Main Report

Unit Tests

Main Character Moves left, right, up and down

- Tested in MainCharacterTVTest
- Covers the feature of the player being able to move with WASD and that the character moves 4 pixels each time a key is pressed in the corresponding direction.

Main Character score goes up by 100 after picking up a power-up

- Tested in MainCharacterTVTest
- Covers the feature of player being able to pick up power-ups and that their score will increase by 100.

Main Character score goes up by 200 after picking up key card

- Tested in MainCharacterTVTest
- Covers the feature of the player being able to pick up key cards and that their score will increase by 200.

Main Character health goes to zero when colliding with an enemy

- Tested in MainCharacterTVTest
- Covers the feature of the player being able to collide with an enemy and that their health will drop to 0 as a result.

Main Character health goes to zero when colliding with trap

- Tested in MainCharacterTVTest
- Covers the feature of the player being able to collide with a trap and that their health will drop to 0 as a result.

Main Character life goes up when power up is collected

- Tested in MainCharacterTVTest
- Covers the feature of the player being able to collect power-ups and that their health will increase

Main Character key card number goes up when collecting key card

- Tested in MainCharacterTVTest
- Covers the feature of the player being able to collect key cards and that the number of key cards that the player has is incremented.

Main Character can’t walk past walls in all 4 directions

- Tested in CollisionCheckerTest
- Covers the feature that the player will not be able to walk past walls and the player’s x or y value will not change when walking in the direction of a wall.

Main Character can collide with enemies

- Tested in CollisionCheckerTest
- Covers the feature of the player being able to collide with enemies in a various of locations.

All of OBJ_Hole are initiated correctly

- Tested in AssetSetterTest
- Covers the feature of the game that the traps are placed correctly in the map

All of OBJ_Battery are initiated correctly

- Tested in AssetSetterTest
- Covers the feature of the game that the power-ups are placed correctly in the map

All of OBJ_Door are initiated correctly

- Tested in AssetSetterTest
- Covers the feature of the game that the exit door is placed correctly in the map

All of OBJ_KeyCard are initiated correctly

- Tested in AssetSetterTest
- Covers the feature of the game that the key cards are placed correctly in the map

Door is interactable with the Main Character

- Tested in CollisionChecker
- Covers the feature of the game that the player is allowed to interact with the door when they have 3 key cards

The game is over when Main Character health goes to zero

- Tested in GamePanelTest
- Covers the feature of the game ending when the player’s life reaches zero

W Key moves MainCharacter up

- Tested in KeyHandlerTest
- Covers the feature that the direction of the character changes to up when W is pressed

S Key moves MainCharacter down

- Tested in KeyHandlerTest
- Covers the feature that the direction of the character changes to down when S is pressed

A Key moves MainCharacter left

- Tested in KeyHandlerTest
- Covers the feature that the direction of the character changes to left when A is pressed

D Key moves MainCharacter right

- Tested in KeyHandlerTest
- Covers the feature that the direction of the character changes to right when D is pressed

P key pauses game (checks game state)

- Tested in KeyHandlerTest
- Covers the feature of the game pausing when the P key is pressed

I key displays instructions (checks game state)

- Tested in KeyHandlerTest
- Covers the feature of the game showing instructions when the I key is pressed

Level is incremented to level 2 after winning level 1

- Tested in GamePanelTest
- Covers the feature of the level being incremented after completing the first level

Level is decremented to level 1 after losing level 1 & retries = 0

- Tested in GamePanelTest
- Covers the feature of the level being decremented after losing level 2 with 0 retries left

Integrations Tests

Interaction between UI and KeyHandler with Main Menu Options

- Tested in IntegrationsTest
- Covers the interaction between the KeyHandler and the UI when the player presses W or S in the main menu. When W or S is pressed, the selection should change on screen.

Interaction between UI, KeyHandler and GamePanel with Main Menu Play Option

- Tested in IntegrationsTest
- Covers the interaction between the KeyHandler, the UI and the GamePanel when the player chooses Play. When Play option is selected and the player presses enter, the game state should change to play state

Interactions between UI and KeyHandler with Lose Screen Options

- Tested in IntegrationsTest
- Covers the interaction between the KeyHandler and the UI when the player presses W or S on the lose screen. When W or S is pressed, the selection should change on screen.

Interactions between UI, KeyHandler and GamePanel with Lose Screen Retry Option

- Tested in IntegrationsTest
- Covers the interaction between the KeyHandler, the UI and the GamePanel when the player chooses Retry on the lose screen. When the Retry option is selected and the player presses enter, the game state should change to play state and return to the current level.

Interactions between UI, KeyHandler and GamePanel with Lose Screen Exit Option

- Tested in IntegrationsTest
- Covers the interaction between the KeyHandler, the UI and the GamePanel when the player chooses Exit on the lose screen. When the Exit option is selected by pressing S and the player presses enter, the game state should change to title state and go back to the main menu screen.

Interactions between UI, KeyHandler with Win screen options

- Tested in IntegrationsTest
- Covers the interaction between the KeyHandler and the UI when the player presses W or S in the win screen. When W or S is pressed, the selection should change on screen.

Interactions between UI, KeyHandler and GamePanel with Win Screen Continue Option

- Tested in IntegrationsTest
- Covers the interaction between the KeyHandler, the UI and the GamePanel when the player chooses Continue Game on the win screen. When the Continue Game option is selected by pressing enter, the game will change to play state and load the next level.

Interactions between UI, KeyHandler and GamePanel with Win Screen Return To Main Menu Option

- Tested in IntegrationsTest
- Covers the interaction between the KeyHandler, the UI and the GamePanel when the player chooses Return to Main Menu on the win screen. When the Return to Main Menu option is selected by pressing either W or S and then pressing enter, the game will change to title state and return to the main menu.

Interactions between UI, KeyHandler, GamePanel and MainCharacter when starting the game, moving the character around and then pausing and resuming game

- Tested in IntegrationsTest
- Covers the interactions between the KeyHandler, the UI, the GamePanel and the MainCharacter when the player first starts the game and chooses the play option in the main menu. After the game begins, the player moves the Main Character around. The player then pauses the game and then resumes the game.

Test Quality and Coverage

We referenced the lecture notes to ensure we were implementing our testing properly. Branch coverage is very important, so we made sure to consider branch coverage, using block and control flow ideas. We then started creating tests based on the basic functionalities of our game - main character movement, main character health, player score, collisions, etc. and moved on from there to create our integrated tests. After, followed the structure of our code and the layout of our classes to implement any key tests that we believed needed to be tested. We have about 69% line coverage for our testing. The remaining lines are used to draw the map and the screens. We believe that our line coverage is fair, as we covered numerous key features of the game in our tests and as stated in the lecture slides “black box testing gives you only about 60-75% coverage”. An aspect that was difficult to test was the drawing methods that are used to draw elements, as there was no concrete way to determine if the output was occurring as expected, therefore our testing was more game logic focused.

We have 40% branch coverage for our testing. This is due to most of our branches covering drawing and movement, which is difficult to test without the game running fully as the user would see it. As mentioned before, drawing methods were not tested, as many of the branches were if/else statements regarding what to draw.

Findings

Writing and running our tests allowed us to find bugs in our code that we were not aware of and not able to detect from testing our game by simply playing. Implementing our tests granted us the ability to cover issues with game logic that could have caused issues in the future. While writing tests, we gained perspective into developing the game further while avoiding bugs that could be found during our testing phase. Some aspects of our game’s functionality that had issues during testing are as follows. Our scoring system did not include the first level score when the total score was displayed, so we modified that. Additionally, the total time of game play was not displayed properly at the end of the game. However, not all issues were related to a bug in the code, or a logic error. The more we played the game, the more we came to recognize issues with game balancing and progression. We moved various enemy starting locations as well as changed walls in different levels. We also felt that 2 levels weren’t enough of a challenge for the player, so we decided to add another level just to keep the player engaged.

In general, testing our game made us really consider how we could improve our design. There were definitely moments where we asked ourselves “why did we do this” during testing and gave us ideas of what we can refactor for Assignment 3.