Phase 1: Use Cases

CMPT 276 Group 8: Darren Jennedy, Samuel Wong, Setu Patel, Steve Lam

Use Case 1:

- Use Case:
  - The player collects a reward (regular or bonus) from a cell
- Iteration:
  - First Iteration
- Primary Actor:
  - Main character (user)
- Goal in context:
  - To create an effect for collecting rewards.
- Preconditions:
  - The player must be alive
  - The game has started, and the rewards are correctly placed (not overlapping with walls or barriers, or unanimated punishments).
- Trigger:
  - Player moves onto a cell that contains a reward (A+ bonus rewards or the P grade regular reward)
- Scenario:
  1) Main character is moving across the map using WASD or the arrow keys
  2) Moves onto a cell that contains a reward (regular or bonus)
  3) Points get added to the player score depending on which reward it is, 2 points if it is a regular reward, and 5 points if it is a bonus reward.
  4) The reward disappears.
- Exceptions:
  - If there is a reward and a moving enemy in the same cell and the player moves to said cell, see **use case 3**.
- Priority:
  - Medium priority, need to implement the board, main character, character movement, rewards first.
- When available:
  - Third Increment
- Frequency of use:
  - High Frequency
- Channel to actor:
  - WASD and Arrow Keys on the user’s keyboard (key listeners)
- Secondary actors:
  - Regular Rewards (P Grade)
  - Bonus Rewards (A+ Grade)
- Channels to secondary actors:
  - Secondary actors are controlled automatically on the desktop application
- Open issues:
  - N/A

<u>Use Case 2:</u>

- Use Case:
  - Running into a punishment (F grades)
- Iteration:
  - First Iteration
- Primary Actor:
  - Main character (user)
- Goal in context:
  - To create a consequence for moving into a cell that contains a punishment.
- Preconditions:
  - The player must be alive
  - The game has started, and the punishments are correctly placed (not overlapping with walls, barriers, or rewards).
- Trigger:
  - Player is on the same cell as the enemy
- Scenario:
  1) Main character is moving across the map using WASD
  2) Player moves onto a cell the contains a punishment
  3) 5 points are deducted from the player's score.
     a) After the point deduction, if the score becomes less than 0, the player loses, and the game goes to the ending screen.
     b) After the point deduction, if the score is greater or equal than 0, the game continues.
- Exceptions:
  - N/A
- Priority:
  - Medium priority, need to first implement board, main character, enemies, character movement.
- When available:
  - Third Increment
- Frequency of Use:
  - Low Frequency
- Channel to actor:
  - WASD and Arrow Keys on the user's keyboard (key listeners)
- Secondary actors:
  - Punishments (F grades)
- Channels to secondary actors:
  - Secondary actors are controlled automatically on the desktop application
- Open issues:
  - N/A

Use Case 3:

- Use Case:
  - Running into a moving enemy (“Evil” Professors)
- Iteration:
  - First Iteration
- Primary Actor:
  - Main character (user)
- Goal in context:
  - To create a consequence for moving into a cell that contains a moving enemy.
- Preconditions:
  - The player must be alive
- Trigger:
  - The player is on the same cell as the moving enemy
- Scenario:
  1) Main character is moving across the map using WASD
  2) Player moves onto a cell which contains an enemy (or moving enemy moves onto the player cell.
  3) The game ends and the player loses.
  4) The game goes to the ending screen.
- Exceptions:
  - If there is a reward and a moving enemy on the same cell and the player moves to said cell, the player will initiate contact with the enemy instead of the reward and the game will be over. The player does not get the points from said reward.
- Priority:
  - Medium priority, need to first implement board, main character, enemies, character movement
- When available:
  - Third Increment
- Frequency of Use:
  - Medium-high frequency
- Channel to actor:
  - WASD and Arrow Keys on the user’s keyboard (key listeners)
- Secondary actors:
  - Moving enemies
- Channels to secondary actors:
  - Secondary actors are controlled automatically on the desktop application
- Open issues:
  - N/A

Use Case 4:

- Use Case:
  - After collecting all the required regular rewards, the player walks through the exit cell and wins the game.
- Iteration:
  - First Iteration
- Primary actor:
  - Main Character (user)
- Goal in context:
  - To create the winning condition for the game.
- Preconditions:
  - The player must be alive.
  - The player must have collected all 40 of the regular rewards (P grades) in order to unlock the exit cell on the boundary wall opposite the starting cell.
- Trigger:
  - Player moves onto the exit cell after unlocking it
- Scenario:
  1) The player collects all 40 of the regular rewards (P grades)
  2) The exit is unlocked
  3) The player walks into the exit cell
  4) The game ends with the player winning, the winning screen pops up displaying the total score and time elapsed.
- Exceptions:
  - If the exit cell is not unlocked because the player has not collected all 40 regular rewards, the exit cell will be treated like a regular wall, refer to **Use Case 7**
- Priority:
  - Low priority, need to first implement board, characters, character movement, rewards, punishments, score system
- When available:
  - Third Increment
- Frequency of use:
  - Only happens at end of game (one time), infrequent
- Channel to actor:
  - WASD and Arrow Keys on the user's keyboard (key listeners)
- Secondary actor:
  - Exit Cell
  - Regular Rewards (P grades)
- Channels to secondary actors:
  - Secondary actors are controlled automatically on the desktop application
- Open Issues:
  - N/A

Use Case 5:

- Use Case:
  - Player loses the game
- Iteration:
  - First Iteration
- Primary actor:
  - Main Character (user)
- Goal in Context:
  - The player is in a state which would result in the user losing the game
- Preconditions:
  - Either:
    - Player has a negative score
    - Player is in contact with the moving enemy
- Trigger:
  - Either:
    - Collecting an F punishment that results in a negative score
    - Running into an enemy
- Scenario:

  A) The player makes contact with a moving enemy

    1) The player moves across the map using WASD or the arrow keys.
    2) The player moves onto a cell that contains a moving enemy.
    3) The game is lost and the game goes to the losing screen.

  B) The player's points become negative

    1) The player moves across the map using WASD or the arrow keys.
    2) The player moves onto a cell that contains a punishment.
    3) 5 points are deducted from the player's score
    4) After the point deduction, if the player has less than 0 points, the game is lost and the game goes to the losing screen.

  - Once the game is lost, a losing screen will appear that says you've lost which displays the score and time as well as a main menu and play again button.
- Exceptions:
  - N/A
- Priority:
  - Low priority, need to first implement board, characters, enemies, character movement, score system.
- When available:
  - Third Increment
- Frequency of Use:
  - Medium frequency, depends how difficult the game is for the player.
- Channel to actor:
  - WASD and Arrow Keys on the user's keyboard (key listeners)
- Secondary actors:
  - Moving enemies ("Evil" Professors)
  - Punishments (F grades)
- Channels to Secondary Actors:
  - Secondary actors are controlled automatically on the desktop application
- Open Issues:
  - N/A

Use Case 6:

- Use Case:
  - Player moves into an empty cell
- Iteration:
  - First Iteration
- Primary actor:
  - Main Character (user)
- Goal in Context:
  - To define general movement of the player
- Preconditions:
  - Player is alive
  - The cell the player is moving towards is not occupied by a barrier
- Trigger:
  - Pressing the WASD key
- Scenario:
  1) The user presses one of the movement keys (W,A,S,D, or arrow keys)
     a) If the player presses the W or the up arrow key, the player moves upwards and occupies the empty cell located upwards of the original position of the player.
     b) If the player presses the A or the left arrow key, the player moves leftwards and occupies the empty cell located leftwards of the original position of the player.
     c) If the player presses the S key or the down arrow key, the player moves downwards and occupies the empty cell located downwards of the original position of the player.
     d) If the player presses the D key or the right arrow key, the player moves rightwards and occupies the empty cell located rightwards of the original position of the player.
- Exceptions:
  - If the player moves to a cell that contains a reward (regular reward or bonus rewards); refer to **Use Case 1**
  - If the player moves to a cell that contains a punishment; refer to **Use Case 2**
  - If the player moves to a cell that contains a moving enemy, refer to **Use Case 3**
  - If the player moves to the ending point after collecting all 40 regular rewards (P grades), refer to **Use Case 4**
  - If the player moves to a wall or barrier, refer to **Use Case 7**
- Priority:
  - High priority, need to first implement the board.
- When available:
  - Second Increment
- Frequency of Use:
  - High frequency
- Channel to actor:
  - WASD and Arrow Keys on the user's keyboard (key listeners)
- Secondary actors:
  - Empty cells of the board.
- Channels to Secondary Actors:
  - Secondary actors are controlled automatically on the desktop application
- Open Issues:
  - N/A

Use Case 7:

- Use Case:
  - Player moves into a barrier (walls or construction zones)
- Iteration:
  - First Iteration
- Primary actor:
  - Main Character (user)
- Goal in Context:
  - To define what would happen if the player walks onto a barrier.
- Preconditions:
  - Player is alive
- Trigger:
  - Player tries to move onto a barrier cell
- Scenario:
  1) Player is moving using WASD or the arrow keys
  2) Player is on a cell next to a barrier
  3) Player attempts to move in the direction of the barrier
  4) Player is not able to move in that direction, stays on same cell
- Exceptions:
  - If the player moves to the ending point after collecting all 40 regular rewards (P grades), refer to **Use Case 4**
- Priority:
  - Medium-high priority, need to first implement board and movable character.
- When available:
  - Second Increment
- Frequency of Use:
  - Medium-high frequency
- Channel to actor:
  - WASD and Arrow Keys on the user's keyboard (key listeners)
- Secondary actors:
  - Barriers (walls or construction zones)
- Channels to Secondary Actors:
  - Secondary actors are controlled automatically on the desktop application
- Open Issues:
  - N/A