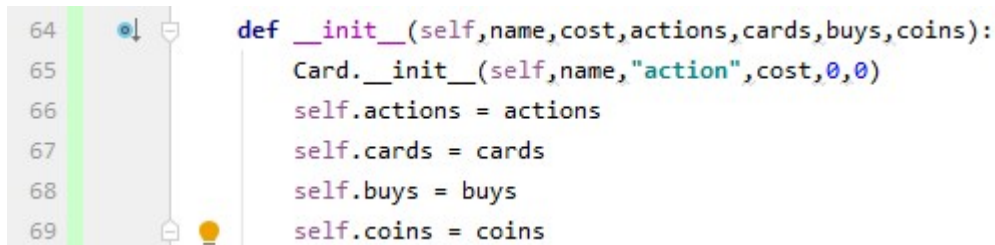


Code Coverage

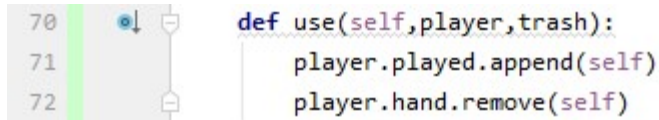
Testing Action_card initialization (100% branch coverage)



```
64 def __init__(self, name, cost, actions, cards, buys, coins):  
65     Card.__init__(self, name, "action", cost, 0, 0)  
66     self.actions = actions  
67     self.cards = cards  
68     self.buys = buys  
69     self.coins = coins
```

The screenshot shows a code editor with line numbers 64 to 69 on the left. A green vertical bar highlights the entire code block. A blue circle with a white arrow points to line 64. A yellow lightbulb icon is at the bottom right of the code block.

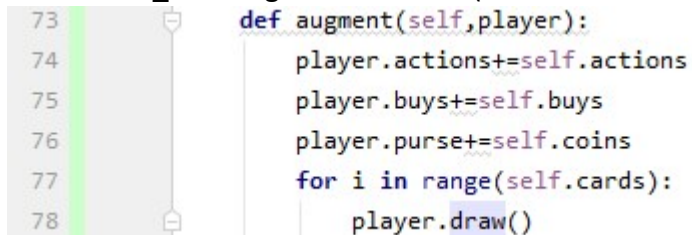
Testing Action_card use function (100% branch coverage)



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

The screenshot shows a code editor with line numbers 70 to 72 on the left. A green vertical bar highlights the entire code block. A blue circle with a white arrow points to line 70. A yellow lightbulb icon is at the bottom right of the code block.

Test Action_card augment function (100% branch coverage)



```
73 def augment(self, player):  
74     player.actions+=self.actions  
75     player.buys+=self.buys  
76     player.purse+=self.coins  
77     for i in range(self.cards):  
78         player.draw()
```

The screenshot shows a code editor with line numbers 73 to 78 on the left. A green vertical bar highlights the entire code block. A blue circle with a white arrow points to line 73. A yellow lightbulb icon is at the bottom right of the code block.

Test Player action_balance function (100% branch coverage)

```
475 def action_balance(self):
476     balance = 0
477     for c in self.stack():
478         if c.category == "action":
479             balance = balance - 1 + c.actions
480     return 70*balance / len(self.stack())
481
```

Test Player calcpoints function (100% branch coverage)

```
492 def calcpoints(self):
493     tally = 0
494     gardens = 0
495     n = 0
496     for c in self.stack():
497         tally += c.vpoints
498         n += 1
499         if c.name == "Gardens":
500             gardens+=1
501     return tally + n//10 * gardens
502
```

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.

```
382 def draw(self, dest=None):
383     #default destination is player's hand
384     if dest==None:
385         dest = self.hand
386     #Replenish deck if necessary.
387     if len(self.deck)==0:
388         self.deck = self.discard
389         self.discard = []
390         random.shuffle(self.deck)
391     #If deck has cards, add card to destination list
392     if len(self.deck)>0:
393         c = self.deck.pop(0)
394         dest.append(c)
395     return c
```

Test Player cardsummary function (100% branch coverage)

```
482 def cardsummary(self):
483     summary = {}
484     for c in self.stack():
485         if c.name in summary:
486             summary[c.name] += 1
487         else:
488             summary[c.name] = 1
489     summary['VICTORY POINTS']=self.calcpoints()
490     return summary
```

Test gameOver function (100% branch coverage)

```
689 def gameOver(supply):
690     if len(supply["Province"])==0:
691         return True
692     out = 0
693     for stack in supply:
694         if len(supply[stack])==0:
695             out+=1
696     if out>=3:
697         return True
698     return False
```