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 (TDD) as a learning goal, 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 (i.e. version control).
We also plan to use branches for different features for organization and show that we are properly using test-driven development from our commit history and a logical branch structure.

Name (ranked in terms of priority - from top to bottom)	How to demo	Notes
Starting the game	Open the application. The window is fixed to dimensions 1280px x 720px. Click "New Game" button in the home screen. Enter a (valid) username. Choose a difficulty. A stage with the corresponding difficulty appears.	When application is opened, the home 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 20 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.
Player movement	Move the player using "WASD", where they represent up, left, down, right movements 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.

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 (TDD) as a learning goal, 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 (i.e. version control).
We also plan to use branches for different features for organization and show that we are properly using test-driven development from our commit history and a logical branch structure.

Name (ranked in terms of priority - from top to bottom)	How to demo	Notes
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. Note that no two stage components can be on the same cell at once.
Player interaction with monsters and obstacles	Hit/got hit by monsters/obstacles. The player dies and a popup screen appears to inform that the player just died. Player is then redirected to the stage selection screen.	A monster is a mobile object that can kill the player and destroy walls upon collision with them. An obstacle is an immobile object that can kill the player. Note that no two stage components can be on the same cell at once.
Quitting the stage	Click the "Quit Stage" button in the stage screen. Player is redirected to the stage selection screen.	After having chosen the stage, the player can proceed as normal by playing the game.
Quitting the game session	Click the "Quit Game Session" button in the stage screen. Player is redirected to the home screen.	The "Quit Game Session" button is next to the "Quit Stage" button on the right.

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 (TDD) as a learning goal, 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 (i.e. version control).
We also plan to use branches for different features for organization and show that we are properly using test-driven development from our commit history and a logical branch structure.

Name (ranked in terms of priority - from top to bottom)	How to demo	Notes
Quitting the game	Click the "Quit Game" button in the home screen or the stage screen. Popup appears to ask for confirmation. If "Yes" is chosen, then the application is closed. If "No" is chosen, then player remains in the same screen as where they have clicked the button.	Popup shows two options for confirmation.
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.

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 (TDD) as a learning goal, 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 (i.e. version control).
We also plan to use branches for different features for organization and show that we are properly using test-driven development from our commit history and a logical branch structure.

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, the stage with the chosen difficulty appears on the screen.	Walls, obstacles, and monsters are randomized on the stage map. 'Easy' stage has a 15x15 stage map (225 cells) and no monsters. 'Medium' stage has a 25x25 stage map (625 cells). 'Hard' stage has a 40x40 stage map (1600 cells). We compute the density of the game elements using weights. - Scaling (s = {0, 1, 2}) is a scalar where the values of s corresponds to the stage difficulty. - Wall weight (w) is a positive real number. - Obstacle weight (o) is a positive real number. - Monster weight (m) is a positive real number. - Empty cells weight (c) is the sum of the wall, obstacle and monster weights. - Total weights (t) is the sum of the all the weights. - The probability of walls and obstacles in a map is: (w * (s + 1) / (t + c)) or (o * (s + 1) / (t + c)). - The probability of monsters is ((m * s) / (t + c)), to account for no monsters in the "Easy" stage. - The probability of empty cells is always 1/2.

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 (TDD) as a learning goal, 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 (i.e. version control).
We also plan to use branches for different features for organization and show that we are properly using test-driven development from our commit history and a logical branch structure.

Name (ranked in terms of priority - from top to bottom)	How to demo	Notes
Completing the stage	Reach the goal in the stage map. A congratulations popup screen appears. Player is then redirected to the leaderboard screen.	The player is spawned in the top left corner of the map. The goal is in the bottom right corner of the stage map.
Accessing the leaderboard	Click the "Leaderboard" button in the home 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 - Chosen stage difficulty - Finished stage time