit the maze.
3. The game displays a warning (that is able to be exited) saying the player still needs to gather the rest of the regular rewards.
4. After reading the message, the player exits the warning and continues playing the game.

Exceptions:

1. The player encounters an enemy - see use case **Failure of the game due to enemy run-in**

Priority: Low priority, this warning can be implemented as the colour of the entrance/exit. The entrance would change from orange to blue once all the regular rewards are obtained.

Open issues:

1. Should the warning time out on its own without the player needing to exit it?