**Software Development Project**

**Report**

|  |  |
| --- | --- |
| **PROJECT TEAM** | |
| **Student No.** | **Student Names** |
| **16138457** | **Lorenzo Cipriani** |
| **16134613** | **Nassima Kara** |
| **16135091** | **Sebastien Zekpa** |

**Morra (Evens or Odds)**

**22nd April 2017**

# Introduction

The main goal of the team was to use, for the development of the project, all and only the topics covered during the lectures. Even if the game itself is not so complicated to implement in java, the team tried to show some key aspects learned; e,g:

* Instantiable classes (the main class is just an entry point then the flow of the program is managed by the GameController, this instantiates the 2 players of the game and manage their states)
* Use of design pattern (GameController is a lightweight implementation of the Chain Of Responsibility pattern)
* Incapsulation (all attributes defined as private/protected and only getters&setters are used to access the state of the objects)
* Validation of user’s input
* Conditional statements (if and switch)
* Loops (while)
* Abstract classes and inheritance (Player is an abstract class extended by Computer and Human)
* Method overloading (e.g.: Player.hasOdds() & Player.hasOdds(boolean).)
* Method overriding (showFingers() in Player is an abstract method that must be overridden by Human and Computer for their own different behaviours when that method is called.
* Use of standard developers tools: Eclipse IDE, Papyrus UML, Git

# Diagrams

* **Overview of the game (class diagram)**

