Google Applied CS with Android

Scarne's Dice

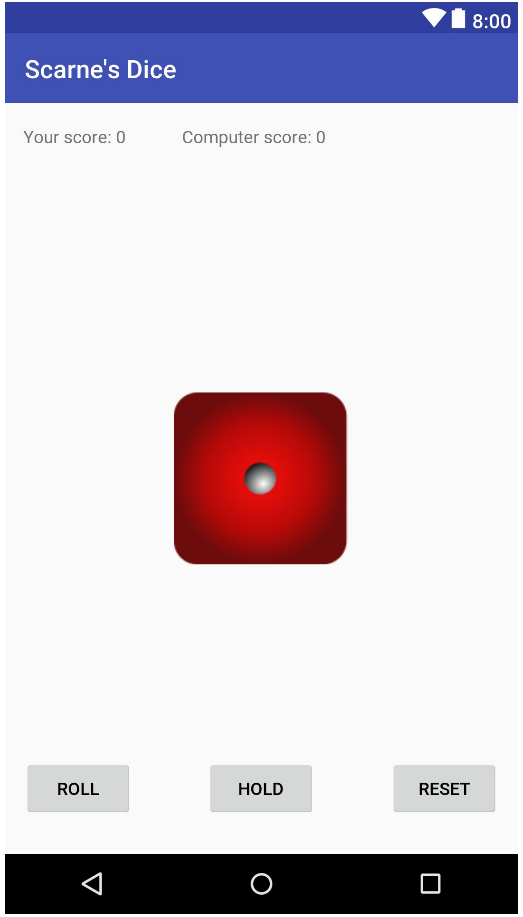
Anupam Das
Google Android with CS facilitator

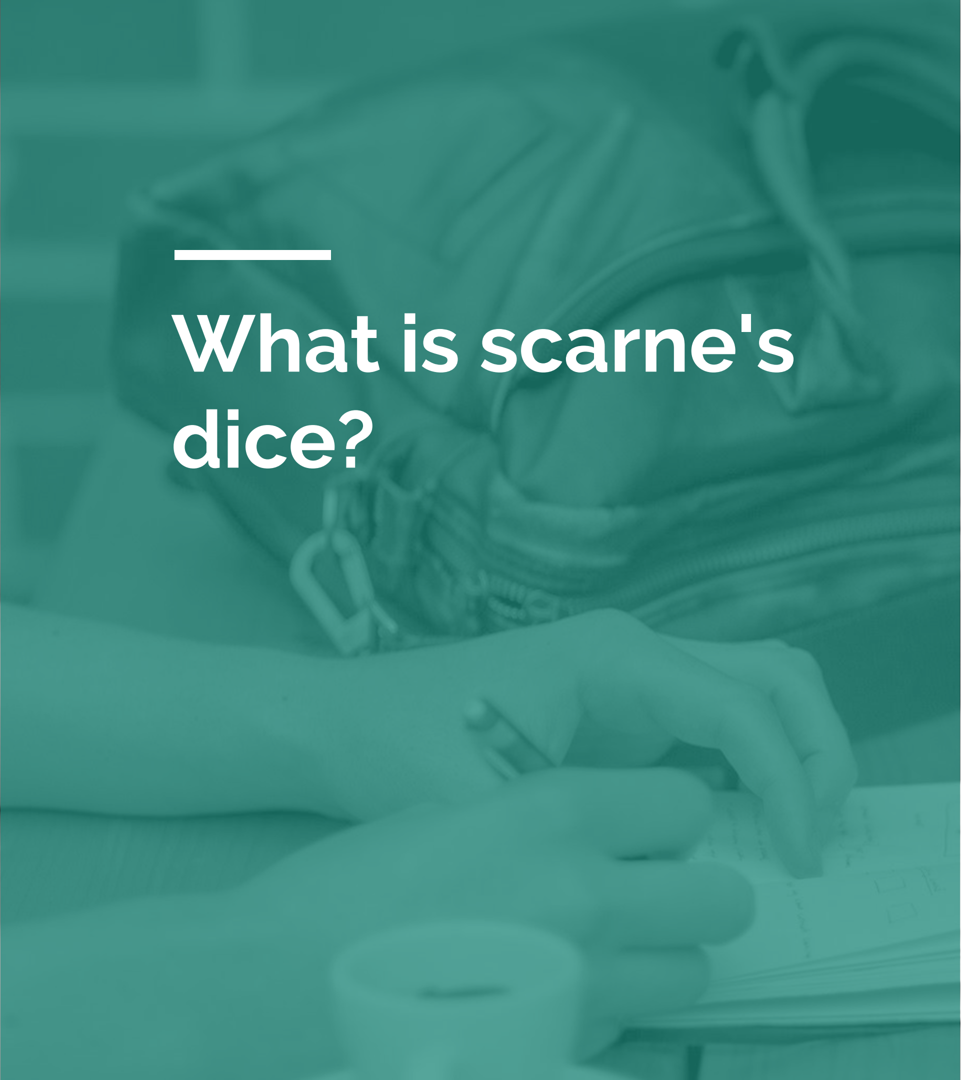
What is Android ?

Android is a mobile operating system developed by Google.

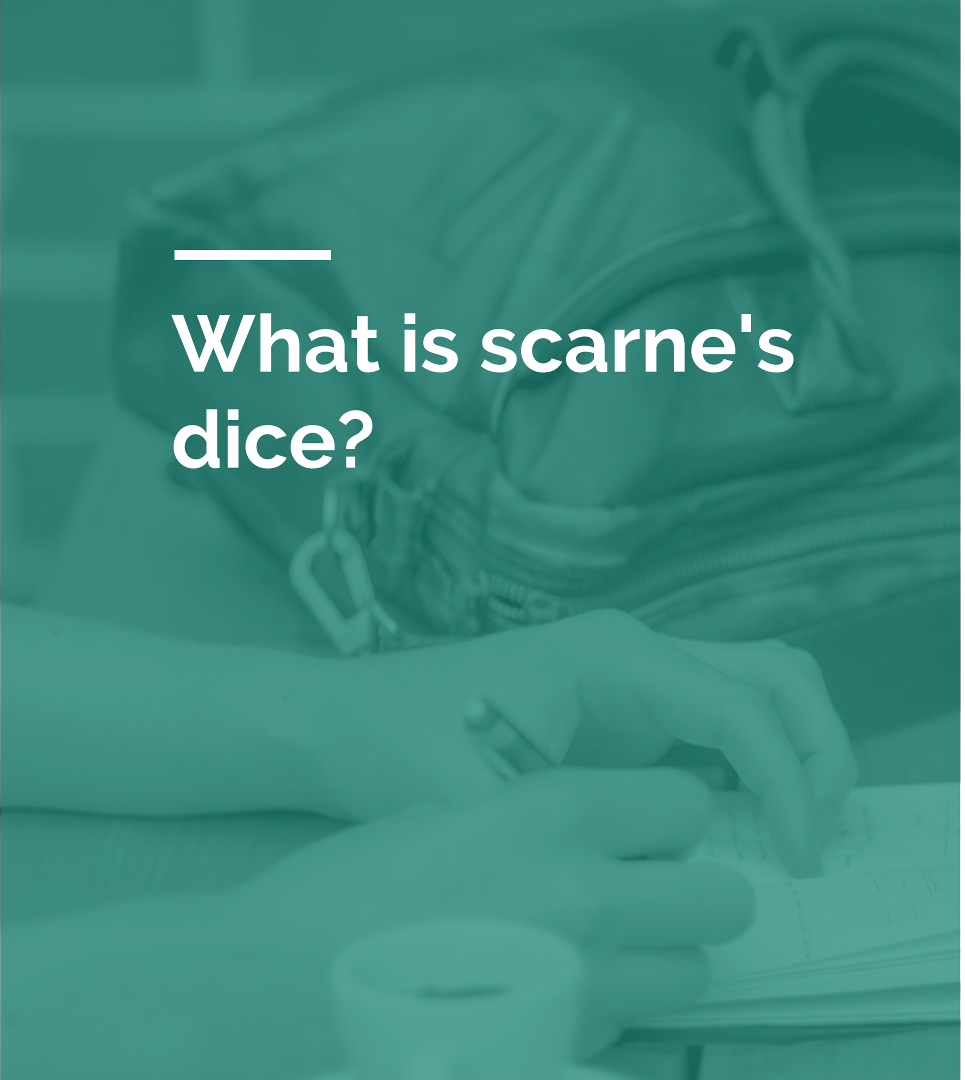
Google

Creating an Android app for scarne's dice.





What is scarne's dice?

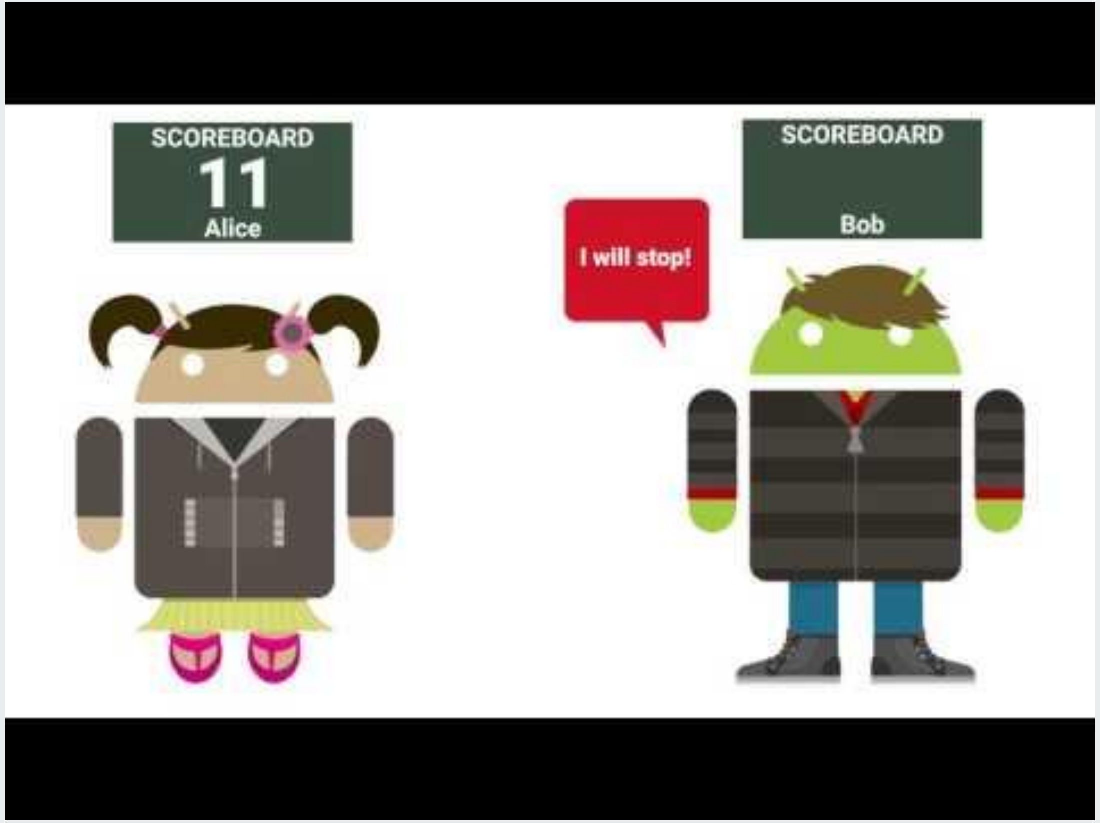


What is scarne's dice?

Scarne’s Dice is a turn-based dice game where players score points by rolling a die and then:

- if they roll a 1, score no points and lose their turn
- if they roll a 2 to 6:
  - add the rolled value to their points
  - choose to either reroll or keep their score and end their turn

The winner is the first player that reaches (or exceeds) 100 points.


SCOREBOARD
11
Alice
I will stop!
SCOREBOARD
Bob

Requirements

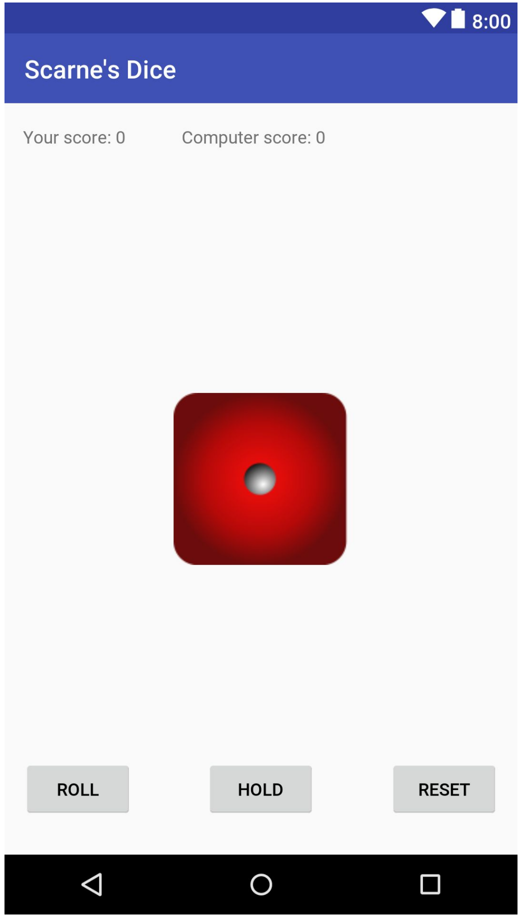
1. Laptop / PC with proper internet connectivity and power.
2. Log in to https://cswithandroid.withgoogle.com/
3. Follow my steps.

Milestone 1:

Implementing the UI

The UI is composed of:

- A *TextView* to display the score and status of the game
- An *ImageView* to display the current die (default to the image of your choice)
- Three *buttons* to either roll the die, end your turn or start over



TextView

Relative Layout

```
<TextView
   android:id="@+id/textView"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_alignParentLeft="true"
   android:layout_alignParentStart="true"
   android:layout_alignParentTop="true"
   android:layout_marginLeft="15dp"
   android:layout_marginStart="15dp"
   android:layout_marginTop="17dp"
   android:text="Your score: 0 computer
score: 0" />
```

Sample Text View

ImageView

Relative Layout

```
<ImageView
   android:id="@+id/imageView"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_centerHorizontal="true"
   android:layout_centerVertical="true"
   app:srcCompat="@drawable/dice1" />
```

Sample Image View

Button

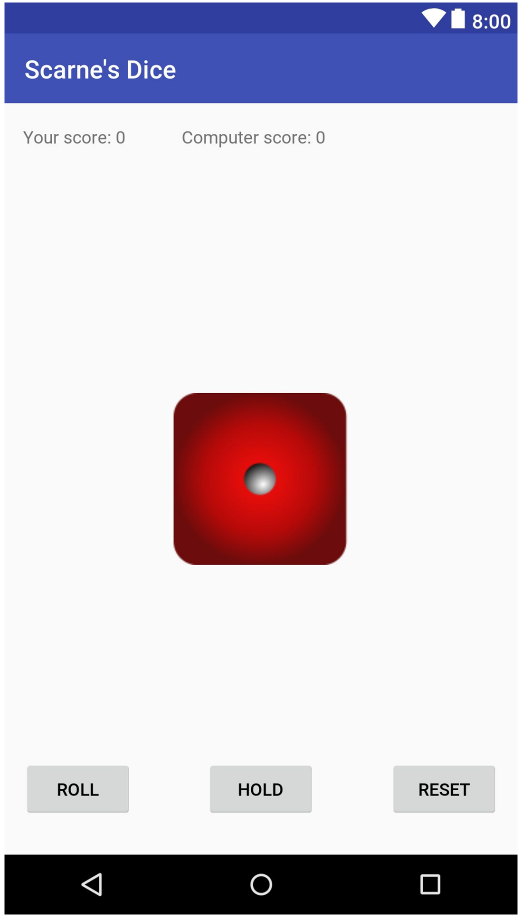
Relative Layout

```
<Button
  android:id="@+id/button_roll"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_alignLeft="@+id/textView"
  android:layout_alignParentBottom="true"

android:layout_alignStart="@+id/textView"
  android:layout_marginBottom="29dp"
  android:text="ROLL" />
```

Sample Button

Hurrah! Milestone 1 completed

Now you have completed the UI.



Milestone 2

Global variables

Four global variables to store:

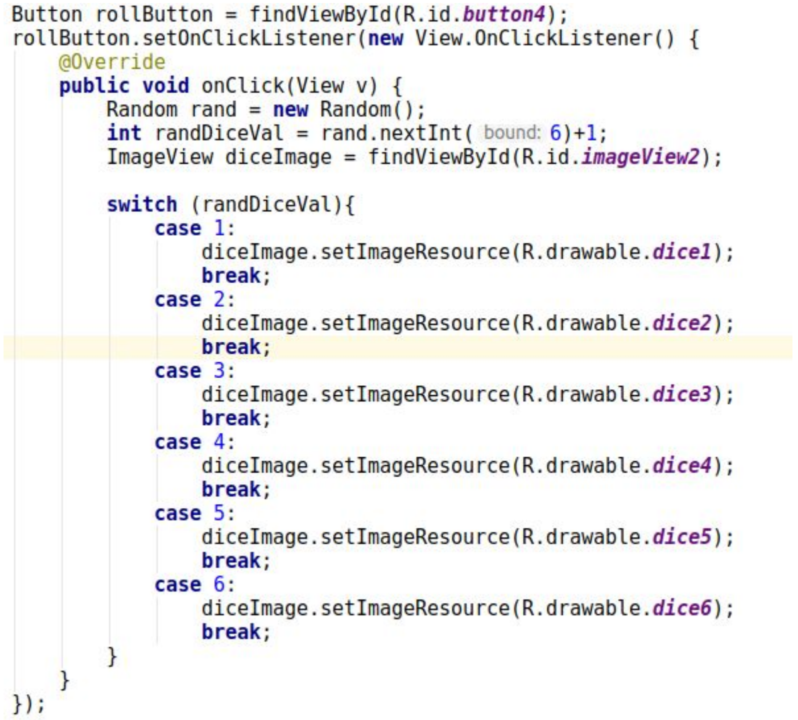
- the user's overall score state
- the user's turn score
- the computer's overall score
- the computer's turn score

```
long userOverallScore;

long userTurnScore;

long computerOverallScore;

long computerTurnScore;

userOverallScore = 0; userTurnScore = 0;

computerOverallScore = 0; computerTurnScore = 0;
```

onClickListener

A onClickListener for the "Roll" button that will:

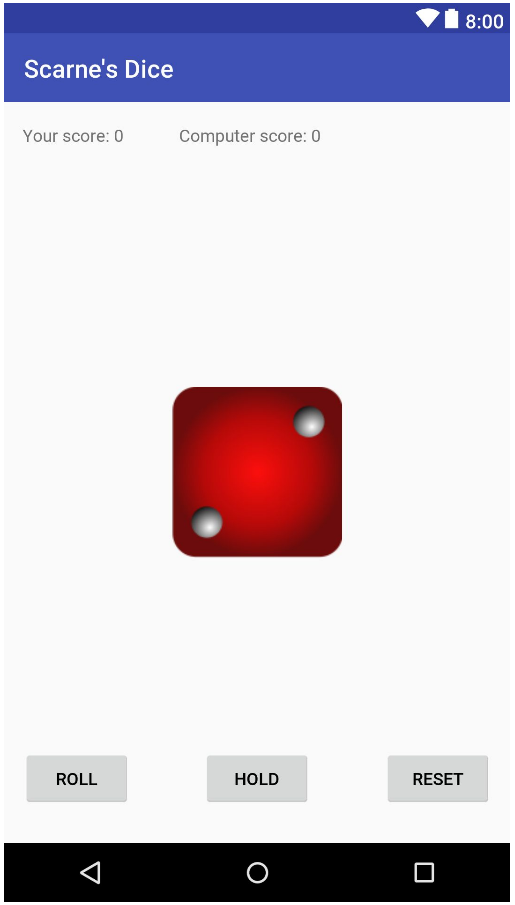
- randomly select a dice value
- update the display to reflect the rolled value



```java
Button rollButton = findViewById(R.id.button4);
rollButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Random rand = new Random();
        int randDiceVal = rand.nextInt( bound: 6)+1;
        ImageView diceImage = findViewById(R.id.imageView2);

        switch (randDiceVal){
            case 1:
                diceImage.setImageResource(R.drawable.dice1);
                break;
            case 2:
                diceImage.setImageResource(R.drawable.dice2);
                break;
            case 3:
                diceImage.setImageResource(R.drawable.dice3);
                break;
            case 4:
                diceImage.setImageResource(R.drawable.dice4);
                break;
            case 5:
                diceImage.setImageResource(R.drawable.dice5);
                break;
            case 6:
                diceImage.setImageResource(R.drawable.dice6);
                break;
        }
    }
});
```

Hurrah! Milestone 2 completed

Now the dice changes with roll.



Milestone 3

Implementing Game Logic

RollButton functionalities:

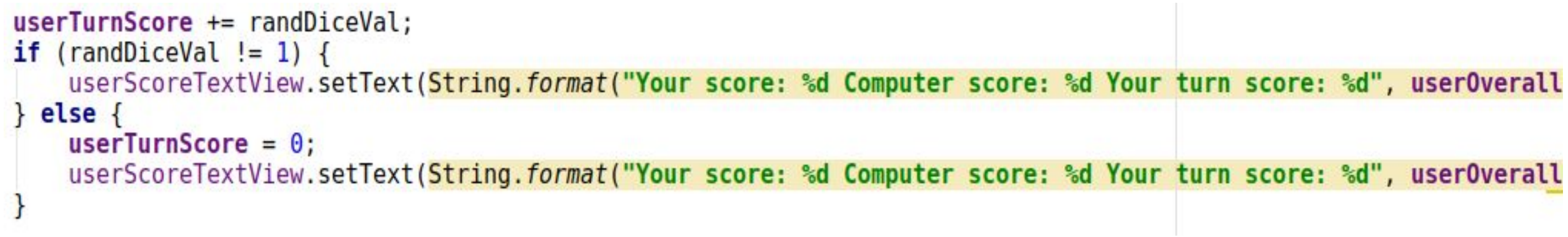
If the roll is not a 1:

- update the user's turn score by the value of the roll
- update the label to "Your score: 0 Computer score: 0 your turn score: X".

If the roll is a 1:

- reset the turn score to 0
- update the label.

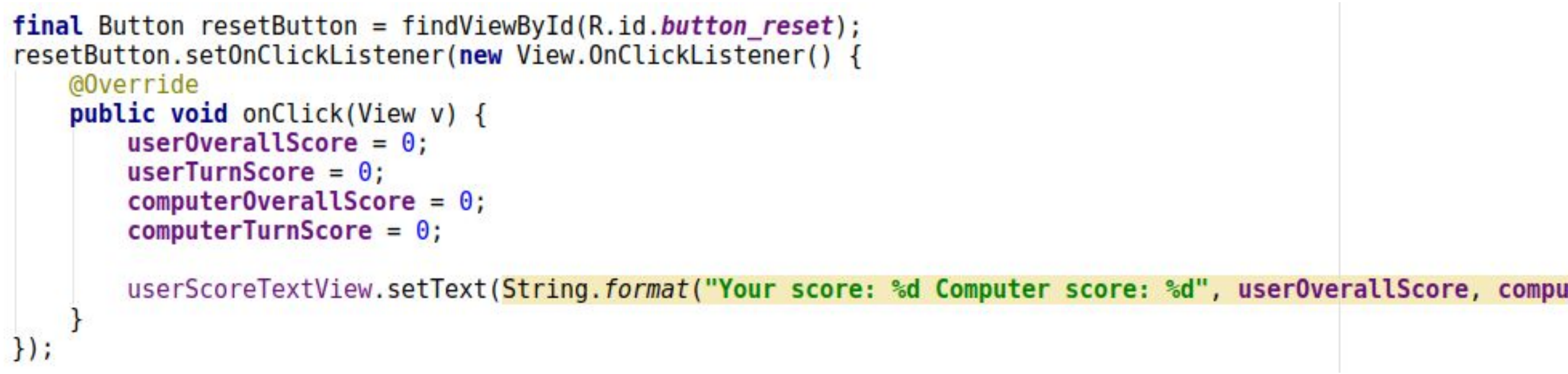
Implementing Game Logic (Continues)



```
userTurnScore += randDiceVal;
if (randDiceVal != 1) {
    userScoreTextView.setText(String.format("Your score: %d Computer score: %d Your turn score: %d", userOverall
} else {
    userTurnScore = 0;
    userScoreTextView.setText(String.format("Your score: %d Computer score: %d Your turn score: %d", userOverall
}
```

Implementing Game Logic (Continues)

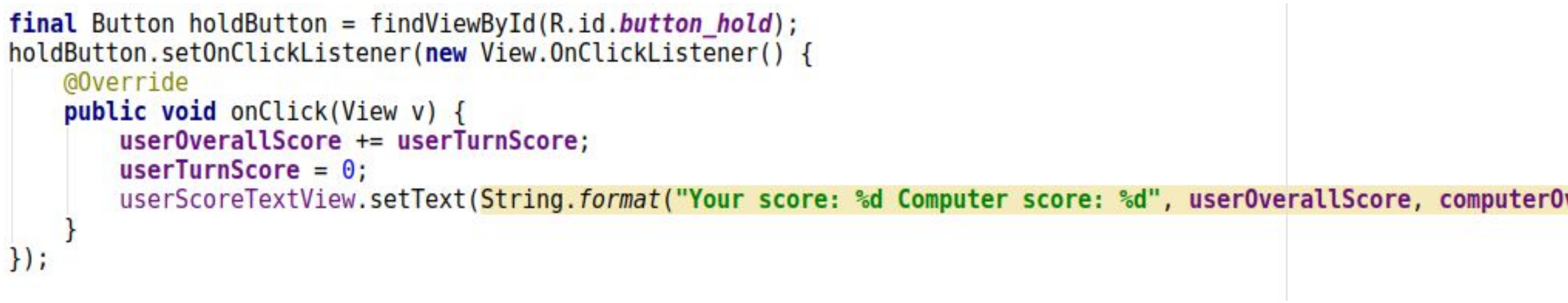
ResetButton functionalities:



```
final Button resetButton = findViewById(R.id.button_reset);
resetButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        userOverallScore = 0;
        userTurnScore = 0;
        computerOverallScore = 0;
        computerTurnScore = 0;

        userScoreTextView.setText(String.format("Your score: %d Computer score: %d", userOverallScore, compu
    }
});
```

Implementing Game Logic (Continues)

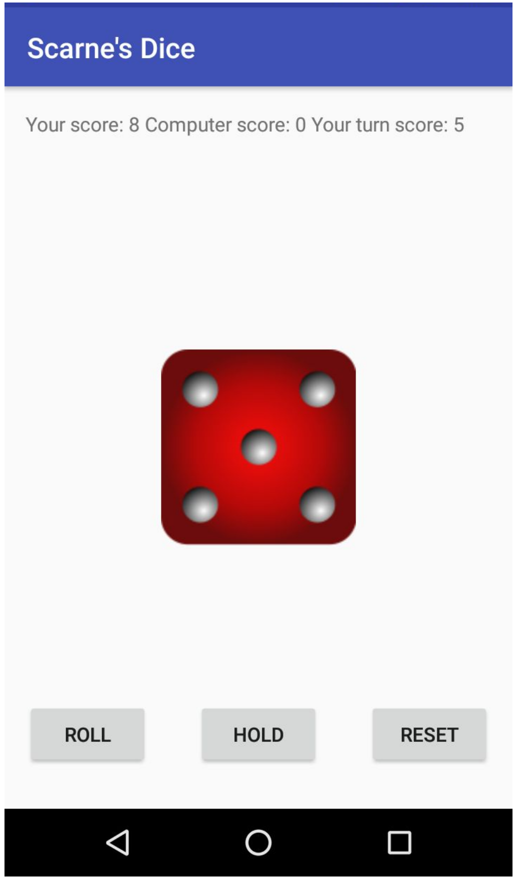
HoldButton functionalities:



```
final Button holdButton = findViewById(R.id.button_hold);
holdButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        userOverallScore += userTurnScore;
        userTurnScore = 0;
        userScoreTextView.setText(String.format("Your score: %d Computer score: %d", userOverallScore, computerO
    }
});
```

Hurrah! Milestone 3 completed

At this stage, the basic user turn functionality is in place.


Scarne's Dice

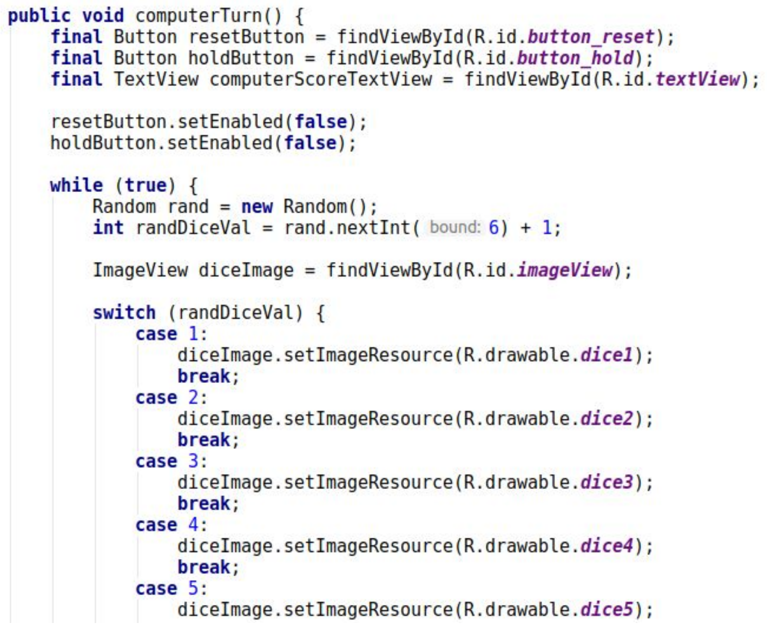
Your score: 8 Computer score: 0 Your turn score: 5

Milestone 4

Implementing computerTurn

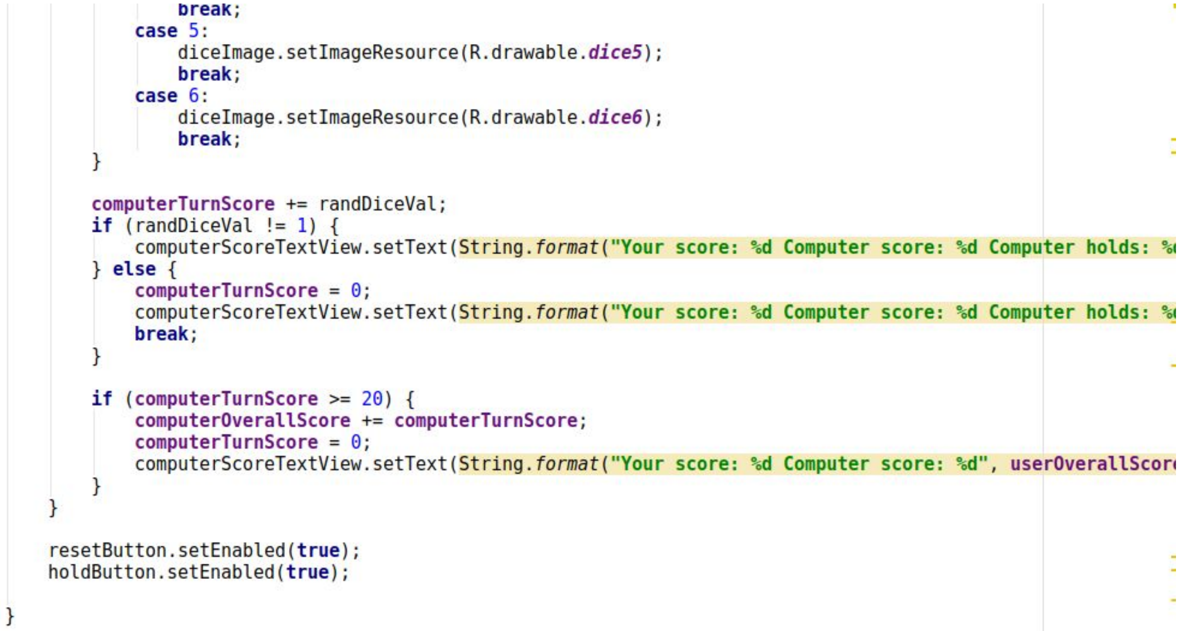
- Disable the roll and hold buttons
- Create a while loop that loops over each of the computer's turn. During each iteration of the loop:
  - pick a random die value and display it
  - follow the game rules depending on the value of the roll.

Implementing computerTurn (continues)



```
public void computerTurn() {
    final Button resetButton = findViewById(R.id.button_reset);
    final Button holdButton = findViewById(R.id.button_hold);
    final TextView computerScoreTextView = findViewById(R.id.textView);

    resetButton.setEnabled(false);
    holdButton.setEnabled(false);

    while (true) {
        Random rand = new Random();
        int randDiceVal = rand.nextInt( bound: 6) + 1;

        ImageView diceImage = findViewById(R.id.imageView);

        switch (randDiceVal) {
            case 1:
                diceImage.setImageResource(R.drawable.dice1);
                break;
            case 2:
                diceImage.setImageResource(R.drawable.dice2);
                break;
            case 3:
                diceImage.setImageResource(R.drawable.dice3);
                break;
            case 4:
                diceImage.setImageResource(R.drawable.dice4);
                break;
            case 5:
                diceImage.setImageResource(R.drawable.dice5);
```

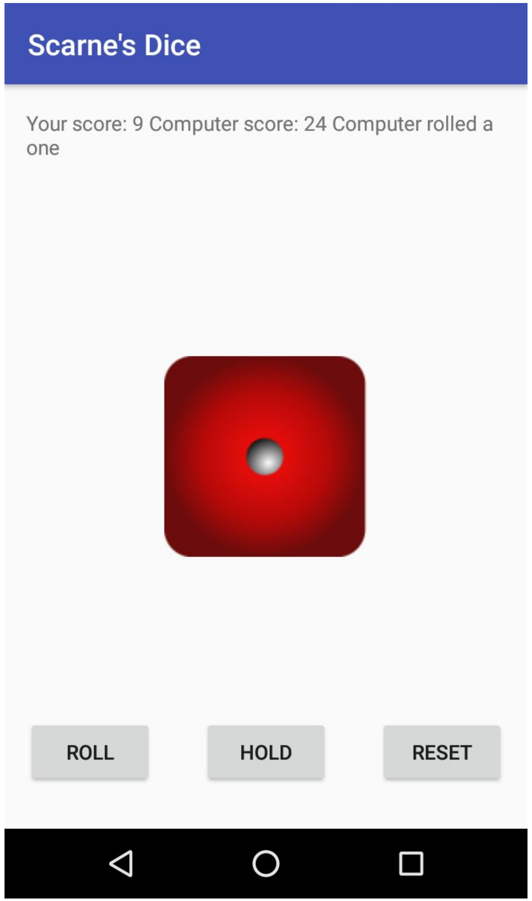
Implementing computerTurn (continues)



```
                break;
            case 5:
                diceImage.setImageResource(R.drawable.dice5);
                break;
            case 6:
                diceImage.setImageResource(R.drawable.dice6);
                break;
        }

        computerTurnScore += randDiceVal;
        if (randDiceVal != 1) {
            computerScoreTextView.setText(String.format("Your score: %d Computer score: %d Computer holds: %
        } else {
            computerTurnScore = 0;
            computerScoreTextView.setText(String.format("Your score: %d Computer score: %d Computer holds: %
            break;
        }

        if (computerTurnScore >= 20) {
            computerOverallScore += computerTurnScore;
            computerTurnScore = 0;
            computerScoreTextView.setText(String.format("Your score: %d Computer score: %d", userOverallScor
        }
    }

    resetButton.setEnabled(true);
    holdButton.setEnabled(true);

}
```

Hurrah! Milestone 4 completed

At this stage, you have an working app.


Scarne's Dice

Your score: 9 Computer score: 24 Computer rolled a one

Milestone 5

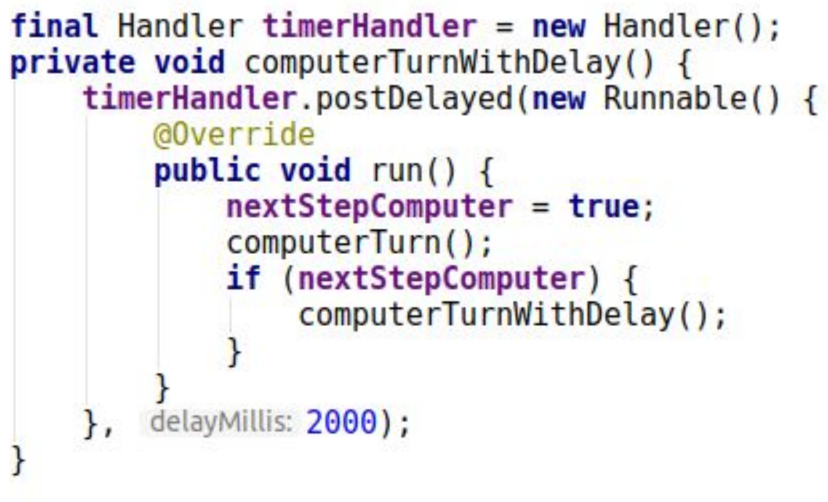
Refactoring

You may find the computer turn to be quite hard to follow as it happens so quickly that you can hardly see the die rolls and the label updates.

Use *Handler.postDelayed()*

Refactoring (continues)

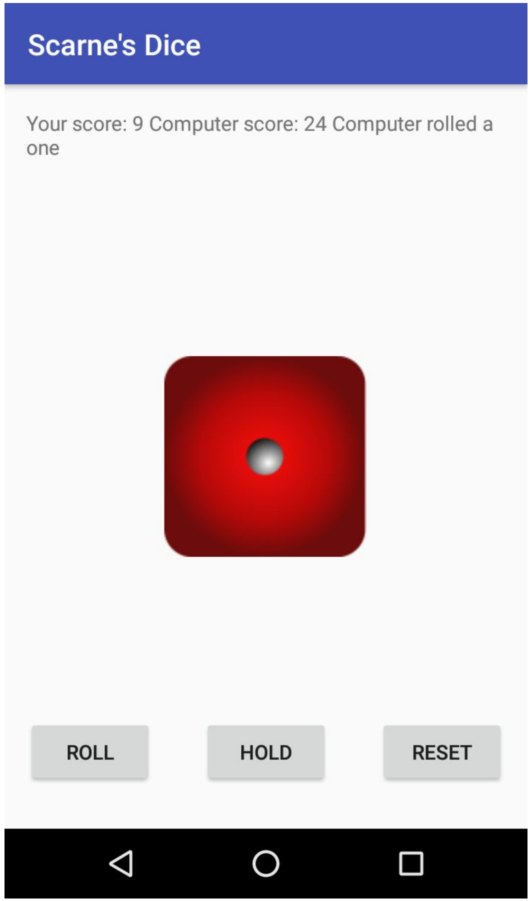
Use boolean check and nextStepComputer = true;



```
final Handler timerHandler = new Handler();
private void computerTurnWithDelay() {
    timerHandler.postDelayed(new Runnable() {
        @Override
        public void run() {
            nextStepComputer = true;
            computerTurn();
            if (nextStepComputer) {
                computerTurnWithDelay();
            }
        }
    }, delayMillis: 2000);
}
```

Hurrah! Milestone 5 completed

Now you have an fully working, faster and simple scarne's dice game app.



Your score: 9 Computer score: 24 Computer rolled a one

References

- Google Applied CS Skills Course Materials
  https://appliedcsskills.withgoogle.com/unit?unit=6&lesson=8
- Google Students Video Links:
  - https://youtu.be/r65MpuDkbh0

Questions?

Thanks!

Feel free to share what you have learned in social media (Facebook, Google +, Twitter.) and with your friends.

Anupam Das
opticod

Google Android with CS Facilitator

http://opticod.com/