Step 1 create a new project

Select empty activity, we’ll start by making a screen with a button that the user can click to go the game activity.

Step 2 back in the main activity find the button, set the on click🡪 playButton.setOnClickListener(this);

Then change the public class mainactivity to implement OnClickListner

Then create the onClick method for button after the onCreate

In the onClick we need to check if the user actually pressed the play button so we use an if statement

if (view.getId() == R.id.playbutton) {

  Intent playbuttonIntent = new Intent(this, HangManGameActivity.class);

  this.startActivity(playbuttonIntent);

}

Step 3 create the second activity that will be the game activity

Step 4

We need store the answers in the xml, in the project resources values folder (you can add more)

<resources>

  <string-array name="words">

    <item>HELLO</item>

  </string-array>

</resources>

Step 5

Back in the game layout we a way to choose a word, where the user can type the answer in, you can use a plain text

Step 6 in the game activity

Start by declaring your variables such as the variable that will hold the array of words.

After that start by need to make a instance of variable that hold the words

Such as🡪 Resources resources = getResources();

randwords = resources.getStringArray(R.array. randwords);

step 7

after the on create

then we the variable that has our words 🡪 words = res.getStringArray(R.array.words);

we need ensure the app will start with a random word so initialize a new random random = new Random

create a string to hold current words 🡪 currentwords=?

Find the view for the answer edit text

Step 8 create a method that we will call, private void playGame(){

}

In this method add a method that ensure the same words wont repeat

🡪

while(newWord.equals(currentword)) newWord = words[rand.nextInt(words.length)];

update the currentword= newWord;

create a text view for each letter for the target word

charactviews = new Textview[CurrentWord.length]();

step 9

we need to remove all textviews removeAllviews();

next use a for loop to iterate over each letter of the awnser, create a text view for each letter and set the text view to the current letter.

for (int c = 0; c < currentWord.length(); c++) {

  charactViews[c] = new TextView(this);

  charactViews[c].setText(""+currWord.charAt(c));

 charactViews [c].setLayoutPara(new LayoutPara(LayoutPara.WRAP\_CONTENT, charactViews.WRAP\_CONTENT));

  charactViews [c].setGravity(Gravity.CENTER);

  charactViews [c].setTextColor(Color.WHITE);

  charactViews [c].setBackgroundResource(R.drawable.letter\_bg);

  //add to layout

  wordLayout.addView(charactViews [c])}

back in the on create call the method

step 10

we a method to know if the user guessed the word correctly you can use a for each loop

step 11 check if the user won use an if statement, after we need another if statement to check if the user guessed all the letters of the target word. If its true then display a message for the user.

Step 11 we need if statement incase the user lost after the first if use an else if statement and display a losing message.

// steps for the animation and transition

For the default transition place the code inside the button or method that will load the second screen

Here is a code that has the fade transition

overridePendingTransition(android.R.anim.fade\_in, android.R.anim.fade\_out);

//animation you would need to create an animation resource folder then inside create a animation resource and name it according to its purpose after that you would need add the code that will execute the animation -🡪 I’ll a rotation method here 🡪 **<rotate  
 android:fromDegrees="0"  
 android:toDegrees="360"  
 android:pivotX="50%"  
 android:pivotY="50%"  
 android:duration="500" />  
</set>**

After you create the resource go back in your activity class and declare the animation variable like so rotateAnimation right beneath the activity extends

After that you would to initialize it in oncreate rotateAnimation = AnimationUtils.LoadAnimation( this, R.anim.rotate);

And finally add the code that will start in within the button/method you want to use --🡪 v.startAnimation(rotateAnimation);