

STAGGERED VIEW IN ANDROID

PURPOSE: To implement a newer gridview called as Staggered view in android. This is to increase the visibility of the content inside the image rather than fixed size crops. The main difference between Staggered view and Grid view is that Staggered view is used when you are browsing through and u dont know what you are exactly looking for while Grid is used when you are navigating.

INTRODUCTION: Most of the smartphones currently used globally have Android pre-installed. Not only have the user's grown to such numbers but also the programmers who code to make better apps and add to the advancement of Android versions over the time. The Android has made the grid view as default for all. So, the decision was to develop a different view apart from grid which would make the view a bit more clear and catchy.

INTERFACE: The IDE used for the programming and testing of the code was Android Studio 1.3 which is pretty much the choice of all coders although, there is also Eclipse . The program source name is StaggeredExample.java

IMPLEMENTATION:

The program name is StaggeredExample.java which will be executed along with other required files.

garden aar dependency libraries has been used into the build. The custom tiles are coded to take dynamic height of the image or text itself.

Custom adapter class is defined to display the image view which is dynamic inside the staggered view. This all is put into the array and displayed accordingly

We have not put in the values for each of the tiles, thus, it won't redirect you to any page.

```
public class StaggeredExample extends AppCompatActivity implements  
AbsListView.OnScrollListener, AbsListView.OnItemClickListener{
```

```
private static final String SAVED_DATA_KEY = "Saved_Data" ;  
private static final String TAG = "StaggeredGridActivity" ;  
private StaggeredGridView mGridView;  
private SampleAdapter mAdapter;  
private boolean mHasMoreDataItem;  
private static int data_declaration;
```

```

ArrayList<String> mData;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_svg);
    mGridView = (StaggeredGridView) findViewById(R.id.grid_view);
    LayoutInflater inflater = getLayoutInflater();
    mAdapter = new SampleAdapter(this, android.R.layout.simple_list_item_1, generateData());
    if(savedInstanceState!=null){
        mData = savedInstanceState.getStringArrayList(SAVED_DATA_KEY);
        Toast.makeText(this, "Delecraed "+data_declartion++, Toast.LENGTH_SHORT).show();
    }
    if(mData==null){
        mData= generateData();
        Toast.makeText(this, "Delecraed "+data_declartion++, Toast.LENGTH_SHORT).show();
    }
    mGridView.setAdapter(mAdapter);
    mGridView.setOnScrollListener(this);
    mGridView.setOnItemClickListener(this);
}

```

```

private ArrayList<String> generateData() {
    ArrayList<String> listData = new ArrayList<String>();
    listData.add("http://i.movie.as/p/230507.jpg");
    listData.add("http://i.movie.as/p/230503.jpg");
    listData.add("http://i.movie.as/p/230501.jpg");
    listData.add("http://i.movie.as/p/230498.jpg");
    listData.add("http://i.movie.as/p/230497.jpg");
    listData.add("http://i.movie.as/p/230493.jpg");
    listData.add("http://i.movie.as/p/230482.jpg");
    listData.add("http://i.movie.as/p/230476.jpg");
    listData.add("http://i.movie.as/p/221644.jpg");
    listData.add("http://i.movie.as/p/219328.jpg");
    listData.add("http://i.movie.as/p/220612.jpg");
    listData.add("http://i.movie.as/p/217255.jpg");

    return listData;
}

```

```

@Override

```

```

    public void onSaveInstanceState(Bundle outState, PersistableBundle outPersistentState) {
        super.onSaveInstanceState(outState, outPersistentState);
        outState.putStringArrayList(SAVED_DATA_KEY,mData);

    }

    @Override
    public void onScrollStateChanged(AbsListView view, int scrollState) {

    }

    @Override
    public void onScroll(AbsListView view, int firstVisibleItem, int visibleItemCount, int
totalItemCount) {
        /*
        if(!mHasMoreDataItem){
            int lastInScreen = firstVisibleItem + visibleItemCount;
            if(lastInScreen<=totalItemCount){
                Log.d(TAG,"onScroll lastInScreen - so load More");
                onLoadItem();
            }
        }*/
    }

    private void onLoadItem() {
        final ArrayList<String> sampleData = generateData();
        for(String data : sampleData){
            mAdapter.add(data);
        }
        mData.addAll(sampleData);
        mAdapter.notifyDataSetChanged();
        Toast.makeText(this,"called",Toast.LENGTH_SHORT).show();
        mHasMoreDataItem=true;
    }

    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        Toast.makeText(this,"Data Clicked at "+position,Toast.LENGTH_SHORT).show();
    }
}

```

Simple Adapter:

An easy adapter to map static data to views defined in an XML file. You can specify the data backing the list as an ArrayList of Maps. Each entry in the ArrayList corresponds to one row in the list. The Maps contain the data for each row.

This is made use by a file called SampleAdapter.java. The code is given below

```
public class SampleAdapter extends ArrayAdapter<String> {
    private static final String LOG_TAG="SampleAdapter";
    Context mAppContext;
    private final LayoutInflater mLayoutInflater;
    private final Random mRandom;
    private static final SparseArray<Double> sPositionHeightRatio=new SparseArray<Double>();
    public SampleAdapter(Context context, int resource,ArrayList<String> object) {
        super(context, resource,object);
        mAppContext =context;
        this.mLayoutInflater = LayoutInflater.from(context);
        this.mRandom = new Random();
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        Toast.makeText(mAppContext, "inside getView", Toast.LENGTH_SHORT).show();
        ViewHolder vh;

        if(convertView==null)
        {
            convertView = mLayoutInflater.inflate(R.layout.row_staggered_layout,parent,false);
            vh =new ViewHolder();
            vh.imgView = (DynamicHeightImageView) convertView.findViewById(R.id.imgView);
            vh.txtView = (TextView) convertView.findViewById(R.id.footer_text_view);
            convertView.setTag(vh);
        }
        else{
            vh= (ViewHolder) convertView.getTag();
        }
        double heightRatio = getPositionRatio(position);

        vh.txtView.setText(getItem(position));
        final TextView textView = vh.txtView;
        vh.imgView.setHeightRatio(heightRatio);
        Glide.with(mAppContext).load(getItem(position)).asBitmap().into(vh.imgView);
        vh.imgView.buildDrawingCache();
    }
}
```

```

        Bitmap bmap = vh.imageView.getDrawingCache();
        if(bmap!=null) {
            Palette.from(bmap).maximumColorCount(15).generate(new
Palette.PaletteAsyncListener() {
                @Override
                public void onGenerated(Palette palette) {
                    Palette.Swatch vibrantSwatch = palette.getVibrantSwatch();
                    if (vibrantSwatch != null) {
                        textView.setBackgroundColor(vibrantSwatch.getRgb());
                    }
                }
            });
        }
        vh.txtView=textView;
        return convertView;
    }

```

```

private double getPositionRatio(int position) {
    double ratio = sPositionHeightRatio.get(position, 0.0);
    if(ratio==0){
        ratio= getRandomHeightRatios();
        sPositionHeightRatio.append(position,ratio);
        Log.d(LOG_TAG,"getPostionRatio:"+position+"ratio");
    }
    return ratio;
}

```

```

private double getRandomHeightRatios() {

    return (mRandom.nextDouble()/2.0)+1.0;
}

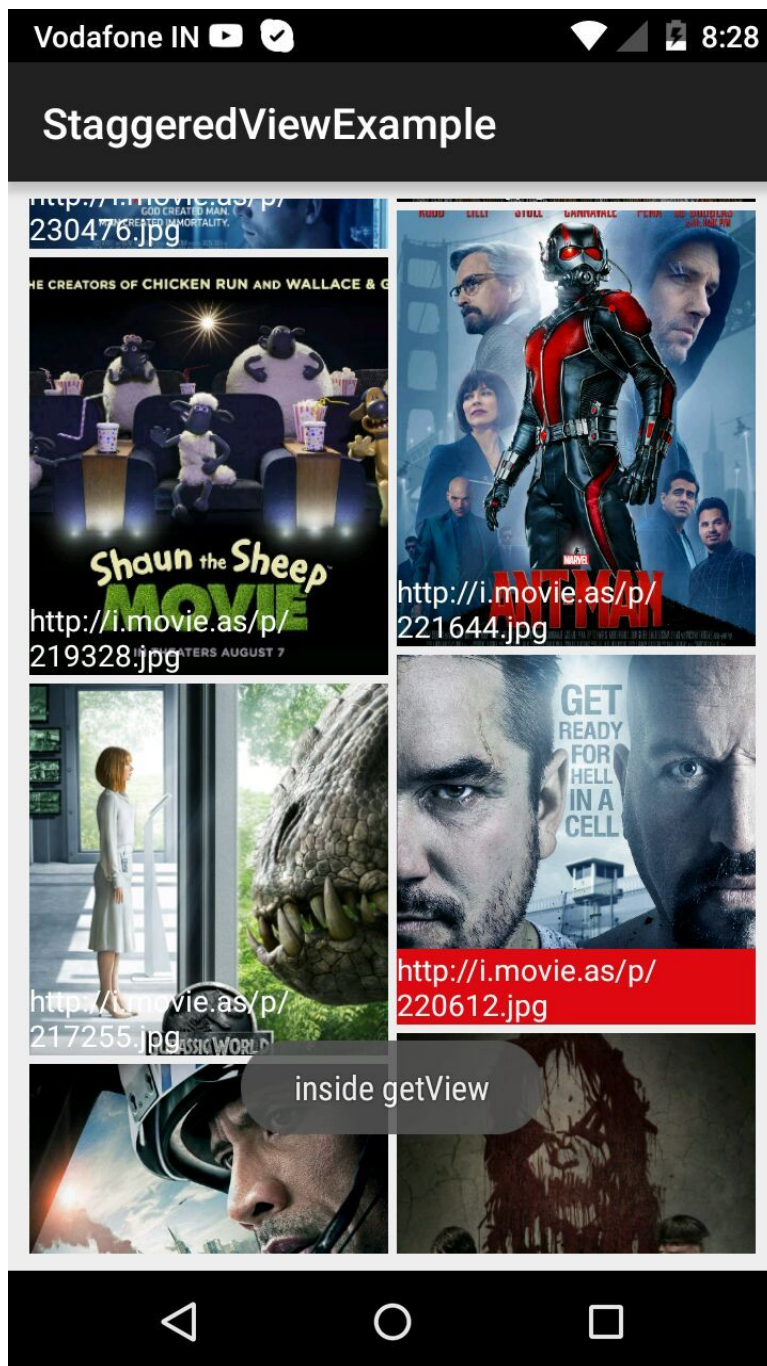
```

```

static class ViewHolder{
    DynamicHeightImageView imageView;
    TextView txtView;
}
}

```

OUTPUT:



StaggeredViewExample

