Janahan Ravichandran (100701645)
February 11, 2021
SOFE 3980 – Software Quality

Assignment 1 – Report

For this assignment I chose to develop a variant of the game snake. This variant of the game consists of a player object moving around a board with pieces of food spawning onto the board one by one. The player is tasked with collecting these food pieces to increase their score. Players do this by utilizing the four directional keys on the keyboard. While doing this, obstacles appear on the board where if the player encounters one of the obstacles the game ends.

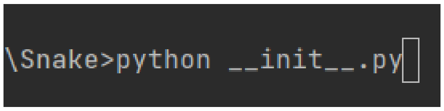
For the implementation the player is represented by a player object which contains a self.x and self.y attribute to hold the current horizontal and vertical position of the player on the board. The object also contains a self.dx and self.dy attribute which keeps track of the direction the player is currently moving. Furthermore, the object contains a move() method for changing the direction the player is moving based on the keyboard input from the player, as well as a draw() method for drawing the player object on the board. Next the classes for the food pieces and the obstacles use similar attributes for x and y coordinates as the player class. Moreover, both the obstacle and snack class use similar methods for drawing as the player class, as well as sharing methods for generating random position for x and y coordinates using the random python module to generate x and y coordinates for each snack and obstacle object on the board.

Within the timeframe given of one week the software engineering process that was applied was the Agile method. I think the Agile model is best for the delivering a high quality in this time frame due to the ability to deliver software quickly while ensuring a working product upon delivery. Furthermore, for a game product the agile approach allows for features to be built incrementally and constantly allow for updates to the game. For example, when developing the snake game updates were made based on features such as movement, and object generation one feature at a time and pushing code to a repository once the feature has been completed. Moreover, by applying this methodology due to the timeframe allows for reduced amount of risk when missing features since this process allows for ongoing releases for the product.

For automated testing I used the python testing framework built into PyCharm along with the python library called unittest. To keep the unit tests organized separate test files were created for testing each object. Problems faced while implementing the testing included working with multiple objects per test. For example, this was caused by testing the generate_obstacle function within the obstacle class, because it required the position of the player to ensure the obstacle wasn’t rendered directly onto the player object. This causes complication due to extra memory usage caused by allocation of extra objects required for testing.

To run the game application:

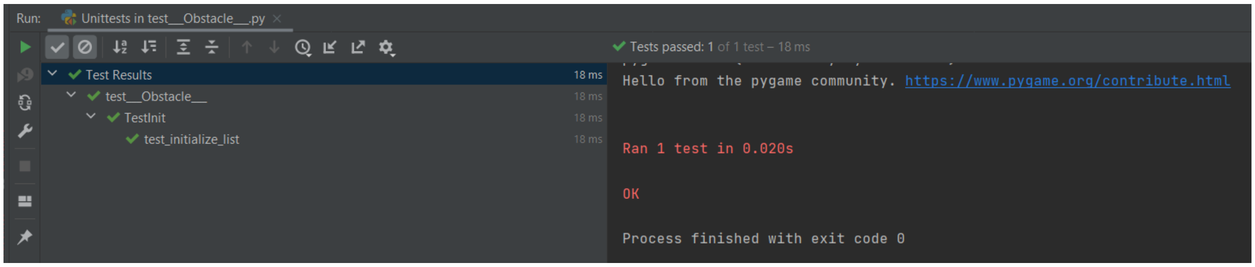
- Navigate to Snake directory and run command python __init__.py from command line.



```
\Snake>python __init__.py
```

Working test cases from PyCharm testing automation:

- Running of the test___Obstacle___.py test file.



Run: Unittests in test__Obstacle__.py

Tests passed: 1 of 1 test – 18 ms

```
Test Results                 18 ms
  test__Obstacle__           18 ms
    TestInit                 18 ms
      test_initialize_list   18 ms
```

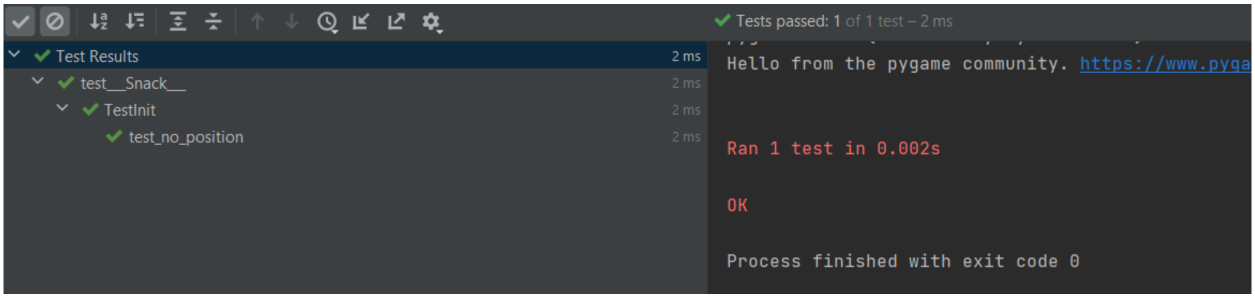
```
Hello from the pygame community. https://www.pygame.org/contribute.html


Ran 1 test in 0.020s

OK

Process finished with exit code 0
```

- Running of the test___Snack___.py test file.



Tests passed: 1 of 1 test – 2 ms

```
Test Results                 2 ms
  test__Snack__              2 ms
    TestInit                 2 ms
      test_no_position       2 ms
```

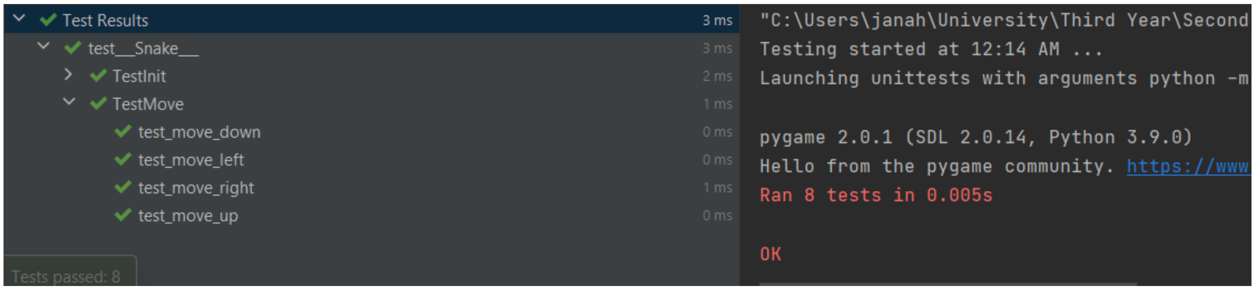
```
Hello from the pygame community. https://www.pyga


Ran 1 test in 0.002s

OK

Process finished with exit code 0
```

- Running of the test___Snake___.py test file.



```
Test Results                 3 ms
  test__Snake__              3 ms
    TestInit                 2 ms
    TestMove                 1 ms
      test_move_down         0 ms
      test_move_left         0 ms
      test_move_right        1 ms
      test_move_up           0 ms
```

Tests passed: 8

```
"C:\Users\janah\University\Third Year\Second
Testing started at 12:14 AM ...
Launching unittests with arguments python -m

pygame 2.0.1 (SDL 2.0.14, Python 3.9.0)
Hello from the pygame community. https://www
Ran 8 tests in 0.005s

OK
```