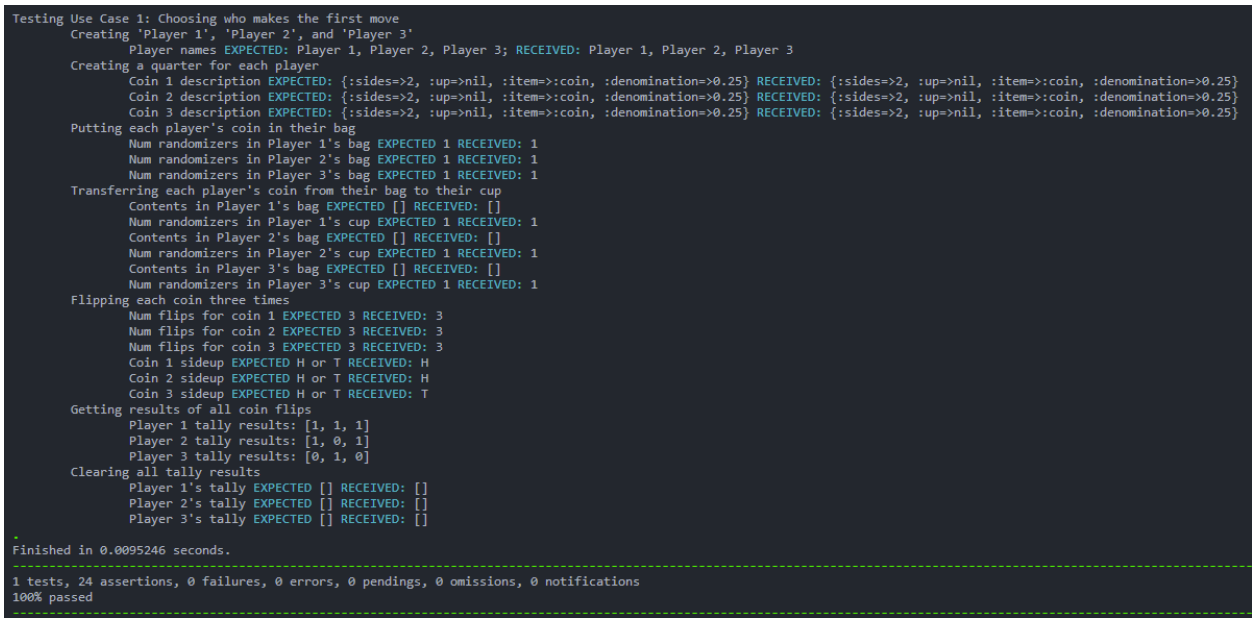
CIS3260 (Software Design IV)
Assignment 1 - Tests
Rabia Qureshi (1046427)
Due date: October 4, 2022 @ 11:30am

Table of Contents

Use Case Tests

Test 1 (UC1): Choosing Who Makes the First Move

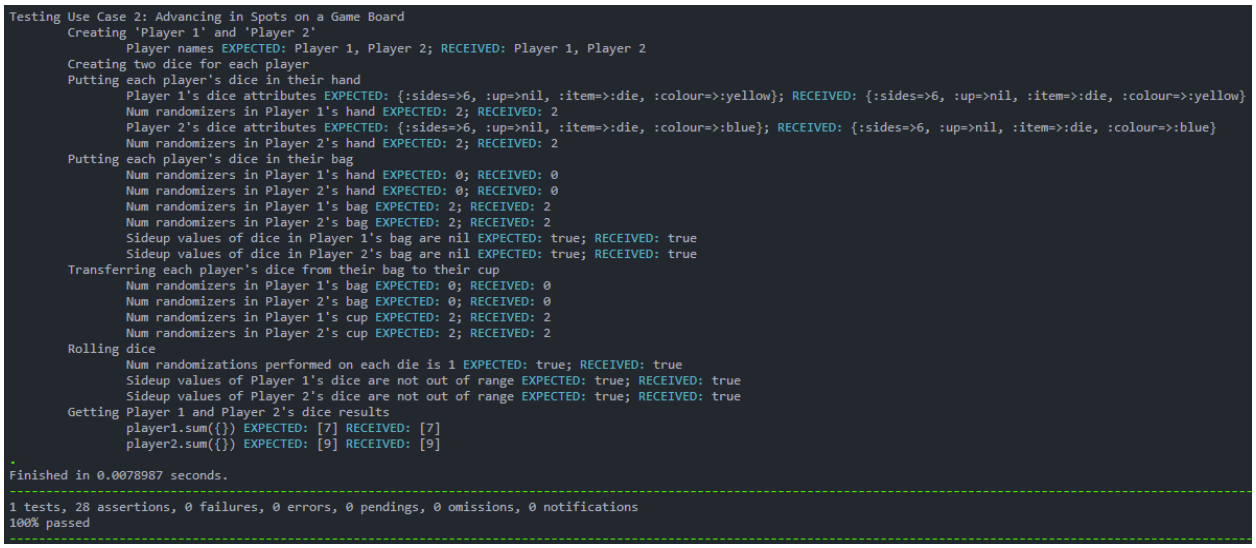


Purpose: Test whether API users can create players and coins, store those coins in players' bags, transfer those coins from bags to cups, and "throw" those cups multiple times to retrieve the tally results of those throws based on a given description.

Methods used:

- Player.store()
- RandomizerContainer.count()
- Player.load()
    - Calls Cup.load() and RandomizerContainer.select()
        - Cup.load() makes use of RandomizerContainer.move_all()
- Player.throw()
    - Calls Cup.throw()
        - Makes use of Randomizer.randomize()
- Randomizer.calls()
- Coin.sideup()
    - Inherits from Randomizer.result()
- Player.tally()
    - Calls Results.description()
- Player.clear()

Test 2 (UC2): Advancing in Spots on a Game Board

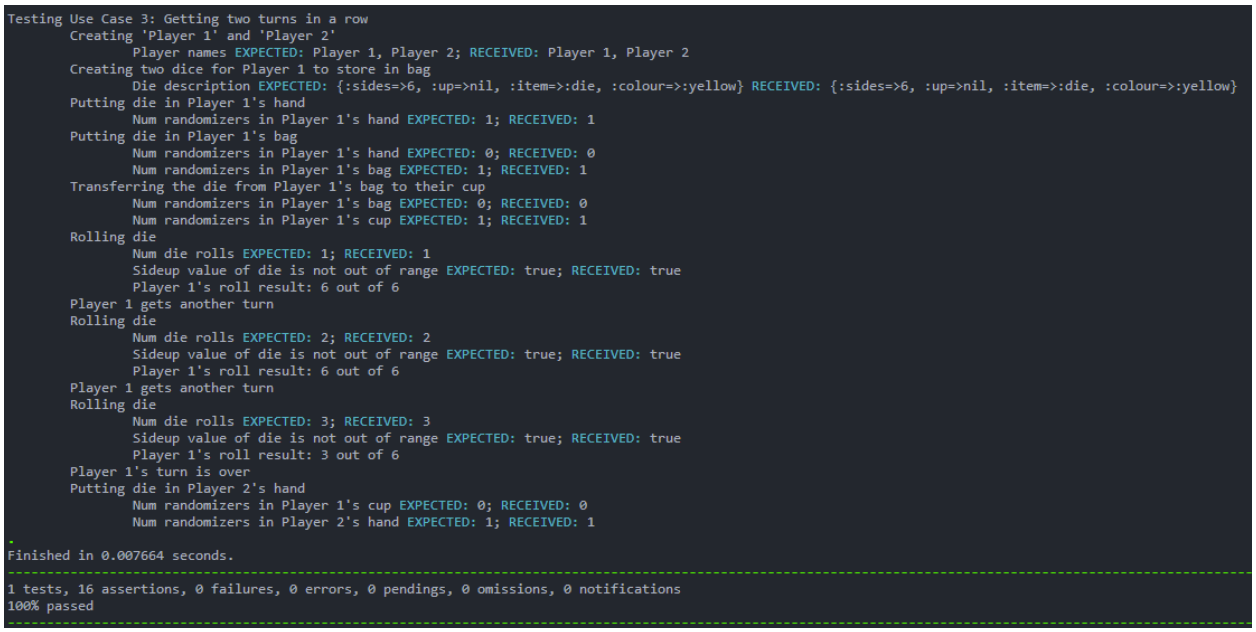


Purpose: Test whether API users can create players and dice, store those dice in their hands and transfer them to their bags and finally to their cups, “throw” those cups, and retrieve the sum of the side-up values of their thrown dice after one throw.

Methods used:

- RandomizerContainer.store_all()
- Player.move_all()
    - Bag.move_all()
        - Inherits from RandomizerContainer.move_all() and calls Randomizer.reset()
- RandomizerContainer.count()
- Die.sideup()
    - Inherits from Randomizer.result()
- Player.load()
    - Calls Cup.load() and RandomizerContainer.select()
        - Cup.load() calls RandomizerContainer.move_all()
- Player.throw()
    - Calls Cup.throw()
        - Calls Randomizer.randomize()
- Player.sum()

Test 3 (UC3): Getting Two Turns in a Row

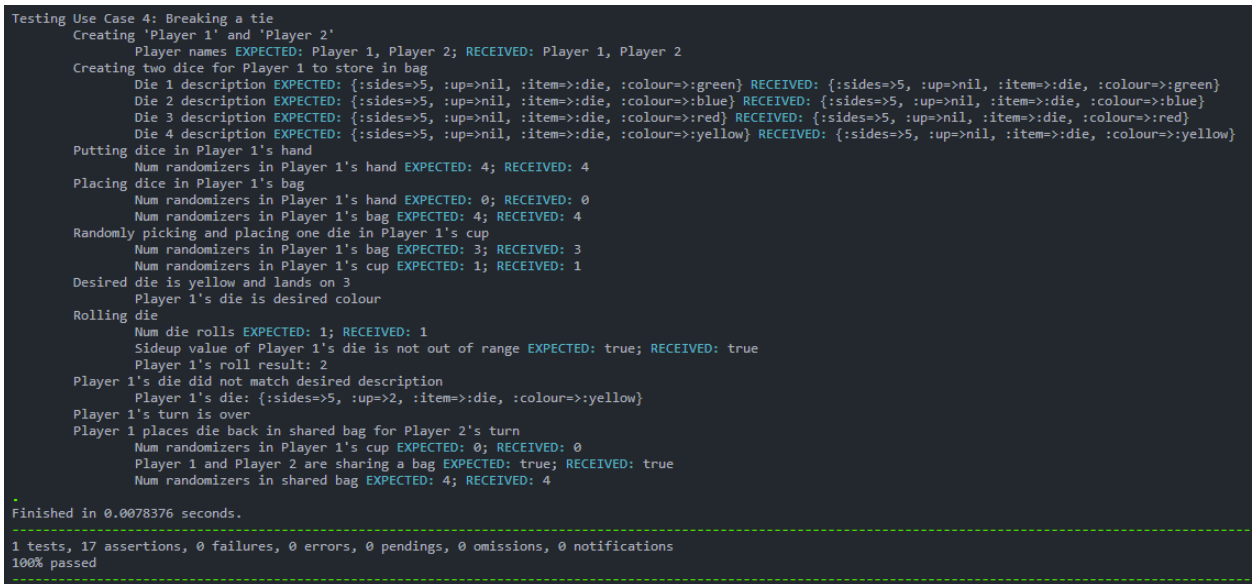


Purpose: Test whether API users can create players and a shared die that is rolled and passed onto the opponent if the player does not get the highest possible side-up value.

Methods used:

- RandomizerContainer.store()
- Player.move_all()
    - Bag.move_all()
        - Inherits from RandomizerContainer.move_all() and calls Randomizer.reset()
- RandomizerContainer.count()
- Player.load()
    - Calls Cup.load() and RandomizerContainer.select()
        - Cup.load() calls RandomizerContainer.move_all()
- Player.throw()
    - Calls Cup.throw()
        - Calls Randomizer.randomize()
- Randomizer.calls()
- Die.sideup()
    - Inherits from Randomizer.result()
- Cup.empty()
- Hand.empty()
- Player.store()

Test 4 (UC4): Breaking a “Tie”



Purpose: Test whether API users can create players with a shared bag that contains dice, one of which is rolled by a player in hopes of meeting certain criteria to break a tie and gain an advantage over the opponent.

Methods used:

- RandomizerContainer.store_all()
- RandomizerContainer.count()
- Player.move_all()
    - Bag.move_all()
        - Inherits from RandomizerContainer.move_all() and calls Randomizer.reset()
- Player.load()
    - Calls Cup.load() and RandomizerContainer.select()
        - Cup.load() calls RandomizerContainer.move_all()
- Player.throw()
    - Calls Cup.throw()
        - Calls Randomizer.randomize()
- Randomizer.calls()
- Die.sideup()
- Die.colour()
- Player.replace()

Untested Method Tests

Test 5: Coin Methods

```
Testing unused Coin class methods
        Before flipping coin...
                coin.sideup() EXPECTED: nil; RECEIVED: nil
                coin.calls() EXPECTED: 0; RECEIVED: 0
                coin.denomination() EXPECTED: 0.25; RECEIVED: 0.25
                {:sides=>2, :up=>nil, :item=>:coin, :denomination=>0.25}
        Flipping coin...
                coin.sideup() EXPECTED: H or T; RECEIVED: T
                coin.calls() EXPECTED: 1; RECEIVED: 1
                {:sides=>2, :up=>:T, :item=>:coin, :denomination=>0.25}
.
Finished in 0.0019713 seconds.
--------------------------------------------------------------------------------
1 tests, 5 assertions, 0 failures, 0 errors, 0 pendings, 0 omissions, 0 notifications
100% passed
--------------------------------------------------------------------------------
```

Untested methods used: Coin.denomination(), Coin.flip()

Test 6: Die Methods

```
Testing unused Die class methods
        Before rolling die...
                die.sideup() EXPECTED: nil; RECEIVED: nil
                die.sides() EXPECTED: 6; RECEIVED: 6
                die.colour() EXPECTED: :red; RECEIVED: red
                die.calls() EXPECTED: 0; RECEIVED: 0
                {:sides=>6, :up=>nil, :item=>:die, :colour=>:red}
        Rolling die...
                die.sideup() EXPECTED: 1-6; RECEIVED: 3
                die.calls() EXPECTED: 1; RECEIVED: 1
                {:sides=>6, :up=>3, :item=>:die, :colour=>:red}
.
Finished in 0.0031464 seconds.
--------------------------------------------------------------------------------
1 tests, 6 assertions, 0 failures, 0 errors, 0 pendings, 0 omissions, 0 notifications
100% passed
--------------------------------------------------------------------------------
```

Untested methods used: Die.sides(), Die.roll()

Test 7: Player Methods

```
Testing Player class methods
        Creating a player
        Creating two quarters for player
        Putting coins in Player's bag
        Transferring Player's coins from their bag to their cup
        Flipping each coin three times
                Flipping coins...
                {:sides=>2, :up=>:H, :item=>:coin, :denomination=>0.25}
                {:sides=>2, :up=>:T, :item=>:coin, :denomination=>0.25}
                Flipping coins...
                {:sides=>2, :up=>:T, :item=>:coin, :denomination=>0.25}
                {:sides=>2, :up=>:T, :item=>:coin, :denomination=>0.25}
                Flipping coins...
                {:sides=>2, :up=>:T, :item=>:coin, :denomination=>0.25}
                {:sides=>2, :up=>:T, :item=>:coin, :denomination=>0.25}
        Getting results of all coin flips
                player.results({}, 0) EXPECTED: [:T, :T] RECEIVED: [:T, :T]
.
Finished in 0.0042931 seconds.
-------------------------------------------------------------------------------------------------
1 tests, 1 assertions, 0 failures, 0 errors, 0 pendings, 0 omissions, 0 notifications
100% passed
-------------------------------------------------------------------------------------------------
```

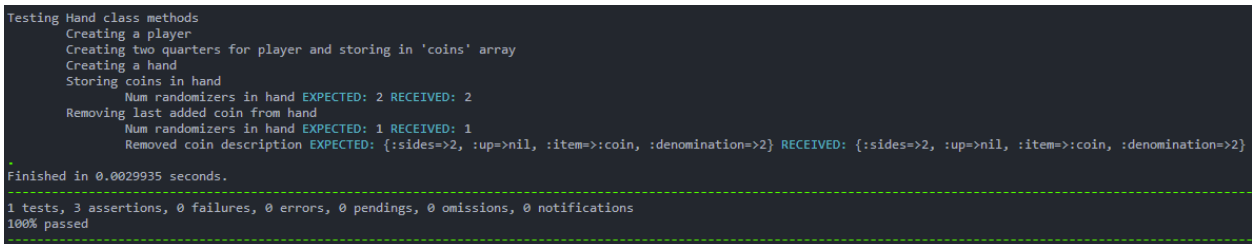
Untested methods used: Player.results()

Test 8: Bag Methods

```
Testing Bag class methods
        Creating a player
        Creating two quarters for player and storing in 'coins' array
        Putting coins in Player's bag
                Num randomizers in bag EXPECTED: 2 RECEIVED: 2
                Num randomizers in 'coins' array EXPECTED: 2 RECEIVED: 2
        Emptying Player's bag
                Num randomizers in bag EXPECTED: 0 RECEIVED: 0
.
Finished in 0.0029274 seconds.
-------------------------------------------------------------------------------------------------
1 tests, 3 assertions, 0 failures, 0 errors, 0 pendings, 0 omissions, 0 notifications
100% passed
-------------------------------------------------------------------------------------------------
```

Untested methods used: Bag.store_all(), Bag.empty()

Test 9: Hand Methods



Untested methods used: Hand.next()