Based on what you learnt this week and the material provided to you, create a dice game. The game contains two dice, one belongs to the player and one belonging to the computer. Dice has to place vertically and has to be identified which one belongs to the user and belongs to the computer.

* Your app has to have two buttons. One said “Higher,” the second says “Lower” it has to be placed horizontally
* By pressing each of these buttons, both dice have to roll.
* If the user selects “higher” and his/her dice is higher than the computer dice; then the user is the winner; otherwise, he or she loses unless the game is a tie.
* If the user selects “lower” and his/her dice is lower than the computer, the user is the winner; otherwise, they lose unless the game is a tie.

Considering the situation, one of the following messages has to be shown.

Computer Win!

User Win!

It’s a tie

You have two options for the message: 1) using an empty text field 2) You can have an image to indicate the situation

\*\* Make sure to apply the id, description and all the necessary attributes in style.xml, string.xml and color.xml.

\*\*make sure that you place the resources into the right folders.

Upload your project on Github and share the link by next week

If you don’t know how to push your project into GitHub, go to this link:

<https://www.geeksforgeeks.org/how-to-upload-project-on-github-from-android-studio/>

DUE DATE FEB 19th before class starts.

**Rubric**

|  |  |  |
| --- | --- | --- |
| **Description** | Possible Mark |  |
| **The building, installing and launching the app** | 30% |  |
| **Code Quality** | 10% |  |
| **Proper implementation of ID** | 10% |  |
| **Proper use of color.xml, strings.xml and style.xml** | 10% |  |
| **Proper layout and android device compatibility** | 20% |  |
| **Proper implementation of resources** | 10% |  |
| **Proper implantation of mipmap (icon has to support square and circle)** | 10% |  |
| **Using API (Extra credit 10%)** |  |  |
|  | 100% |  |

Github Project Url:

https : <https://github.com/samanth93/DiceGame.git>

SSH: [git@github.com:samanth93/DiceGame.git](mailto:git@github.com:samanth93/DiceGame.git)

GitHub CLI: gh repo clone samanth93/DiceGame

Paste your java code here//

**package** com.example.dicegame;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.ImageView;  
**import** android.widget.TextView;  
  
**import** java.util.Random;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
*// we have two images left dice and right dice* ImageView topDice = findViewById(R.id.***top\_dice***);  
 ImageView bottomDice = findViewById(R.id.***bottom\_dice***);  
 Button rollHigherButton = findViewById(R.id.***higher\_btn\_roll***);  
 Button rollLowerButton = findViewById(R.id.***lower\_btn\_roll***);  
 TextView winnerString = findViewById(R.id.***winner\_string***);  
 **int**[] diceArray = {R.drawable.***dice1***,  
 R.drawable.***dice2***,  
 R.drawable.***dice3***,  
 R.drawable.***dice4***,  
 R.drawable.***dice5***,  
 R.drawable.***dice6***};  
*// click on higher button* rollHigherButton.setOnClickListener(**new** View.OnClickListener(){  
 @Override  
 **public void** onClick(View v) {  
 Random rand = **new** Random();  
 **int** randTopDie = rand.nextInt(6);  
 **int** randBottomDie = rand.nextInt(6);  
 Log.*d*(**"left die"**, String.*valueOf*(randTopDie));  
 Log.*d*(**"right die"**, String.*valueOf*(randBottomDie));  
 **if**(randTopDie > randBottomDie){  
 Log.*d*(**"Higher Button"**, **"Computer Won"**);  
 winnerString.setText(**"COMPUTER WIN!"**);  
 }**else if**(randTopDie == randBottomDie){  
 Log.*d*(**"Higher Button"**, **"Tie"**);  
 winnerString.setText(**"IT'S A TIE"**);  
 }**else**{  
 Log.*d*(**"Higher Button"**, **"User Won"**);  
 winnerString.setText(**"USER WIN!"**);  
 }  
 topDice.setImageResource(diceArray[randTopDie]);  
 bottomDice.setImageResource(diceArray[randBottomDie]);  
 }  
 });  
*// click on lower button* rollLowerButton.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Random rand = **new** Random();  
 **int** randTopDie = rand.nextInt(6);  
 **int** randBottomDie = rand.nextInt(6);  
 Log.*d*(**"left die"**, String.*valueOf*(randTopDie));  
 Log.*d*(**"right die"**, String.*valueOf*(randBottomDie));  
 **if**(randTopDie > randBottomDie){  
 Log.*d*(**"Higher Button"**, **"User Won"**);  
 winnerString.setText(**"USER WIN!"**);  
 }**else if**(randTopDie == randBottomDie){  
 Log.*d*(**"Higher Button"**, **"Tie"**);  
 winnerString.setText(**"IT'S A TIE"**);  
 }**else**{  
 Log.*d*(**"Higher Button"**, **"Computer Won"**);  
 winnerString.setText(**"COMPUTER WIN!"**);  
 }  
 topDice.setImageResource(diceArray[randTopDie]);  
 bottomDice.setImageResource(diceArray[randBottomDie]);  
 }  
 });  
 }  
}

Paste the snapshot of your simulator hereGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated