

Lab 2 Task 3 Solution

3) Become familiar with function *main*

1. Where is a *NumberGame* constructed?

- By observation, we can conclude a *NumberGame* constructed inside function *main*

```
1      def main() -> None:
2          ...
3          while True:
4              g = NumberGame(goal, minimum, maximum, (p1, p2)) #<-
Here!!
5              winner = g.play()
6              print(f'And {winner} is the winner!!!')
7              print(p1)
8              print(p2)
9              again = input('Again? (y/n) ')
10             if again != 'y':
11                 return
12
13
```

2. This function calls *g.play* repeatedly in a loop. What about the game can change each time *g.play* is called: the goal, the min or max move, the players, the moves?

- By observation, we can conclude that
 1. the goal doesn't change
 2. the min or max move don't change
 3. the current player change as a result of whose_turn method.

```
1      def play(self) -> str:
2          ...
3          while self.current < self.goal:
4              self.play_one_turn() # <- In here
5          ...
6          winner = self.whose_turn(self.turn - 1)
7          return winner.namePlayers
8
9      def play_one_turn(self) -> None:
10         ...
```

```

11         next_player = self.whose_turn(self.turn) # <-
    Here!!
12         amount = next_player.move(
13             self.current,
14             self.min_step,
15             self.max_step,
16             self.goal
17         )
18         self.current += amount
19         self.turn += 1
20
21         print(f'{next_player.name} moves {amount}.')
22         print(f'Total is now {self.current}.')
23
24
25     def whose_turn(self, turn: int) -> Player:
26         ...
27         if turn % 2 == 0:
28             return self.players[0]
29         else:
30             return self.players[1]
31
32

```

4. the move changes by the *move* method in *play_one_turn*.

```

1     def play(self) -> str:
2         ...
3         while self.current < self.goal:
4             self.play_one_turn()
5         ...
6         winner = self.whose_turn(self.turn - 1)
7         return winner.namePlayers
8
9     def play_one_turn(self) -> None:
10        ...
11        next_player = self.whose_turn(self.turn)
12        amount = next_player.move( # <- Here!!
13            self.current,
14            self.min_step,
15            self.max_step,
16            self.goal
17        )
18        self.current += amount
19        self.turn += 1
20
21        print(f'{next_player.name} moves {amount}.')
22        print(f'Total is now {self.current}.')
23
24

```

3. List all the places in this function where a *Player* is stored, an instance attribute of *Player* is accessed or set, or a method is called on a *Player*.

Rough Work:

We need to find all places in this function where *Player* is stored, where an instance attribute of *Player* is accessed or set, or where a method is called on a *Player*.

1. Find where *Player* is stored.

First, we need to find where *Player* is stored.

Because we know from code that the third argument in *NumberGame* is of type `Tuple[Player, Player]`, we can conclude *Player* is stored inside variables *p1* and *p2*

```
1      def main() -> None:
2          """Play multiple rounds of a NumberGame based
3          on user input settings.
4          """
5          goal = int(input('Enter goal amount: '))
6          minimum = int(input('Enter minimum move: '))
7          maximum = int(input('Enter maximum move: '))
8          p1 = make_player('p1') # <- Here!!
9          p2 = make_player('p2') # <- Here!!
10         while True:
11             g = NumberGame(goal, minimum, maximum, (p1,
12             p2)) # <- Here!!
13             winner = g.play()
14             print(f'And {winner} is the winner!!!')
15             print(p1)
16             print(p2)
17             again = input('Again? (y/n) ')
18             if again != 'y':
19                 return
```

2. Find where the instance attribute of *Player* is accessed or set.

Second, we need to find where the instance attribute of *Player* is accessed or set.

By observation, we can conclude there are two places where one or more instance attributes of *Player* is accessed or set.

The first one is inside *play* method.

```
1      class NumberGame:
2          ...
3          def play(self) -> str:
4              ...
5              winner = self.whose_turn(self.turn - 1)
6              return winner.name # <- Here!!
7
```

The second one is inside *play_one_turn* method.

```
1      class NumberGame:
2          ...
3          def play_one_turn(self) -> None:
4              ...
5              print(f'{next_player.name} moves {amount}.')
6              # <- Here!!
7              print(f'Total is now {self.current}.')
```

3. Find where a method of *Player* is called.

Finally, we need to find where method of *Player* is called.

By observation, we can conclude one method of *Player* is used, and it is called inside *play_one_turn* method.

```
1      class NumberGame:
2          ...
3          def play_one_turn(self) -> None:
4              next_player = self.whose_turn(self.turn)
5              amount = next_player.move( # <- Here!!
6                  self.current,
7                  self.min_step,
8                  self.max_step,
9                  self.goal
10             )
11
```