Ekansh Chawla
Felmer Macadangdang
Nicholas Lagasse
Sundus Subuktegin

CMPT 276 PROJECT PHASE 1: USE CASES

Use cases (different scenarios in which user interacts with the game)

- User Interface
    - **Primary Actor:** User
    - **Goal in context:** Launch the game window to display game and information, and begin running game logic and taking user input
    - **Preconditions:** All code must be written
    - **Trigger:** Main.java is launched
    - **Scenario:**
        I. User launches main.java
        II. User chooses difficulty from start menu
        III. Game window launches depending on difficulty
        IV. Numerical data such as current points, current game time, number of enemies on the board, if the player has the special item, as well as game board itself is displayed in window
        V. Player either wins or loses game
        VI. End screen is shown, including time, score, win/lose status, and end game button
        VII. End game button closes game and window

    - **Exceptions:**
        I. Game window didn't launch properly
        II. User never selects game difficulty
        III. User never selects quit game button
    - **Priority:** High
    - **When available:** After launch of Main.java
    - **Frequency of use:** Once a game
    - **Channel to Actor:** Keyboard
    - **Secondary Actors:**
        I. Mouse
    - **Open Issues**
        - Resize game window
        - Close game window at any time
        - Launch game on different computers

- Start Game
    - **Primary Actor:** Player
    - **Goal in context:** Initialize game to begin state
    - **Preconditions:** Game window launched successfully
    - **Trigger:** Select game difficulty from start menu
    - **Scenario:**
        I. User selects difficulty
        II. All enemies, items, objects, and scoring are spawned in properly
        III. Player spawns on start square
        IV. Player plays the game
    - **Exceptions:**
        I. Player times out game
        II. Enemies not spawned properly
        III. Items not spawned properly
        IV. Objects not spawned properly
        V. Game board generated is impossible to beat
    - **Priority:** High
    - **When available:** After Selecting game difficulty
    - **Frequency of use:** Once a game
    - **Channel to Actor:** Game window
    - **Secondary Actors:**
        I. Numerical data
        II. Game board
    - **Open Issues**
        I. Get user input properly

- Player Movement
    - **Primary Actor:** Player
    - **Goal in context:** To allow the player to move their character on the game board, interacting with the map.
    - **Preconditions:** Game started, the player is alive, the player is on the board.
    - **Trigger:** The player initiates movement by pressing a directional input.
    - **Scenario:**
        i. Player observes game board and decided on movement
        ii. Player presses directional control reflecting desired movement
        iii. System detects input and checks if direction chosen is valid
        iv. System updates player position
        v. Game checks for collisions with enemies or powerups.

vi. Game state reflects changes depending on the state of the target cell (Health Loss, Powerup Gained)

- **Exceptions:**
  i. Barrier in target cell: System cancels Movement
  ii. Hole in target cell: tbd
- **Priority:** Essential, main game mechanic must be implemented
- **When available:** First increment
- **Frequency of use:** Used throughout gameplay
- **Channel to Actor:** Keyboard input
- **Secondary Actors:**
  i. Game system
- **Open Issues**
  ~~i. Diagonal Movement~~
  ~~ii. Hole Powerup Movement~~

- Get Mandatory Item
  - **Primary Actor:** Player
  - **Goal in context:** The player must collect all mandatory objects as they move through the board in order to win the game.
  - **Preconditions:** Game started, the player is alive and moving through the board
  - **Trigger:** The player moves onto a square with a regular item on it
  - **Scenario:**
    i. The player collects the item on the square
    ii. The item is removed from the game and added to the player's inventory
    iii. The reward amount of the item is added to the player's score
    iv. The player's score is updated and reflects the reward picked up
  - **Exceptions:**
    i. Impossible to reach item -> barriers block player from reaching it
    ii. Item spawns on the same block as a barrier
  - **Priority:** High
  - **When available:** from first increment, all mandatory items should be on the board
  - **Frequency of use:** 3 mandatory items in each level
  - **Channel to Actor:** Keyboard input
  - **Secondary Actors:**
    i. Game system
  - **Open Issues**

        i. Player reaches the end but does not pick up one or all mandatory items

- Get Optional Item
    - **Primary Actor:** Player
    - **Goal in context:** player can collect an optional item to boost their score and be able to jump over a hole
    - **Preconditions:** Game started, the player is alive and moving through the board
    - **Trigger:** The player moves onto a square with an optional item on it
    - **Scenario:**
        i. Player collects the optional item on the square
        ii. The item is removed from the game and added to the player's inventory
        iii. The reward amount of the item is added to the player's score
        iv. The player's score is updated and reflects the reward picked up
        v. While the player has the bonus item in their inventory, they can "jump" over a hole - They'll be able to cross a hole with no punishment
        vi. Once they jump over a hole, the item will disappear from their inventory and they will no longer have this special ability
        vii. After the optional item disappears from the player's inventory, it will be allowed to respawn
    - **Exceptions:**
        i. Impossible to reach item -> barriers block player from reaching it
        ii. Item spawns on the same block as a barrier
        iii. Player does not pick up item before it despawns
    - **Priority:** medium
    - **When available:** spawns randomly during the game and despawns within a certain time if not picked up
    - **Frequency of use:** only 1 optional item spawns at a time
    - **Channel to Actor:** keyboard input
    - **Secondary Actors:**
        i. Game system
    - **Open Issues**
        i. Ensure that having the special item allows player to move through holes unharmed

- Player Death
    - **Primary Actor:** Player

- **Goal in context:** Main character death by falling in hole (stationary enemy) or by moving enemy attacks -> lose game
- **Preconditions:**
    i. Player is playing the game
    ii. Health reached 0
- **Trigger:** Health depletes to 0
- **Scenario:**
    i. Encounters hole enemy (health reaches 0) or multiple enemy attacks that lead to health reaching 0
    ii. System checks for player death condition
    iii. Trigger death animation, ragdoll falling and screen change
    iv. New UI display to change to initial game menu (essentially respawn)
    v. Restore player, restart at default predetermined state (health, position)
    vi. Resume game
- **Exceptions:**
    i. Game crash (player not spawned correctly or game not ending properly)
    ii. TBA
- **Priority:** Essential (follows game flow)
- **When available:** Throughout game
- **Frequency of use:** once per game
- **Channel to Actor:** Visual and audio feedback (screen fade, ragdoll animation)
- **Secondary Actors:**
    i. Game engine, health system, physics system, UI and sounds system
- **Open Issues**
    i. Ensuring spawn with predetermined health and spawn position.
    ii. Game glitch death handling

- Win Game
    - **Primary Actor:** Player
    - **Goal in context:** Player wins by collecting all the regular rewards and reaching the exit cell
    - **Preconditions:**
        i. Game has started
        ii. Main character is alive
        iii. Collected all regular awards on the board
        iv. Navigating towards exit cell

- **Trigger:** Main character reaches the exit cell with all the regular rewards.
- **Scenario:**
    i. Player navigates main character avoiding enemies and picking up rewards
    ii. Player gathers all regular awards across board game
    iii. System verifies that all regular rewards were gathered
    iv. Player guides the main character towards the exit cell
    v. System checks if character reached exit and all winning conditions are met
    vi. Game displays win screen with data: {time taken}
    vii. Restarts to first game screen
- **Exceptions:**
    i. Players score/health drops below 0 -> game is over and player loses
    ii. Encounters another player before reaching the exit -> game is over and player loses
- **Priority:** Main priority, goal to win game
- **When available:** Final phase of gameplay, when all rewards have been collected
- **Frequency of use:** Once per game session, given the completion of preconditions
- **Channel to Actor:** Visual display (UI)
- **Secondary Actors:**
    i. Game environment (board, rewards, enemies, barriers).
    ii. Score System
- **Open Issues:**
    i. If main character collects all regular rewards but doesn't reach the exit cell
    ii. If the main character reach the exit cell but doesn't collect all regular rewards
    iii. A clear UI to notify the stage -> whether all rewards have been picked up.

- Hide On Bush
    - **Primary Actor:** Player
    - **Goal in context:** Hide from enemies in the bush to avoid damage causing enemies to enter a confused state.
    - **Preconditions:** Player is adjacent to a cell containing a bush.
    - **Trigger:** Player enters cell containing a bush
    - **Scenario:**

    i. Player is being chased by enemy
    ii. The player observes a bush on the game board
    iii. Player moves into bush cell
    iv. System recognizes player is in the bush
    v. Player enters stealth state and UI or game elements informs player of this state
    vi. Enemies enter confused state where the enemy then moves randomly
    vii. Player remains hidden until leaving the bush cell
- **Exceptions:**
    i. Player is already in a bush cell, the player's position is updated but the game state remains unchanged.
- **Priority:**
    i. Medium, it adds a feature that makes the game unique but is not essential.
- **When available:** First Increment
- **Frequency of use:** Used frequently whenever bush available.
- **Channel to Actor:** Keyboard input is used to move character and enter stealth state
- **Secondary Actors:**
    i. Game system
- **Open Issues**
    i. Deterrent for players who stay in bush for too long