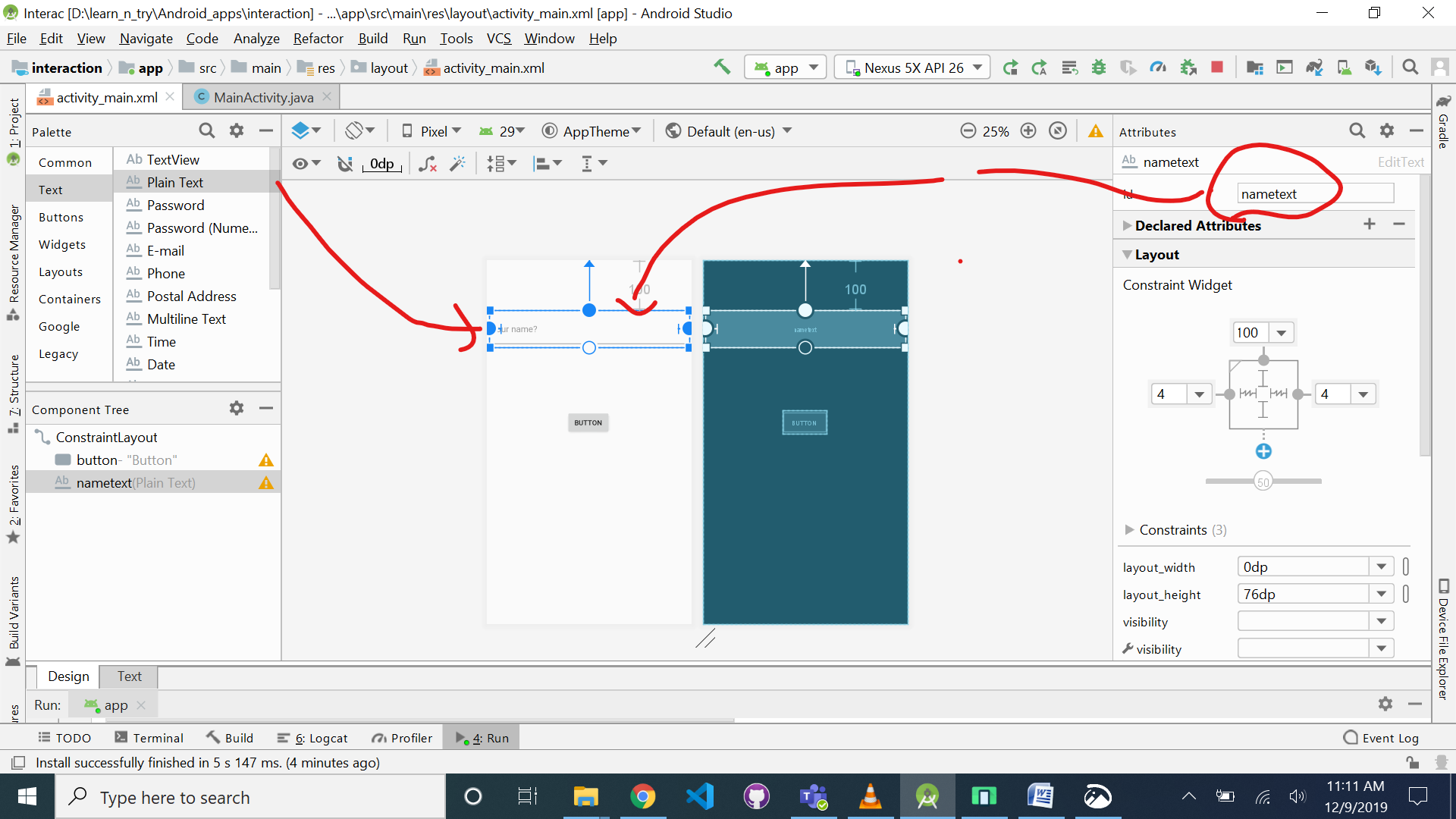
****

**In main activity add click function which is the onClick function for our button.**

**public void** click(View view)  
{  
 EditText nameEditText = (EditText) findViewById(R.id.***nametext***);  
 Log.*i*(**"Holaaa"**,**"Welcome"**);  
 Log.*i*(**"text"**, nameEditText.getText().toString());  
}

To give alternate chance to players and not letting overwrite each other’s move, follow this code.

Use imageView.getDrawable()!=null

**package** com.example.connect3;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.ImageView;  
**import** android.widget.Toast;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 **boolean re**=**true**;  
 **public void** dropIn(View view){  
 ImageView counter = (ImageView)view;  
 **if**(counter.getDrawable()==**null**) {  
 counter.setTranslationY(-1500);  
 **if** (**re**) {  
 counter.setImageResource(R.drawable.***red***);  
 **re** = **false**;  
 } **else** {  
 counter.setImageResource(R.drawable.***yellow***);  
 **re** = **true**;  
 }  
 counter.animate().translationYBy(1500).setDuration(1000);  
 }  
 }  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 }  
}

If resource name is required follow this link -<https://stackoverflow.com/questions/23357706/how-to-check-which-current-image-resource-is-attached-to-imageview-in-android-xm>

**To import a video, first make a folder “raw” inside res and paste video in it. Then create a VideoView and write this code in OnCreate method :-**

VideoView vid = (VideoView ) findViewById(R.id.videoView);

vid.setVideoPath(“android.resource://”+getPackageName()+”/”+R.raw.demovideo);

*//ADD MEDIA CONTROLLER TO ADD PAUSE PLAY SEEK OPTION:*

MediaController med = new MediaController(this);

med.setAnchorView(vid);

vid.setMediaController(med);

vid.start();

**TO run an Audio, we don’t have to add any view, just adding media controller will do.**

MediaPlayer med = MediaPlayer.create(context:this,R.raw.demoaudio);

med.start();

//to add controls, u have to add buttons on your own and add this function to play button function…

//for seekbar, chk out audio seeking video mod -4 vid -14

//If filename is stored in some variable, or saved as tag of the button being tapped, we can use this code snippet :

**public void** play(View view){  
 Button btn= (Button) view;  
 Log.*i*(**"Button pressed"**,btn.getTag().toString());  
 MediaPlayer m = MediaPlayer.*create*(**this**, getResources().getIdentifier(btn.getTag().toString(), **"raw"**, getPackageName()));  
 m.start();  
}

in this code, we had different buttons which were supposed to play different audio on being tapped.

All the buttons had the tag set as their corresponding audio filenames. And to avoid writing separate functions for each, we used a generic func play which was called whenever a button was tapped.

The view shared by the button helped in playing the respective audio file.

The processes run on threads, there is this main thread where we usually put in all the functions.

But it is often advised to make use of other threads to run things simultaneously in the background as well, background thread comes real handy when downloading data.

Use AsyncTask

