Ingeniería del software

Ejercicio Opcional TDD

Luis Fernando Paz-1567369
Adrián Rodríguez-1428612
Fco. Javier Honrubia -1564963

Contenido

Use Cases

Informacion:

* -> Determina que el código ya pasa ese test.

TO-DO LIST

TESTS UNITARIOS PLAYER:

* playerName init works

* empty init works

* eq operator works

TESTS UNITARIOS GAME:

* Can call game get_score()

* Before the game score is "Love-All"

* Test init

* won_point adds 1 point to the correct player

TESTS INTEGRACION GAME Y PLAYER:

* When player1 wins first point score is "Fifteen-Love"

* When player1 wins first point and player2 wins second point score is "Fifteen-All"

* When player1 wins first and third point and player2 wins second point score is "Thirty-Fifteen"

* When player1 wins second point score is "Thirty-Love"

* When player1 wins third point score is "Forty-Love"

* When player1 wins fourth point score is "Win for player1"

* When player2 wins first point score is "Love-Fifteen"

TESTS EXPLORACIÓN:

* wonpoint on a completed/won game should show an error

* player1 and player2 from game should be different

* create_player() works correctly

* create_game() works correctly

Conjuntos de Test

Para garantizar que el programa realiza correctamente el tanteo, hemos realizado los siguientes tipos de test:

- ***Unitarios*** *para asegurar el correcto funcionamiento de las clases Game y Player.*
- ***Integración*** *para testear la correcta integración entre las clases Game y Player para simular un futuro uso por un usuario final.*
- ***Exploración*** *para detectar y solucionar situaciones extrañas que pueden derivar en errores en la aplicación.*

Resultados del TDD

Los objetos creados con el objetivo de poder realzar el tanteo son las siguientes clases, **Game** Y **Player.**

```
class Game:
    def __init__(self, player_1: Player, player_2: Player):
        self.__player1__ = player_1
        self.__player2__ = player_2
        self.__points_player1__ = 0
        self.__points_player2__ = 0
        self.__combinations__ = [["Love-All", "Love-Fifteen",
"Love-Thirty", "Love-Forty", "Win for player2"],
                                 ["Fifteen-Love", "Fifteen-All",
"Fifteen-Thirty", "Fifteen-Forty", "Win for player2"],
                                 ["Thirty-Love", "Thirty-Fifteen",
"Thirty-All", "Thirty-Forty", "Win for player2"],
                                 ["Forty-Love", "Forty-Fifteen",
"Forty-Thirty", "Deuce", "Advantage player2"],
                                 ["Win for player1", "Win for
player1", "Win for player1", "Advantage player1", "Deuce"]]

    def get_score(self):
        if (self.__points_player1__ + self.__points_player2__)
<= 6:
            return
self.__combinations__[self.__points_player1__][self.__points_p
layer2__]
        else:
            if self.__points_player1__ ==
self.__points_player2__:
                return "Deuce"
            elif (self.__points_player1__ -
self.__points_player2__) == 1:
                return "Advantage player1"
```

```
            elif (self.__points_player1__ -
self.__points_player2__) == -1:
                return "Advantage player2"
            elif (self.__points_player1__ -
self.__points_player2__) == 2:
                return "Win for player1"
            elif (self.__points_player1__ -
self.__points_player2__) == -2:
                return "Win for player2"


    def won_point(self, player_obj: str):
        if (self.get_score() == "Win for player1"):
            print("Could not won_point--> GAME is won for
player1")
        elif (self.get_score() == "Win for player2"):
            print("Could not won_point--> GAME is won for
player2")
        else:
            if self.__player1__.__playerName__ == player_obj:
                self.__points_player1__ =
self.__points_player1__ + 1
            else:
                self.__points_player2__ =
self.__points_player2__ + 1
```

```
class Player:
    def __init__(self, name = None):
        if name is not None:
            self.__playerName__ = name
        else:
            self.__playerName__ = ""

    def __eq__(self, other: str):
        return self.__playerName__ == other.__playerName__
```

Enlace al repositorio de GitHub

https://github.com/es2021uab-1567369/ExerciciTDD.git