CMPT 276

Introduction to Software Engineering

Phase 1 : Use Cases

Group 22

- Gurnoor Chahal (Garry)
- Thomas Kingsman
- Owen Liu
- Yoobin Nam

1) Simple Cases:

a) Main Actor: Teaching Assistant (TA)

Use case: moveCharacter
Primary actor: TA (player)
Goal in context: To move the character in the map when the user presses the keys
Preconditions: Code has been set for key
Trigger: User presses the keys
Scenario:

1. User: press the key (among up,down,left,right)
2. Character: move depending on the key

Exceptions:

1. Press invalid keys: character will not move until user presses a valid key
2. When pressing keys in opposite direction at the same time: character will not move
3. When the barrier is in front of character: character cannot move forward, have to go other directions

Use case: getRegularReward
Primary actor: TA
Goal in context: To get maximum sanity to finish the game
Preconditions: Regular rewards must be distributed
Trigger: When the character reaches the reward position (x,y)
Scenario:

1. Character: move to get reward
2. Character: reach the target location
3. Character: get some sanity to finish the game

Exceptions:

1. Already got maximum sanity: will not get sanity

Use case: getBonusReward
Primary actor: TA
Goal in context: To get bonus sanity in a few seconds
Preconditions: Code has been set so that the bonus comes out at random time
Trigger: When the character reaches the bonus reward position (x,y)
Scenario:

1. Bonus Reward: appear in random location at random time
2. Character: move to get reward
3. Character: reach the target location
4. Character: get extra bonus sanity

Exceptions:

1. Reach the location after reward disappeared: cannot get sanity
2. Already got maximum sanity: will not get sanity

Use case: meetMovingEnemies
Primary actor: TA
Goal in context: To end (lose) the game
Preconditions: Need to set code to reset everything when reaching the same location as the student
Trigger: When the character reaches the same position (x,y) as the student
Scenario:

1. students: walk around the map towards player
2. Character: walk around the map to get reward by user
3. Character: reach the student at the same location
4. User Interface: end and lose the game

Exceptions:

No exception

Use case: meetStaticEnemies
Primary actor: TA
Goal in context: To lose sanity
Preconditions: Need to set code to reduce the sanity when reaching the same location as the static enemies
Trigger: When the character reaches the same position (x,y) as the static enemies
Scenario:

1. Static enemies: appear on random locations when game is started
2. Character: walk around the map to get reward by user
3. Character: meet the static enemies at the same location
5. User Interface: lose some sanity

Exceptions:

1. If do not have sanity: end and lose the game

Use case: reachEndLocation (Office)
Primary actor: TA
Goal in context: To end the game (or go to the next stage)
Preconditions: Need to set code that check earning sanity, and character’s position to finish the game
Trigger: When the character reaches the end location with maximum sanity
Scenario:

1. Character: collect reward to get maximum sanity
2. Character: move to the end location with maximum sanity
3. Character: reach the end location (x,y)
4. U.I: show the sanity and time

Exceptions:

1. Character reaches end location without maximum sanity: nothing happen

b) Main Actor: Enemies

Use case: makeMovingEnemies
Primary actor: movingEnemies
Goal in context: To make user lose the game
Preconditions: Need to set code that creates some number of students and moving 1 value at a time towards player
Trigger: When game is started
Scenario:

1. When user press the start button
2. Show the game screen and creates students

Exceptions:

1. When the game is not running: do not create students

Use case: makeStaticEnemies
Primary actor: staticEnemies
Goal in context: To reduce sanity
Preconditions: Need to set code that create some number of punishments on the random location
Trigger: When game is started
Scenario:

1. When user press the start button
2. Show the game screen and creates punishments

Exceptions:

1. When the game is not running: do not create punishments

c) Main Actor: Rewards

Use case: makeRegularReward
Primary actor: regularReward
Goal in context: To get sanity
Preconditions: Need to set code that create some number of rewards
Trigger: When game is started
Scenario:

1. When user press the start button
2. Show the game screen and creates regular rewards

Exceptions:

1. When the game is not running: do not create rewards

Use case: makeBonusReward
Primary actor: regularReward
Goal in context: To get bonus sanity
Preconditions: Need to set code that create bonus reward randomly during the playing
Trigger: While the game is running
Scenario:

1. When user press the start button
2. User plays the game
3. Bonus reward appears
4. Disappear if player gets it, or after a few seconds later

Exceptions:

1. When the game is not running: do not create bonus

d) Main Actor: Time

Use case: checkTime
Primary actor: Time (timer)
Goal in context: To display how long it takes to finish the game
Preconditions: Need to set code that check time after starting game
Trigger: After game is started
Scenario:

1. Starting timer after starting game
2. When game is ended (lose or win), calculate time consumption

Exceptions:

1. Quit game during the play: cannot use the time value

e) Main Actor: User Interface

Use case: displayStartScene
Primary actor: UI, Screen
Goal in context: To deliver our output
Preconditions: Need to set code that create UI with start and setting button
Trigger: When code is executed
Scenario:

3. Run the program: start to display our start scene

Exceptions:

2. When there is errors: our scene do not display on the screen

Use case: displayPlayScene
Primary actor: UI, Screen
Goal in context: To deliver our output
Preconditions: Need to set code that create UI with our entities
Trigger: After start button is pressed
Scenario:

1. Start button is pressed from the start scene: change the UI to play scene
2. Starting some above cases: make player, enemies, rewards and barriers

Exceptions:

1. When there is errors: the scene does not display on the screen

Details for all of above cases:

- priority: Essential to play game, must be set
- When available: After code is set and during the play game
- Frequency of use: It depends on how much each case occurs during the game
- Channel to actor: Via screen (monitor) with set code
- Secondary actors: All the main characters a)~d) can be secondary actors depending on the cases
- Channel to secondary actors: Via screen (monitor) with set code

The ultimate case

Main actor: user

Use case: playGame
Primary actor: User (with screen)
Goal in context: To play the game
Preconditions: All of above cases must be set
Trigger: User want to play our game
Scenario:

1. Our program is started
2. Start scene is displayed on the screen with buttons
3. User press the start button
4. Our game is started, change the scene to play scene
5. Create player, enemies, rewards, and barriers on the screen
6. User need to get maximum score to finish the game
    - avoiding enemies and punishments
    - get reward to earn sanity
    - if player meet student, the game is ended
    - if sanity become negative value, the game is ended
7. After got maximum sanity, try to move the character to the end location
8. If character reaches at the end location with maximum sanity, screen show the sanity and time

Exceptions:

1. If there is a error: game will not be started

Priority: Essential to play and check the game
When available: After code is set
Frequency of use: Everytime when code is run
Channel to actor: Monitor (screen)
Secondary actors: Keyboard