

LUND UNIVERSITY

FACULTY OF ENGINEERING

APPLIED ARTIFICIAL INTELLIGENCE

Assignment 1: The Reversi Game

Author:

Nicolas BURGUER

Juan Gabriel MARTÍNEZ PERIS



LUND
UNIVERSITY

Contents

| | | |
|----------|---------------------------------------|----------|
| 1 | Introduction | 2 |
| 2 | Design | 2 |
| 2.1 | UML Diagram | 3 |
| 2.2 | Board.java | 3 |
| 2.3 | Player.java | 4 |
| 2.4 | ArtificialIntelligence.java | 4 |
| 2.5 | HumanPlayer.java | 4 |
| 2.6 | Game.java | 4 |
| 3 | Execution | 4 |
| 4 | Conclusion | 4 |
| 5 | References | 4 |

1 Introduction

Artificial Intelligence(AI) and the videogame industry have a close relationship. Programmers working in that industry try to find better algorithms capable of winning a human player. There are different techniques depending on the type of game, but usually behind all of them there is a search algorithm that allows the computer to take the best decision depending on the state of the game.

The aim of this report is to explain our implementation of the *Reversi/Othello* game, a popular board game created in 1883. It is played on an 8x8 board by two players, one of them will use dark disks and the other light ones. For the *Othello* variant, the game starts with four discs placed in a square in the middle of the grid, two facing light side up, two pieces with the dark side up, with same-colored disks on a diagonal with each other. In order to win the game, you must have more discs of your color than your opponent. You can achieve this by turning over any discs from your opponent that are in a straight line and bounded by the disk you have just placed and another of your disks.

In the following sections are described our design with a UML diagram and the different Java classes with their methods. Furthermore, you will have explained how to execute our program and where to find it. The last section has the conclusions that we have reached after finishing the assignment.

2 Design

In this section we have described the design of our implementation. First, you will find the UML diagram with the classes and methods that we thought were necessary for the assignment. Then, we have highlighted some parts of the code in each class that are relevant in order to understand how the program works.

2.1 UML Diagram

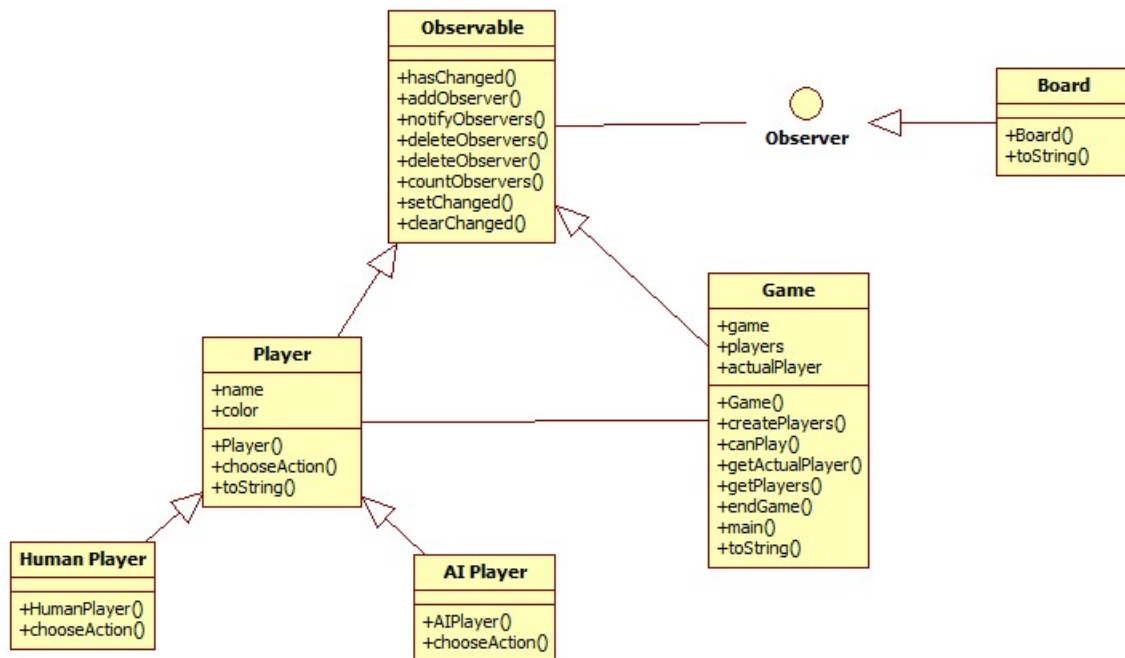


Figure 1: Othello's UML Diagram

Figure 1 shows the UML diagram of our *Othello*'s implementation. As shown, we have 5 classes: **Board**, **Game**, **Player**, **HumanPlayer** and **ArtificialIntelligence**. We will describe the purpose of each of them briefly as we will highlight relevant pieces of code in the following subsections:

- **Board**
- **Game**
- **Player**
- **HumanPlayer**
- **ArtificialIntelligence**

2.2 Board.java

```
System.out.println("Hello_World");
```

2.3 Player.java

2.4 ArtificialIntelligence.java

2.5 HumanPlayer.java

2.6 Game.java

3 Execution

4 Conclusion

5 References

[1] Reversi - Wikipedia entry: <http://en.wikipedia.org/wiki/Reversi>

[2] Russel, Stuart; Norvig, Peter, *Artificial Intelligence, A Modern Approach*, Pearson Education, 3rd Edition.