**Exercise 3 - A Terminal-based Tutto Game - Testing**

**Testingmethology**

For the Testing of the game, we followed the approach of chapter 5 and created for each Class in the Implementation a TestingCalss named with “Test” as ending. For example, for the class Player we created a PlayerTest class. Each Test Class contains various individual test which can be run as standalone, or one can run the entire test class with all included test.

**Branch coverage**

We were able to achieve a total branch coverage of 63% for our implementation of the Tuttogame and our Testing suite. The branch coverage varied in between the implemented classes, especially in the card classes.

Our achieved branch coverage of 63% was below the required 70%.

**Explanation**

We were able to pinpoint reasons for a lower test coverage. The lower Test coverage can be attributed to two main factors. These being while loops in combination with the random method of the DICE class. Especially in the various Card classes aswell as others (Gamemode) there is a multitude of while(true) loops. These while(true) loops in combination with the Random function that is used in the roll() method of the dice Class to assign a random values to Objects of the Class dice leads to the fact that one does not know what to select in the while loop and exiting it is impossible.

Additionally, to the above stated reasons a multitude of methods in the GameModel Class are declared as private. This makes writing tests that cover all branches even more challenging for said class and further lowered our branch coverage.

Therefore, we were not able to achieve the required branch coverage.