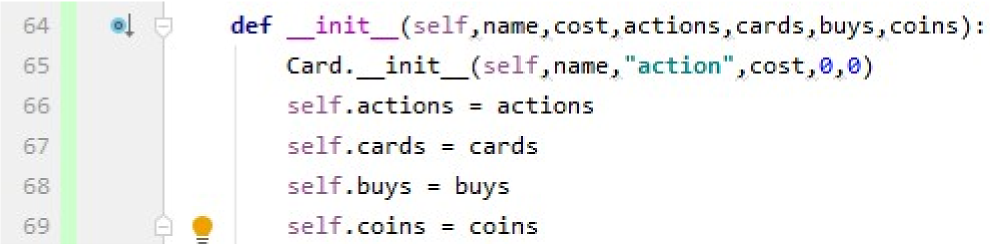
Brendan Jang
CS 362
Assignment 3b

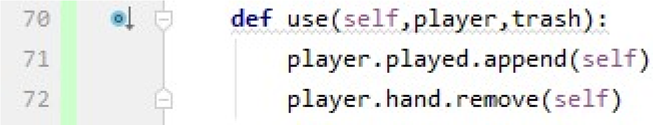
Code Coverage

Testing Action_card initialization (100% branch coverage)



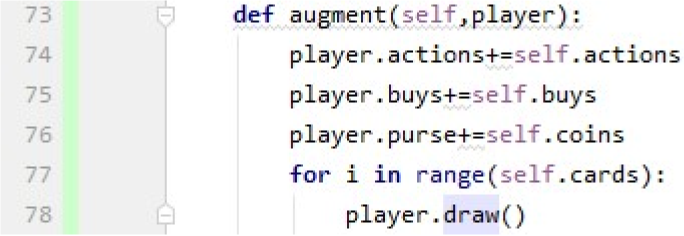
```
def __init__(self,name,cost,actions,cards,buys,coins):
    Card.__init__(self,name,"action",cost,0,0)
    self.actions = actions
    self.cards = cards
    self.buys = buys
    self.coins = coins
```

Testing Action_card use function (100% branch coverage)



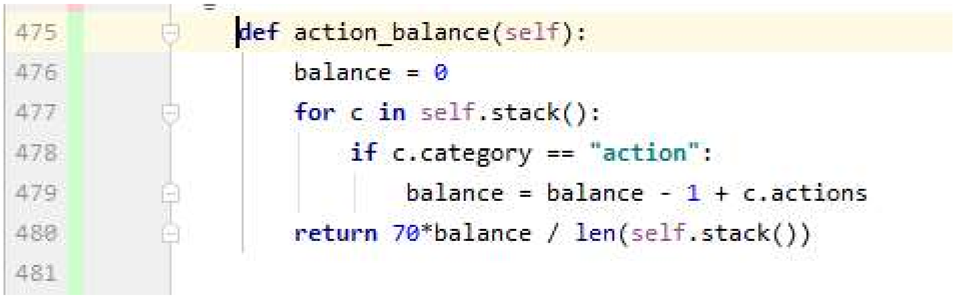
```
def use(self,player,trash):
    player.played.append(self)
    player.hand.remove(self)
```

Test Action_card augment function (100% branch coverage)



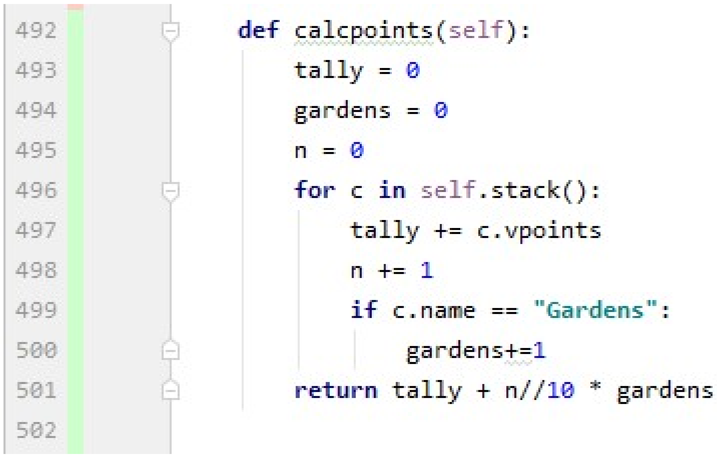
```
def augment(self,player):
    player.actions+=self.actions
    player.buys+=self.buys
    player.purse+=self.coins
    for i in range(self.cards):
        player.draw()
```

Test Player action_balance function (100% branch coverage)



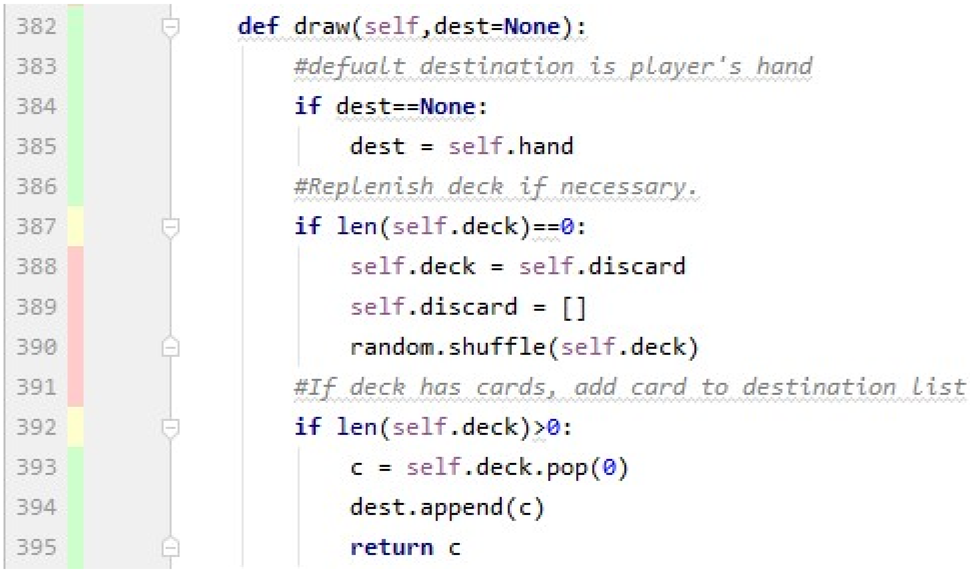
```python
    def action_balance(self):
        balance = 0
        for c in self.stack():
            if c.category == "action":
                balance = balance - 1 + c.actions
        return 70*balance / len(self.stack())
```

Test Player calcpoints function (100% branch coverage)



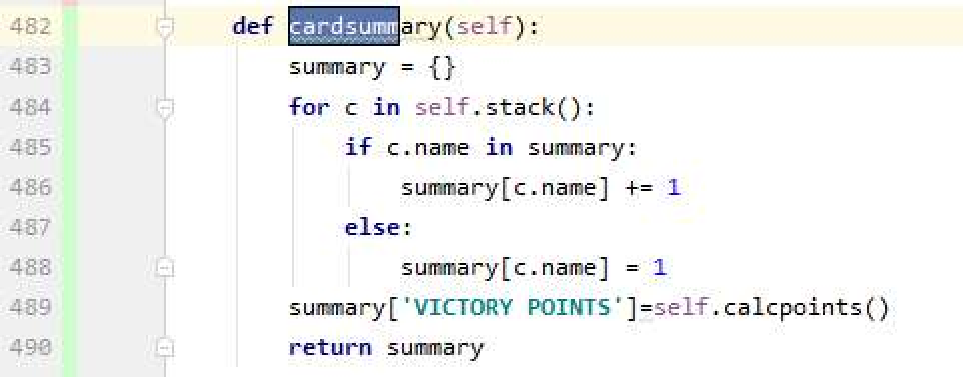
```python
    def calcpoints(self):
        tally = 0
        gardens = 0
        n = 0
        for c in self.stack():
            tally += c.vpoints
            n += 1
            if c.name == "Gardens":
                gardens+=1
        return tally + n//10 * gardens
```

Test Player draw function (50% branch coverage). My tests for the draw function did not cover the branch where the code checks for an empty deck. In order to cover this branch in my tests, I need to add a test case where the deck is reset to an empty list, the draw function is called, and finally I need to assert my assumptions.



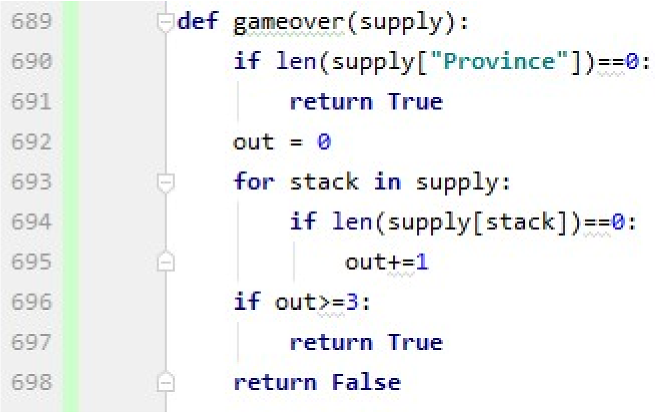
```python
def draw(self,dest=None):
    #defualt destination is player's hand
    if dest==None:
        dest = self.hand
    #Replenish deck if necessary.
    if len(self.deck)==0:
        self.deck = self.discard
        self.discard = []
        random.shuffle(self.deck)
    #If deck has cards, add card to destination list
    if len(self.deck)>0:
        c = self.deck.pop(0)
        dest.append(c)
        return c
```

Test Player cardsummary function (100% branch coverage)



```python
def cardsummary(self):
    summary = {}
    for c in self.stack():
        if c.name in summary:
            summary[c.name] += 1
        else:
            summary[c.name] = 1
    summary['VICTORY POINTS']=self.calcpoints()
    return summary
```

Test gameOver function (100% branch coverage)



```python
def gameover(supply):
    if len(supply["Province"])==0:
        return True
    out = 0
    for stack in supply:
        if len(supply[stack])==0:
            out+=1
    if out>=3:
        return True
    return False
```