

## Important Links

1. View Shapes
  2. Button Shapes
  3. imgbak
- 

## Images in android:

- Bitmap (raster) images:
    - Png,jpg,gif: lots of details and various colors
    - Depends on pixels.
    - Takes less time to draw.
  - Xml files:
    - Are drawn using math calculations.
    - They have the same quality on screens with different screen density and they have the same quality on screens with different sizes.
  - Svg (scalable vector graphics) files:
    - Are drawn using math calculations.
    - They have the same quality on screens with different screen density and they have the same quality on screens with different sizes.
    - Takes more time to draw because calculations need time.
- 

**Notes:** for images that do not have lots of details and colors, try to look for SVG images.

If you could not find one,try converting from your bitmap image to SVG. if the quality is what you want, then use the converted SVG.

Otherwise, use the raster (bitmap) image.

---

**SVG is best for images that have two conditions:**

1. little details and small variety colors.
2. Small images such as icons.

## Event Handling:

- Events: the action on a view
  - Event Listener: the components that waits and listen for the event to occur
  - Event Handler: the component that says what happens when the event occurs
- 

## Fragment vs Activity:

- Activity is normally used as the entry point of a program.
- Activity is used for main screens that have no navigation relationship.
- Activity cannot be reused.

	Activity	Fragment
Used as entry point of application	yes	no
Main screen in application (go from it to many other screens)	yes	no
Code reusability	no	yes
How heavy	Heavy weight	Light weight
Lives by itself	yes	No, it should exit in another activity or fragment

**Rule:** use activity as entry point and in main screens from which we can navigate to other destinations. Otherwise, use fragments.

**Fragments & JetPack Navigation:** we use jetpack navigation components to navigate from one screen to another.

1. Create 3 fragments.
2. Add the **jetpack navigation component Dependencies** on **build.gradle**
3. Create **Navigation graph**: resources (right click → new → Android Resource File(resource type = **Navigation**))
4. Add from Containers : **FragmentContainerView**: and inside of it the **navigation graph**
5. Add the 3 fragment pages to the **Navigation graph**.
6. On the override View view = ..... **then** return view; **then** view.findViewById();
7. And to the button **onClickListener** =>  
Navigation.findNavController(view).navigate(R.id.fragment id on graph);

**Fragment & Fragment Manager:** we use fragment manager if we want to navigate between fragments inside activity.

1. Inside MainActivity put fragmentContainer inside of it put 3 fragments.
2. BlankFragment1 firstFrag = new BlankFragment1();
3. BlankFragment2 secondFrag = new BlankFragment2();
4. FragmentManager manager = getChildFragmentManager();
5. Button onClick => manager.beginTransaction().replace(R.id.Container Id,timerFrag).commit();
6. Private boolean mode = true; + Button onClick => mode != mode +  
if (mode)  
{  
    manager.beginTransaction().replace(R.id.Container Id,firstFrag).commit();  
}  
else  
{  
    manager.beginTransaction().replace(R.id.Container Id,secondFrag).commit();  
}

**Access innerFragmentValues from ParentFragment:**

**Inner Fragment.java:**

1. TextView tv = view.findViewById(R.id.text1);
2. String info = tv.getText().toString();
3. Bundle bundle = new Bundle();
4. bundle.putString("info",info);

5. setArguments(bundle);

**Parent Fragment.java:**

1. Fragment frag = manager.findFragmentById(R.id.InnerFragment);
2. Bundle bundle = frag.getArguments();
3. String info = bundle.getString("info");

**Action:** is the line between a fragment and another(Connector).

## Layouts

1. Constraint Layout
2. Linear Layout
  - a. Horizontal
  - b. Vertical

## Important View Design Properties

1. Gravity : start,center,end
2. Margin
3. Padding
4. Weight & Weight sum
5. Background: → res → drawable → shapeName

## Button Properties

1. Button → attributes → onClick → add method
2. MainActivity.java: Button Name.SetOnClickListener(new view)
3. Tag: give the button tag → if (tag.equals(the tag we give him)){ do this }

## Text Properties

1. setText
2. getText
3. toString