WORD GAME

Intermediate Programming Project

Faculty of Engineering

CSE 1046

OWNERS:	Ali Çalış	150712025
	Oğuzhan Çalıkkasap	150714841

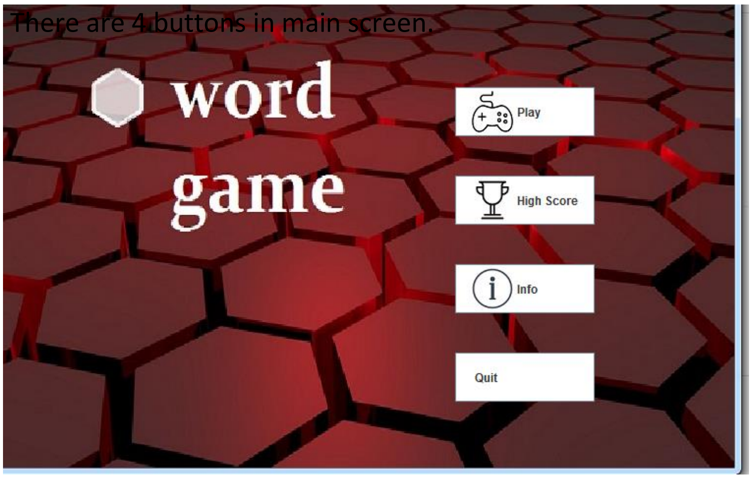
DATE: 18/05/2014

WORD GAME

“Kelime Oyunu” was a TV show, which contestants were trying to find words from their definitions. We adapted this show to a computer game

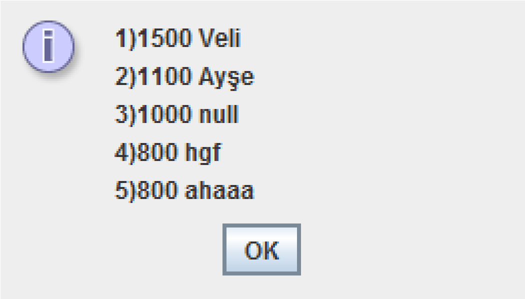
Main Screen:

There are 4 buttons in main screen.



Play button has an action listener which dispose this frame and went to “Oyun Ekrani”s constructor.

High Score calls method from High Score class, which writes top 5 High Scores to screen. Action listener shows Message Dialog, which shown below:



1)1500 Veli
2)1100 Ayşe
3)1000 null
4)800 hgf
5)800 ahaaa

Info Button gives some information about game. Quit button exits.

Game Screen:

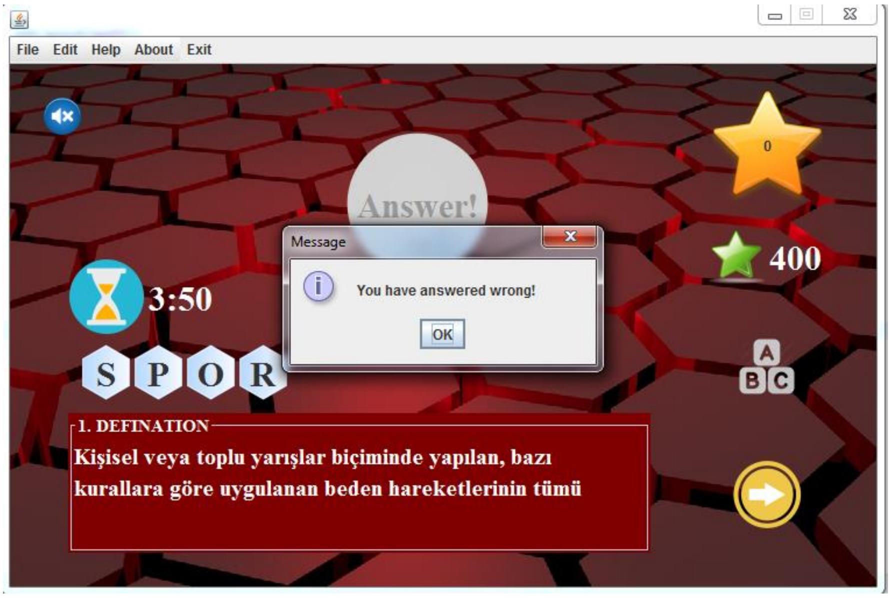
When play button is clicked, Option Pane comes to screen and asks name to user.

There are some conditions for this option pane. If name is empty, option pane asks user’s name, repeatedly.

When a String is entered, frame is disposed.

In the “Oyun Ekrani” class, timer object was created and there are counter methods which controls with an boolean variable named ContinuetoCount.

When answer button is clicked, action listener is read by program. In answer’s action listener, time on screen is stopped because boolean value is false. According to player’s answer there are 2 option pane which notified your success or fail.



Message

You have answered wrong!

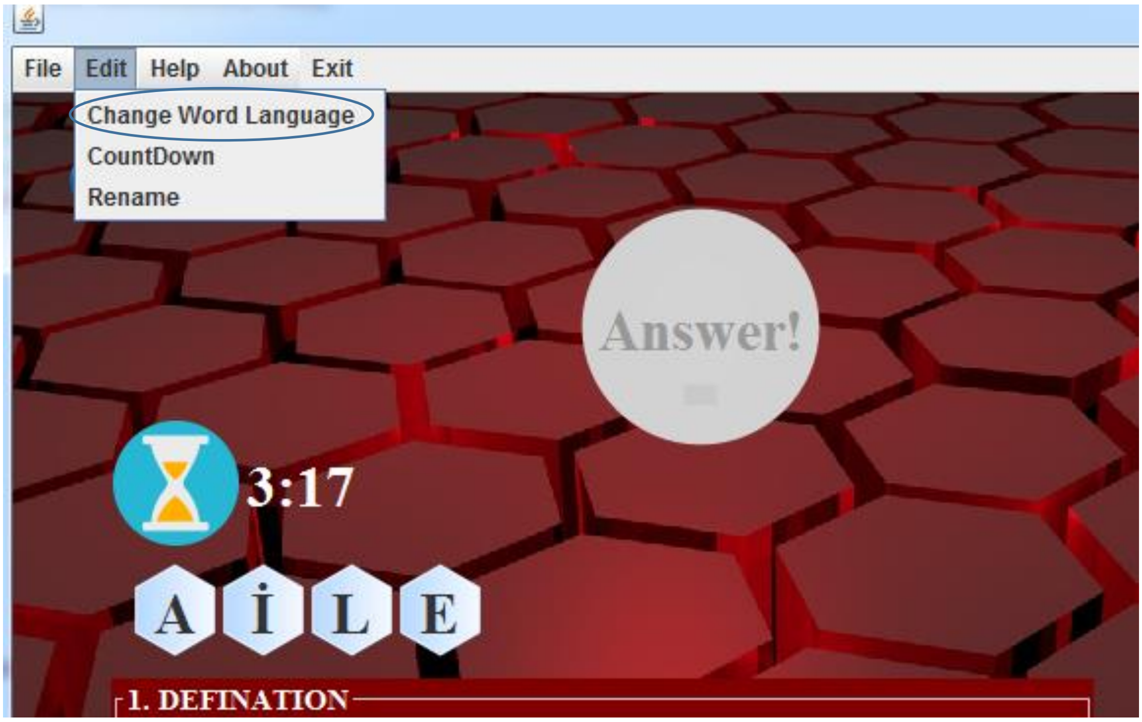
1. DEFINATION

Kişisel veya toplu yarışlar biçiminde yapılan, bazı kurallara göre uygulanan beden hareketlerinin tümü

After that, player had to click Next button to get the next question.

When question is asked to player, he/she has a chance to give a hint letter. If player gets all letters, he/she cannot get point from the question.

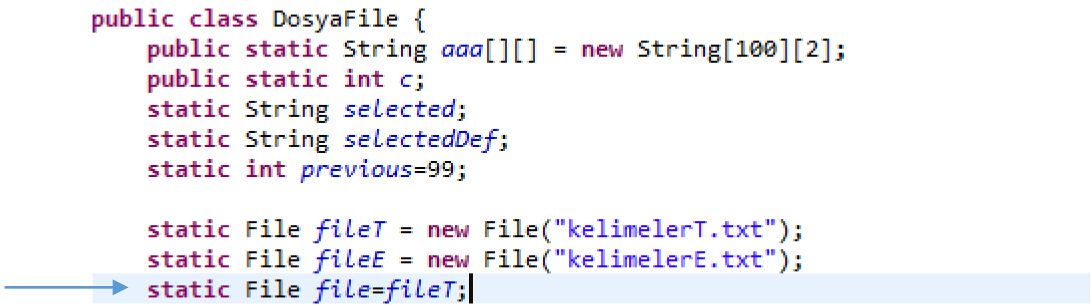
From Menu Bar, player also can change words’ language. Program reads words from a text file. When language is changed, another text file is read.



1. DEFINATION

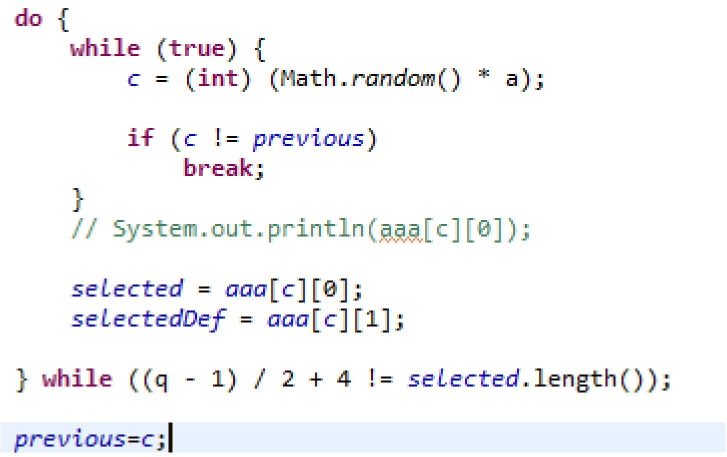
There is also one more class which we used to read words from text file and pick a word, named “DosyaFile” class.

File Input Class



```
public class DosyaFile {
    public static String aaa[][] = new String[100][2];
    public static int c;
    static String selected;
    static String selectedDef;
    static int previous=99;

    static File fileT = new File("kelimelerT.txt");
    static File fileE = new File("kelimelerE.txt");
    static File file=fileT;
```

Our variables were defined in this class and Turkish was set default. Then, we write our words and definitions to 2 dimensional String array.



```
do {
    while (true) {
        c = (int) (Math.random() * a);

        if (c != previous)
            break;
    }
    // System.out.println(aaa[c][0]);

    selected = aaa[c][0];
    selectedDef = aaa[c][1];

} while ((q - 1) / 2 + 4 != selected.length());

previous=c;
```

In this do while loop, first we get a random variable between 0 and length of our word array and stored in a variable named previous.

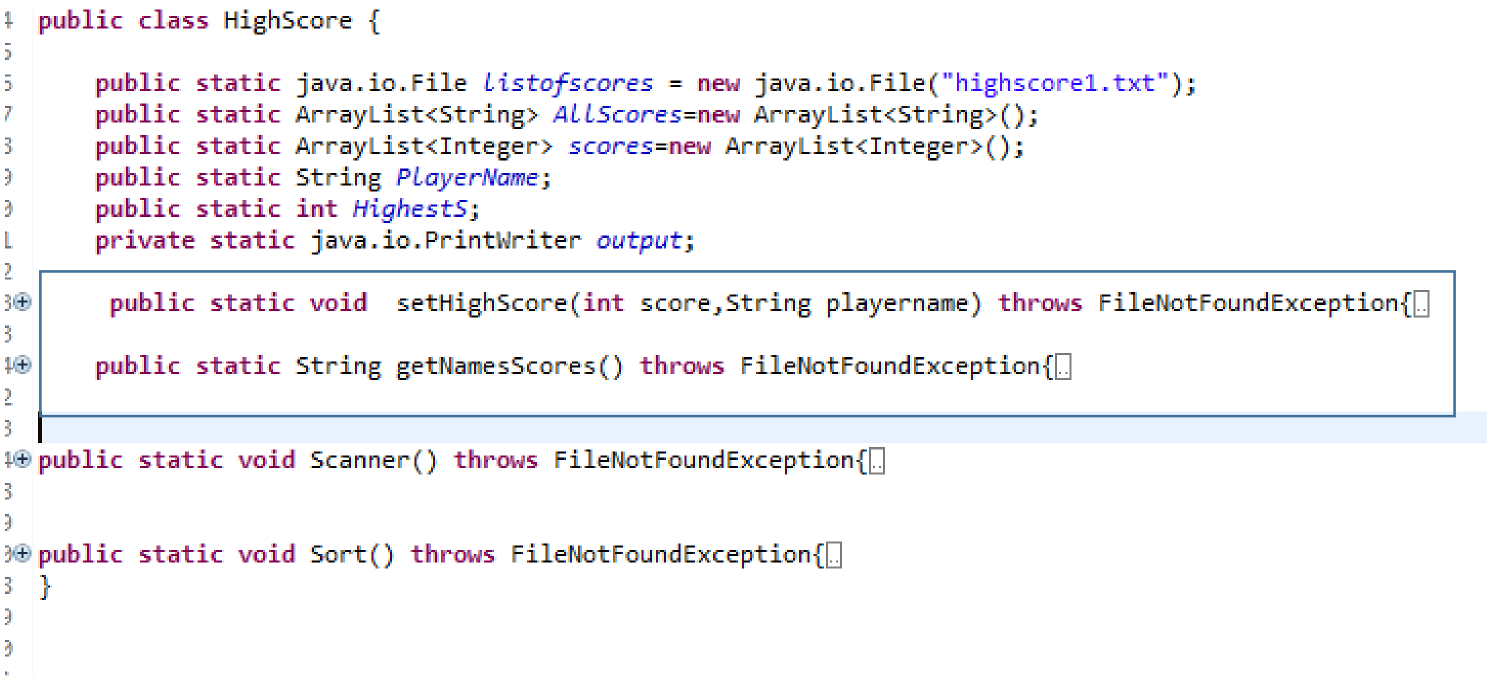
High Score Class

In this class we created an array list which reads our previous scores from high scores text file and lists in order. There are 4 different methods but 2 of them calling by other classes.

Set & Get methods used in different points.

When high scores button clicked, get method is called by a class and this method turns String, which is read from the text file and, to write on a Dialog Pane.

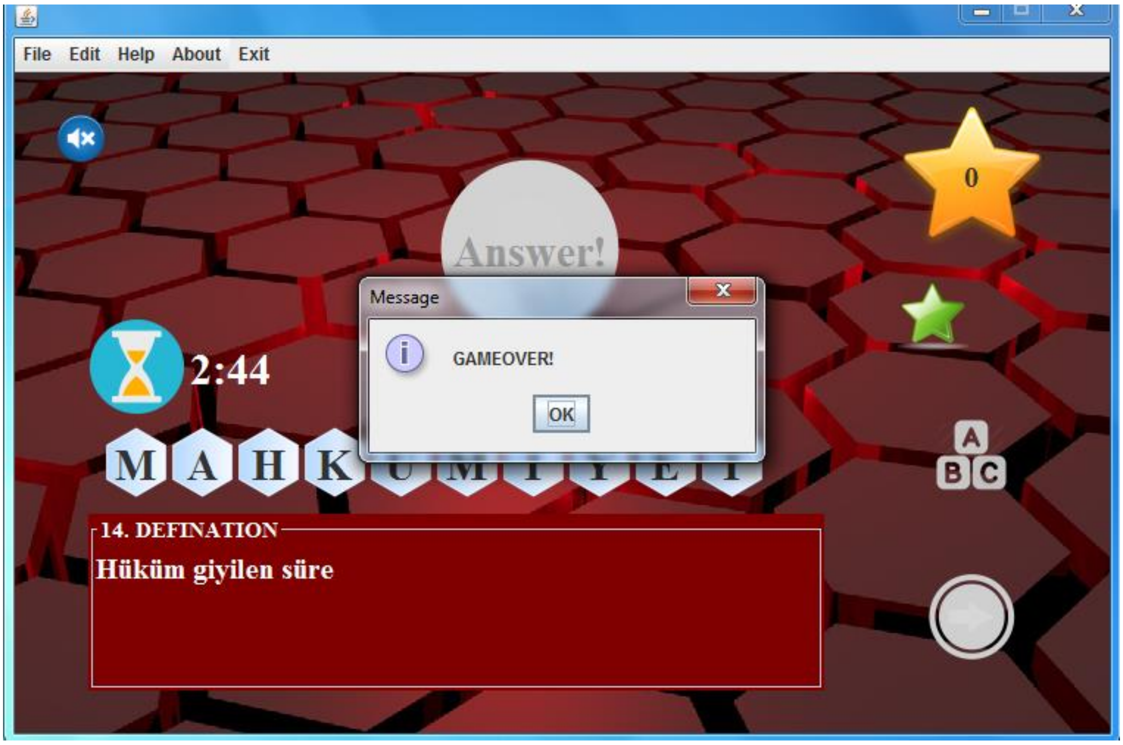
Other method is used by storing. When game ends, method is called and new score is stored with player name.



```
public class HighScore {

    public static java.io.File listofscores = new java.io.File("highscore1.txt");
    public static ArrayList<String> AllScores=new ArrayList<String>();
    public static ArrayList<Integer> scores=new ArrayList<Integer>();
    public static String PlayerName;
    public static int HighestS;
    private static java.io.PrintWriter output;

     public static void  setHighScore(int score,String playername) throws FileNotFoundException{□

    public static String getNamesScores() throws FileNotFoundException{□


public static void Scanner() throws FileNotFoundException{□


public static void Sort() throws FileNotFoundException{□
}
```

Last Screen

We had 2 conditions to end game. As I said before, time is first condition. The other one is number of questions. When amount of questions reaches 14, game also ends.

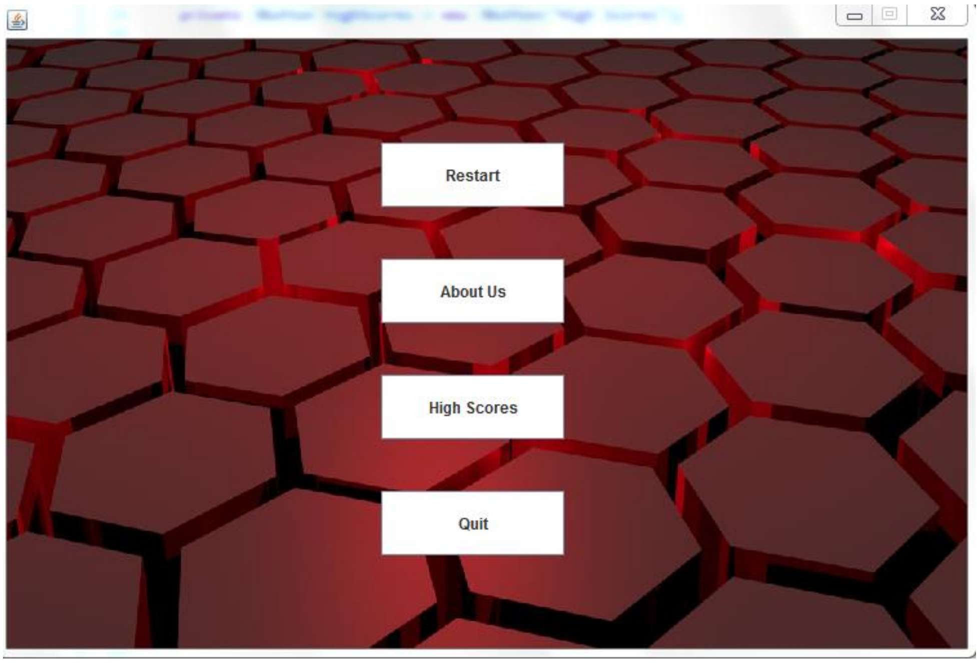


GAMEOVER!

14. DEFINATION

Hüküm giyilen süre

After then, we had one more screen which is named Last Screen. In this screen same action listeners added to buttons. Restart button disposes Last Screen and goes Main Menu's frame constructor. Also high score does same actions with High Score button on Main Screen.



Also we had an Active Menu Bar which is written in Menu Bar Class.

Game totally end with Quit Button.