Brendan Jang
CS 362
Assignment 3a

Documentation

In this assignment, I wrote unit tests to evaluate and verify the performance of the Python code in the Dominion Python program. This code is split into three Python Unit Test modules:

- TestAction_card.py – Unit Tests for Action_card class
- TestPlayer.py – Unit Tests for Player class
- TestGameOver – UnitTests for gameover function

I. Action_card class Unit Tests

a. test_initialization

This code tests the initialization function of the Action_card class. It runs a number of randomized iterations in which it creates an Action_card object by calling the Action_card constructor with randomized parameters. It then checks that the values in the Action_card object are set correctly by comparing them to the object's parameters to verify that they have been set properly.

b. test_use

This code tests the use function of the Action_card class. It runs a number of randomized iterations in which it creates a Player object by calling the Player constructor. It then creates an Action_card object by calling the Action_Card constructor with randomized parameters. The card is added to the player's hand, and the use function of the Action_card object is called with the Player object passed as a parameter. The code checks that the card is then removed from the player's hand and is correctly passed to the played attribute of the Player object.

c. test_augment

This code tests the augment function of the Action_card class. It runs a number of randomized iteractions in which it creates a Player object by calling the Player constructor. It then creates an Action_card by calling the Action_card constructor with randomized parameters. The augment method of the Action_card object is called with the Player objected passed as a parameter. The code checks that the Player attributes are then updated correctly, checking the values of actions, buys, purse, and the size of the Player hand.

II. Player class Unit Tests

a. test_action_balance

This code tests the action_balance function of the Player class. It begins by constructing a Player object and verifying that the function initially returns zero for the starting action balance. It then runs a number of randomized iterations in which it creates an Action_card object by calling the Action_card constructor with randomized parameters. These cards are added to the Player object's deck. The balance of the Player is calculated and compared against the value returned by the action_balance function to verify functionality.

b. test_calcpoints

This code tests the calcpoints function of the Player class. It begins by constructing a Player object and verifying that the function initially returns 3 for the starting number of points. It then runs a number of randomized iterations in which it creates a Card object by calling the Card constructor with randomized victory points. These cards are added to the Player object's deck. The number of points of the Player is calculated and compared against the value returned by the calcpoints function to verify functionality. Additionally, the code then runs a number of randomized iterations in which it creates a Gardens object by calling the Gardens constructor. These cards are added to the Player object's deck. The number of points of the Player is again calculated and compared against the value returned by the calcpoints function to verify functionality.

c. test_draw

This code tests the draw function of the Player class. It begins by constructing a Player object. It then calls the draw function of the Player object until the Deck is empty. For each card drawn, it verifies that the size of the deck decreases by 1 and that the card is transferred to the player hand. The code next constructs a new Player object and an empty list: dest It then calls the draw function of the Player object with argument dest until the Deck is empty. For each card drawn, it verifies that the size of the deck decreases by 1 and that the card is transferred to the dest list.

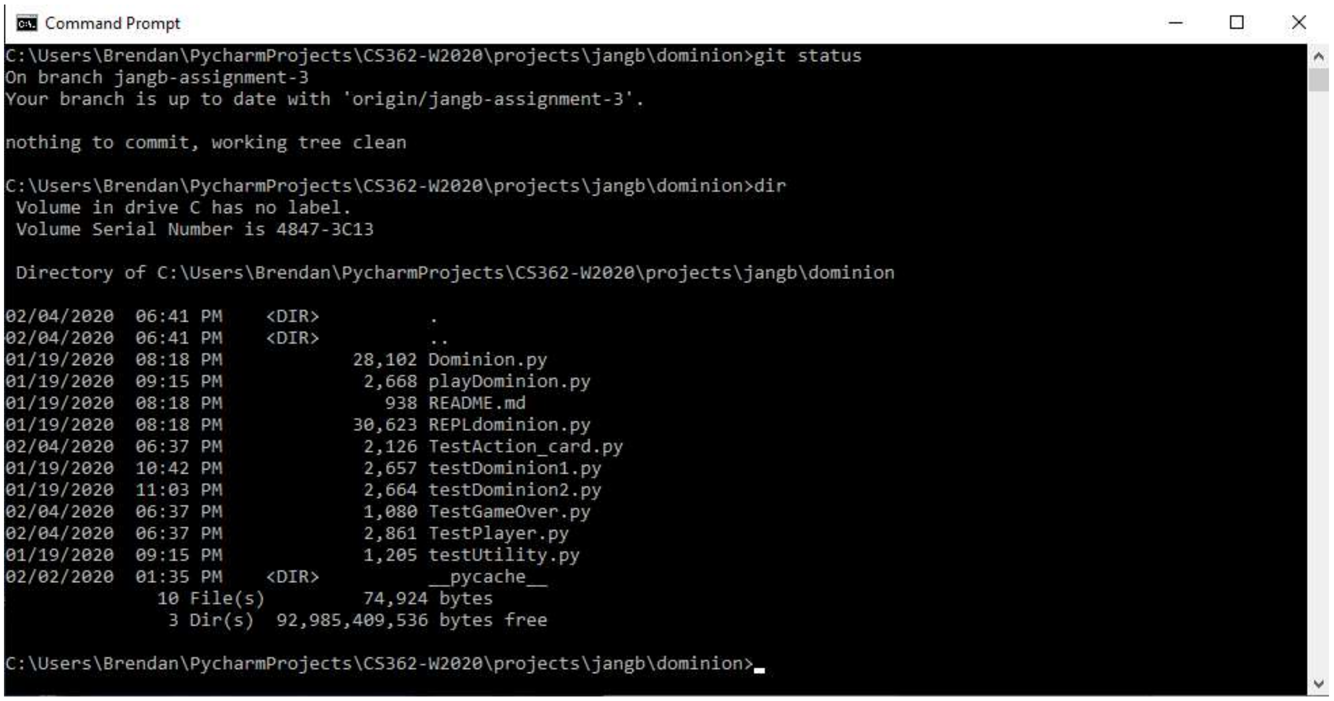
d. test_cardsummary

This code tests the cardsummary function of the Player class. It begins by constructing a Player object and verifying that the initial dictionary returned by the cardsummary method produces a dictionary with the correct default values. It then runs a number of randomized iterations in which it creates a Card object by calling the Card constructor with randomized victory points. These cards are added to the Player object's deck. The code calculates the expected dictionary output and calls the cardsummary function to verify that the function produces the correct desired output.

III. Gameover function Unit Tests

a. test_gameover

This code tests the gameover function in the Dominion module. It begins by creating the supply dictionary with a randomized number of elements: Copper, Silver, Gold, Estate, Duchy, Province, and Curse. These are randomly created such that it could either be possible that the game is over or that the game is not over. The code then calculates the outcome to determine if the game should be over or not, to determine the expected value. The gameover function is then called, and its output is compared against the expected output to verify functionality.



Command Prompt

```
C:\Users\Brendan\PycharmProjects\CS362-W2020\projects\jangb\dominion>git status
On branch jangb-assignment-3
Your branch is up to date with 'origin/jangb-assignment-3'.

nothing to commit, working tree clean

C:\Users\Brendan\PycharmProjects\CS362-W2020\projects\jangb\dominion>dir
 Volume in drive C has no label.
 Volume Serial Number is 4847-3C13

 Directory of C:\Users\Brendan\PycharmProjects\CS362-W2020\projects\jangb\dominion

02/04/2020  06:41 PM    <DIR>          .
02/04/2020  06:41 PM    <DIR>          ..
01/19/2020  08:18 PM            28,102 Dominion.py
01/19/2020  09:15 PM             2,668 playDominion.py
01/19/2020  08:18 PM               938 README.md
01/19/2020  08:18 PM            30,623 REPLdominion.py
02/04/2020  06:37 PM             2,126 TestAction_card.py
01/19/2020  10:42 PM             2,657 testDominion1.py
01/19/2020  11:03 PM             2,664 testDominion2.py
02/04/2020  06:37 PM             1,080 TestGameOver.py
02/04/2020  06:37 PM             2,861 TestPlayer.py
01/19/2020  09:15 PM             1,205 testUtility.py
02/02/2020  01:35 PM    <DIR>          __pycache__
              10 File(s)         74,924 bytes
               3 Dir(s)  92,985,409,536 bytes free

C:\Users\Brendan\PycharmProjects\CS362-W2020\projects\jangb\dominion>
```