2IP90 Programming - CBL Assignment Backlog (Project Group 176)

Huy Hong (1935569), Long Pham (2000954)

Learning goals of choice: Test-driven development, version control

Throughout the project, we plan to implement test-driven development, which we first do the following:
- Writing tests for features before implementing the methods
- Running the tests (which would fail to indicate that testing is running properly)
- Write the simplest code that would pass the tests
- Refactor the code as necessary (implement the methods).

We plan to show this via Github, which will also be another learning goal of ours.
We also plan to use branches for different features for organization and reflect that we are properly using test-driven development from our commit history.

Name (ranked in terms of priority - from top to bottom)	How to demo	Notes
Player movement	Move the player using "WASD"/arrow keys, where they represent up, left, down, right arrow keys respectively. Each time when a direction key is clicked once, the player moves 1 cell in the corresponding direction of the arrow key.	Since each click only advances the player by 1 cell, if you click for instance, the "W" key and the left arrow key subsequently (while very quickly), then the player can appear to move faster.
Player interaction with walls	Attempt to move in the direction of a wall. Then the player cannot move until they click an arrow key in the direction away from the wall.	Walls do not kill the player and stop the player from moving towards the wall. This is the same for repeatedly going into the wall. Does not fall into the 'obstacles' category.
Player interaction with monsters and obstacles	Hit/got hit by monsters/obstacles. Then the player dies and respawns at the entrance.	A monster is a mobile object that can kill the player. An obstacle is an immobile object that can kill the player.

2IP90 Programming - CBL Assignment Backlog (Project Group 176)

Huy Hong (1935569), Long Pham (2000954)

Learning goals of choice: Test-driven development, version control

Throughout the project, we plan to implement test-driven development, which we first do the following:
- Writing tests for features before implementing the methods
- Running the tests (which would fail to indicate that testing is running properly)
- Write the simplest code that would pass the tests
- Refactor the code as necessary (implement the methods).

We plan to show this via Github, which will also be another learning goal of ours.
We also plan to use branches for different features for organization and reflect that we are properly using test-driven development from our commit history.

Name (ranked in terms of priority - from top to bottom)	How to demo	Notes
Starting the game	Open the application. Maximize the window. Click "New Game" button in the title screen. Enter a username. Choose a difficulty. A stage with the corresponding difficulty appears. The player is spawned at the map entrance after 3 seconds.	When application is opened, the title screen appears with "New Game", "Leaderboard" and "Quit Game" buttons. After "New Game", a username input screen appears. Name input must be of type String and be less than 35 chars (spaces included). If the input is longer or not of type String, the game will ask for input again until the correct input is there.
Completing the stage	Reach the exit in the stage. A congratulations screen appears. Click the "Restart" button to restart the stage. Click the "Leaderboard" button to be directed to the leaderboard screen. Click the "Home" button to be directed to the title screen.	"Restart", "Home", and "Leaderboard" buttons are next to each other in the congratulations screen. (Main priority is to create the stage for the Normal difficulty. Once all the components up to "Quitting the application" work, then we will create the stages for the other 2 difficulties.)

2IP90 Programming - CBL Assignment Backlog (Project Group 176)

Huy Hong (1935569), Long Pham (2000954)

Learning goals of choice: Test-driven development, version control

Throughout the project, we plan to implement test-driven development, which we first do the following:
- Writing tests for features before implementing the methods
- Running the tests (which would fail to indicate that testing is running properly)
- Write the simplest code that would pass the tests
- Refactor the code as necessary (implement the methods).

We plan to show this via Github, which will also be another learning goal of ours.
We also plan to use branches for different features for organization and reflect that we are properly using test-driven development from our commit history.

Name (ranked in terms of priority - from top to bottom)	How to demo	Notes
Quitting the stage	Click the "Quit Stage" button in the stage screen. Player is redirected to the stage selection screen.	"Quit Stage" button is in the top-left corner of the stage screen. After having chosen the stage, the player can proceed as normal by playing the game.
Quitting the game	Click the "Quit Session" button in the stage screen/title screen. Player is redirected to the title screen.	The "Quit Session" button is next to the "Quit Stage" button on the right.
Quitting the application	Click the "Quit Game" button in the title screen. The application is closed.	The "Quit Game" button is in the home screen.
Adding a timer	Start a stage. Then a timer begins as soon as the player is spawned in the stage.	Timer stops either when player completes the stage, or dies, or quits the stage, or quits the game. Time will only be saved when completing the stage. Timer appears in the bottom-left corner of the stage screen.
Player interaction with points	Multiple little pellets will be placed on the path that the player can go on. Each pellet equate to 1 point.	The player obtain points by going on the same position as the pellets. The player loses all their points when they die.

2IP90 Programming - CBL Assignment Backlog (Project Group 176)

Huy Hong (1935569), Long Pham (2000954)

Learning goals of choice: Test-driven development, version control

Throughout the project, we plan to implement test-driven development, which we first do the following:
- Writing tests for features before implementing the methods
- Running the tests (which would fail to indicate that testing is running properly)
- Write the simplest code that would pass the tests
- Refactor the code as necessary (implement the methods).

We plan to show this via Github, which will also be another learning goal of ours.
We also plan to use branches for different features for organization and reflect that we are properly using test-driven development from our commit history.

Name (ranked in terms of priority - from top to bottom)	How to demo	Notes
Adding difficulties	Choose a difficulty between 3 different difficulties: Easy, Medium, Hard after you have inputted the player name. There are 3 buttons for each difficulty that you can click, and you can only choose one difficulty. After choosing the difficulty, a screen with the description of the stage is shown for 8 seconds. The stage with the chosen difficulty appears on the screen.	Obstacles and walls remain the same every time the stage is played. Point pellets are randomly generated across the stage, where the points are scattered differently for every time the stage is played. The stage terrain is made up of a number of cells. "Easy" mode has: - Small stage terrain (100 cells) - Obstacles (5) - Walls (25) - No monsters - Points (50) "Medium" mode has: - Normal stage terrain (250 cells) - Obstacles (50) - Walls (50) - Monsters (15), and they move at a relative speed (1 cell / sec) - Points (125) "Hard" mode has: - Normal stage terrain (500 cells) - Obstacles (150) - Walls (100) - Monsters (50), and they move at a faster speed (3 cells / sec) - Points (250) Note that the number of cells in the stage terrain include the cells where the obstacles, walls,

2IP90 Programming - CBL Assignment Backlog (Project Group 176)

Huy Hong (1935569), Long Pham (2000954)

Learning goals of choice: Test-driven development, version control

Throughout the project, we plan to implement test-driven development, which we first do the following:

- Writing tests for features before implementing the methods
- Running the tests (which would fail to indicate that testing is running properly)
- Write the simplest code that would pass the tests
- Refactor the code as necessary (implement the methods).

We plan to show this via Github, which will also be another learning goal of ours.
We also plan to use branches for different features for organization and reflect that we are properly using test-driven development from our commit history.

Name (ranked in terms of priority - from top to bottom)	How to demo	Notes
Accessing the leaderboard	Click the "Leaderboard" button in the title screen. Player is redirected to the leaderboard screen.	Leaderboard shows 10 entries at once and contains: - Rank in leaderboard (from 1 to 10) - Username input - Time - Number of points
Sound effects and music	Each stage difficulty has a different song in the background. Click any button in the application, then a clicking sound effect follows.	The song stops when you are no longer on one of the stage screens ("Easy", "Medium", or "Hard" modes)