**Hangman Game  
Documentation**

developed by **Team "Apple"**(Martin Tsekov, Yavor Atov, Georgi Georgiev, Mariela Kasovska)

**1. Introduction of the product**

**Hangman Game** is a basic object-oriented web application which is implemented thanks to the newest trends in the web development - **HTML5,** **CSS3**, **JavaScript** and **JQuery**. The game is really simple - the user must guess the word given by entering letters. If the word is guessed with less than 9 mistakes - the user wins the game, if not - the game is considered lost.

**2. Implementation details**

* **Index.html -** the main web page of the application
* **wordTests.html** - the web page which contains the tests of the word class
* **drawHangman.js** - the class draws the gallow and the hangman by given number of mistakes. If the user has made one mistake only the gallow is drawn, than if he or she has made two mistakes the method begins to draw the head and when more than two mistakes are made - the body of the hangman. When the user makes nine or more mistakes the game is over.
* **facebookApi** .**js**- the class enables the sharing of the application in Facebook by clicking on the Facebook icon.
* **gameStatus.js** - the class checks if the game is lost or won and shows the user an appropriate message. It consists of four functions:  
  - **check -** checks the number of the user's mistakes; if they are less or equal to eight and the whole word is revealed the game is won, and if the mistakes are more than eight the games is considered lost;  
  - **looseGame** - shows a message to the user that he or she has lost the game  
  - **winGame -** shows a message to the user that he or she has won the game  
   - **showMessage** - takes care of the messages; an optional implementation of a score board has been taken under consideration but due to the little time till the public defense of the project has not been developed.
* **generateRandomWord** - consists of a **randomWordGenerator** function which generates a random word; the word is chosen from a previously defined array;
* **keyboardGenerator** - consists of a single function - keyBoardGenerator which takes care of the initialization of the keyboard in the user interface;
* **word -** takes care of the initialization of a word in the game; consists of four functions:  
  - **init** - it plays the role of a class constructor  
  - **drawWord** - draws the word in the user interface  
  - **contains** - checks if the word contains the user's input letter  
  - **isRevealed** - returns true if the word has been revealed and false if not
* **gameEngine.js** - the class enables the generation of the game objects and takes care of the user interface. It consists of:  
  - an instance of the **drawer** class which draws the hangman  
  - an instance of the **word** class which generates a random word that is chosen from a previously defined array of words  
  - an instance of the **gameStatus** class which informs the user if the game is lost or won
* **wordTests.js** - contains the actual tests of the word class