Conor Hartigan Object Orientated Programming Project Documentation

## Requirement Specification

For my OOP project I made the card game ‘Lives’. This is an overview of the rules of the game:

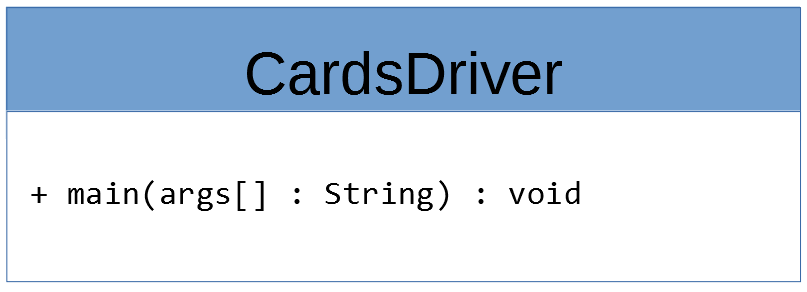
* There are two players, the user and the CPU.
* Both the player and the CPU (player 2) start with 7 cards.
* Players take turns placing their card.
* If the player gets more than 20 cards they lose.
* The card played
* To win one player must be out of cards and call ‘Last Card’ or the other player must have more than 20 cards.
* There are function cards, Two, Jack, King Ace and Queen, the Two +2’s the next player, the Jack skips the next player, the King reverses the direction, the Ace is the wildcard, it changes suits to whatever the player selects, the Queen makes the next player pick up 4 cards.

The program begins by running the CardsDriver class, the CardsDriver class plays the song until the program is closed and opens the MainMenuGUI. From here you can select ‘Play’, ‘Statistics’ or ‘Exit’. The ‘Play’ button runs the LivesGUI, the ‘Statistics’ button opens the data file so the player can see the number of wins and loses they’ve had, and the ‘Exit’ button exits the program.

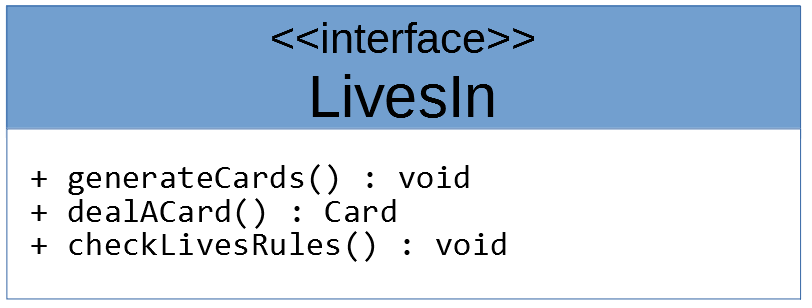
When the player selects ‘Play’ the LivesGUI constructor generates the playing area and the game starts. When the user presses the ‘Statistics’ button it calls the readInfo() method. The readInfo() method opens the save data from the playerstats.dat file. Selecting stats will show the user the current values for wins and loses. The ‘Exit’ button closes the program.

When the player selects play the LivesGUI constructor generates the playing field which consist of 3 panels, one to contain the player cards, one to contain the player 2 cards and one to display the last played card, there is 20 JButtons for each player cards, the game starts with 7 visible and with cards on them (using the dealACard method). The stackOfCards panel has a random card on top at random, after that the card on top is the card previously played by either player or player 2.

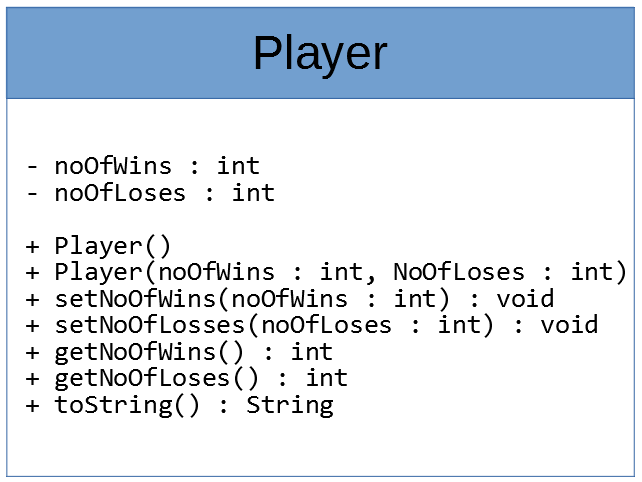
## UML Diagram - CardsDriver class



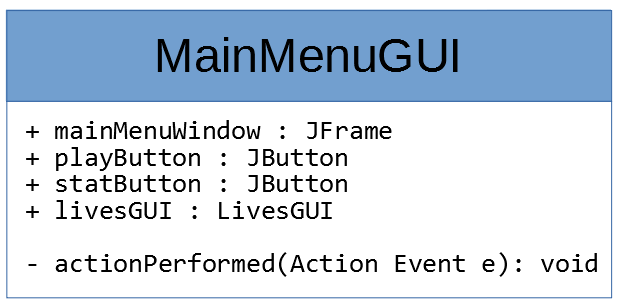
## UML Diagram - LivesIn interface



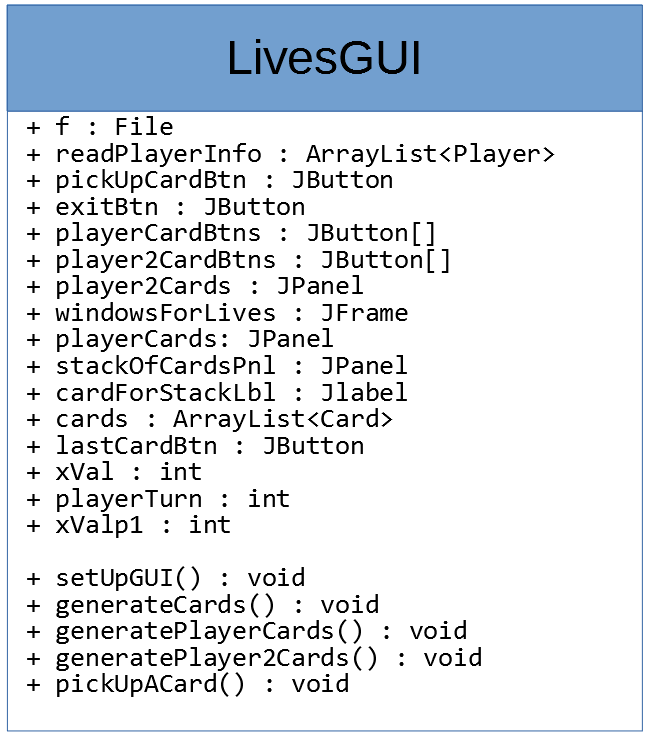
## UML Diagram - Player class



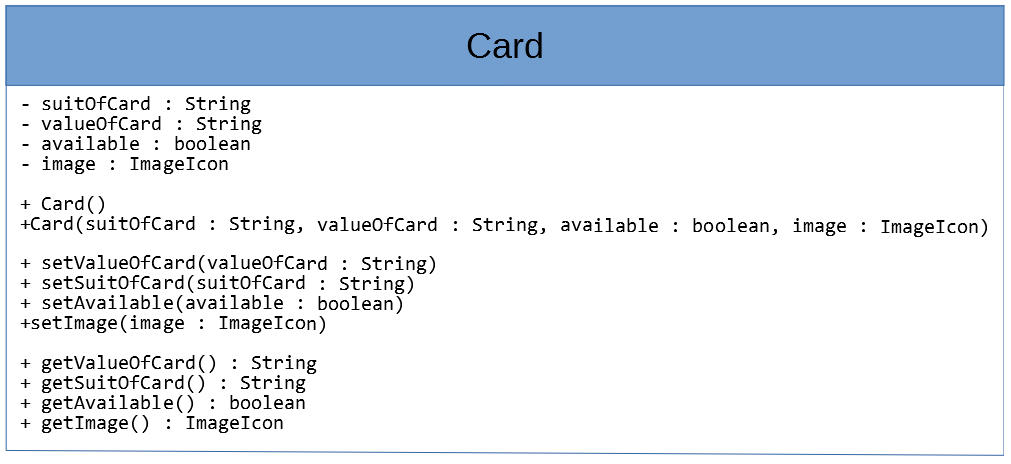
## UML Diagram – MainMenuGUI class



## UML Diagram – LivesGUI Class



## UML Diagram - Card Class



## UML Diagram

