# Cours Projet de Programmation

# Fiche Projet

## Project title : **Word Wheel Game (without graphics)**

## Student names

**CHAIEB Meryam** **SASSI Houssem**

**JEMAI Siwar MARZOUKY Najem**

## Functional Requirements Specification

1. Who are the users (called *actors*) of your application? List the different types of users. (Typically there are only one or two types)

There’s only one type of useres which is the player .

1. How will each type of user interact with the application?

* Our application will allow the user to enter his name first to verify if he already played the game :
* If he’s an old player, his last score will apear else he’s name will be saved
* Set language
* Our application will allow the user to choose the level game (easy,medium,hard)
* Our application will allow the user to find words from the list on the screen
* Our application will display the score and show him the correct answers when the timing end .

## Program Design (Conception Détaillée)

**class opening : this class represents the opening of the game . it’s composed of 4 methods :**

1. **method welcome :**

* **It will pin up a message like "welcome in this application! the creators are......"**

1. **method userManagement :**

* **it will ask the user to enter his name first . then it will check if he’s an old palyer or not :**
  + **if he’s an old player it will display he’s last score**
  + **if he isn’t , it will save he’s name on the text file**

1. **method setLanguage :**

* **it will ask the user to choose a language (french,english,arab)**

1. **method level :**

* **it will ask the user to choose the level game (hard,medium,easy)**

**As a conclusion, we can say there’s 3 main attributes :**

* **name : the user’s name**
* **language**
* **level**

**class gameSetup : it’s composed of 4 methods :**

1. **method generateWord :**

* **It searchs randomly a word of 9 letters from the dictionary and save it in the text file just in front of the user’s name.**

1. **Method verifWord :**

* **It sets a list of correct words .**
* **It checks if the word written by the user is correct** 
  + **Having a meaning**
  + **Not a proper nounes, names**
  + **The shortest word should be composed by 3 letters**
  + **The longest word should be composed by 9 letters**

1. **Method score :**

* **Calculate the score and save it**

1. **Method result :**

* **Show the score to the player**
* **Save it in the text file.**
* **Show the longest word.**

**As a conclusion, this class contains 3 main attributs :**

* **Score**
* **writtenWord : the word given by the player.**
* **List : the list of the correct words .**

**Class Timer : to count down the time. It depends of the game level :**

* **Easy : 5 minutes**
* **Midium : 3 minutes**
* **Hard : 2 minutes**