Georgian CANADA | ILAC
Georgian@ILACToronto

COMP 1030

Final Project
Hangman game

Igor Guimaraes

Navjot Kaur

Yuvraj Singla

Rodolfo Silva

Hangman game

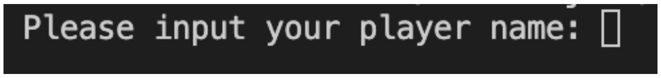
Hangman is a game that is widely used mainly in teaching environments, as it is a game where we have involved discovering words through letter choices, making it a fun challenge for students and teachers.

Firstly, we have to point out that one of the best ways of teaching new words and disseminating new knowledge to children or students involved in learning new languages, is with the use of games and scavenger hunts, making the absorption of new knowledge fun, for this we can use the use of games like hangman, where players need to discover the hidden word, through trial and error.

Our hangman game project was developed using the Java object-oriented programming language in its version JDK 17.0.6 2023-01-17 and using the libraries java.util.Scanner, java.util.Random and java.util.ArrayList, where we have a group of words stored in an array along with their hints through the java.util.ArrayList library, where when accessing the system, the user will visualize a set of spaces for the letters of the word selected through a random by the java.util library .Random, when typing your choice of letter captured by the java.util.Scanner library, the system checks if the word has the typed letter, if the selected letter exists, the result will be displayed on the screen with the filling of the space or the display of that the choice is wrong.

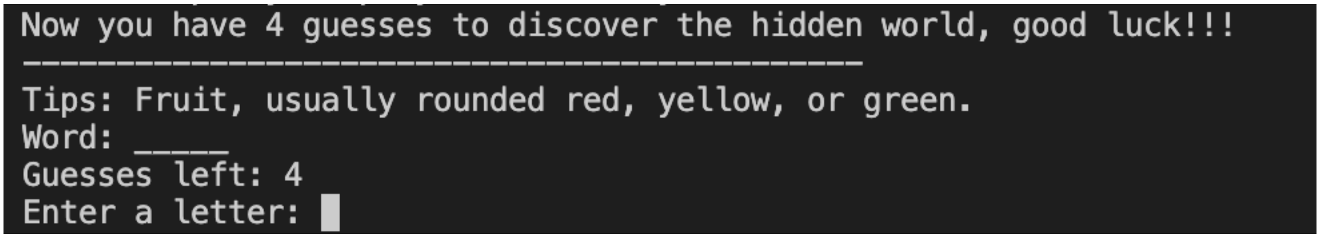
At the end of the round with up to four attempts, the player will see the information that the process has been completed, and will be asked if he wants to continue or end the game, if he wants to continue he will press the letter Y or y and a new round will begin, if you choose to end the game by pressing the letter N or n, you will receive on the screen the total number of rounds played, which words were displayed with your hints and which was the total score achieved in all plays.

Firstly the user will asked to input the name:



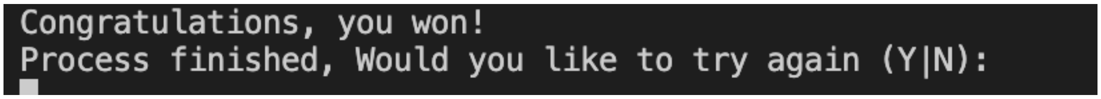
```
Please input your player name:
```

Second, the player will input letters until discovery the world:



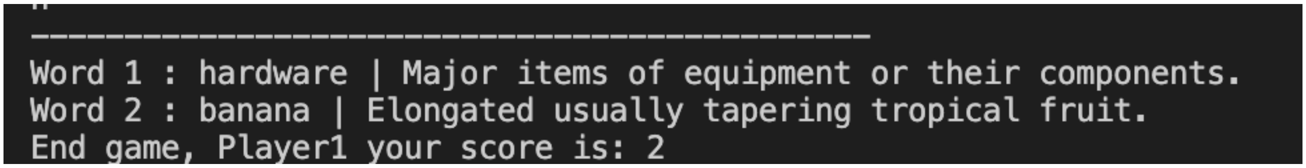
```
Now you have 4 guesses to discover the hidden world, good luck!!!
---------------------------------------------
Tips: Fruit, usually rounded red, yellow, or green.
Word: _____
Guesses left: 4
Enter a letter:
```

Third, player will see the message asking if would like to try again or don’t:



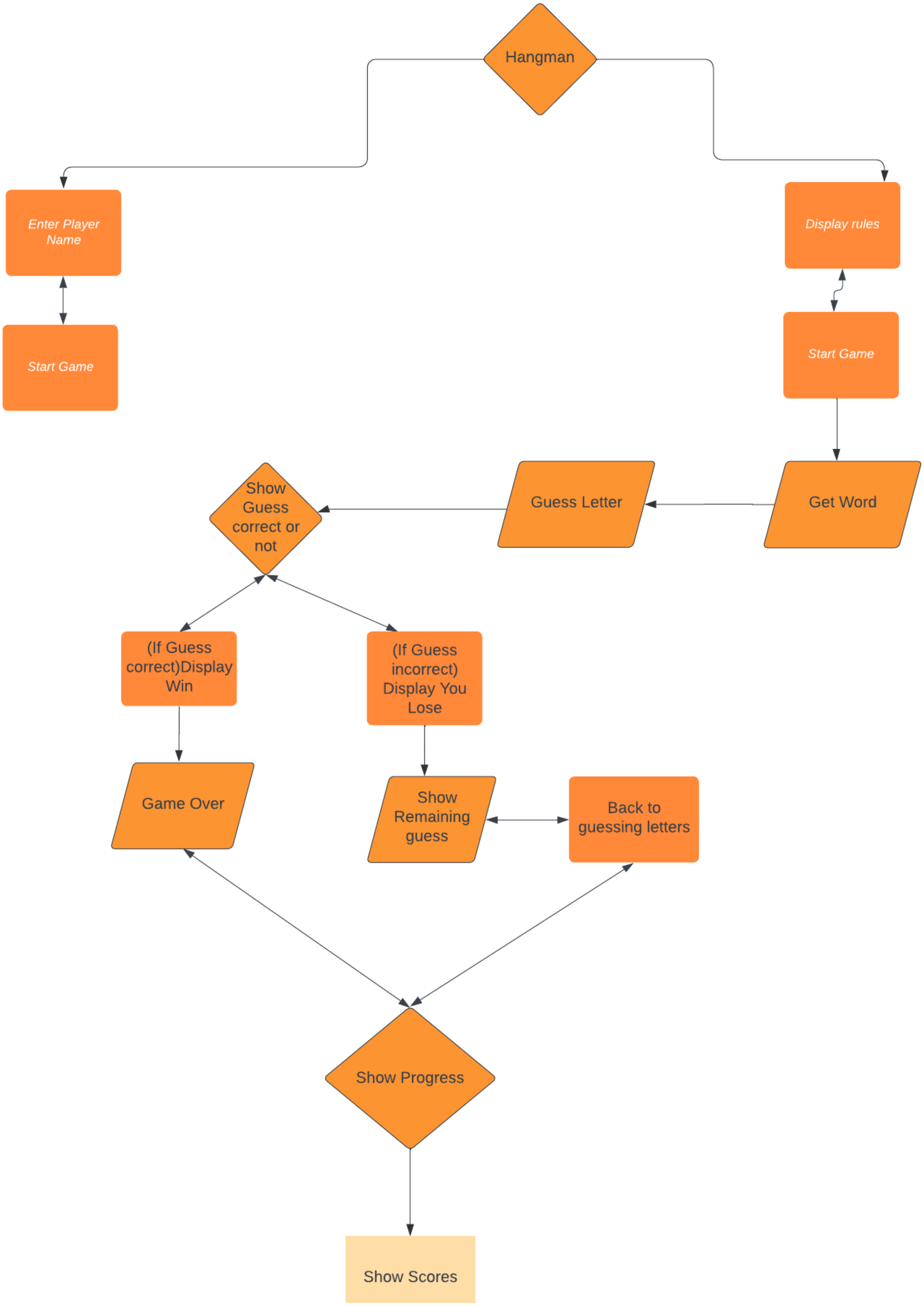
```
Congratulations, you won!
Process finished, Would you like to try again (Y|N):
```

Finally the player will receive your score with words and tips:



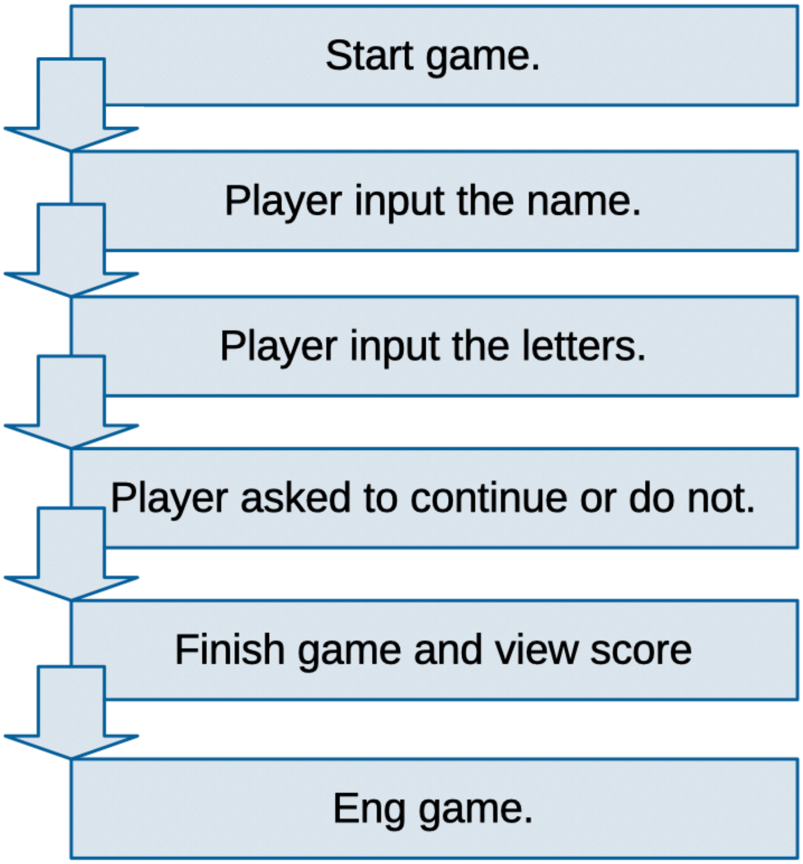
```
 ---------------------------------------------
 Word 1 : hardware | Major items of equipment or their components.
 Word 2 : banana | Elongated usually tapering tropical fruit.
 End game, Player1 your score is: 2
```

Hangman Flowchart:



Caption

Hangman diagram:



Start game.

Player input the name.

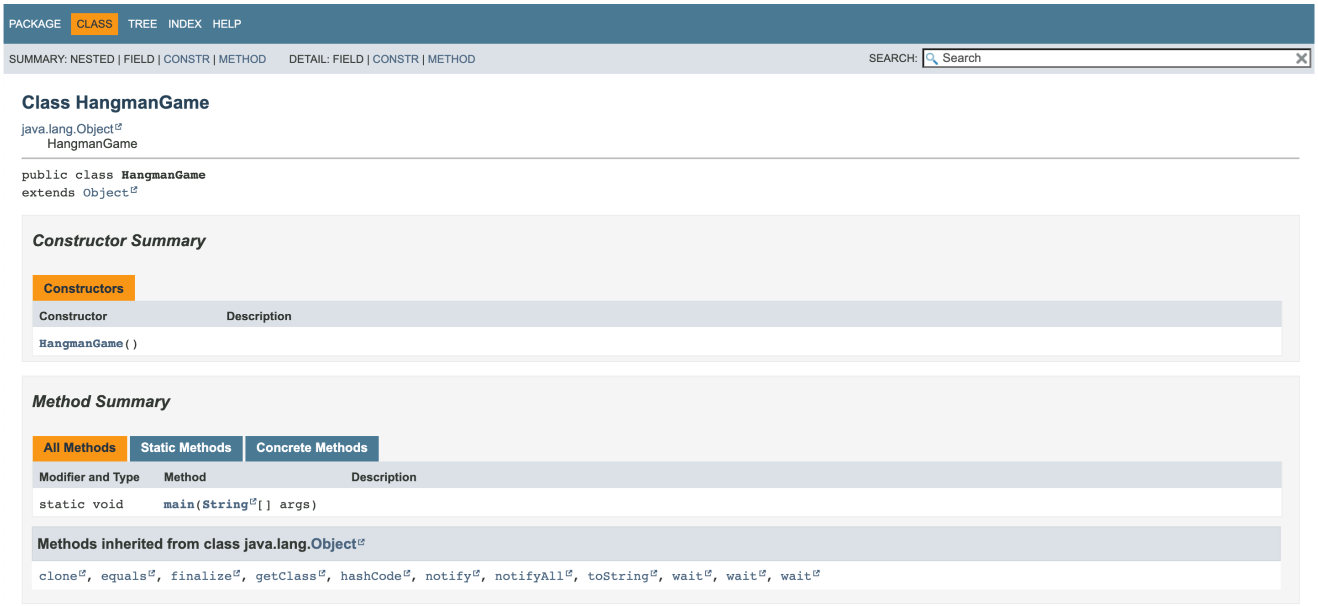
Player input the letters.

Player asked to continue or do not.

Finish game and view score

Eng game.

Javadoc Hangmagame.java



PACKAGE CLASS TREE INDEX HELP
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD
SEARCH: Search

Class HangmanGame

java.lang.Object
HangmanGame

```
public class HangmanGame
extends Object
```

Constructor Summary

Constructors

Constructor	Description
HangmanGame()	

Method Summary

All Methods | Static Methods | Concrete Methods

Modifier and Type	Method	Description
static void	main(String[] args)	

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Source code:

Class Hangman:

```
/**
 * Final Project Hangmangame.
 * @author Igor Guimaraes 200540189 - Navjot Kaur 200547472 - Yuvraj Singla - Rodolfo Carvalho 200536943
 * @date March 18, 2023
 * @time 09:00pm
 */


public class Hangman {
    private String wordRec;
    private String tipsRec;


    public Hangman(String wordRec, String tipsRec) {
        this.wordRec = wordRec;
        this.tipsRec = tipsRec;
    }


    public String getWordRec() {
        return this.wordRec;
    }


    public void setWordRec(String wordRec) {
        this.wordRec = wordRec;
    }


    public String getTipsRec() {
        return this.tipsRec;
    }

    public void setTipsRec(String tipsRec) {
        this.tipsRec = tipsRec;
    }

}
```

Class HandmanGame:

```
/**
 * Final Project Hangmangame.
 * @author Igor Guimaraes 200540189 - Navjot Kaur 200547472 - Yuvraj Singla - Rodolfo Carvalho 200536943
 * @date March 18, 2023
 * @time 09:00pm
 */
import java.util.Scanner;
import java.util.Random;
import java.util.ArrayList;

public class HangmanGame {

    private static final String[] WORDS = { "apple", "banana", "artificial", "funny", "hardware" };
    private static final String[] TIPS = { "Fruit, usually rounded red, yellow, or green.",
            "Elongated usually tapering tropical fruit.", "humanly contrived", "causing amusement or laughter",
            "Major items of equipment or their components." };
    private static final int MAX_GUESSES = 4;
```

```
public static void main(String[] args) {
    ArrayList<Hangman> list = new ArrayList<>();
    int x;
    int y;
    int position = 0;
    Random random = new Random();
    // x = random.nextInt(5);
    Scanner scanner = new Scanner(System.in);

    int score = 0;
    char opt = 'y';
    String name;
    System.out.print("Please input your player name: ");// User input
    name = (scanner.nextLine());// User input record
    System.out.println("Now you have 4 guesses to discover the hidden world, good luck!!!");
    do {
        try {
            x = random.nextInt(5);
            String word = selectRandomWord(x);
            String tips = selectRandomTips(x);
            Hangman recorder = new Hangman(word, tips);
            list.add(recorder);
            position = position + 1;

            int remainingGuesses = MAX_GUESSES;
            StringBuilder guessedLetters = new StringBuilder();

            while (remainingGuesses > 0) {
                System.out.println("-------------------------------------------");
    System.out.println("Tips: " + tips);
    System.out.println("Word: " + getHiddenWord(word, guessedLetters));
    System.out.println("Guesses left: " + remainingGuesses);
    System.out.print("Enter a letter: ");
    String letter = scanner.nextLine();

    if (guessedLetters.indexOf(letter) != -1) {
        System.out.println("You already guessed that letter!");
    } else {
        guessedLetters.append(letter);

        if (word.indexOf(letter) != -1) {
            System.out.println("Correct!");
        } else {
            System.out.println("Wrong!");
            remainingGuesses--;
        }
    }

    if (getHiddenWord(word, guessedLetters).equals(word)) {
        System.out.println("Congratulations, you won!");
        score = score + 1;
        break;
    }
}

if (remainingGuesses == 0) {
    System.out.println("Sorry, you lost. The word was " + word);
}
System.out.println("Process finished, Would you like to try again (Y|N): ");// User input
```

```
            opt = (scanner.nextLine().charAt(0));// User input record
        } // Exceptions treatment
        catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("The exception is: " + e.getMessage());
        } catch (NullPointerException e) {
            System.out.println("The exception is: " + e.getMessage());
        } catch (RuntimeException e) {
            System.out.println("The exception is: " + e.getMessage());
        }
    } while (opt == 'y' || opt == 'Y');

    System.out.println("--------------------------------------------");
    for (int i = 0; i < position; i++) {
        String word = list.get(i).getWordRec();
        String tips = list.get(i).getTipsRec();

        System.out.println("Word " + (i+1) + " : " + word + " | " + tips);
    }
    System.out.println("End game, "+name+" your score is: " + score);
}

// --------

static String selectRandomWord(int x) {
    return WORDS[x];
}

static String selectRandomTips(int x) {
    return TIPS[x];
}
```

```
    static private String getHiddenWord(String word, StringBuilder guessedLetters) {
        StringBuilder hiddenWord = new StringBuilder();

        for (int i = 0; i < word.length(); i++) {
            char c = word.charAt(i);

            if (guessedLetters.indexOf(String.valueOf(c)) != -1) {
                hiddenWord.append(c);
            } else {
                hiddenWord.append("_");
            }
        }

        return hiddenWord.toString();
    }
}
```