Use Cases Phase 1 CMPT 226

Main Program

Use Case: Open program
Primary actor: Player
Goal in Context: Initiate the game program
Preconditions: Have the Java code compiled into an executable file
Trigger: The player decides to launch the game
Scenario:

1. The player sits at a computer
2. Player logs into the computer
3. Player double clicks executable

Exceptions: N/A
When available: Anytime
Frequency of use: Any
Channel to actor: Via mouse and keyboard
Secondary actors: N/A
Channels to Secondary actors: N/A

Main Menu

Use Case: Start Game
Primary actor: Player
Goal in Context: On the main menu of the game
Preconditions: Open program Use Case
Trigger: The player decides to begin a round
Scenario:

1. The player clicks the Start button,MenuLogic.difficultySelectionMenu() is called which triggers display.difficultySelectionMenu() to open selection menu visuals
2. The player selects the preferred difficulty and MenuLogic.startGame creates an instance of gameLogic to generate hero, enemies, and board objects.
3. Display.GameRender() is called to display the game board, hero, and enemies

Exceptions: N/A
When available: On the main game menu
Frequency of use: Any
Channel to actor: Via mouse and keyboard
Secondary actors: N/A
Channels to Secondary actors: N/A

Use Case: Exit Game
Primary actor: Player
Goal in Context: Terminate the program
Preconditions: in the main menu
Trigger: The player wishes to close game
Scenario:

1. The player presses the exit button, the main game loop ends and the program closes

Exceptions: N/A
When available: in the main menu
Frequency of use: once per game launch
Channel to actor: Via mouse and keyboard:
Secondary actors: N/A
Channels to Secondary actors: N/A

Use Case: Open the Settings menu
Primary actor: Player
Goal in Context: Open the settings menu
Preconditions: On the main menu of the game
Trigger: The player decides to open the settings menu
Scenario:

1. The player clicks the settings button
2. MenuLogic.SettingsMenu() is called which triggers Display.SettingsMenu() which changes visuals

Exceptions: N/A
When available: On the main game menu
Frequency of use: Any
Channel to actor: Via mouse and keyboard:
Secondary actors: N/A
Channels to Secondary actors: N/A

Settings Menu

Use Case: Change game volume
Primary actor: Player
Goal in Context: to adjust the game volume
Preconditions: In the settings menu
Trigger: The player wishes to change the game volume
Scenario:

1. The player presses the ‘+’ or ‘-’ buttons
2. These will call MenuLogic.adjustVolume with a + or - parameter to raise or lower the volume respectively

Exceptions: the game is muted volume will not change

When available: in the settings menu
Frequency of use: anytime
Channel to actor: Via mouse and keyboard:
Secondary actors: N/A
Channels to Secondary actors: N/A

Use Case: Mute/unmute Game Audio
Primary actor: Player
goal in Context: to mute or unmute the game audio
Preconditions: in the settings menu
Trigger: The player wishes to mute or unmute the game's audio
Scenario:

1. The player presses the mute button this will call MenuLogic.muteToggle() which will mute or unmute the game audio based on the previous setting

Exceptions: N/A
When available: in the settings menu
Frequency of use: Any
Channel to actor: Via mouse and keyboard:
Secondary actors: N/A
Channels to Secondary actors: N/A

Use Case: Change Characters skin
Primary actor: Player
Goal in Context: Change the skin of the character
Preconditions: In the settings menu
Trigger: player wishes to change the look of their character
Scenario:

1. The player clicks on the character icon in the settings menu
2. MenuLogic.changeCharSkin() is called to switch the the next skin
3. Display.SettingsMenu will be called to re-render the menu with new skin

Exceptions: N/A
When available: in the settings menu
Frequency of use: any
Channel to actor: Via mouse and keyboard:
Secondary actors: N/A
Channels to Secondary actors: N/A

Gameplay

Use Case: Choose restart after game-over by loss
Primary actor: Player

Goal in Context: game resets to the state of a new game; a new round begins with the same difficulty
Preconditions: A round of the game must be played, and the game must be terminated via intended game mechanics i.e not exiting the program
Trigger: The round ends
Scenario:

1. The round ends
2. Display.GameOverMenu is called to open game over menu
3. The Player chooses the restart option
4. MenuLogic.startGame() is called again with the same difficulty parameter

Exceptions: N/A
When available: after the end of a round ended by either collision with a moving enemy or negative score

Frequency of use: once per round
Channel to actor: Via mouse and keyboard:
Secondary actors: N/A
Channels to Secondary actors: N/A

Use Case: Choose “menu” after game-over by loss
Primary actor: Player
Goal in Context: game brings UI back to the main menu display
Preconditions: A round of the game must be played, and the game must be terminated via intended game mechanics i.e not exiting the program
Trigger: The round ends
Scenario:

1. The round ends with a loss
2. Display.GameOverMenu is called to open game over menu
3. The Player chooses the “back to main menu option”
4. GameLogic object is deleted and the variable is set to null
5. Display.MainMenu() is called to render the main menu

Exceptions: N/A
When available: after the end of a round ended by either collision with a moving enemy or negative score

Frequency of use: once per round
Channel to actor: Via mouse and keyboard:
Secondary actors: N/A
Channels to Secondary actors: N/A

Use Case: character movement
Primary actor: Player

Goal in Context: player decides to move the character
Preconditions: the game must be running
Trigger: player presses “w”, “a”, “s”, or “d” for up left down or right movements respectively
Scenario:

1. Player inputs the movement key
2. GameLogic.processPlayerMovement() is called with the direction, this method makes the considerations below
3. If the square is empty, the player moves
4. If the square has a barrier, the player remains in the current space
5. If the square has a reward, the player moves and collects the reward (see collected reward use case)
6. If the square has a stationary enemy, the player moves and takes damage (see stationary enemy use case)
7. If the square has a moving enemy, game over (see moving enemy use case & game-over use case)
8. If the square is the exit and all rewards are collected, the game is won (see win condition use case)

Exceptions: the game is paused
When available: during gameplay
Frequency of use: once per movement tick
Channel to actor: Via mouse and keyboard
Secondary actors: N/A
Channels to Secondary actors: N/A

Use Case: collect reward
Primary actor: Player
Goal in Context: collect a reward
Preconditions: the game is running, the player is beside a reward tile
Trigger: the player moves the character onto the reward tile
Scenario:

1. The player moves the character onto the reward tile
2. The Reward is removed from the board array (which means it will not appear in the next call of Display.GameRender())
3. The score from the reward is added to the total score using GameLogic.adjustScore()

Exceptions: N/A
When available: during gameplay
Frequency of use: however many rewards are spawned for the current difficulty
Channel to actor: Via mouse and keyboard:
Secondary actors: N/A
Channels to Secondary actors: N/A

Use Case: stationary enemy

Primary actor: Player
Goal in Context: take damage from stationary enemy
Preconditions: the game is running, the player is beside a stationary enemy tile
Trigger: the player moves the character onto the stationary enemy tile
Scenario:

1. The player moves the character onto the stationary enemy tile
2. The stationary enemy is removed from the board nxm array(this means it will not appear in the next Display.GameRender() call)
3. The score is deducted from the total score using GameLogic.adjustScore
4. If the score is negative, game over (see the game-over use case)

Exceptions: N/A
When available: during gameplay
Frequency of use: however many stationary enemies are spawned for the current difficulty
Channel to actor: Via mouse and keyboard:
Secondary actors: N/A
Channels to Secondary actors: N/A

Use Case: moving enemy
Primary actor: Player
Goal in Context: lose the game by touching a moving enemy
Preconditions: the game is running
Trigger: either the moving enemy moves onto the same square as the character or vice versa
Scenario:

1. Moving enemy moves onto the same tile as the main character or vice versa
2. Game-over use case is triggered

Exceptions: N/A
When available: during gameplay
Frequency of use: once per game
Channel to actor: Via mouse and keyboard:
Secondary actors: N/A
Channels to Secondary actors: N/A

Use Case: game over
Primary actor: Player
Goal in Context: end the game due to a loss
Preconditions: the game is running
Trigger: player gets a negative total score or touches a moving enemy
Scenario:

1. Stationary enemy use case or moving enemy use case

2. Display.GameOver() is called to render the intermediate screen that displays a “Game Over” message, players' score, the final time, “try again”, and “back”
3. If the player presses “try again” then “choose restart after game-over by loss” use case, if the player presses “back” then “choose menu after game-over by loss” use case

Exceptions: The player wins the game
When available: during gameplay
Frequency of use: once per round
Channel to actor: Via mouse and keyboard:
Secondary actors: N/A
Channels to Secondary actors: N/A

Use Case: Win condition
Primary actor: Player
Goal in Context: win the game
Preconditions: Player has collected all regular rewards (see collect reward use case)
Trigger: player walks through the exit gate
Scenario:

1. All rewards are collected
2. GameLogic.openExit is called to open the exit for the player
3. The player moves the character to the exit
4. Display.GameOverMenu is called to open the game-over menu which displays Congratulations, total score, time, try again, and main menu options
5. If the Player chooses to try again option
6. MenuLogic.startGame() is called again with the same difficulty parameter
7. . if the player presses “back” the player is taken back to the main menu
8. GameLogic object is deleted and the variable is set to null
9. Display.MainMenu() is called to render the main menu

Exceptions: game-over use case
When available: during gameplay
Frequency of use: once per game
Channel to actor: Via mouse and keyboard:
Secondary actors: N/A
Channels to Secondary actors: N/A

Use Case: GameTick
Primary actor: System
Goal in Context: process a game tick
Preconditions: the game is running
Trigger: a certain amount of time passes tbd base on testing
Scenario:

1. Process player movement based on character movement use case

2. GameLogic.processEnemyMovment() is called to move all enemies in the direction closest to the character, direction will be determined via a breadth-first search on the board nxm array
3. Display.GameRender() called to refresh visuals

Exceptions: game-over use case
When available: during gameplay
Frequency of use: once per game
Channel to actor: Via mouse and keyboard:
Secondary actors: N/A
Channels to Secondary actors: N/A