CMPT 276 D100
FALL 2021
GROUP 21
PHASE 3

LEI GONG
STEVEN QUINN
JUSTIN TANG
WENHUI WU

TEST CASES

- **Agents**
  - AgentTest
    - Create movable Enemy (1)
    - Create immovable Enemy (2)
    - Create MainCharacter
    - Verify Enemy (1) is movable
    - Verify Enemy (2) is immovable
  - DirectionTest
    - Check the enum of each direction is correct
  - EnemyTest
    - Check the Constructor method of two types of enemies
    - Test getCountdown() method
    - Test minusCountdown() method
    - Test getPunishment() method
    - Test setRetired() and isRetired() methods
    - Test setStuck() and isStuck() methods
  - MainCharacterTest
    - Check the Constructor method
    - Test getHealth() method
    - Test getCharacterScore() and addCharacterScore() methods
    - Test getMaxHealth() and setMaxHealth() methods
    - Test setHealth() method
- **Display**
  - AssetTest
    - Test if asset map loads well
- **Fixed**
  - BonusRewardTest
    - Check reduceLifespan() method reduced the lifespan or not
    - Check if isRetired() method checking the item has retired or not
    - Test setRetired() method
    - Test BonusRewar() method
    - Test if getLifespan() method get correct value
  - EventTileTest
    - Check the Constructor method
    - Test reveal() method
    - Test toString() method

    - RewardTest
        - Check if getScore() method returns correct score
        - Test setScore() method
        - Test if Reward() set initial score.
    - WallTest
        - Test if Wall isPermeable() method
        - Test if setX and setY are set the value
        - Test if Wall contains correct X value and Y value
- **Game**
    - CellTest
        - Test if cell contains permeable wall
        - Test getX() and getY() methods get position correctly.
        - Tested if method contain object in the list
    - MazeTest
        - Create mazes
        - Verify interactions between tiles
          (MainCharacter → Wall, EnemyB → EnemyA, etc.)
        - Check exit conditions
        - Check exceptions

TEST COVERAGE

Tests	Line Coverage (%)	Branch Coverage (%)
AgentTest	100	100
DirectionTest	100	100
EnemyTest	100	100
MainCharacterTest	100	100
AssetTest	100	100
BonusRewardTest	100	100
EventTleTest	100	100
RewardTest	100	100
WallTest	100	100
CellTest	100	100
IntegrationTest - **MazeTest** - **MazeObject**	 80 100	 84 100

TEST RESULTS

- addMazeObjectToHomeCell not covered (potentially useful)
- addMazeObjectToCell not covered (potentially useful)
- getMazeObjectAsString and getCellsArrayAsString are not tested
    - Both functions were used for printing a text-based version of the Maze to visualize object positions, and are not used in the final rendition of the game
- WinPage, EndPage, and IntroductionText are not tested
    - All functions are final; there are no other possible way to test these functions other than running the existing ones, so test cases are not created
- Clock is not tested
    - It is a class that extends Pane and there is no way to modify or call its internal variables. Since it runs automatically, test cases are not needed. In other words, since there are no variables to be changed, the method Clock is not tested.

FINDINGS & CODE ADJUSTMENTS

Through IntegrationTest, there are several findings which require adjustments or overhauls.

For instance, in one of the tests, mainCharacter tries to move into the end tile while the exit condition is false. This should have resulted in the main character not being able to move and remain in the same tile. However, the main character managed to move into the end tile despite the exit condition being false; this is due to the fact that the end tile was made with the assumption that it will be created at the outer wall - end tile itself is permeable and since it shares the same tile as the wall tile, the wall tile gave it the impermeable attribute.

Another finding has been found during initial tests for exceptions. In those tests, assertions for exceptions do not seem to work. It turns out that it has not thrown an exception; from this, an adjustment of throwing new exceptions has been added, improving the code.

The process of testing the game also gave us an opportunity to remove unused classes that had been left in from previous iterations. One such example is the Space class, which was originally used to signify an empty tile before the group switched to a Cell-object-based three dimensional Maze.