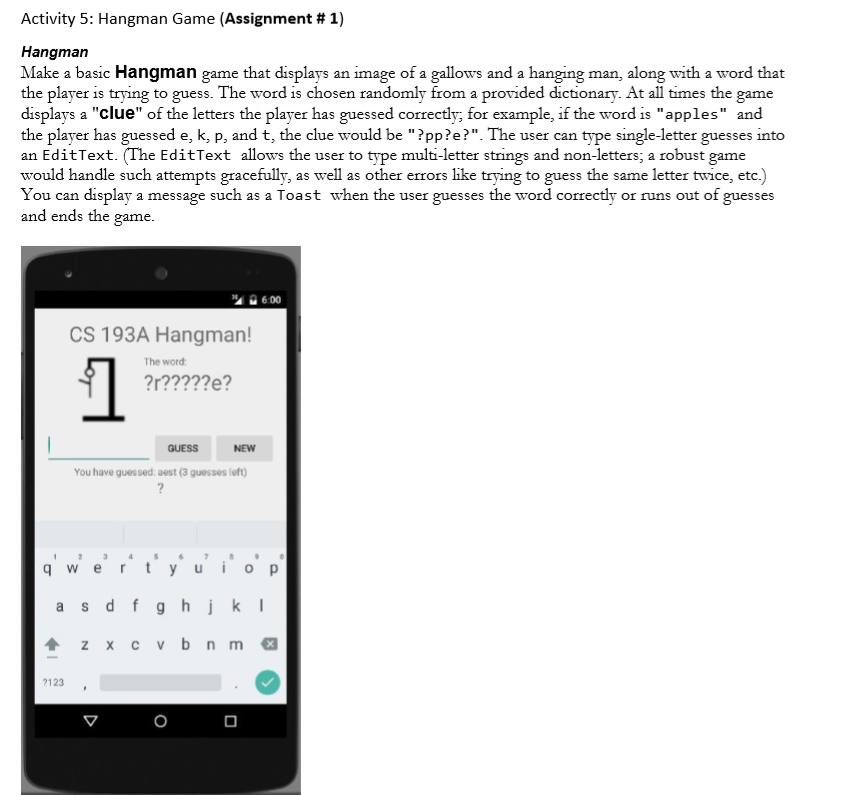
|  |
| --- |
| Mobile Application Development |
| Assignment-1 |

|  |
| --- |
| Safdar Hussain, BSCS-VII-A, CMS: 0231-18-0059  11-2-2021 |



Code

package com.example.hangmangame;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;

import android.os.Bundle;

import android.view.View;

import android.widget.EditText;

import android.widget.ImageView;

import android.widget.TextView;

import java.util.ArrayList;

import java.util.Random;

public class MainActivity extends AppCompatActivity {

    Context context;

    String[] WORDS;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

        img = findViewById(R.id.img);

        context=getApplicationContext();

        WORDS = context.getResources().getStringArray(R.array.words);

        wordTv = findViewById(R.id.letters);

        newGame();

    }

    public static final Random RANDOM = new Random();

    public static final int MAX\_ERRORS = 6;

    private String wordToFind;

    private char[] wordFound;

    private int nbErrors;

    private ArrayList < String > letters = new ArrayList < > ();

    private ImageView img;

    private EditText wordTv;

    private TextView wordToFindTv;

    public void creatNewGame(View view) {

            newGame();

    }

        private String nextWordToFind() {

        return WORDS[RANDOM.nextInt(WORDS.length)];

    }

    public void newGame() {

        nbErrors = 0;

        letters.clear();

        wordToFind = nextWordToFind();

        wordFound = new char[wordToFind.length()];

        for (int i = 0; i < wordFound.length; i++) {

            wordFound[i] = '?';

        }

        updateImg(nbErrors);

        wordTv.setText(wordFoundContent());

    }

    public boolean wordFound() {

        return wordToFind.contentEquals(new String(wordFound));

    }

    public void clickedGuess(View view) {

        EditText txt=findViewById(R.id.txt);

        String letter=txt.getText().toString();

        if (letter.length() == 1 && nbErrors < MAX\_ERRORS) {

            TextView leftGueses=findViewById(R.id.leftGuesses);

            int num=6-nbErrors;

            leftGueses.setText("You have guessed asset (" + num + " guesses left)");

            enter(letter);

            wordTv.setText(wordFoundContent());

            updateImg(nbErrors);

            TextView message=findViewById(R.id.message);

                        if (wordFound()) {

                Toast.makeText(context, R.string.you\_win, Toast.LENGTH\_SHORT).show();

                message.setText(R.string.you\_win);

            } else {

                if (nbErrors >= MAX\_ERRORS) {

                    Toast.makeText(this, R.string.you\_lose, Toast.LENGTH\_SHORT).show();

                    message.setText(R.string.you\_lose);

                }

            }

        } else {

            Toast.makeText(this, R.string.game\_is\_ended, Toast.LENGTH\_SHORT).show();

        }

    }

    private void enter(String c) {

                if (!letters.contains(c)) {

                        if (wordToFind.contains(c)) {

                int index = wordToFind.indexOf(c);

                while (index >= 0) {

                    wordFound[index] = c.charAt(0);

                    index = wordToFind.indexOf(c, index + 1);

                }

            } else {

                                nbErrors++;

                Toast.makeText(this, R.string.try\_an\_other, Toast.LENGTH\_SHORT).show();

            }

                        letters.add(c);

        } else {

            Toast.makeText(this, R.string.letter\_already\_entered, Toast.LENGTH\_SHORT).show();

        }

    }

        private String wordFoundContent() {

        StringBuilder builder = new StringBuilder();

        for (int i = 0; i < wordFound.length; i++) {

            builder.append(wordFound[i]);

            if (i < wordFound.length - 1) {

                builder.append(" ");

            }

        }

        return builder.toString();

    }

    private void updateImg(int play) {

        int resImg = getResources().getIdentifier("hangman" + play, "drawable",

                getPackageName());

        img.setImageResource(resImg);

    }

}

Output

