Software Development

Jose Carlos Kuzolitz Garcia (x20164840)

Dublin - Ireland
December 2020

1. Project

This is a game called “Find Computer Words Game” which is a turn-based game. For this project this game is played by 2 players. As soon as the application is executed by the 1st player, it displays the game description and rules.

For this assignment, at the end of every turn the players are asked whether they would like to keep playing or not. Besides that, the player who guesses a word that has a vowel as its first character scores 3 points. And for a correct guesses word which starts in consonant, the player scores 1 point.

The points scored are displayed in every turn depending whose turn it is. When the players decide for not starting a new turn, the final result is displayed.

2. IPO

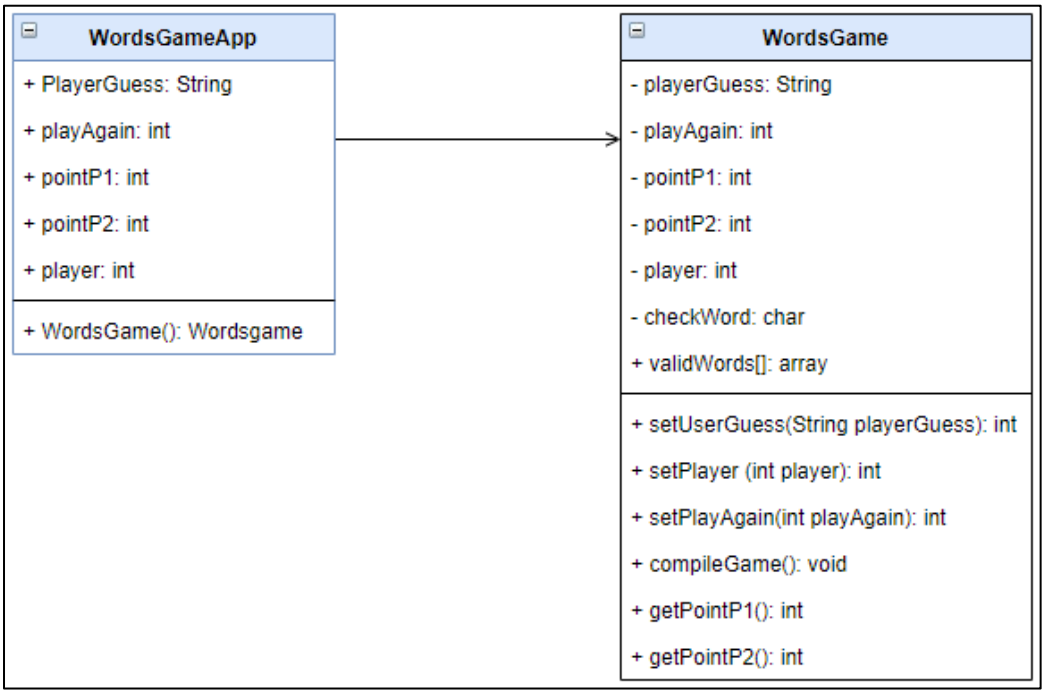
Find below the IPO created to shows how to information is handled by the game application.

Input	Process	Output

1. Player executes the application.
 	
2. The game displays Descriton and Rules as per below:
==========
= Welcome to Find Computer Words Game =
==========
 	
Rules:
12 Letters will be presented and each player has to guess words using them
 	
Points:
For each word correctly guessed starting by a vowel, the player gets 1 point
For each word correctly guessed starting by a consonant, the player gets 3 points
 	
==========
 	
3. Player is asked to type a word with 12 random letters choosen by the application. See example below:
 	
Player 1
Create a word with some of the letters listed below:
 	
k t q o s a w w m o a z
 	
Player 1, try to guess a word:
 	
5. Game askes to Player 2 to submit a word based on
the 12 letters random displayed
(a new set of words is generated everytime)
 	
6. Player 2 submits his/her word
	if(player == 1){
	for(int i=0, i < validWords.length ; 1++){
	if(playerGuess.equals(validWords)){
	checkWord = playerGuess.charAt(0);
	if(checkWord == vowel(A,a,E,e,I,i,O,o,U,u))
	pointP1 = pointP1 + 3;
	}
	else{
	pointP1 = pointP1 + 1;
	}
	}
	}
	}
	else{
	for(int i=0, i < validWords.length ; 1++){
	if(playerGuess.equals(validWords)){
	checkWord = playerGuess.charAt(0);
	if(checkWord == vowel(A,a,E,e,I,i,O,o,U,u))
	pointP2 = pointP2 + 3;
	}
	else{
	pointP2 = pointP2 + 1;
	}
	}
	}
	}
		1. Displays how many points Player 1 scored
		
		2. Displays how many points Player 2 scored
		
		3. Displays a message asking if the players would like to start a new turn, as per below:
		"Would you like to start a new turn? (1 - Yes or 2 - No)"
		
		4. Displays the Final Result as per below:
		
		FINAL RESULT:
		---------- 	
		
		Player 1: 1 points Player 2: 0 points
		
		Thank you for playing with us!

3. Class Diagram

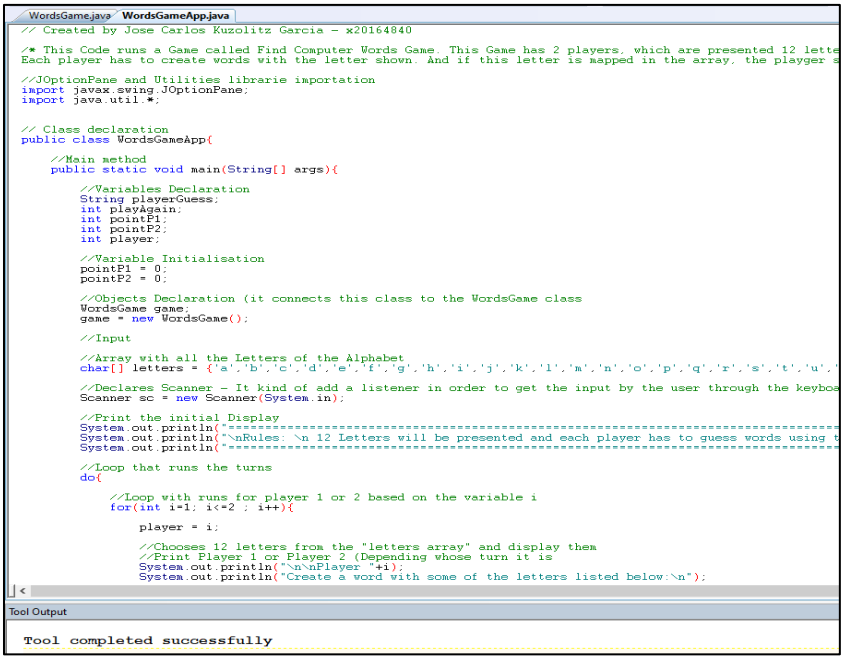
In the UML (Unified Modeling Language), the Class Diagram is used to describe a structure of an application based on Classes and its details. Basically, this Diagram shows information such as Classes with their Attributes, operations, relationships and objects.



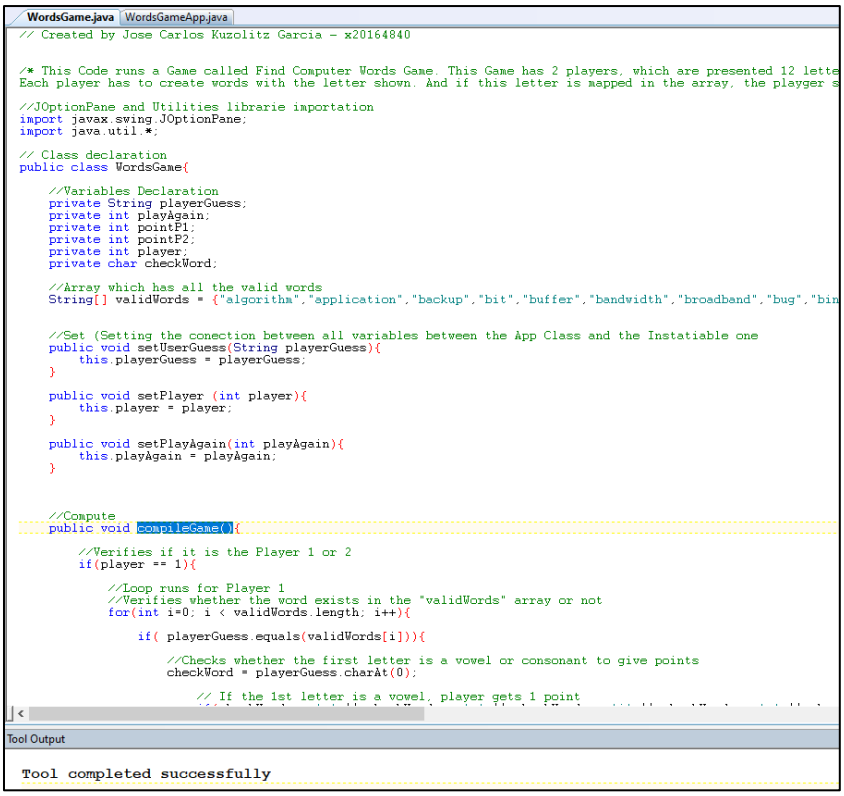
4. Testing

4.1 Compiling the Code

The following images show that the code is being compiled without any errors.



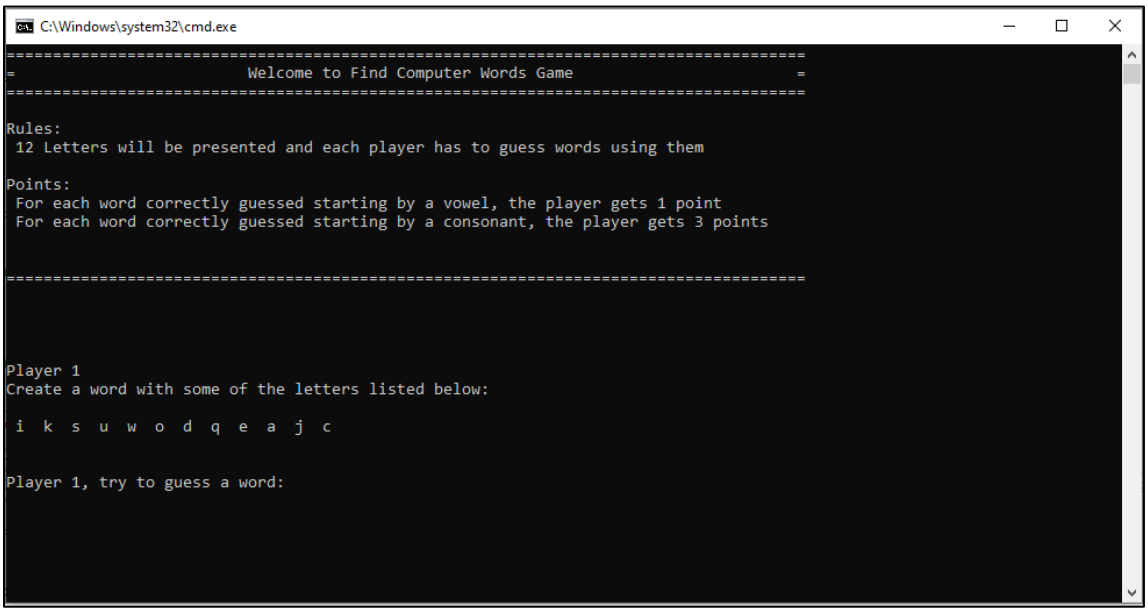
```
WordsGame.java  WordsGameApp.java
// Created by Jose Carlos Kuzolitz Garcia - x20164840

/* This Code runs a Game called Find Computer Words Game. This Game has 2 players, which are presented 12 lette
Each player has to create words with the letter shown. And if this letter is mapped in the array, the playger s

//JOptionPane and Utilities librarie importation
import javax.swing.JOptionPane;
import java.util.*;


// Class declaration
public class WordsGameApp{

    //Main method
    public static void main(String[] args){

        //Variables Declaration
        String playerGuess;
        int playAgain;
        int pointP1;
        int pointP2;
        int player;

        //Variable Initialisation
        pointP1 = 0;
        pointP2 = 0;

        //Objects Declaration (it connects this class to the WordsGame class
        WordsGame game;
        game = new WordsGame();

        //Input

        //Array with all the Letters of the Alphabet
        char[] letters = {'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','

        //Declares Scanner - It kind of add a listener in order to get the input by the user through the keyboa
        Scanner sc = new Scanner(System.in);

        //Print the initial Display
        System.out.println("
        System.out.println("\nRules: \n 12 Letters will be presented and each player has to guess words using t
        System.out.println("

        //Loop that runs the turns
        do{

            //Loop with runs for player 1 or 2 based on the variable i
            for(int i=1; i<=2 ; i++){

                player = i;

                //Chooses 12 letters from the "letters array" and display them
                //Print Player 1 or Player 2 (Depending whose turn it is
                System.out.println("\n\nPlayer "+i);
                System.out.println("Create a word with some of the letters listed below:\n");
<
Tool Output

Tool completed successfully
```



```
WordsGame.java  WordsGameApp.java
// Created by Jose Carlos Kuzolitz Garcia - x20164840


/* This Code runs a Game called Find Computer Words Game. This Game has 2 players, which are presented 12 lette
Each player has to create words with the letter shown. And if this letter is mapped in the array, the playger s

//JOptionPane and Utilities librarie importation
import javax.swing.JOptionPane;
import java.util.*;

// Class declaration
public class WordsGame{

    //Variables Declaration
    private String playerGuess;
    private int playAgain;
    private int pointP1;
    private int pointP2;
    private int player;
    private char checkWord;

    //Array which has all the valid words
    String[] validWords = {"algorithm","application","backup","bit","buffer","bandwidth","broadband","bug","bin


    //Set (Setting the conection between all variables between the App Class and the Instatiable one
    public void setUserGuess(String playerGuess){
        this.playerGuess = playerGuess;
    }

    public void setPlayer (int player){
        this.player = player;
    }

    public void setPlayAgain(int playAgain){
        this.playAgain = playAgain;
    }


    //Compute
    public void compileGame(){

        //Verifies if it is the Player 1 or 2
        if(player == 1){

            //Loop runs for Player 1
            //Verifies whether the word exists in the "validWords" array or not
            for(int i=0; i < validWords.length; i++){

                if( playerGuess.equals(validWords[i])){

                    //Checks whether the first letter is a vowel or consonant to give points
                    checkWord = playerGuess.charAt(0);

                        // If the 1st letter is a vowel, player gets 1 point
<
Tool Output

Tool completed successfully
```

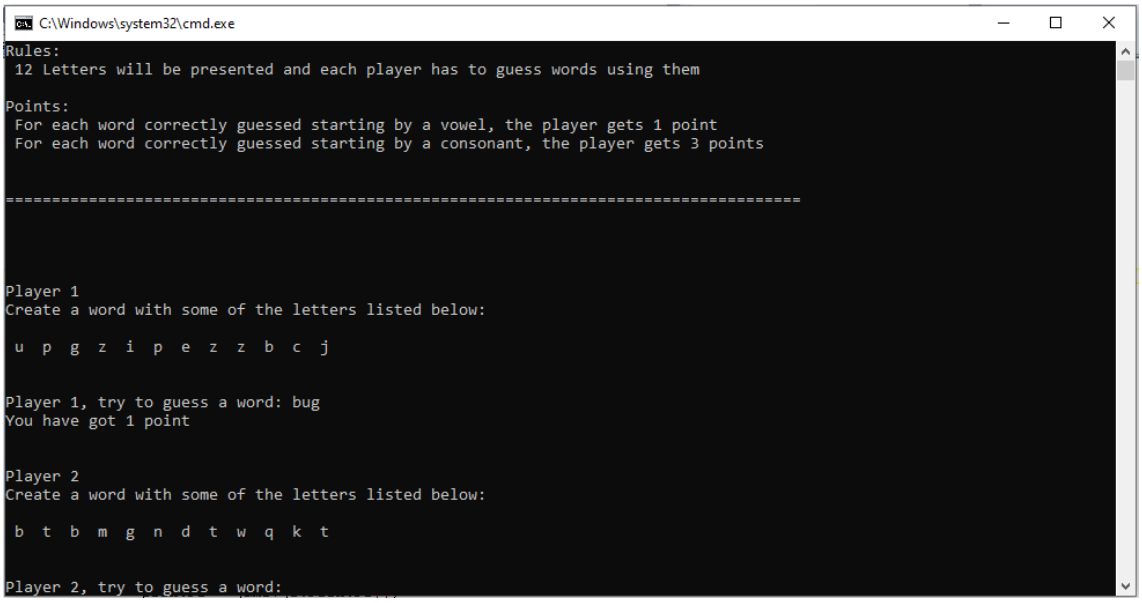
4.2 Executing the Game

The following image shows what the Game displays as soon as it is executed.

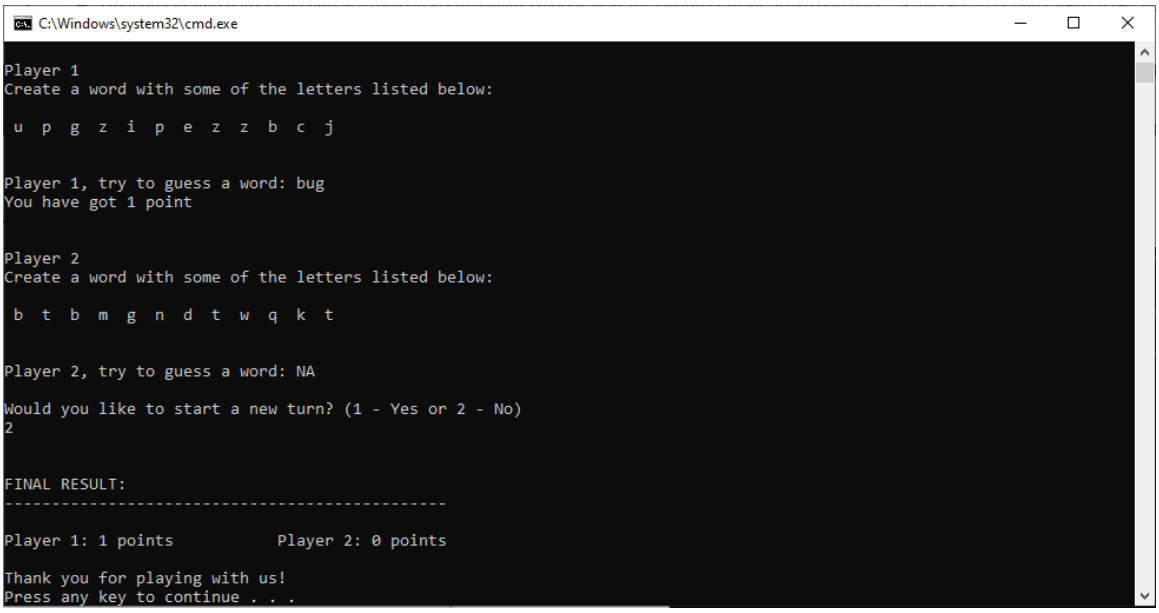


4.3 Guessing Words

At this moment, the Player 1 is requested to insert a word. The word inserted is “bug”. The Player 1 has scored 1 point.



Then, it is the Player 2 turn. The Player 2 sets “NA”. No information about points is displayed, because the Player 2 did not score. And the application asks the players if they would like to start another turn. The option selected was “2 - No” in order to finish the game.



5. Source Code

5.1 WordsGameApp.java

```
// Created by Jose Carlos Kuzolitz Garcia - x20164840

/* This Code runs a Game called Find Computer Words Game. This Game has 2 players, which are presented 12 letters randomly selected.

Each player has to create words with the letter shown. And if this letter is mapped in the array, the player  score points*/

//Utilities library importation
import java.util.*;

// Class declaration
public class WordsGameApp{

//Main method
public static void main(String[] args){

        //Variables Declaration
        String playerGuess;
        int playAgain;
        int pointP1;
        int pointP2;
        int player;
```

```
//Variable Initialisation
pointP1 = 0;
pointP2 = 0;


//Objects Declaration (it connects this class to the WordsGame class
WordsGame game;
game = new WordsGame();


//Input
//Array with all the Letters of the Alphabet
char[] letters = {'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z'};

//Declares Scanner - It kind of add a listener in order to get the input by the user through the keyboard
Scanner sc = new Scanner(System.in);


//Print the initial Display
        System.out.println("============================================================
==============================\n=                  Welcome to Find Computer Words Game
=\n============================================================================
==========");

System.out.println("\nRules: \n 12 Letters will be presented, and each player has to guess words using them \n\nPoints: \n For each word correctly guessed starting by a vowel, the player gets 1 point \n For each word correctly guessed starting by a consonant, the player gets 3 points \n\n");
        System.out.println("============================================================
==============================\n\n");

//Loop that runs the turns
do{
    //Loop with runs for player 1 or 2 based on the variable i
    for(int i=1; i<=2 ; i++){
        player = i;

        //Chooses 12 letters from the "letters array" and display them
        //Print Player 1 or Player 2 (Depending whose turn it is)
        System.out.println("\n\nPlayer "+i);

        System.out.println("Create a word with some of the letters listed below:\n");

        //Loop displays random letters
            for(int j=0; j<12; j++){
            char letter = letters[(int) (Math.random() * letters.length)];
            System.out.print(" "+letter+ " ");
            }
```

```
                    //Asks for a guess

                    System.out.print("\n\n\nPlayer "+i+", try to guess a word: ");

                    //reads the information that the user inserted and allocates it to the
                    userGuess variable
                    playerGuess = sc.next();

                    //SET
                    //Execute function on WordsGame giving PlayerGuess and Player as
                    parameters
                    game.setUserGuess(playerGuess);
                    game.setPlayer(player);


                    //Compiles Word set by player
                    game.compileGame();


                    //Outputs the points got by players
                    //This clause executes getPointP1 or getPointP2 depending on the current
                    player
                    if(player == 1){

                            pointP1 = game.getPointP1();

                    }

                    else{

                            pointP2 = game.getPointP2();

                    }

            }

        // Verifies of the player wants to start another turn
        System.out.println("\nWould you like to start a new turn? (1 - Yes or 2 - No)");
        playAgain = sc.nextInt();
        System.out.println("\n");

        //While user types 1, the game keeps running
        }while(playAgain == 1);
         //Outputs the Final Result
        System.out.println("FINAL RESULT:");
        System.out.println("-----------------------------------------------");
        System.out.println("\nPlayer 1: " +pointP1+ " points          Player 2: " +pointP2+ " points");
        System.out.println("\nThank you for playing with us!");

        //Close Scanner
        sc.close();
        }
}
```

5.2 WordsGame.java

```
// Created by Jose Carlos Kuzolitz Garcia - x20164840

/* This Code runs a Game called Find Computer Words Game. This Game has 2 players, which are presented 12 letters randomly selected.

Each player has to create words with the letter shown. And if this letter is mapped in the array, the player    score points*/

//Utilities library importation
import java.util.*;

//Class declaration
public class WordsGame{

//Variables Declaration
private String playerGuess;
private int playAgain;
private int pointP1;
private int pointP2;
private int player;
private char checkWord;

//Array which has all the valid words
String[] validWords =
{"algorithm","application","backup","bit","buffer","bandwidth","broadband","bug","binary","browser","bus","cache","command","computer","cookie","compiler","cyberspace","compress","configure","database","digital","data","debug","desktop","disk","domain","decompress","development","download","dynamic","email","encryption","firewall","flowchart","file","folder","graphics","hyperlink","host","hardware","icon","inbox","internet","kernel","keyword","keyboard","laptop","login","logic","malware","motherboard","mouse","mainframe","memory","monitor","multimedia","network","node","offline","online","path","process","protocol","password","phishing","platform","program","portal","privacy","programmer","queue","resolution","root","restore","router","reboot","runtime","screen","security","shell","snapshot","spam","screenshot","server","script","software","spreadsheet","storage","syntax","table","template","thread","terminal","username","virtual","virus","web","website","window","wireless"};

//Set (Setting the conection between all variables between the App Class and the Insatiable one
public void setUserGuess(String playerGuess){
this.playerGuess = playerGuess;
}

public void setPlayer (int player){
this.player = player;
}

public void setPlayAgain(int playAgain){
this.playAgain = playAgain;
}
```

```
//Compute
public void compileGame(){

//Verifies if it is the Player 1 or 2
if(player == 1){

        //Loop runs for Player 1
        //Verifies whether the word exists in the "validWords" array or not
        for(int i=0; i < validWords.length; i++){
          if(playerGuess.equals(validWords[i])){

              //Checks whether the first letter is a vowel or consonant to give points
              checkWord = playerGuess.charAt(0);

              //If the 1st letter is a vowel, player gets 1 point
              if(checkWord == 'a' || checkWord == 'e' || checkWord == 'i' || checkWord ==
                 'o' || checkWord == 'u' || checkWord == 'A' || checkWord == 'E' ||
                 checkWord == 'I' || checkWord == 'O' || checkWord == 'U' ){

                      //Displays the output
                      System.out.println("You have got 3 points");
                      //Adds 3 points to the variable "pointP1"
                              pointP1 = pointP1 + 3;

               }
               //Else the 1st letter is a consonant, player get 3 points
               else{
                       //Displays the output
                       System.out.println("You have got 1 point");
                       //Adds 1 point to the variable "pointP1"
                       pointP1 = pointP1 + 1;
                }
           }
        }
}
else{
        //Verifies whether the word exists in the "validWords" array or not
        for(int i=0; i < validWords.length; i++){
          if(playerGuess.equals(validWords[i])){

           //Checks whether the first letter is a vowel or consonant to give points
            checkWord = playerGuess.charAt(0);
```

```
                    //If the 1st letter is a vowel, player gets 1 point
                    if(checkWord == 'a' || checkWord == 'e' || checkWord == 'i' || checkWord ==
                                'o' || checkWord == 'u' || checkWord == 'A' || checkWord == 'E' ||
                                        checkWord == 'I' || checkWord == 'O' || checkWord == 'U' ){

                    //Displays the output
                    System.out.println("You have got 3 points");
                    //Adds 3 points to the variable "pointP2"
                    pointP2 = pointP2 + 3;

                    }
                    //Else the 1st letter is a consonant, player get 3 points
                    else{
                        //Displays the output
                        System.out.println("You have got 1 point");
                        //Adds 1 point to the variable "pointP2"
                        pointP2 = pointP2 + 1;
                    }
                }
            }
        }
    }

    //GET Returns the Points from the compute and return it to the App Class
    public int getPointP1(){
            return pointP1;
    }

    public int getPointP2(){
            return pointP2;
     }

}
```