package com.example.acer\_pc.pokecats.gamedemo;  
import android.app.AlertDialog;  
import android.content.Context;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.graphics.Bitmap;  
import android.graphics.Canvas;  
import android.graphics.Paint;  
import android.graphics.Rect;  
import android.graphics.drawable.ColorDrawable;  
import android.graphics.drawable.Drawable;  
import android.media.AudioAttributes;  
import android.media.AudioManager;  
import android.media.SoundPool;  
import android.os.Build;  
import android.os.Handler;  
import android.os.Message;  
import android.support.annotation.RequiresApi;  
import android.util.Log;  
import android.view.MotionEvent;  
import android.view.View;  
import android.view.WindowManager;  
import android.widget.Button;  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.Comparator;  
import java.util.Random;  
import java.util.TimerTask;  
*/\*\*  
 \* Created by acer-pc on 2018/1/23.  
 \*/*public class GameView extends View {  
 private static final int *INIT\_HP*=5;//初始HP  
 private static final int *INIT\_SCORE*=0;//初始score  
 private static final int *catAuthorId*=3;//代表作者的喵的id  
 private static final int []*bonus*={1,3,5,10};//每打到一只喵喵增加的分数  
 private static final int *gridLength*=380;//一个格子的长宽  
 private static final int *catKindNum*=4;//喵喵种类  
 private static final int *putCatDelay* =10;//延迟多久开始放喵，延迟时长为putCatDelay\*putCatPeriod  
 private static final int *putCatPeriod* =300;//放喵的周期  
 private static final int *framePeriod* =100;//相邻两帧动画的时间间隔  
 private static final int *infiniteModeRowNum*=3;//无限模式行数  
 private static final int *infiniteModeColNum*=3;//无限模式列数  
 private static final int []*passScore*={  
 0,  
 100,  
 100,  
 100,  
 100,  
 100,  
 200,  
 200,  
 200,  
 200,  
 200,  
 300,  
 300,  
 300,  
 300,  
 300,  
 400,  
 400,  
 400,  
 400,  
 400,  
 500,  
 500,  
 600,  
 600,  
 700,  
 700,  
 800,  
 800,  
 900,  
 1000//level30  
 };  
// private static final int []passScore={  
// 0,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100,  
// 100//level30  
// };  
 private int rowNum=0;//多少行  
 private int colNum=0;//多少列  
 private StartActivity myContext;  
 private int HP=*INIT\_HP*;//生命值  
 private int score=*INIT\_SCORE*;//得分  
 private boolean isUpdateing=false;  
 private Random random= new Random();  
 private Hole [][]holes;  
 private boolean isOver=false;  
 public SoundPool soundPool=null;  
 private int catPokedSoundId =-1;  
 private int bombPokedSoundId =-1;  
 private int []catSoundId;  
 private int restPutCatDelay=0;//还剩多久时间放喵  
 private final Handler handler= new Handler(){  
 @Override  
 public void handleMessage(Message msg) {  
 if (msg.what==0x123&&!isUpdateing&&!isOver){//更新画布  
 invalidate();  
 }  
 super.handleMessage(msg);  
 }  
 };//处理事件  
 public class MyPoint{  
 public float x,y;  
 public MyPoint(float x, float y) {  
 this.x = x;  
 this.y = y;  
 }  
 }  
 public GameView(Context context) {  
 super(context);  
 myContext= (StartActivity)context;  
 }  
 @RequiresApi(api = Build.VERSION\_CODES.*LOLLIPOP*)  
 private void initSounds(){  
 catPokedSoundId =-1;  
 catSoundId=new int[*catKindNum*];  
 for (int i=0;i<*catKindNum*;i++)  
 catSoundId[i]=-1;  
 SharedPreferences prefs= getContext().getSharedPreferences(Macro.*PREFS\_FILE*,Context.*MODE\_PRIVATE*);  
 if (prefs.getString(Macro.*SOUNDS*,Macro.*CLOSE*).equals(Macro.*OPEN*)) {  
 SoundPool.Builder spb = new SoundPool.Builder();  
 spb.setMaxStreams(100);  
 AudioAttributes.Builder attrBuilder = new AudioAttributes.Builder();  
 //设置音频流的合适属性  
 attrBuilder.setLegacyStreamType(AudioManager.*STREAM\_MUSIC*);  
 spb.setAudioAttributes(attrBuilder.build()); //转换音频格式  
 soundPool = spb.build(); //创建SoundPool对象  
 catPokedSoundId = soundPool.load(getContext(), R.raw.*duang*, 1);  
 bombPokedSoundId= soundPool.load(getContext(),R.raw.*boom*,1);  
 catSoundId[0] = soundPool.load(getContext(), R.raw.*cat1*, 1);  
 catSoundId[1] = soundPool.load(getContext(), R.raw.*cat2*, 1);  
 catSoundId[2] = soundPool.load(getContext(), R.raw.*cat3*, 1);  
 catSoundId[3] = soundPool.load(getContext(), R.raw.*cat4*, 1);  
 }  
 }  
 private void initAnimation0(){  
 Hole.*animationOutAndIn*[0]= new Bitmap[20];  
 Hole.*animationOutAndIn*[0][0]=getBitmap(R.drawable.*cat1\_1*,1\**gridLength*/6);//1\_1  
 Hole.*animationOutAndIn*[0][1]=getBitmap(R.drawable.*cat1\_2*,2\**gridLength*/6);//2\_2  
 Hole.*animationOutAndIn*[0][2]=getBitmap(R.drawable.*cat1\_3*,3\**gridLength*/6);//3\_3  
 Hole.*animationOutAndIn*[0][3]=getBitmap(R.drawable.*cat1\_4*,4\**gridLength*/6);//4\_4  
 Hole.*animationOutAndIn*[0][4]=getBitmap(R.drawable.*cat1\_5*,5\**gridLength*/6);//5\_5  
 Hole.*animationOutAndIn*[0][5]=getBitmap(R.drawable.*cat1\_1*,6\**gridLength*/6);//1\_6  
 Hole.*animationOutAndIn*[0][6]=getBitmap(R.drawable.*cat1\_2*,6\**gridLength*/6);//2\_6  
 Hole.*animationOutAndIn*[0][7]=getBitmap(R.drawable.*cat1\_3*,6\**gridLength*/6);//3\_6  
 Hole.*animationOutAndIn*[0][8]=getBitmap(R.drawable.*cat1\_4*,6\**gridLength*/6);//4\_6  
 Hole.*animationOutAndIn*[0][9]=getBitmap(R.drawable.*cat1\_5*,6\**gridLength*/6);//5\_6  
 Hole.*animationOutAndIn*[0][10]=Hole.*animationOutAndIn*[0][5];//1\_6  
 Hole.*animationOutAndIn*[0][11]=Hole.*animationOutAndIn*[0][6];//2\_6  
 Hole.*animationOutAndIn*[0][12]=Hole.*animationOutAndIn*[0][7];//3\_6  
 Hole.*animationOutAndIn*[0][13]=Hole.*animationOutAndIn*[0][8];//4\_6  
 Hole.*animationOutAndIn*[0][14]=Hole.*animationOutAndIn*[0][9];//5\_6  
 Hole.*animationOutAndIn*[0][15]=getBitmap(R.drawable.*cat1\_1*,5\**gridLength*/6);//1\_5  
 Hole.*animationOutAndIn*[0][16]=getBitmap(R.drawable.*cat1\_2*,4\**gridLength*/6);//2\_4  
 Hole.*animationOutAndIn*[0][17]=Hole.*animationOutAndIn*[0][2];//3\_3  
 Hole.*animationOutAndIn*[0][18]=getBitmap(R.drawable.*cat1\_4*,2\**gridLength*/6);//4\_2  
 Hole.*animationOutAndIn*[0][19]=getBitmap(R.drawable.*cat1\_5*,1\**gridLength*/6);//5\_1  
 }  
 private void initAnimation1(){  
 Hole.*animationOutAndIn*[1]= new Bitmap[16];  
 Hole.*animationOutAndIn*[1][0]=getBitmap(R.drawable.*cat2\_1*,1\**gridLength*/6);//1\_1  
 Hole.*animationOutAndIn*[1][1]=getBitmap(R.drawable.*cat2\_2*,2\**gridLength*/6);//2\_2  
 Hole.*animationOutAndIn*[1][2]=getBitmap(R.drawable.*cat2\_3*,3\**gridLength*/6);//3\_3  
 Hole.*animationOutAndIn*[1][3]=getBitmap(R.drawable.*cat2\_4*,6\**gridLength*/6);//4\_6  
 Hole.*animationOutAndIn*[1][4]=getBitmap(R.drawable.*cat2\_1*,6\**gridLength*/6);//1\_6  
 Hole.*animationOutAndIn*[1][5]=getBitmap(R.drawable.*cat2\_2*,6\**gridLength*/6);//2\_6  
 Hole.*animationOutAndIn*[1][6]=getBitmap(R.drawable.*cat2\_3*,6\**gridLength*/6);//3\_6  
 Hole.*animationOutAndIn*[1][7]=Hole.*animationOutAndIn*[1][3];//4\_6  
 Hole.*animationOutAndIn*[1][8]=Hole.*animationOutAndIn*[1][4];//1\_6  
 Hole.*animationOutAndIn*[1][9]=Hole.*animationOutAndIn*[1][5];//2\_6  
 Hole.*animationOutAndIn*[1][10]=Hole.*animationOutAndIn*[1][6];//3\_6  
 Hole.*animationOutAndIn*[1][11]=Hole.*animationOutAndIn*[1][3];//4\_6  
 Hole.*animationOutAndIn*[1][12]=Hole.*animationOutAndIn*[1][4];//1\_6  
 Hole.*animationOutAndIn*[1][13]=getBitmap(R.drawable.*cat2\_2*,3\**gridLength*/6);//2\_3  
 Hole.*animationOutAndIn*[1][14]=getBitmap(R.drawable.*cat2\_3*,2\**gridLength*/6);//3\_2  
 Hole.*animationOutAndIn*[1][15]=getBitmap(R.drawable.*cat2\_4*,1\**gridLength*/6);//4\_1  
 }  
 private void initAnimation2(){  
 Hole.*animationOutAndIn*[2]= new Bitmap[10];  
 Hole.*animationOutAndIn*[2][0]=getBitmap(R.drawable.*cat3\_1*,1\**gridLength*/6);//1\_1  
 Hole.*animationOutAndIn*[2][1]=getBitmap(R.drawable.*cat3\_2*,2\**gridLength*/6);//2\_2  
 Hole.*animationOutAndIn*[2][2]=getBitmap(R.drawable.*cat3\_3*,3\**gridLength*/6);//3\_3  
 Hole.*animationOutAndIn*[2][3]=getBitmap(R.drawable.*cat3\_4*,6\**gridLength*/6);//4\_6  
 Hole.*animationOutAndIn*[2][4]=getBitmap(R.drawable.*cat3\_5*,6\**gridLength*/6);//5\_6  
 Hole.*animationOutAndIn*[2][5]=Hole.*animationOutAndIn*[2][3];//4\_6  
 Hole.*animationOutAndIn*[2][6]=Hole.*animationOutAndIn*[2][4];//5\_6  
 Hole.*animationOutAndIn*[2][7]=getBitmap(R.drawable.*cat3\_6*,3\**gridLength*/6);//6\_3  
 Hole.*animationOutAndIn*[2][8]=getBitmap(R.drawable.*cat3\_7*,2\**gridLength*/6);//7\_2  
 Hole.*animationOutAndIn*[2][9]=getBitmap(R.drawable.*cat3\_8*,1\**gridLength*/6);//8\_1  
 }  
 private void initAnimation3(){  
 Hole.*animationOutAndIn*[3]= new Bitmap[20];  
 Hole.*animationOutAndIn*[3][0]=getBitmap(R.drawable.*cat4\_1*,1\**gridLength*/6);//1\_1  
 Hole.*animationOutAndIn*[3][1]=getBitmap(R.drawable.*cat4\_1*,2\**gridLength*/6);//1\_2  
 Hole.*animationOutAndIn*[3][2]=getBitmap(R.drawable.*cat4\_1*,3\**gridLength*/6);//1\_3  
 Hole.*animationOutAndIn*[3][3]=getBitmap(R.drawable.*cat4\_1*,4\**gridLength*/6);//1\_4  
 Hole.*animationOutAndIn*[3][4]=getBitmap(R.drawable.*cat4\_1*,5\**gridLength*/6);//1\_5  
 Hole.*animationOutAndIn*[3][5]=getBitmap(R.drawable.*cat4\_1*,6\**gridLength*/6);//1\_6  
 Hole.*animationOutAndIn*[3][6]=getBitmap(R.drawable.*cat4\_2*,6\**gridLength*/6);//2\_6  
 Hole.*animationOutAndIn*[3][7]=Hole.*animationOutAndIn*[3][6];//2\_6  
 Hole.*animationOutAndIn*[3][8]=Hole.*animationOutAndIn*[3][5];//1\_6  
 Hole.*animationOutAndIn*[3][9]=Hole.*animationOutAndIn*[3][5];//1\_6  
 Hole.*animationOutAndIn*[3][10]=Hole.*animationOutAndIn*[3][6];//2\_6  
 Hole.*animationOutAndIn*[3][11]=Hole.*animationOutAndIn*[3][6];//2\_6  
 Hole.*animationOutAndIn*[3][12]=Hole.*animationOutAndIn*[3][5];//1\_6  
 Hole.*animationOutAndIn*[3][13]=Hole.*animationOutAndIn*[3][5];//1\_6  
 Hole.*animationOutAndIn*[3][14]=Hole.*animationOutAndIn*[3][5];//1\_6  
 Hole.*animationOutAndIn*[3][15]=Hole.*animationOutAndIn*[3][4];//1\_5  
 Hole.*animationOutAndIn*[3][16]=Hole.*animationOutAndIn*[3][3];//1\_4  
 Hole.*animationOutAndIn*[3][17]=Hole.*animationOutAndIn*[3][2];//1\_3  
 Hole.*animationOutAndIn*[3][18]=Hole.*animationOutAndIn*[3][1];//1\_2  
 Hole.*animationOutAndIn*[3][19]=Hole.*animationOutAndIn*[3][0];//1\_1  
 }  
 private void initAnimationBomb(){  
 Hole.*animationBomb*= new Bitmap[20];  
 Hole.*animationBomb*[0]=getBitmap(R.drawable.*bomb*,1\**gridLength*/6);//0\_1  
 Hole.*animationBomb*[1]=getBitmap(R.drawable.*bomb*,2\**gridLength*/6);//0\_2  
 Hole.*animationBomb*[2]=getBitmap(R.drawable.*bomb*,3\**gridLength*/6);//0\_3  
 Hole.*animationBomb*[3]=getBitmap(R.drawable.*bomb*,5\**gridLength*/6);//0\_5  
 Hole.*animationBomb*[4]=Hole.*animationBomb*[3];//0\_5  
 Hole.*animationBomb*[5]=Hole.*animationBomb*[3];//0\_5  
 Hole.*animationBomb*[6]=Hole.*animationBomb*[3];//0\_5  
 Hole.*animationBomb*[7]=Hole.*animationBomb*[3];//0\_5  
 Hole.*animationBomb*[8]=getBitmap(R.drawable.*bomb*,4\**gridLength*/6);//0\_4  
 Hole.*animationBomb*[9]=Hole.*animationBomb*[2];//0\_3  
 Hole.*animationBomb*[10]=Hole.*animationBomb*[2];//0\_3  
 Hole.*animationBomb*[11]=Hole.*animationBomb*[8];//0\_4  
 Hole.*animationBomb*[12]=Hole.*animationBomb*[3];//0\_5  
 Hole.*animationBomb*[13]=Hole.*animationBomb*[3];//0\_5  
 Hole.*animationBomb*[14]=Hole.*animationBomb*[3];//0\_5  
 Hole.*animationBomb*[15]=Hole.*animationBomb*[3];//0\_5  
 Hole.*animationBomb*[16]=Hole.*animationBomb*[3];//0\_5  
 Hole.*animationBomb*[17]=Hole.*animationBomb*[2];//0\_3  
 Hole.*animationBomb*[18]=Hole.*animationBomb*[1];//0\_2  
 Hole.*animationBomb*[19]=Hole.*animationBomb*[0];//0\_1  
 }  
 private void initAnimation(){  
 Hole.*animationOutAndIn*= new Bitmap[*catKindNum*][];  
 initAnimation0();  
 initAnimation1();  
 initAnimation2();  
 initAnimation3();  
 initAnimationBomb();  
 }  
 private void initMatrixLayout(){//矩阵型的布局  
 holes= new Hole[rowNum][colNum];  
 WindowManager wm= (WindowManager) getContext().getSystemService(Context.*WINDOW\_SERVICE*);  
 int gridSpace= 0;//格子间的间隔  
 int firstX=(wm.getDefaultDisplay().getWidth()-gridSpace\*(colNum-1)-colNum\**gridLength*)/2;  
 int firstY= 50;//第一个格子的Y  
 firstY= (wm.getDefaultDisplay().getHeight()-rowNum\**gridLength*)/2+firstY;  
 for (int i=0;i<rowNum;i++)  
 for (int j=0;j<colNum;j++)  
 holes[i][j]=new Hole(firstX+j\*(*gridLength*+gridSpace),firstY+i\*(*gridLength*-50));  
 }  
 private void initRandomLayout(){//随机布局  
 holes= new Hole[rowNum][colNum];  
 WindowManager wm= (WindowManager) getContext().getSystemService(Context.*WINDOW\_SERVICE*);  
 int width= wm.getDefaultDisplay().getWidth();//屏幕宽  
 int height= wm.getDefaultDisplay().getHeight();//屏幕高  
 int gridSpace= 10;//格子间的间隔  
 int firstX= gridSpace;//与左边沿的间距  
 int firstY= 200;//与上边沿的间距  
 ArrayList<MyPoint> pos= new ArrayList<MyPoint>();  
 for (int i=0;i<rowNum\*colNum;i++){  
 float x,y;  
 boolean flag=true;  
 do {  
 flag=false;  
 x = random.nextInt(width - 2 \* firstX - *gridLength*) + firstX;//随机一个坐标  
 y = random.nextInt(height - firstY - 100-*gridLength*) + firstY;  
 for (int j = 0; j <i&&!flag; j++) {  
 float xx=pos.get(j).x;  
 float yy=pos.get(j).y;  
 if (Math.*abs*(xx-x)<gridSpace||  
 Math.*abs*(yy-y)<gridSpace){  
 flag= true;  
 break;  
 }  
 }  
 }while (flag);  
 pos.add(new MyPoint(x,y));  
 }  
 Collections.*sort*(pos, new Comparator<MyPoint>() {  
 @Override  
 public int compare(MyPoint a, MyPoint b) {  
 return ((Float)a.y).compareTo((Float)b.y);  
 }  
 });  
 for (int i=0;i<rowNum;i++)  
 for (int j=0;j<colNum;j++) {  
 //Log.i("yaoling1997","y:"+pos.get(i \* rowNum + j).y);  
 holes[i][j] = new Hole(pos.get(i \* colNum + j).x, pos.get(i \* colNum + j).y);  
 }  
 }  
 private void initLayout(){//初始布局  
 switch (myContext.getLevel()){  
 case 0:  
 rowNum=*infiniteModeRowNum*;  
 colNum=*infiniteModeColNum*;  
 holes= new Hole[rowNum][colNum];  
 initRandomLayout();  
 break;  
 case 1://矩阵布局  
 rowNum=1;  
 colNum=1;  
 initMatrixLayout();  
 break;  
 case 2://矩阵布局  
 rowNum=1;  
 colNum=2;  
 initMatrixLayout();  
 break;  
 case 3://矩阵布局  
 rowNum=3;  
 colNum=1;  
 initMatrixLayout();  
 break;  
 case 4://矩阵布局  
 rowNum=2;  
 colNum=2;  
 initMatrixLayout();  
 break;  
 case 5://矩阵布局  
 rowNum=2;  
 colNum=2;  
 initMatrixLayout();  
 break;  
 case 6://矩阵布局  
 rowNum=2;  
 colNum=2;  
 initMatrixLayout();  
 break;  
 case 7://矩阵布局  
 rowNum=2;  
 colNum=2;  
 initMatrixLayout();  
 break;  
 case 8://矩阵布局  
 rowNum=2;  
 colNum=2;  
 initMatrixLayout();  
 break;  
 case 9://矩阵布局  
 rowNum=4;  
 colNum=1;  
 initMatrixLayout();  
 break;  
 case 10://随机布局  
 rowNum=5;  
 colNum=1;  
 initRandomLayout();  
 break;  
 case 11://矩阵布局  
 rowNum=1;  
 colNum=3;  
 initMatrixLayout();  
 break;  
 case 12://随机布局  
 rowNum=2;  
 colNum=2;  
 initMatrixLayout();  
 break;  
 case 13://矩阵布局  
 rowNum=2;  
 colNum=3;  
 initMatrixLayout();  
 break;  
 case 14://随机布局  
 rowNum=3;  
 colNum=2;  
 initRandomLayout();  
 break;  
 case 15://矩阵布局  
 rowNum=2;  
 colNum=3;  
 initMatrixLayout();  
 break;  
 case 16://矩阵布局  
 rowNum=4;  
 colNum=2;  
 initMatrixLayout();  
 break;  
 case 17://随机布局  
 rowNum=1;  
 colNum=8;  
 initRandomLayout();  
 break;  
 case 18://随机布局  
 rowNum=1;  
 colNum=8;  
 initRandomLayout();  
 break;  
 case 19://随机布局  
 rowNum=1;  
 colNum=8;  
 initRandomLayout();  
 break;  
 case 20://矩阵布局  
 rowNum=3;  
 colNum=3;  
 initMatrixLayout();  
 break;  
 case 21://矩阵布局  
 rowNum=3;  
 colNum=3;  
 initMatrixLayout();  
 break;  
 case 22://随机布局  
 rowNum=1;  
 colNum=9;  
 initRandomLayout();  
 break;  
 case 23://随机布局  
 rowNum=1;  
 colNum=9;  
 initRandomLayout();  
 break;  
 case 24://随机布局  
 rowNum=1;  
 colNum=10;  
 initRandomLayout();  
 break;  
 case 25://随机布局  
 rowNum=1;  
 colNum=10;  
 initRandomLayout();  
 break;  
 case 26://随机布局  
 rowNum=1;  
 colNum=11;  
 initRandomLayout();  
 break;  
 case 27://矩阵布局  
 rowNum=4;  
 colNum=3;  
 initMatrixLayout();  
 break;  
 case 28://矩阵布局  
 rowNum=4;  
 colNum=3;  
 initMatrixLayout();  
 break;  
 case 29://随机布局  
 rowNum=1;  
 colNum=12;  
 initRandomLayout();  
 break;  
 case 30://随机布局  
 rowNum=1;  
 colNum=12;  
 initRandomLayout();  
 break;  
 }  
 }  
 private void playSound(int id){  
 if (id<0)  
 return;  
 soundPool.play(id, 1, 1, 0, 0, 1);  
 Log.*i*("yaoling1997","play Sound Successfully id:"+id);  
 }  
 private void restart(){  
 HP= *INIT\_HP*;  
 score=*INIT\_SCORE*;  
 isOver=false;  
 initLayout();  
 restPutCatDelay=*putCatDelay*;  
 }  
 private void loseHP(){  
 HP = Math.*max*(0, HP - 1);  
 }  
 private int getNextId(){  
 int cat4=100,cat3=90,cat2=60,cat1=10;  
 int tmp = random.nextInt(100);  
 switch (myContext.getLevel()){  
 case 0://只有喵1、喵2、喵3、喵4和炸弹，无限模式  
 cat4=99;  
 cat3=90;  
 cat2=60;  
 cat1=10;  
 break;  
 case 1://只有喵1  
 cat3=100;  
 cat2=100;  
 cat1=0;  
 break;  
 case 2://只有喵1  
 cat3=100;  
 cat2=100;  
 cat1=0;  
 break;  
 case 3://只有喵1  
 cat3=100;  
 cat2=100;  
 cat1=0;  
 break;  
 case 4://只有喵1，矩阵布局  
 cat3=100;  
 cat2=100;  
 cat1=0;  
 break;  
 case 5://只有喵1和炸弹，矩阵布局  
 cat3=100;  
 cat2=100;  
 cat1=20;  
 break;  
 case 6://只有喵2，矩阵布局  
 cat3=100;  
 cat2=0;  
 cat1=0;  
 break;  
 case 7://只有喵1和喵2，矩阵布局  
 cat3=100;  
 cat2=50;  
 cat1=0;  
 break;  
 case 8://只有喵1、喵2和炸弹，矩阵布局  
 cat3=100;  
 cat2=70;  
 cat1=20;  
 break;  
 case 9://只有喵1、喵2和炸弹，随机布局  
 cat3=100;  
 cat2=70;  
 cat1=10;  
 break;  
 case 10://只有喵1、喵2和炸弹，随机布局  
 cat3=100;  
 cat2=80;  
 cat1=40;  
 break;  
 case 11://只有喵3，随机布局  
 cat3=0;  
 cat2=0;  
 cat1=0;  
 break;  
 case 12://只有喵1、喵2、喵3，随机布局  
 cat3=90;  
 cat2=60;  
 cat1=10;  
 break;  
 case 13://只有喵1、喵2、喵3，矩阵布局  
 cat3=90;  
 cat2=60;  
 cat1=10;  
 break;  
 case 14://只有喵1、喵2、喵3，随机布局  
 cat3=90;  
 cat2=60;  
 cat1=20;  
 break;  
 case 15://只有喵1、喵2、喵3和炸弹，矩阵布局  
 cat4=99;  
 cat3=90;  
 cat2=60;  
 cat1=20;  
 break;  
 case 16://只有喵1、喵2、喵3和炸弹，矩阵布局  
 cat4=99;  
 cat3=80;  
 cat2=60;  
 cat1=40;  
 break;  
 case 17://只有喵1、喵2、喵3和炸弹，随机布局  
 cat4=99;  
 cat3=90;  
 cat2=80;  
 cat1=20;  
 break;  
 case 18://只有喵1、喵2、喵3和炸弹，随机布局  
 cat4=99;  
 cat3=60;  
 cat2=30;  
 cat1=10;  
 break;  
 case 19://只有喵1、喵2、喵3和炸弹，随机布局  
 cat4=99;  
 cat3=90;  
 cat2=60;  
 cat1=20;  
 break;  
 case 20://只有喵1、喵2、喵3和炸弹，矩阵布局  
 cat4=99;  
 cat3=80;  
 cat2=50;  
 cat1=30;  
 break;  
 case 21://只有喵1、喵2、喵3和炸弹，随机布局  
 cat4=99;  
 cat3=90;  
 cat2=40;  
 cat1=20;  
 break;  
 case 22://只有喵1、喵2、喵3和炸弹，随机布局  
 cat4=99;  
 cat3=90;  
 cat2=40;  
 cat1=10;  
 break;  
 case 23://只有喵1、喵2、喵3和炸弹，随机布局  
 cat4=99;  
 cat3=90;  
 cat2=50;  
 cat1=20;  
 break;  
 case 24://只有喵1、喵2、喵3和炸弹，随机布局  
 cat4=99;  
 cat3=60;  
 cat2=40;  
 cat1=20;  
 break;  
 case 25://只有喵1、喵2、喵3和炸弹，随机布局  
 cat4=99;  
 cat3=80;  
 cat2=50;  
 cat1=20;  
 break;  
 case 26://只有喵1、喵2、喵3和炸弹，随机布局  
 cat4=99;  
 cat3=90;  
 cat2=50;  
 cat1=20;  
 break;  
 case 27://只有喵1、喵2、喵3和炸弹，随机布局  
 cat4=99;  
 cat3=80;  
 cat2=60;  
 cat1=20;  
 break;  
 case 28://只有喵1、喵2、喵3和炸弹，随机布局  
 cat4=99;  
 cat3=90;  
 cat2=70;  
 cat1=20;  
 break;  
 case 29://只有喵1、喵2、喵3和炸弹，随机布局  
 cat4=99;  
 cat3=70;  
 cat2=50;  
 cat1=20;  
 break;  
 case 30://只有喵1、喵2、喵3和炸弹，随机布局  
 cat4=99;  
 cat3=70;  
 cat2=50;  
 cat1=20;  
 break;  
 }  
 if (tmp>=cat4)  
 tmp = 3;  
 else if (tmp >= cat3)  
 tmp = 2;  
 else if (tmp >= cat2)  
 tmp = 1;  
 else if (tmp>=cat1)  
 tmp = 0;  
 else  
 tmp = -1;  
 return tmp;  
 }  
 @RequiresApi(api = Build.VERSION\_CODES.*LOLLIPOP*)  
 public void initGame(){  
 initSounds();  
 initAnimation();  
 Hole.*catPoked* =getBitmap(R.drawable.*cat\_poked*,*gridLength*);  
 Hole.*bombPoked* =getBitmap(R.drawable.*bomb\_poked*,*gridLength*);  
 Hole.*stubFront*=getBitmap(R.drawable.*stub\_front*,*gridLength*);  
 Hole.*stubBack*=getBitmap(R.drawable.*stub\_back*,*gridLength*);  
 Hole.*giftBoxClose*=getBitmap(R.drawable.*giftbox\_close*,*gridLength*);  
 Hole.*giftBoxOpen*=getBitmap(R.drawable.*giftbox\_open*,*gridLength*);  
 Log.*i*("yaoling1997","Hole ok");  
 //initMatrixLayout();  
 restart();  
 StartActivity.*timer*.schedule(new TimerTask() {//随机放喵喵  
 @Override  
 public void run() {  
 if (restPutCatDelay>0)  
 restPutCatDelay--;  
 if (!isOver&&restPutCatDelay==0) {  
 //Log.i("yaoling1997","put");  
 int row = random.nextInt(rowNum);  
 int col = random.nextInt(colNum);  
 if (holes[row][col].getStatus() == Hole.*EMPTY*) {  
 int tmp= getNextId();  
 holes[row][col].begin(tmp);  
 handler.sendEmptyMessage(0x123);  
 if (tmp>=0)//出来的是喵  
 playSound(catSoundId[tmp]);  
 }  
 }  
 }  
 },0, *putCatPeriod*);  
 StartActivity.*timer*.schedule(new TimerTask() {  
 @Override  
 public void run() {  
 if (!isOver) {  
 //Log.i("yaoling1997","update");  
 for (int i = 0; i < rowNum; i++)  
 for (int j = 0; j < colNum; j++) {  
 if (!holes[i][j].next())  
 loseHP();  
 }  
 handler.sendEmptyMessage(0x123);  
 }  
 }  
 },0,*framePeriod*);  
 }  
 private Bitmap getBitmap(int id,int height){  
 //height:从上往下截多长  
 Drawable image= getResources().getDrawable(id);  
 Bitmap bitmap=Bitmap.*createBitmap*(*gridLength*,*gridLength*, Bitmap.Config.*ARGB\_8888*);  
 Rect rect= new Rect(0,0,*gridLength*,*gridLength*);  
 image.setBounds(rect);//以设定的比例画到bitmap的画布上  
 Canvas canvas= new Canvas(bitmap);  
 image.draw(canvas);  
 //bitmap=Bitmap.createBitmap(bitmap,0,0,gridLength,height);//截取特定部分  
 bitmap=Bitmap.*createBitmap*(bitmap,0,0,*gridLength*,Math.*max*(height,1));//截取特定部分  
 return bitmap;  
 }  
 @Override  
 protected void onDraw(Canvas canvas) {  
 super.onDraw(canvas);  
 drawInfo(canvas);  
 for (int i= 0;i<rowNum;i++)  
 for (int j=0;j<colNum;j++) {  
 int status=holes[i][j].getStatus();  
 int id=holes[i][j].getId();  
 Bitmap bitmapEvent;  
 Paint paintEvent= new Paint();  
 if (status==Hole.*EMPTY*){  
 bitmapEvent=Hole.*stubBack*;  
 }else if (status<Hole.*EMPTY*){  
 if (id>=0)//是喵被打了  
 bitmapEvent=Hole.*catPoked*;  
 else//是炸弹被打了  
 bitmapEvent=Hole.*bombPoked*;  
 paintEvent.setAlpha((Hole.*EMPTY*-status)\*255/(Hole.*EMPTY*- Hole.*POKED*));  
 }else if (id>=0){//是喵  
 bitmapEvent= Hole.*animationOutAndIn*[id][status];  
 }else {//是炸弹  
 bitmapEvent= Hole.*animationBomb*[status];  
 }  
 float x= holes[i][j].getX();  
 float y= holes[i][j].getY();  
 float catY= y+*gridLength*-bitmapEvent.getHeight();  
 if (myContext.getLevel()>=30){  
 if (status==Hole.*EMPTY*)  
 canvas.drawBitmap(Hole.*giftBoxClose*, x, y, null);  
 else {  
 canvas.drawBitmap(bitmapEvent, x, catY, paintEvent);  
 canvas.drawBitmap(Hole.*giftBoxOpen*, x, y, null);  
 }  
 }else {  
 canvas.drawBitmap(Hole.*stubBack*, x, y, null);  
 canvas.drawBitmap(bitmapEvent, x, catY, paintEvent);  
 canvas.drawBitmap(Hole.*stubFront*, x, y, null);  
 }  
 }  
 if (HP<=0||  
 (myContext.getLevel()>0&&score>=*passScore*[myContext.getLevel()])){  
 gameOver();  
 return;  
 }  
 }  
 private void drawInfo(Canvas canvas){  
 myContext.tvHP.mySetText(""+HP);  
 if (myContext.getLevel()==0)  
 myContext.tvScore.mySetText(""+score);  
 else//闯关模式要显示目标得分  
 myContext.tvScore.mySetText(""+score+"/"+*passScore*[myContext.getLevel()]);  
 }  
 private void updateScoreboard(){//更新积分榜  
 SharedPreferences prefs= getContext().getSharedPreferences(Macro.*PREFS\_FILE*,Context.*MODE\_PRIVATE*);  
 int tmpScore= score;  
 for (int i=0;i<ScoreboardActivity.*NoNum*;i++){  
 int oldScore=prefs.getInt(Macro.*NO*[i],0);  
 if (oldScore<tmpScore){  
 SharedPreferences.Editor editor = prefs.edit();  
 editor.putInt(Macro.*NO*[i],tmpScore);  
 editor.commit();  
 tmpScore=oldScore;  
 }  
 }  
 }  
 @RequiresApi(api = Build.VERSION\_CODES.*LOLLIPOP*)  
 private void gameOver(){  
 if (isOver)  
 return;  
 isOver=true;  
 boolean isPass= myContext.getLevel()>0&&score>=*passScore*[myContext.getLevel()];//是否过关，决定用哪种对话框，无限模式不存在过关  
 if (myContext.getLevel()==0)//是无限模式，需要更新积分榜  
 updateScoreboard();  
 AlertDialog.Builder builder= new AlertDialog.Builder(getContext());  
 View alertDialogView;  
 if (isPass) {  
 alertDialogView = View.*inflate*(getContext(), R.layout.*alertdialog\_gamepass*, null);  
 }else {  
 alertDialogView = View.*inflate*(getContext(), R.layout.*alertdialog\_gameover*, null);  
 }  
 StrokeTextView tvContent= (StrokeTextView)alertDialogView.findViewById(R.id.*tvContent*);  
 Button btnStart= (Button)alertDialogView.findViewById(R.id.*btnStart*);  
 Button btnEnd= (Button)alertDialogView.findViewById(R.id.*btnEnd*);  
 builder.setView(alertDialogView);  
 builder.setCancelable(false);  
 final AlertDialog dialog= builder.create();  
 if (!isPass) {//没过关对话框  
 tvContent.setText("得分：" + score + " ");  
 btnStart.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 restart();  
 dialog.dismiss();  
 }  
 });  
 }else {//过关对话框  
 tvContent.setText("恭喜过关！");  
 if (myContext.getLevel()<Macro.*LEVEL\_NUM*) {  
 SharedPreferences prefs= myContext.getSharedPreferences(Macro.*PREFS\_FILE*,Context.*MODE\_PRIVATE*);//解锁下一关  
 SharedPreferences.Editor editor= prefs.edit();  
 editor.putString(Macro.*LEVEL*+(myContext.getLevel()+1),Macro.*OPEN*);  
 editor.commit();  
 btnStart.setText("下一关");  
 btnStart.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 myContext.updateLevel(myContext.getLevel()+1);;//跳至下一关  
 restart();  
 dialog.dismiss();  
 }  
 });  
 }else {  
 btnStart.setText("恭喜通关");  
 btnStart.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 Intent intent= new Intent();  
 intent.setAction(Macro.*ACTION\_PASS\_ALL\_LEVELS\_ACTIVITY*);  
 getContext().startActivity(intent);  
 ((StartActivity)getContext()).finish();  
 dialog.dismiss();  
 }  
 });  
 }  
 }  
 btnEnd.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 ((StartActivity)getContext()).finish();  
 dialog.dismiss();  
 }  
 });  
 MainActivity.*addBtnAnimation*(btnStart);  
 MainActivity.*addBtnAnimation*(btnEnd);  
 dialog.getWindow().setBackgroundDrawable(new ColorDrawable());  
 dialog.show();//show必须放前面，不知道为啥  
 WindowManager.LayoutParams lp= dialog.getWindow().getAttributes();  
 lp.width=Macro.*ALERTDIALOG\_WIDTH*;  
 lp.height=Macro.*ALERTDIALOG\_HEIGHT*;  
 dialog.getWindow().setAttributes(lp);  
 }  
 @Override  
 public boolean onTouchEvent(MotionEvent event) {  
 if (event.getAction()!=MotionEvent.*ACTION\_DOWN*)//只有手指点击才算poke成功  
 return true;  
 float x= event.getX();  
 float y= event.getY();  
 for (int i=rowNum-1;i>=0;i--)  
 for (int j=colNum-1;j>=0;j--){//倒序判断，确保在最外面的喵喵先被打到  
 float holeX=holes[i][j].getX();  
 float holeY=holes[i][j].getY();  
 if (holeX<=x&&x<=holeX+*gridLength*&&  
 holeY<=y&&y<=holeY+*gridLength*){  
 if (holes[i][j].poke()) {  
 if (holes[i][j].getId()>=0){//敲到的是喵，加分  
 score+=*bonus*[holes[i][j].getId()]\*Hole.*sameNum*;//连续敲到同一种喵，奖励更多分  
 if (holes[i][j].getId()==*catAuthorId*)//敲到作者加血  
 HP++;  
 Log.*i*("yaoling1997","catPokedSoundId:"+ catPokedSoundId);  
 if (catPokedSoundId >=0)  
 playSound(catPokedSoundId);  
 }else {  
 loseHP();  
 if (bombPokedSoundId >=0)  
 playSound(bombPokedSoundId);  
 }  
 invalidate();  
 return true;  
 }  
 }  
 }  
 return true;  
 }  
}

package com.example.acer\_pc.pokecats.gamedemo;  
import android.graphics.Bitmap;  
*/\*\*  
 \* Created by acer-pc on 2018/1/23.  
 \*/*public class Hole {  
 public static final int *BEGIN*=0;//喵喵开始钻出  
 public static final int *EMPTY*=-1;//洞洞为空  
 public static final int *POKED*=-4;//喵喵被打了，(EMPTY-POKED)\*period为敲击画面持续的时间  
 public static final int *INIT\_ID*=-2;//初始id  
 public static Bitmap [][]*animationOutAndIn*;//喵喵的进出动画  
 public static Bitmap []*animationBomb*;//炸弹的动画（）  
 public static Bitmap *catPoked*;//喵喵被打到的图片  
 public static Bitmap *bombPoked*;//炸弹被打到的图片  
 public static Bitmap *stubBack*;//整个树桩  
 public static Bitmap stubFront;//只保留树桩的前部分  
 public static Bitmap *giftBoxClose*;//礼物盒关闭状态  
 public static Bitmap *giftBoxOpen*;//礼物盒打开状态  
 public static int *preId*=*INIT\_ID*;//前一个被敲的东东的id  
 public static int *sameNum*=0;//连续敲击同一种东东多少次  
 private float x,y;//洞洞左上角坐标  
 private int status=*EMPTY*;//洞洞的状态  
 private int id=0;//喵喵或炸弹的id  
 public Hole(float x, float y) {  
 this.x = x;  
 this.y = y;  
 status=*EMPTY*;  
 id=0;//0,1,2代表喵，-1代表炸弹  
 *preId*=*INIT\_ID*;  
 *sameNum*=0;  
 }  
 public float getX() {  
 return x;  
 }  
 public float getY() {  
 return y;  
 }  
 public int getStatus() {  
 return status;  
 }  
 public void begin(int id){//开始钻出  
 status= *BEGIN*;  
 this.id=id;  
 }  
 public boolean next(){//下一个状态，返回值表示为false表示掉血  
 if (status>=*BEGIN*){  
 status++;  
 if (id>=0) {//是喵喵  
 if (status >= *animationOutAndIn*[id].length) {  
 //回到洞洞里，掉血  
 status = *EMPTY*;  
 return false;  
 }  
 }else {//是炸弹  
 if (status>=*animationBomb*.length){  
 //回到洞洞里，不掉血  
 status = *EMPTY*;  
 return true;  
 }  
 }  
 }  
 if (status<*EMPTY*){  
 status++;  
 }  
 return true;  
 }  
 public boolean poke(){//敲击事件，true表示敲击到东西了  
 if (status>=*BEGIN*){  
 status=*POKED*;  
 if (*preId*==id){  
 *sameNum*++;  
 }else {  
 *sameNum*=1;  
 *preId*=id;  
 }  
 return true;  
 }  
 return false;  
 }  
 public int getId() {  
 return id;  
 }  
}

package com.example.acer\_pc.pokecats.gamedemo;  
import android.app.Activity;  
import android.content.Intent;  
import android.os.Bundle;  
import android.support.annotation.Nullable;  
import android.view.View;  
import android.view.Window;  
import android.widget.Button;  
*/\*\*  
 \* Created by acer-pc on 2018/1/28.  
 \*/*public class InfiniteModeActivity extends Activity {  
 Button btnStart,btnScoreboard;  
 @Override  
 protected void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 requestWindowFeature(Window.*FEATURE\_NO\_TITLE*);  
 setContentView(R.layout.*activity\_infinite\_mode*);  
 bindViews();  
 }  
 private void bindViews(){  
 btnStart = (Button)findViewById(R.id.*btnStart*);  
 btnScoreboard= (Button)findViewById(R.id.*btnScoreboard*);  
 MainActivity.*addBtnAnimation*(btnStart);  
 MainActivity.*addBtnAnimation*(btnScoreboard);  
 }  
 public void onClick(View view){  
 if (view.getId()==R.id.*btnStart*){  
 Intent intent= new Intent();  
 intent.putExtra(Macro.*GAME\_MODE*,0);//0代表无限模式  
 intent.setAction(Macro.*ACTION\_START\_ACTIVITY*);  
 startActivity(intent);  
 }else if (view.getId()==R.id.*btnScoreboard*){  
 Intent intent= new Intent();  
 intent.setAction(Macro.*ACTION\_SCOREBOARD\_ACTIVITY*);  
 startActivity(intent);  
 }  
 }  
  
}

package com.example.acer\_pc.pokecats.gamedemo;  
import android.app.Activity;  
import android.os.Bundle;  
import android.support.annotation.Nullable;  
import android.view.Window;  
*/\*\*  
 \* Created by acer-pc on 2018/1/27.  
 \*/*public class IntroductionActivity extends Activity {  
 @Override  
 protected void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 requestWindowFeature(Window.FEATURE\_NO\_TITLE);  
 setContentView(R.layout.*activity\_introduction*);  
 }  
}

package com.example.acer\_pc.pokecats.gamedemo;  
*/\*\*  
 \* Created by acer-pc on 2018/1/22.  
 \*/*public class Macro {  
 public static final String *ACTION\_SETTING\_ACTIVITY*="lyj.action.setting";  
 public static final String *ACTION\_START\_ACTIVITY*="lyj.action.start";  
 public static final String *ACTION\_SCOREBOARD\_ACTIVITY*="lyj.action.scoreboard";  
 public static final String *ACTION\_INTRODUCTION\_ACTIVITY*="lyj.action.introduction";  
 public static final String *ACTION\_INFINITE\_MODE\_ACTIVITY*="lyj.action.infiniteMode";  
 public static final String *ACTION\_PASS\_MODE\_ACTIVITY*="lyj.action.passMode";  
 public static final String ACTION\_PASS\_ALL\_LEVELS\_ACTIVITY="lyj.action.passAllLevels";  
 public static final String *PREFS\_FILE*="PokeCatsPrefs";  
 public static final String *OPEN*="open";  
 public static final String *CLOSE*="close";  
 public static final String *BG\_MUSIC*="bgMusic";  
 public static final String *SOUNDS*="sounds";  
 public static final String []*NO*={"No.1","No.2","No.3"};  
 public static final String *GAME\_MODE*="gameMode";//游戏模式  
 public static final String *LEVEL*="level";//关卡  
 public static final int *LEVEL\_NUM*=30;//总关卡数，是PassModeAdapter.colNum的倍数  
 public static final int *ALERTDIALOG\_WIDTH*=440\*2;//对话框宽度  
 public static final int *ALERTDIALOG\_HEIGHT*=210\*2;//对话框高度  
}

package com.example.acer\_pc.pokecats.gamedemo;  
import android.app.Activity;  
import android.app.ActivityManager;  
import android.content.Context;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.os.Build;  
import android.os.Bundle;  
import android.support.annotation.RequiresApi;  
import android.util.Log;  
import android.view.MotionEvent;  
import android.view.View;  
import android.view.Window;  
import android.widget.Button;  
import android.widget.RelativeLayout;  
*/\*\*  
 \* Created by acer-pc on 2018/1/22.  
 \*/*public class MainActivity extends Activity {  
 SharedPreferences prefs;  
 RelativeLayout mainLayout;  
 Button btnPassMode,btnInfiniteMode,btnSetting,btnIntroduction,btnExit;  
 @RequiresApi(api = Build.VERSION\_CODES.*LOLLIPOP*)  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 requestWindowFeature(Window.*FEATURE\_NO\_TITLE*);  
 setContentView(R.layout.*activity\_main*);  
 bindViews();  
 ActivityManager activityManager = (ActivityManager)this.getSystemService(Context.*ACTIVITY\_SERVICE*);  
 Log.*i*("yaoling1997","最大内存：" + activityManager.getMemoryClass());  
 prefs=getSharedPreferences(Macro.*PREFS\_FILE*,*MODE\_PRIVATE*);  
// SharedPreferences.Editor editor = prefs.edit();//清空用户保存的数据  
// editor.clear();  
// editor.commit();  
 unlockLevel();//解锁关卡  
 initBackgroundMusicAndSoundsPrefs();//第一次玩，默认音效和背景音乐打开  
 Intent intent= new Intent(this,MusicService.class);  
 intent.putExtra(Macro.*BG\_MUSIC*,prefs.getString(Macro.*BG\_MUSIC*,Macro.*CLOSE*));//open or close  
 startService(intent);  
 }  
 void initBackgroundMusicAndSoundsPrefs(){  
 if (prefs.getString(Macro.*BG\_MUSIC*,"").equals("")&&prefs.getString(Macro.*SOUNDS*,"").equals("")) {  
 SharedPreferences.Editor editor = prefs.edit();  
 editor.putString(Macro.*BG\_MUSIC*, Macro.*OPEN*);  
 editor.putString(Macro.*SOUNDS*, Macro.*OPEN*);  
 editor.commit();  
 }  
 }  
 private void unlockLevel(){  
 SharedPreferences.Editor editor = prefs.edit();  
 for (int i=1;i<=1;i++){  
 editor.putString(Macro.*LEVEL*+i,Macro.*OPEN*);  
 }  
 editor.commit();  
 }  
 public static void addBtnAnimation(final Button btn){  
 btn.setOnTouchListener(new View.OnTouchListener() {  
 @Override  
 public boolean onTouch(View view, MotionEvent motionEvent) {//返回false是为了能够调用onclick事件  
 //Log.i("yaoling1997","This is onTouch: "+motionEvent.getAction());  
 if (motionEvent.getAction()==MotionEvent.*ACTION\_DOWN*){  
 btn.setScaleX((float)0.8);  
 btn.setScaleY((float)0.8);  
 btn.invalidate();  
 return false;  
 }  
 if (motionEvent.getAction()==MotionEvent.*ACTION\_UP*){  
 btn.setScaleX(1);  
 btn.setScaleY(1);  
 btn.invalidate();  
 return false;  
 }  
 return false;  
 }  
 });  
 }  
 private void bindViews(){  
 mainLayout= (RelativeLayout)findViewById(R.id.*mainLayout*);  
 btnPassMode = (Button)findViewById(R.id.*btnPassMode*);  
 btnInfiniteMode = (Button)findViewById(R.id.*btnInfiniteMode*);  
 btnSetting= (Button)findViewById(R.id.*btnSetting*);  
 btnIntroduction= (Button)findViewById(R.id.*btnIntroduction*);  
 btnExit= (Button)findViewById(R.id.*btnExit*);  
 *addBtnAnimation*(btnPassMode);  
 *addBtnAnimation*(btnInfiniteMode);  
 *addBtnAnimation*(btnSetting);  
 *addBtnAnimation*(btnIntroduction);  
 *addBtnAnimation*(btnExit);  
 //mainLayout.setBackground(getDrawable(R.drawable.background2));  
 }  
 @RequiresApi(api = Build.VERSION\_CODES.*LOLLIPOP*)  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 Intent intent= new Intent(this,MusicService.class);  
 stopService(intent);  
 }  
 public void onClick(View view){  
 if (view.getId()==R.id.*btnPassMode*){  
 Intent intent= new Intent();  
 intent.setAction(Macro.*ACTION\_PASS\_MODE\_ACTIVITY*);  
 startActivity(intent);  
 }else if (view.getId()==R.id.*btnInfiniteMode*){  
 Intent intent= new Intent();  
 intent.setAction(Macro.*ACTION\_INFINITE\_MODE\_ACTIVITY*);  
 startActivity(intent);  
 }else if (view.getId()==R.id.*btnSetting*){  
 Intent intent= new Intent();  
 intent.setAction(Macro.*ACTION\_SETTING\_ACTIVITY*);  
 startActivity(intent);  
 }else if (view.getId()==R.id.*btnIntroduction*){  
 Intent intent= new Intent();  
 intent.setAction(Macro.*ACTION\_INTRODUCTION\_ACTIVITY*);  
 startActivity(intent);  
 }else if (view.getId()==R.id.*btnExit*){  
 this.finish();  
 }  
 }  
}

package com.example.acer\_pc.pokecats.gamedemo;  
import android.app.Service;  
import android.content.Intent;  
import android.content.res.AssetFileDescriptor;  
import android.media.MediaPlayer;  
import android.os.IBinder;  
import android.support.annotation.Nullable;  
import android.util.Log;  
*/\*\*  
 \* Created by acer-pc on 2018/1/22.  
 \*/*public class MusicService extends Service {  
 MediaPlayer player=new MediaPlayer();  
 @Override  
 public void onCreate() {  
 Log.*i*("yaoling1997","This is onCreate");  
 super.onCreate();  
 }  
 @Override  
 public int onStartCommand(Intent intent, int flags, int startId) {  
 if (intent!=null) {  
 String bgMusic = intent.getStringExtra(Macro.*BG\_MUSIC*);  
 if (bgMusic.equals(Macro.*OPEN*)) {  
 play();  
 } else {  
 stop();  
 }  
 }  
 return super.onStartCommand(intent, flags, startId);  
 }  
 @Nullable  
 @Override  
 public IBinder onBind(Intent intent) {  
 return null;  
 }  
 private void play(){  
 Log.*i*("yaoling1997","This is play");  
 try {  
 AssetFileDescriptor fileDescriptor=getAssets().openFd("backgroundMusic.mp3");  
 player.setDataSource(fileDescriptor.getFileDescriptor(),fileDescriptor.getStartOffset(), fileDescriptor.getLength());  
 player.setLooping(true);  
 player.prepare();  
 player.start();  
 } catch (Exception e) {  
 //e.printStackTrace();  
 }  
 }  
 private void stop(){  
 player.stop();  
 player.reset();  
 }  
 @Override  
 public void onDestroy() {  
 stop();  
 Log.*i*("yaoling1997","This is MusicService's onDestroy");  
 super.onDestroy();  
 }  
}

package com.example.acer\_pc.pokecats.gamedemo;  
import android.app.Activity;  
import android.graphics.drawable.AnimationDrawable;  
import android.os.Bundle;  
import android.support.annotation.Nullable;  
import android.view.Window;  
import android.widget.ImageView;  
*/\*\*  
 \* Created by acer-pc on 2018/1/29.  
 \*/*public class PassAllLevelsActivity extends Activity {  
 private ImageView ivPlayWithCats;  
 @Override  
 protected void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 requestWindowFeature(Window.*FEATURE\_NO\_TITLE*);  
 setContentView(R.layout.*activity\_pass\_all\_levels*);  
 bindViews();  
 }  
 private void bindViews(){  
 ivPlayWithCats= (ImageView)findViewById(R.id.*ivPlayWithCats*);  
 AnimationDrawable ad= (AnimationDrawable)ivPlayWithCats.getBackground();  
 ad.start();  
 }  
}

package com.example.acer\_pc.pokecats.gamedemo;  
import android.app.Activity;  
import android.os.Bundle;  
import android.support.annotation.Nullable;  
import android.view.Window;  
import android.widget.ListView;  
import java.util.LinkedList;  
import java.util.List;  
*/\*\*  
 \* Created by acer-pc on 2018/1/28.  
 \*/*public class PassModeActivity extends Activity {  
 private ListView lvPassMode;  
 private List<Integer> myData;  
 private PassModeAdapter passModeAdapter;  
 @Override  
 protected void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 requestWindowFeature(Window.*FEATURE\_NO\_TITLE*);  
 setContentView(R.layout.*activity\_pass\_mode*);  
 bindViews();  
 }  
 @Override  
 protected void onRestart() {  
 getPassModeAdapter();  
 lvPassMode.setAdapter(passModeAdapter);  
 super.onRestart();  
 }  
 private void getPassModeAdapter(){  
 myData= new LinkedList<Integer>();  
 for (int i=0;i<Macro.*LEVEL\_NUM*/PassModeAdapter.*colNum*;i++)  
 myData.add(i);  
 passModeAdapter= new PassModeAdapter(myData,this);  
 }  
 private void bindViews(){  
 lvPassMode= (ListView)findViewById(R.id.*lvPassMode*);  
 getPassModeAdapter();  
 lvPassMode.setAdapter(passModeAdapter);  
 }  
}

package com.example.acer\_pc.pokecats.gamedemo;  
import android.content.Context;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.BaseAdapter;  
import android.widget.Button;  
import java.util.List;  
*/\*\*  
 \* Created by acer-pc on 2018/1/28.  
 \*/*public class PassModeAdapter extends BaseAdapter {  
 public static final int *colNum*=3;//每行显示多少关  
 private List<Integer> myData;  
 private Context myContext;  
 public PassModeAdapter(List<Integer> myData,Context myContext) {  
 this.myData = myData;  
 this.myContext= myContext;  
 }  
 @Override  
 public int getCount() {  
 return myData.size();  
 }  
 @Override  
 public Object getItem(int i) {  
 return null;  
 }  
 @Override  
 public long getItemId(int i) {  
 return i;  
 }  
 private void setBtn(Button btn, final int level){  
 MainActivity.*addBtnAnimation*(btn);  
 SharedPreferences prefs= myContext.getSharedPreferences(Macro.*PREFS\_FILE*,Context.*MODE\_PRIVATE*);  
 if (prefs.getString(Macro.*LEVEL*+level,Macro.*CLOSE*).equals(Macro.*OPEN*)) {//true代表关卡已解锁  
 btn.setText("" + (level));  
 btn.setBackground(myContext.getDrawable(R.drawable.*pass\_mode\_unlock*));  
 btn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 Intent intent= new Intent();  
 intent.putExtra(Macro.*GAME\_MODE*,level);//0代表无限模式  
 intent.setAction(Macro.*ACTION\_START\_ACTIVITY*);  
 myContext.startActivity(intent);  
 }  
 });  
 }else {  
 btn.setText("");  
 btn.setBackground(myContext.getDrawable(R.drawable.*pass\_mode\_lock*));  
 }  
 }  
 @Override  
 public View getView(int i, View view, ViewGroup viewGroup) {  
 int rowId= myData.get(i);  
 int level1=rowId\*3+1,level2=rowId\*3+2,level3=rowId\*3+3;  
 view= LayoutInflater.*from*(myContext).inflate(R.layout.*pass\_mode\_adapter*,viewGroup,false);  
 Button btn1= view.findViewById(R.id.*btn1*);  
 Button btn2= view.findViewById(R.id.*btn2*);  
 Button btn3= view.findViewById(R.id.*btn3*);  
 setBtn(btn1,level1);  
 setBtn(btn2,level2);  
 setBtn(btn3,level3);  
 return view;  
 }  
}

package com.example.acer\_pc.pokecats.gamedemo;  
import android.app.Activity;  
import android.app.AlertDialog;  
import android.content.SharedPreferences;  
import android.graphics.drawable.ColorDrawable;  
import android.os.Bundle;  
import android.support.annotation.Nullable;  
import android.view.View;  
import android.view.Window;  
import android.view.WindowManager;  
import android.widget.Button;  
*/\*\*  
 \* Created by acer-pc on 2018/1/24.  
 \*/*public class ScoreboardActivity extends Activity {//查看积分界面  
 public static final int *NoNum*=3;  
 private SharedPreferences prefs;  
 private StrokeTextView []tvNo;  
 private Button btnReset;  
 @Override  
 protected void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 requestWindowFeature(Window.*FEATURE\_NO\_TITLE*);  
 setContentView(R.layout.*activity\_scoreboard*);  
 bindViews();  
 }  
 private void bindViews(){  
 tvNo= new StrokeTextView[*NoNum*];  
 tvNo[0]=(StrokeTextView)findViewById(R.id.*tvNo1*);  
 tvNo[1]=(StrokeTextView)findViewById(R.id.*tvNo2*);  
 tvNo[2]=(StrokeTextView)findViewById(R.id.*tvNo3*);  
 btnReset= (Button)findViewById(R.id.*btnReset*);  
 MainActivity.*addBtnAnimation*(btnReset);  
 btnReset.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 AlertDialog.Builder builder= new AlertDialog.Builder(ScoreboardActivity.this);  
 View alertDialogView;  
 alertDialogView = View.*inflate*(ScoreboardActivity.this, R.layout.*alertdialog\_reset\_scoreboard*, null);  
 StrokeTextView tvContent= (StrokeTextView)alertDialogView.findViewById(R.id.*tvContent*);  
 Button btnConfirm= (Button)alertDialogView.findViewById(R.id.*btnConfirm*);  
 Button btnCancel= (Button)alertDialogView.findViewById(R.id.*btnCancel*);  
 builder.setView(alertDialogView);  
 builder.setCancelable(false);  
 final AlertDialog dialog= builder.create();  
 btnConfirm.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 prefs=getSharedPreferences(Macro.*PREFS\_FILE*,*MODE\_PRIVATE*);  
 SharedPreferences.Editor editor = prefs.edit();//清空用户分数  
 for (int i=0;i<NoNum;i++) {  
 editor.remove(Macro.NO[i]);  
 }  
 editor.commit();  
 updateNo();  
 dialog.dismiss();  
 }  
 });  
 btnCancel.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 dialog.dismiss();  
 }  
 });  
 MainActivity.*addBtnAnimation*(btnConfirm);  
 MainActivity.*addBtnAnimation*(btnCancel);  
 dialog.getWindow().setBackgroundDrawable(new ColorDrawable());  
 dialog.show();//show必须放前面，不知道为啥  
 WindowManager.LayoutParams lp= dialog.getWindow().getAttributes();  
 lp.width=Macro.*ALERTDIALOG\_WIDTH*;  
 lp.height=Macro.*ALERTDIALOG\_HEIGHT*;  
 dialog.getWindow().setAttributes(lp);  
 }  
 });  
 updateNo();  
 }  
 private void updateNo() {  
 prefs=getSharedPreferences(Macro.*PREFS\_FILE*,*MODE\_PRIVATE*);  
 for (int i=0;i<*NoNum*;i++) {  
 int score = prefs.getInt(Macro.*NO*[i], 0);  
 tvNo[i].mySetText(Macro.*NO*[i] + ": " + score);  
 }  
 }  
}

package com.example.acer\_pc.pokecats.gamedemo;  
import android.annotation.SuppressLint;  
import android.app.Activity;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.support.annotation.Nullable;  
import android.view.View;  
import android.view.Window;  
import android.widget.Button;  
*/\*\*  
 \* Created by acer-pc on 2018/1/23.  
 \*/*public class SettingActivity extends Activity {  
 Button btnBgMusic,btnSounds;  
 SharedPreferences prefs;  
 @Override  
 protected void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 requestWindowFeature(Window.FEATURE\_NO\_TITLE);  
 setContentView(R.layout.activity\_setting);  
 bindViews();  
 btnBgMusic.setOnClickListener(new View.OnClickListener() {  
 @SuppressLint("NewApi")  
 @Override  
 public void onClick(View view) {  
 prefs=getSharedPreferences(Macro.PREFS\_FILE,MODE\_PRIVATE);  
 SharedPreferences.Editor editor = prefs.edit();  
 if (prefs.getString(Macro.BG\_MUSIC,Macro.CLOSE).equals(Macro.CLOSE)){//close->open  
 editor.putString(Macro.BG\_MUSIC,Macro.OPEN);  
 }else {//open->close  
 editor.putString(Macro.BG\_MUSIC,Macro.CLOSE);  
 }  
 editor.commit();  
 applyPrefs();  
 }  
 });  
 btnSounds.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 prefs=getSharedPreferences(Macro.PREFS\_FILE,MODE\_PRIVATE);  
 SharedPreferences.Editor editor = prefs.edit();  
 if (prefs.getString(Macro.SOUNDS,Macro.CLOSE).equals(Macro.CLOSE)){//close->open  
 editor.putString(Macro.SOUNDS,Macro.OPEN);  
 }else {//open->close  
 editor.putString(Macro.SOUNDS,Macro.CLOSE);  
 }  
 editor.commit();  
 applyPrefs();  
 }  
 });  
 }  
 @SuppressLint("NewApi")  
 private void bindViews(){  
 btnBgMusic= (Button)findViewById(R.id.btnBgMusic);  
 btnSounds= (Button)findViewById(R.id.btnSounds);  
 applyPrefs();  
 }  
 @SuppressLint("NewApi")  
 private void applyPrefs(){  
 prefs=getSharedPreferences(Macro.PREFS\_FILE,MODE\_PRIVATE);  
 Intent intent= new Intent(this,MusicService.class);  
 if (prefs.getString(Macro.BG\_MUSIC,Macro.CLOSE).equals(Macro.OPEN)) {  
 btnBgMusic.setBackgroundDrawable(getDrawable(R.drawable.btn\_on));  
 intent.putExtra(Macro.BG\_MUSIC,prefs.getString(Macro.BG\_MUSIC,Macro.OPEN));//open or close  
 }else {  
 btnBgMusic.setBackgroundDrawable(getDrawable(R.drawable.btn\_off));  
 intent.putExtra(Macro.BG\_MUSIC,prefs.getString(Macro.BG\_MUSIC,Macro.CLOSE));//open or close  
 }  
 if (prefs.getString(Macro.SOUNDS,Macro.CLOSE).equals(Macro.OPEN)) {  
 btnSounds.setBackgroundDrawable(getDrawable(R.drawable.btn\_on));  
 }else {  
 btnSounds.setBackgroundDrawable(getDrawable(R.drawable.btn\_off));  
 }  
 btnBgMusic.invalidate();  
 startService(intent);  
 }  
}

package com.example.acer\_pc.pokecats.gamedemo;  
import android.app.Activity;  
import android.os.Build;  
import android.os.Bundle;  
import android.support.annotation.Nullable;  
import android.support.annotation.RequiresApi;  
import android.util.Log;  
import android.view.Window;  
import android.widget.RelativeLayout;  
import java.util.Timer;  
*/\*\*  
 \* Created by acer-pc on 2018/1/23.  
 \*/*public class StartActivity extends Activity {  
 public static Timer timer;  
 private RelativeLayout layout;  
 private GameView gameView;  
 public StrokeTextView tvLevel,tvHP,tvScore;  
 private int level;//玩家选择的关卡，0表示无限模式  
 @RequiresApi(api = Build.VERSION\_CODES.LOLLIPOP)  
 @Override  
 protected void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 requestWindowFeature(Window.FEATURE\_NO\_TITLE);  
 setContentView(R.layout.activity\_start);  
 bindViews();  
 level=getIntent().getIntExtra(Macro.GAME\_MODE,-1);  
 Log.i("yaoling1997","StartActivity,level: "+level);  
 if (level<0)  
 finish();  
 if (level>0)  
 updateLevel(level);  
 timer= new Timer();  
 Log.i("yaoling1997","startActivity");  
 gameView= new GameView(this);  
 Log.i("yaoling1997","before initGame");  
 gameView.initGame();  
 Log.i("yaoling1997","before afterInitGame");  
 layout.addView(gameView);  
 }  
 private void bindViews(){  
 layout= (RelativeLayout)findViewById(R.id.startLayout);  
 tvLevel= (StrokeTextView)findViewById(R.id.tvLevel);  
 tvHP= (StrokeTextView)findViewById(R.id.*tvHP*);  
 tvScore= (StrokeTextView)findViewById(R.id.*tvScore*);  
 Log.*i*("yaoling1997","R.id.tvHP:"+R.id.*tvHP*);  
 }  
 public int getLevel() {  
 return level;  
 }  
 public void updateLevel(int newLevel){  
 Log.*i*("yaoling1997","updateLevel: "+newLevel);  
 level=newLevel;  
 tvLevel.mySetText("关卡 "+level+" ");  
 if (level>=30){//关卡>=30换背景  
 layout.setBackground(getDrawable(R.drawable.*start\_background2*));  
 Log.*i*("yaoling1997","change startBackground");  
 }  
 }  
 @Override  
 protected void onDestroy() {  
 *timer*.cancel();  
 if (gameView.soundPool!=null)  
 gameView.soundPool.release();  
 super.onDestroy();  
 }  
}

package com.example.acer\_pc.pokecats.gamedemo;  
import android.content.Context;  
import android.graphics.Canvas;  
import android.graphics.Color;  
import android.graphics.Paint;  
import android.text.TextPaint;  
import android.util.AttributeSet;  
import android.view.ViewGroup;  
import android.widget.TextView;  
*/\*\*  
 \* Created by acer-pc on 2018/1/23.  
 \*/*public class StrokeTextView extends android.support.v7.widget.AppCompatTextView {  
 private TextView outlineTextView = null;  
 public StrokeTextView(Context context)  
 {  
 super(context);  
 outlineTextView = new TextView(context);  
 init();  
 }  
 public StrokeTextView(Context context, AttributeSet attrs)  
 {  
 super(context, attrs);  
 outlineTextView = new TextView(context, attrs);  
 init();  
 }  
 public StrokeTextView(Context context, AttributeSet attrs, int defStyle)  
 {  
 super(context, attrs, defStyle);  
 outlineTextView = new TextView(context, attrs, defStyle);  
 init();  
 }  
 public void init()  
 {  
 TextPaint paint = outlineTextView.getPaint();  
 paint.setStrokeWidth(5);// 描边宽度  
 paint.setStyle(Paint.Style.*STROKE*);  
 outlineTextView.setTextColor(Color.*parseColor*("#000000"));// 描边颜色  
 outlineTextView.setGravity(getGravity());  
 }  
 public void mySetText(String s){  
 this.setText(s);  
 outlineTextView.setText(s);  
 }  
 @Override  
 public void setLayoutParams (ViewGroup.LayoutParams params)  
 {  
 super.setLayoutParams(params);  
 outlineTextView.setLayoutParams(params);  
 }  
 @Override  
 protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec)  
 {  
 super.onMeasure(widthMeasureSpec, heightMeasureSpec);  
 // 设置轮廓文字  
 CharSequence outlineText = outlineTextView.getText();  
 if (outlineText == null || !outlineText.equals(this.getText()))  
 {  
 outlineTextView.setText(getText());  
 postInvalidate();  
 }  
 outlineTextView.measure(widthMeasureSpec, heightMeasureSpec);  
 }  
 @Override  
 protected void onLayout (boolean changed, int left, int top, int right, int bottom)  
 {  
 super.onLayout(changed, left, top, right, bottom);  
 outlineTextView.layout(left, top, right, bottom);  
 }  
 @Override  
 protected void onDraw(Canvas canvas)  
 {  
 outlineTextView.draw(canvas);  
 super.onDraw(canvas);  
 }  
}

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@drawable/background"  
 android:id="@+id/mainLayout">  
 <ImageView  
 android:id="@+id/ivInfiniteModeTitle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_alignParentTop="true"  
 android:src="@drawable/infinite\_mode\_title" />  
 <Button  
 android:id="@+id/btnStart"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@id/ivInfiniteModeTitle"  
 android:layout\_marginTop="150dp"  
 android:text="@string/startGame"  
 android:onClick="onClick"  
 android:background="@drawable/btn\_background"  
 style="@style/myButtonStyle"/>  
 <Button  
 android:id="@+id/btnScoreboard"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@id/btnStart"  
 android:layout\_marginTop="10dp"  
 android:text="@string/scoreboard"  
 android:onClick="onClick"  
 android:background="@drawable/btn\_background"  
 style="@style/myButtonStyle"/>  
</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@drawable/background"  
 android:id="@+id/settingLayout">  
 <ImageView  
 android:id="@+id/ivIntroductionTitle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_alignParentTop="true"  
 android:src="@drawable/introduction\_title" />  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@+id/ivIntroductionTitle"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="100dp"  
 >  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 >  
 <TextView  
 android:id="@+id/tvCat1"  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginLeft="10dp"  
 android:background="@drawable/cat1\_introduction" />  
 <com.example.acer\_pc.pokecats.gamedemo.StrokeTextView  
 android:id="@+id/tvCat1Info"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textAllCaps="true"  
 android:text="@string/cat1Introduction"  
 android:textSize="15sp"  
 android:textColor="#EA5246"  
 android:textStyle="bold|italic"  
 android:gravity="center\_vertical|left"  
 android:layout\_alignTop="@+id/tvCat1"  
 android:layout\_toRightOf="@+id/tvCat1"  
 android:layout\_marginTop="30dp"  
 android:layout\_marginLeft="10dp"  
 />  
 <TextView  
 android:id="@+id/tvCat2"  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:layout\_alignParentLeft="true"  
 android:layout\_below="@+id/tvCat1"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginLeft="10dp"  
 android:background="@drawable/cat2\_introduction" />  
 <com.example.acer\_pc.pokecats.gamedemo.StrokeTextView  
 android:id="@+id/tvCat2Info"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textAllCaps="true"  
 android:text="@string/cat2Introduction"  
 android:textSize="15sp"  
 android:textColor="#EA5246"  
 android:textStyle="bold|italic"  
 android:gravity="center\_vertical|left"  
 android:layout\_alignTop="@+id/tvCat2"  
 android:layout\_toRightOf="@+id/tvCat2"  
 android:layout\_marginTop="30dp"  
 android:layout\_marginLeft="10dp"  
 />  
 <TextView  
 android:id="@+id/tvCat3"  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:layout\_alignParentLeft="true"  
 android:layout\_below="@+id/tvCat2"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginLeft="10dp"  
 android:background="@drawable/cat3\_introduction" />  
 <com.example.acer\_pc.pokecats.gamedemo.StrokeTextView  
 android:id="@+id/tvCat3Info"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textAllCaps="true"  
 android:text="@string/cat3Introduction"  
 android:textSize="15sp"  
 android:textColor="#EA5246"  
 android:textStyle="bold|italic"  
 android:gravity="center\_vertical|left"  
 android:layout\_alignTop="@+id/tvCat3"  
 android:layout\_toRightOf="@+id/tvCat3"  
 android:layout\_marginTop="30dp"  
 android:layout\_marginLeft="10dp"  
 />  
 <TextView  
 android:id="@+id/tvCat4"  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:layout\_alignParentLeft="true"  
 android:layout\_below="@+id/tvCat3"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginLeft="10dp"  
 android:background="@drawable/cat4\_1" />  
 <com.example.acer\_pc.pokecats.gamedemo.StrokeTextView  
 android:id="@+id/tvCat4Info"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textAllCaps="true"  
 android:text="@string/cat4Introduction"  
 android:textSize="15sp"  
 android:textColor="#EA5246"  
 android:textStyle="bold|italic"  
 android:gravity="center\_vertical|left"  
 android:layout\_alignTop="@+id/tvCat4"  
 android:layout\_toRightOf="@+id/tvCat4"  
 android:layout\_marginTop="30dp"  
 android:layout\_marginLeft="10dp"  
 />  
 <TextView  
 android:id="@+id/tvBomb"  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:layout\_alignParentLeft="true"  
 android:layout\_below="@+id/tvCat4"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginLeft="10dp"  
 android:background="@drawable/bomb" />  
 <com.example.acer\_pc.pokecats.gamedemo.StrokeTextView  
 android:id="@+id/tvBombInfo"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textAllCaps="true"  
 android:text="@string/bombIntroduction"  
 android:textSize="15sp"  
 android:textColor="#EA5246"  
 android:textStyle="bold|italic"  
 android:gravity="center\_vertical|left"  
 android:layout\_alignTop="@+id/tvBomb"  
 android:layout\_toRightOf="@+id/tvBomb"  
 android:layout\_marginTop="30dp"  
 android:layout\_marginLeft="10dp"  
 />  
 <com.example.acer\_pc.pokecats.gamedemo.StrokeTextView  
 android:id="@+id/tvBonus"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textAllCaps="true"  
 android:text="@string/bonusIntroduction"  
 android:textSize="15sp"  
 android:textColor="#EA5246"  
 android:textStyle="bold|italic"  
 android:gravity="center\_vertical|left"  
 android:layout\_alignParentLeft="true"  
 android:layout\_below="@+id/tvBomb"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginLeft="30dp"  
 />  
 </RelativeLayout>  
 </ScrollView>  
</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@drawable/background"  
 android:id="@+id/mainLayout">  
 <ImageView  
 android:id="@+id/ivMainTitle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_alignParentTop="true"  
 android:src="@drawable/main\_title" />  
 <Button  
 android:id="@+id/btnPassMode"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@id/ivMainTitle"  
 android:layout\_marginTop="150dp"  
 android:text="@string/passMode"  
 android:onClick="onClick"  
 android:background="@drawable/btn\_background"  
 style="@style/myButtonStyle"/>  
 <Button  
 android:id="@+id/btnInfiniteMode"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@id/btnPassMode"  
 android:layout\_marginTop="10dp"  
 android:text="@string/infiniteMode"  
 android:onClick="onClick"  
 android:background="@drawable/btn\_background"  
 style="@style/myButtonStyle"/>  
 <Button  
 android:id="@+id/btnSetting"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@id/btnInfiniteMode"  
 android:layout\_marginTop="10dp"  
 android:text="@string/setting"  
 android:onClick="onClick"  
 android:background="@drawable/btn\_background"  
 style="@style/myButtonStyle"/>  
 <Button  
 android:id="@+id/btnIntroduction"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@id/btnSetting"  
 android:layout\_marginTop="10dp"  
 android:text="@string/introduction"  
 android:onClick="onClick"  
 android:background="@drawable/btn\_background"  
 style="@style/myButtonStyle"/>  
 <Button  
 android:id="@+id/btnExit"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@id/btnIntroduction"  
 android:layout\_marginTop="10dp"  
 android:text="@string/exit"  
 android:onClick="onClick"  
 android:background="@drawable/btn\_background"  
 style="@style/myButtonStyle"/>  
</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@drawable/start\_background"  
 >  
 <ImageView  
 android:id="@+id/ivPlayWithCats"  
 android:layout\_width="300dp"  
 android:layout\_height="300dp"  
 android:layout\_alignParentTop="true"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="50dp"  
 android:background="@drawable/pass\_all\_levels"  
 />  
 <com.example.acer\_pc.pokecats.gamedemo.StrokeTextView  
 android:id="@+id/tvTheEnd"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textAllCaps="true"  
 android:text="@string/theEnd"  
 android:textSize="15sp"  
 android:textColor="#EA5246"  
 android:textStyle="bold|italic"  
 android:gravity="center\_vertical|left"  
 android:layout\_below="@id/ivPlayWithCats"  
 android:layout\_marginTop="10dp"  
 android:layout\_centerHorizontal="true"  
 />  
 <ImageView  
 android:id="@+id/ivTheEnd"  
 android:layout\_width="100dp"  
 android:layout\_height="44dp"  
 android:layout\_below="@id/tvTheEnd"  
 android:layout\_alignParentRight="true"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginRight="3dp"  
 android:background="@drawable/the\_end"  
 />  
</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@drawable/background"  
 android:id="@+id/mainLayout">  
 <ImageView  
 android:id="@+id/ivPassModeTitle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_alignParentTop="true"  
 android:src="@drawable/pass\_mode\_title" />  
 <ListView  
 android:id="@+id/lvPassMode"  
 android:layout\_centerHorizontal="true"  
 android:layout\_below="@id/ivPassModeTitle"  
 android:layout\_marginTop="100dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center\_horizontal"  
 android:footerDividersEnabled="false"  
 android:headerDividersEnabled="false"  
 android:divider="@null">  
 </ListView>  
</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@drawable/background"  
 android:id="@+id/settingLayout">  
 <ImageView  
 android:id="@+id/ivScoreboardTitle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_alignParentTop="true"  
 android:src="@drawable/scoreboard\_title" />  
 <com.example.acer\_pc.pokecats.gamedemo.StrokeTextView  
 android:id="@+id/tvNo1"  
 android:layout\_above="@id/tvNo2"  
 android:layout\_marginBottom="30dp"  
 android:text="@string/No1"  
 style="@style/myTextViewStyle"  
 />  
 <com.example.acer\_pc.pokecats.gamedemo.StrokeTextView  
 android:id="@+id/tvNo2"  
 android:layout\_centerVertical="true"  
 android:text="@string/No2"  
 style="@style/myTextViewStyle"  
 />  
 <com.example.acer\_pc.pokecats.gamedemo.StrokeTextView  
 android:id="@+id/tvNo3"  
 android:layout\_below="@id/tvNo2"  
 android:layout\_marginTop="30dp"  
 android:text="@string/No3"  
 style="@style/myTextViewStyle"  
 />  
 <Button  
 android:id="@+id/btnReset"  
 android:layout\_width="96dp"  
 android:layout\_height="40dp"  
 android:background="@drawable/btn\_background"  
 android:layout\_below="@id/tvNo3"  
 android:layout\_marginTop="30dp"  
 android:layout\_alignParentRight="true"  
 android:layout\_marginRight="30dp"  
 android:textColor="#000055"  
 android:textStyle="bold"  
 android:textSize="15sp"  
 android:textAllCaps="false"  
 android:text="@string/reset"  
 />  
</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@drawable/background"  
 android:id="@+id/settingLayout">  
 <ImageView  
 android:id="@+id/ivSettingTitle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_alignParentTop="true"  
 android:src="@drawable/setting\_title" />  
 <com.example.acer\_pc.pokecats.gamedemo.StrokeTextView  
 android:id="@+id/tvBgMusic"  
 android:layout\_above="@id/tvSounds"  
 android:layout\_marginBottom="30dp"  
 android:text="@string/bgMusic"  
 style="@style/myTextViewStyle"  
 />  
 <Button  
 android:id="@+id/btnBgMusic"  
 android:layout\_width="50dp"  
 android:layout\_height="25dp"  
 android:layout\_alignTop="@+id/tvBgMusic"  
 android:layout\_alignParentRight="true"  
 android:layout\_marginRight="30dp"  
 android:background="@drawable/btn\_off"  
 />  
 <com.example.acer\_pc.pokecats.gamedemo.StrokeTextView  
 android:id="@+id/tvSounds"  
 android:layout\_centerVertical="true"  
 android:text="@string/sounds"  
 style="@style/myTextViewStyle"  
 />  
 <Button  
 android:id="@+id/btnSounds"  
 android:layout\_width="50dp"  
 android:layout\_height="25dp"  
 android:layout\_alignTop="@+id/tvSounds"  
 android:layout\_alignParentRight="true"  
 android:layout\_marginRight="30dp"  
 android:background="@drawable/btn\_off"  
 />  
</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@drawable/start\_background"  
 android:id="@+id/startLayout">  
 <com.example.acer\_pc.pokecats.gamedemo.StrokeTextView  
 android:id="@+id/tvLevel"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginTop="3dp"  
 android:layout\_marginLeft="3dp"  
 android:layout\_alignParentLeft="true"  
 android:textSize="15sp"  
 android:textColor="#EA5246"  
 android:textStyle="bold|italic"  
 android:gravity="center\_vertical|left"  
 />  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="3dp"  
 android:layout\_alignParentTop="true"  
 android:layout\_centerHorizontal="true"  
 >  
 <com.example.acer\_pc.pokecats.gamedemo.StrokeTextView  
 android:id="@+id/tvHP"  
 android:layout\_width="120dp"  
 android:layout\_height="70dp"  
 android:layout\_centerVertical="true"  
 android:layout\_alignParentLeft="true"  
 android:gravity="center"  
 android:text="10"  
 android:textColor="#ff0000"  
 android:textStyle="bold|italic"  
 android:textSize="18sp"  
 android:background="@drawable/info\_background"  
 android:drawablePadding="0dp"  
 android:drawableLeft="@drawable/red\_heart"  
 />  
 <com.example.acer\_pc.pokecats.gamedemo.StrokeTextView  
 android:id="@+id/tvScore"  
 android:layout\_width="120dp"  
 android:layout\_height="70dp"  
 android:layout\_centerVertical="true"  
 android:layout\_alignParentLeft="true"  
 android:gravity="center"  
 android:text="100000"  
 android:textColor="#ff0000"  
 android:textStyle="bold|italic"  
 android:textSize="18sp"  
 android:background="@drawable/info\_background"  
 android:drawableLeft="@drawable/yellow\_star"  
 />  
 </LinearLayout>  
</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="220dp"  
 android:layout\_height="105dp"  
 android:background="@drawable/alert\_background"  
 >  
 <TextView  
 android:id="@+id/tvTitle"  
 android:layout\_width="150dp"  
 android:layout\_height="30dp"  
 android:background="@drawable/game\_over"  
 android:layout\_marginLeft="5dp"  
 android:layout\_marginTop="5dp"/>  
 <com.example.acer\_pc.pokecats.gamedemo.StrokeTextView  
 android:id="@+id/tvContent"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textAllCaps="true"  
 android:text="@string/score"  
 android:textSize="15sp"  
 android:textColor="#EA5246"  
 android:textStyle="bold|italic"  
 android:gravity="center\_vertical|left"  
 android:layout\_below="@id/tvTitle"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginLeft="10dp"  
 />  
 <Button  
 android:id="@+id/btnStart"  
 android:layout\_width="96dp"  
 android:layout\_height="40dp"  
 android:background="@drawable/btn\_background"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentRight="true"  
 android:layout\_marginRight="8dp"  
 android:layout\_marginTop="92dp"  
 android:textColor="#000055"  
 android:textStyle="bold"  
 android:textSize="15sp"  
 android:textAllCaps="false"  
 android:text="@string/restart"  
 />  
 <Button  
 android:id="@+id/btnEnd"  
 android:layout\_width="96dp"  
 android:layout\_height="40dp"  
 android:background="@drawable/btn\_background"  
 android:layout\_alignTop="@+id/btnStart"  
 android:layout\_toLeftOf="@+id/btnStart"  
 android:layout\_marginRight="5dp"  
 android:textColor="#000055"  
 android:textStyle="bold"  
 android:textSize="15sp"  
 android:textAllCaps="false"  
 android:text="@string/end"  
 />  
</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="220dp"  
 android:layout\_height="105dp"  
 android:background="@drawable/alert\_background"  
 >  
 <TextView  
 android:id="@+id/tvTitle"  
 android:layout\_width="60dp"  
 android:layout\_height="60dp"  
 android:background="@drawable/pass"  
 android:layout\_marginLeft="5dp"  
 android:layout\_marginTop="5dp"/>  
 <com.example.acer\_pc.pokecats.gamedemo.StrokeTextView  
 android:id="@+id/tvContent"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textAllCaps="true"  
 android:text="@string/score"  
 android:textSize="15sp"  
 android:textColor="#EA5246"  
 android:textStyle="bold|italic"  
 android:gravity="center\_vertical|left"  
 android:layout\_below="@id/tvTitle"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginLeft="10dp"  
 />  
 <Button  
 android:id="@+id/btnStart"  
 android:layout\_width="96dp"  
 android:layout\_height="40dp"  
 android:background="@drawable/btn\_background"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentRight="true"  
 android:layout\_marginRight="8dp"  
 android:layout\_marginTop="92dp"  
 android:textColor="#000055"  
 android:textStyle="bold"  
 android:textSize="15sp"  
 android:textAllCaps="false"  
 android:text="@string/restart"  
 />  
 <Button  
 android:id="@+id/btnEnd"  
 android:layout\_width="96dp"  
 android:layout\_height="40dp"  
 android:background="@drawable/btn\_background"  
 android:layout\_alignTop="@+id/btnStart"  
 android:layout\_toLeftOf="@+id/btnStart"  
 android:layout\_marginRight="5dp"  
 android:textColor="#000055"  
 android:textStyle="bold"  
 android:textSize="15sp"  
 android:textAllCaps="false"  
 android:text="@string/end"  
 />  
</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="220dp"  
 android:layout\_height="105dp"  
 android:background="@drawable/alert\_background"  
 >  
 <TextView  
 android:id="@+id/tvTitle"  
 android:layout\_width="150dp"  
 android:layout\_height="30dp"  
 android:layout\_marginLeft="5dp"  
 android:layout\_marginTop="5dp"/>  
 <com.example.acer\_pc.pokecats.gamedemo.StrokeTextView  
 android:id="@+id/tvContent"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textAllCaps="true"  
 android:text="@string/resetScoreboard"  
 android:textSize="15sp"  
 android:textColor="#EA5246"  
 android:textStyle="bold|italic"  
 android:gravity="center\_vertical|left"  
 android:layout\_below="@id/tvTitle"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginLeft="10dp"  
 />  
 <Button  
 android:id="@+id/btnConfirm"  
 android:layout\_width="96dp"  
 android:layout\_height="40dp"  
 android:background="@drawable/btn\_background"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentRight="true"  
 android:layout\_marginRight="8dp"  
 android:layout\_marginTop="92dp"  
 android:textColor="#000055"  
 android:textStyle="bold"  
 android:textSize="15sp"  
 android:textAllCaps="false"  
 android:text="@string/confirm"  
 />  
 <Button  
 android:id="@+id/btnCancel"  
 android:layout\_width="96dp"  
 android:layout\_height="40dp"  
 android:background="@drawable/btn\_background"  
 android:layout\_alignTop="@+id/btnConfirm"  
 android:layout\_toLeftOf="@+id/btnConfirm"  
 android:layout\_marginRight="5dp"  
 android:textColor="#000055"  
 android:textStyle="bold"  
 android:textSize="15sp"  
 android:textAllCaps="false"  
 android:text="@string/cancel"  
 />  
</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/mainLayout">  
 <Button  
 android:id="@+id/btn1"  
 android:layout\_width="60dp"  
 android:layout\_height="60dp"  
 android:textColor="#3f4fef"  
 android:textStyle="bold"  
 android:textSize="18sp"  
 android:textAllCaps="false"  
 android:layout\_toLeftOf="@id/btn2"  
 android:layout\_marginRight="20dp"  
 android:background="@drawable/pass\_mode\_unlock"  
 />  
 <Button  
 android:id="@+id/btn2"  
 android:layout\_width="60dp"  
 android:layout\_height="60dp"  
 android:textColor="#3f4fef"  
 android:textStyle="bold"  
 android:textSize="18sp"  
 android:textAllCaps="false"  
 android:layout\_centerHorizontal="true"  
 android:background="@drawable/pass\_mode\_unlock"  
 />  
 <Button  
 android:id="@+id/btn3"  
 android:layout\_width="60dp"  
 android:layout\_height="60dp"  
 android:textColor="#3f4fef"  
 android:textStyle="bold"  
 android:textSize="18sp"  
 android:textAllCaps="false"  
 android:layout\_toRightOf="@id/btn2"  
 android:layout\_marginLeft="20dp"  
 android:background="@drawable/pass\_mode\_unlock"  
 />  
</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="colorPrimary">#3F51B5</color>  
 <color name="colorPrimaryDark">#303F9F</color>  
 <color name="colorAccent">#FF4081</color>  
</resources>

<resources>  
 <string name="app\_name">PokeCats</string>  
 <string name="btnSpace">50dp</string>  
 <string name="bgMusic">背景音乐：</string>  
 <string name="sounds">音效：</string>  
 <string name="infiniteMode">无限模式</string>  
 <string name="setting">设置</string>  
 <string name="scoreboard">积分榜</string>  
 <string name="exit">退出</string>  
 <string name="startGame">开始游戏</string>  
 <string name="No1">No.1：</string>  
 <string name="No2">No.2：</string>  
 <string name="No3">No.3：</string>  
 <string name="score">得分：</string>  
 <string name="restart">再来一局</string>  
 <string name="end">歇会儿</string>  
 <string name="introduction">游戏介绍</string>  
 <string name="cat1Introduction">稳重的新加坡猫，捉到加一分。</string>  
 <string name="cat2Introduction">可靠的俄罗斯蓝猫，捉到加三分。</string>  
 <string name="cat3Introduction">淘气的曼切堪猫，捉到加五分。</string>  
 <string name="cat4Introduction">作者：李奕君。热爱编程的稀有猫，捉到生命值加一。</string>  
 <string name="bombIntroduction">炸弹，捉到生命值减一。</string>  
 <string name="bonusIntroduction">连续捉到同一种猫有额外分数加成！</string>  
 <string name="passMode">闯关模式</string>  
 <string name="resetScoreboard">您确定要重置积分榜吗？</string>  
 <string name="confirm">确定</string>  
 <string name="cancel">取消</string>  
 <string name="reset">重置</string>  
 <string name="theEnd">恭喜您通过重重考验，成为一位优秀的撸猫达人！</string>  
</resources>

<resources>  
 <!-- Base application theme. -->  
 <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">  
 <!-- Customize your theme here. -->  
 <item name="colorPrimary">@color/colorPrimary</item>  
 <item name="colorPrimaryDark">@color/colorPrimaryDark</item>  
 <item name="colorAccent">@color/colorAccent</item>  
 </style>  
 <style name="myButtonStyle">  
 <item name="android:textColor">#000055</item>  
 <item name="android:textStyle">bold</item>  
 <item name="android:textSize">18sp</item>  
 <item name="android:textAllCaps">false</item>  
 <item name="android:layout\_width">200dp</item>  
 <item name="android:layout\_height">50dp</item>  
 </style>  
 <style name="myTextViewStyle">  
 <item name="android:textColor">#EA5246</item>  
 <item name="android:textStyle">bold|italic</item>  
 <item name="android:gravity">center\_vertical|left</item>  
 <item name="android:textSize">18sp</item>  
 <item name="android:layout\_alignParentLeft">true</item>  
 <item name="android:layout\_marginLeft">30dp</item>  
 <item name="android:layout\_width">200dp</item>  
 <item name="android:layout\_height">25dp</item>  
 </style>  
</resources>

<?xml version="1.0" encoding="utf-8"?>  
<animation-list xmlns:android="http://schemas.android.com/apk/res/android"  
 android:oneshot="false"  
 >  
 <item  
 android:drawable="@drawable/pass\_all\_levels\_1"  
 android:duration="80"  
 />  
 <item  
 android:drawable="@drawable/pass\_all\_levels\_2"  
 android:duration="80"  
 />  
 <item  
 android:drawable="@drawable/pass\_all\_levels\_3"  
 android:duration="80"  
 />  
 <item  
 android:drawable="@drawable/pass\_all\_levels\_4"  
 android:duration="80"  
 />  
 <item  
 android:drawable="@drawable/pass\_all\_levels\_5"  
 android:duration="80"  
 />  
 <item  
 android:drawable="@drawable/pass\_all\_levels\_6"  
 android:duration="80"  
 />  
</animation-list>

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.acer\_pc.pokecats.gamedemo">  
 <application  
 android:allowBackup="true"  
 android:icon="@drawable/icon"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"  
 android:largeHeap="true">  
 <activity android:name=".MainActivity"  
 android:screenOrientation="portrait"  
 >  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 <service android:name=".MusicService">  
 <intent-filter>  
 <action android:name="lyj.action.MusicService"/>  
 </intent-filter>  
 </service>  
 <activity android:name=".StartActivity"  
 android:screenOrientation="portrait"  
 >  
 <intent-filter>  
 <action android:name="lyj.action.start" />  
 <category android:name="android.intent.category.DEFAULT"/>  
 </intent-filter>  
 </activity>  
 <activity android:name=".SettingActivity"  
 android:screenOrientation="portrait"  
 >  
 <intent-filter>  
 <action android:name="lyj.action.setting" />  
 <category android:name="android.intent.category.DEFAULT"/>  
 </intent-filter>  
 </activity>  
 <activity android:name=".ScoreboardActivity"  
 android:screenOrientation="portrait"  
 >  
 <intent-filter>  
 <action android:name="lyj.action.scoreboard" />  
 <category android:name="android.intent.category.DEFAULT"/>  
 </intent-filter>  
 </activity>  
 <activity android:name=".IntroductionActivity"  
 android:screenOrientation="portrait"  
 >  
 <intent-filter>  
 <action android:name="lyj.action.introduction" />  
 <category android:name="android.intent.category.DEFAULT"/>  
 </intent-filter>  
 </activity>  
 <activity android:name=".PassModeActivity"  
 android:screenOrientation="portrait">  
 <intent-filter>  
 <action android:name="lyj.action.passMode" />  
 <category android:name="android.intent.category.DEFAULT"/>  
 </intent-filter>  
 </activity>  
 <activity android:name=".InfiniteModeActivity"  
 android:screenOrientation="portrait">  
 <intent-filter>  
 <action android:name="lyj.action.infiniteMode" />  
 <category android:name="android.intent.category.DEFAULT"/>  
 </intent-filter>  
 </activity>  
 <activity android:name=".PassAllLevelsActivity"  
 android:screenOrientation="portrait">  
 <intent-filter>  
 <action android:name="lyj.action.passAllLevels" />  
 <category android:name="android.intent.category.DEFAULT"/>  
 </intent-filter>  
 </activity>  
 </application>  
</manifest>

<?xml version="1.0" encoding="utf-8"?>  
<vector xmlns:android="http://schemas.android.com/apk/res/android"  
 android:width="108dp"  
 android:height="108dp"  
 android:viewportHeight="108"  
 android:viewportWidth="108">  
 <path  
 android:fillColor="#26A69A"  
 android:pathData="M0,0h108v108h-108z" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M9,0L9,108"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M19,0L19,108"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M29,0L29,108"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M39,0L39,108"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M49,0L49,108"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M59,0L59,108"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M69,0L69,108"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M79,0L79,108"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M89,0L89,108"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M99,0L99,108"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M0,9L108,9"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M0,19L108,19"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M0,29L108,29"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M0,39L108,39"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M0,49L108,49"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M0,59L108,59"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M0,69L108,69"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M0,79L108,79"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M0,89L108,89"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M0,99L108,99"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M19,29L89,29"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M19,39L89,39"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M19,49L89,49"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M19,59L89,59"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M19,69L89,69"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M19,79L89,79"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M29,19L29,89"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M39,19L39,89"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M49,19L49,89"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M59,19L59,89"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M69,19L69,89"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M79,19L79,89"  
 android:strokeColor="#33FFFFFF"  
 android:strokeWidth="0.8" />  
</vector>

<vector xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:aapt="http://schemas.android.com/aapt"  
 android:width="108dp"  
 android:height="108dp"  
 android:viewportHeight="108"  
 android:viewportWidth="108">  
 <path  
 android:fillType="evenOdd"  
 android:pathData="M32,64C32,64 38.39,52.99 44.13,50.95C51.37,48.37 70.14,49.57 70.14,49.57L108.26,87.69L108,109.01L75.97,107.97L32,64Z"  
 android:strokeColor="#00000000"  
 android:strokeWidth="1">  
 <aapt:attr name="android:fillColor">  
 <gradient  
 android:endX="78.5885"  
 android:endY="90.9159"  
 android:startX="48.7653"  
 android:startY="61.0927"  
 android:type="linear">  
 <item  
 android:color="#44000000"  
 android:offset="0.0" />  
 <item  
 android:color="#00000000"  
 android:offset="1.0" />  
 </gradient>  
 </aapt:attr>  
 </path>  
 <path  
 android:fillColor="#FFFFFF"  
 android:fillType="nonZero"  
 android:pathData="M66.94,46.02L66.94,46.02C72.44,50.07 76,56.61 76,64L32,64C32,56.61 35.56,50.11 40.98,46.06L36.18,41.19C35.45,40.45 35.45,39.3 36.18,38.56C36.91,37.81 38.05,37.81 38.78,38.56L44.25,44.05C47.18,42.57 50.48,41.71 54,41.71C57.48,41.71 60.78,42.57 63.68,44.05L69.11,38.56C69.84,37.81 70.98,37.81 71.71,38.56C72.44,39.3 72.44,40.45 71.71,41.19L66.94,46.02ZM62.94,56.92C64.08,56.92 65,56.01 65,54.88C65,53.76 64.08,52.85 62.94,52.85C61.8,52.85 60.88,53.76 60.88,54.88C60.88,56.01 61.8,56.92 62.94,56.92ZM45.06,56.92C46.2,56.92 47.13,56.01 47.13,54.88C47.13,53.76 46.2,52.85 45.06,52.85C43.92,52.85 43,53.76 43,54.88C43,56.01 43.92,56.92 45.06,56.92Z"  
 android:strokeColor="#00000000"  
 android:strokeWidth="1" />  
</vector>

<?xml version="1.0" encoding="utf-8"?>  
<adaptive-icon xmlns:android="http://schemas.android.com/apk/res/android">  
 <background android:drawable="@drawable/ic\_launcher\_background" />  
 <foreground android:drawable="@drawable/ic\_launcher\_foreground" />  
</adaptive-icon>

<?xml version="1.0" encoding="utf-8"?>  
<adaptive-icon xmlns:android="http://schemas.android.com/apk/res/android">  
 <background android:drawable="@drawable/ic\_launcher\_background" />  
 <foreground android:drawable="@drawable/ic\_launcher\_foreground" />  
</adaptive-icon>

<?xml version="1.0" encoding="UTF-8"?>  
<module external.linked.project.id=":app" external.linked.project.path="$MODULE\_DIR$" external.root.project.path="$MODULE\_DIR$/.." external.system.id="GRADLE" type="JAVA\_MODULE" version="4">  
 <component name="FacetManager">  
 <facet type="android-gradle" name="Android-Gradle">  
 <configuration>  
 <option name="GRADLE\_PROJECT\_PATH" value=":app" />  
 </configuration>  
 </facet>  
 <facet type="android" name="Android">  
 <configuration>  
 <option name="SELECTED\_BUILD\_VARIANT" value="debug" />  
 <option name="ASSEMBLE\_TASK\_NAME" value="assembleDebug" />  
 <option name="COMPILE\_JAVA\_TASK\_NAME" value="compileDebugSources" />  
 <afterSyncTasks>  
 <task>generateDebugSources</task>  
 </afterSyncTasks>  
 <option name="ALLOW\_USER\_CONFIGURATION" value="false" />  
 <option name="MANIFEST\_FILE\_RELATIVE\_PATH" value="/src/main/AndroidManifest.xml" />  
 <option name="RES\_FOLDER\_RELATIVE\_PATH" value="/src/main/res" />  
 <option name="RES\_FOLDERS\_RELATIVE\_PATH" value="file://$MODULE\_DIR$/src/main/res" />  
 <option name="ASSETS\_FOLDER\_RELATIVE\_PATH" value="/src/main/assets" />  
 </configuration>  
 </facet>  
 </component>  
 <component name="NewModuleRootManager" LANGUAGE\_LEVEL="JDK\_1\_7">  
 <output url="file://$MODULE\_DIR$/build/intermediates/classes/debug" />  
 <output-test url="file://$MODULE\_DIR$/build/intermediates/classes/test/debug" />  
 <exclude-output />  
 <content url="file://$MODULE\_DIR$">  
 <sourceFolder url="file://$MODULE\_DIR$/build/generated/source/apt/debug" isTestSource="false" generated="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/build/generated/source/r/debug" isTestSource="false" generated="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/build/generated/source/aidl/debug" isTestSource="false" generated="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/build/generated/source/buildConfig/debug" isTestSource="false" generated="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/build/generated/source/rs/debug" isTestSource="false" generated="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/build/generated/res/rs/debug" type="java-resource" />  
 <sourceFolder url="file://$MODULE\_DIR$/build/generated/res/resValues/debug" type="java-resource" />  
 <sourceFolder url="file://$MODULE\_DIR$/build/generated/source/apt/androidTest/debug" isTestSource="true" generated="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/build/generated/source/r/androidTest/debug" isTestSource="true" generated="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/build/generated/source/aidl/androidTest/debug" isTestSource="true" generated="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/build/generated/source/buildConfig/androidTest/debug" isTestSource="true" generated="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/build/generated/source/rs/androidTest/debug" isTestSource="true" generated="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/build/generated/res/rs/androidTest/debug" type="java-test-resource" />  
 <sourceFolder url="file://$MODULE\_DIR$/build/generated/res/resValues/androidTest/debug" type="java-test-resource" />  
 <sourceFolder url="file://$MODULE\_DIR$/build/generated/source/apt/test/debug" isTestSource="true" generated="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/debug/res" type="java-resource" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/debug/resources" type="java-resource" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/debug/assets" type="java-resource" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/debug/aidl" isTestSource="false" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/debug/java" isTestSource="false" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/debug/rs" isTestSource="false" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/debug/shaders" isTestSource="false" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/androidTestDebug/res" type="java-test-resource" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/androidTestDebug/resources" type="java-test-resource" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/androidTestDebug/assets" type="java-test-resource" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/androidTestDebug/aidl" isTestSource="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/androidTestDebug/java" isTestSource="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/androidTestDebug/rs" isTestSource="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/androidTestDebug/shaders" isTestSource="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/testDebug/res" type="java-test-resource" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/testDebug/resources" type="java-test-resource" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/testDebug/assets" type="java-test-resource" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/testDebug/aidl" isTestSource="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/testDebug/java" isTestSource="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/testDebug/rs" isTestSource="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/testDebug/shaders" isTestSource="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/main/res" type="java-resource" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/main/resources" type="java-resource" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/main/assets" type="java-resource" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/main/aidl" isTestSource="false" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/main/java" isTestSource="false" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/main/rs" isTestSource="false" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/main/shaders" isTestSource="false" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/test/res" type="java-test-resource" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/test/resources" type="java-test-resource" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/test/assets" type="java-test-resource" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/test/aidl" isTestSource="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/test/java" isTestSource="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/test/rs" isTestSource="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/test/shaders" isTestSource="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/androidTest/res" type="java-test-resource" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/androidTest/resources" type="java-test-resource" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/androidTest/assets" type="java-test-resource" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/androidTest/aidl" isTestSource="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/androidTest/java" isTestSource="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/androidTest/rs" isTestSource="true" />  
 <sourceFolder url="file://$MODULE\_DIR$/src/androidTest/shaders" isTestSource="true" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/assets" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/blame" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/build-info" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/builds" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/check-manifest" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/classes" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/incremental" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/incremental-classes" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/incremental-runtime-classes" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/incremental-verifier" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/instant-run-resources" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/instant-run-support" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/javaPrecompile" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/jniLibs" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/manifests" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/prebuild" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/reload-dex" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/res" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/rs" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/shaders" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/split-apk" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/splits-support" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/symbols" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/tmp" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/intermediates/transforms" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/outputs" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/reports" />  
 <excludeFolder url="file://$MODULE\_DIR$/build/tmp" />  
 </content>  
 <orderEntry type="jdk" jdkName="Android API 26 Platform" jdkType="Android SDK" />  
 <orderEntry type="sourceFolder" forTests="false" />  
 <orderEntry type="library" name="com.android.support:support-media-compat-26.1.0" level="project" />  
 <orderEntry type="library" scope="TEST" name="junit:junit:4.12@jar" level="project" />  
 <orderEntry type="library" name="android.arch.lifecycle:common:1.0.0@jar" level="project" />  
 <orderEntry type="library" scope="TEST" name="com.android.support.test.espresso:espresso-core-3.0.1" level="project" />  
 <orderEntry type="library" name="com.android.support:support-fragment-26.1.0" level="project" />  
 <orderEntry type="library" scope="TEST" name="javax.inject:javax.inject:1@jar" level="project" />  
 <orderEntry type="library" scope="TEST" name="com.squareup:javawriter:2.1.1@jar" level="project" />  
 <orderEntry type="library" name="com.android.support:support-vector-drawable-26.1.0" level="project" />  
 <orderEntry type="library" name="android.arch.lifecycle:runtime-1.0.0" level="project" />  
 <orderEntry type="library" name="com.android.support:appcompat-v7-26.1.0" level="project" />  
 <orderEntry type="library" name="com.android.support:support-annotations:26.1.0@jar" level="project" />  
 <orderEntry type="library" name="com.android.support.constraint:constraint-layout-solver:1.0.2@jar" level="project" />  
 <orderEntry type="library" name="com.android.support:support-core-utils-26.1.0" level="project" />  
 <orderEntry type="library" name="com.android.support.constraint:constraint-layout-1.0.2" level="project" />  
 <orderEntry type="library" name="com.android.support:support-core-ui-26.1.0" level="project" />  
 <orderEntry type="library" scope="TEST" name="com.android.support.test:runner-1.0.1" level="project" />  
 <orderEntry type="library" scope="TEST" name="com.android.support.test:rules-1.0.1" level="project" />  
 <orderEntry type="library" scope="TEST" name="com.google.code.findbugs:jsr305:2.0.1@jar" level="project" />  
 <orderEntry type="library" scope="TEST" name="com.android.support.test.espresso:espresso-idling-resource-3.0.1" level="project" />  
 <orderEntry type="library" scope="TEST" name="org.hamcrest:hamcrest-core:1.3@jar" level="project" />  
 <orderEntry type="library" name="com.android.support:support-compat-26.1.0" level="project" />  
 <orderEntry type="library" name="android.arch.core:common:1.0.0@jar" level="project" />  
 <orderEntry type="library" scope="TEST" name="org.hamcrest:hamcrest-library:1.3@jar" level="project" />  
 <orderEntry type="library" scope="TEST" name="org.hamcrest:hamcrest-integration:1.3@jar" level="project" />  
 <orderEntry type="library" name="com.android.support:support-v4-26.1.0" level="project" />  
 <orderEntry type="library" scope="TEST" name="net.sf.kxml:kxml2:2.3.0@jar" level="project" />  
 <orderEntry type="library" name="com.android.support:animated-vector-drawable-26.1.0" level="project" />  
 </component>  
</module>

apply plugin: 'com.android.application'  
  
android {  
 compileSdkVersion 26  
 defaultConfig {  
 applicationId "com.example.acer\_pc.pokecats"  
 minSdkVersion 22  
 targetSdkVersion 26  
 versionCode 1  
 versionName "1.0"  
 testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"  
 }  
 buildTypes {  
 release {  
 minifyEnabled false  
 proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'  
 }  
 }  
 productFlavors {  
 }  
}  
  
dependencies {  
 implementation fileTree(include: ['\*.jar'], dir: 'libs')  
 implementation 'com.android.support:appcompat-v7:26.1.0'  
 implementation 'com.android.support.constraint:constraint-layout:1.0.2'  
 testImplementation 'junit:junit:4.12'  
 androidTestImplementation 'com.android.support.test:runner:1.0.1'  
 androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'  
}

<?xml version='1.0' encoding='UTF-8'?>  
<metadata>  
 <com.android.support.constraint/>  
 <com.android.databinding/>  
 <com.android.support/>  
 <com.android.support.test/>  
 <com.android.support.test.janktesthelper/>  
 <com.android.support.test.uiautomator/>  
 <com.android.support.test.espresso/>  
 <android.arch.persistence.room/>  
 <android.arch.lifecycle/>  
 <android.arch.core/>  
 <com.google.android.instantapps/>  
 <com.google.android.instantapps.thirdpartycompat/>  
 <com.android.java.tools.build/>  
 <com.android.tools/>  
 <com.android.tools.layoutlib/>  
 <com.android.tools.ddms/>  
 <com.android.tools.external.com-intellij/>  
 <com.android.tools.build/>  
 <com.android.tools.analytics-library/>  
 <com.android.tools.internal.build.test/>  
 <com.android.tools.lint/>  
 <com.android.tools.external.org-jetbrains/>  
 <com.android.support.test.espresso.idling/>  
 <com.android.support.test.services/>  
 <com.google.firebase/>  
 <com.google.android.gms/>  
 <com.google.gms/>  
 <android.arch.paging/>  
 <com.crashlytics.sdk.android/>  
 <io.fabric.sdk.android/>  
 <android.arch.persistence/>  
 <com.google.android.wearable/>  
 <com.google.android.support/>  
 <com.android.installreferrer/>  
 <com.google.ar/>  
</metadata>

<?xml version='1.0' encoding='UTF-8'?>  
<com.android.support>  
 <support-compat versions="24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <leanback-v17 versions="21.0.0,21.0.2,21.0.3,22.0.0,22.1.0,22.1.1,22.2.0,22.2.1,23.0.0,23.0.1,23.1.0,23.1.1,23.2.0,23.2.1,23.3.0,23.4.0,24.0.0-alpha1,24.0.0-alpha2,24.0.0-beta1,24.0.0,24.1.0,24.1.1,24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <recommendation versions="23.0.1,23.1.0,23.1.1,23.2.0,23.2.1,23.3.0,23.4.0,24.0.0-alpha1,24.0.0-alpha2,24.0.0-beta1,24.0.0,24.1.0,24.1.1,24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <support-tv-provider versions="26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <support-vector-drawable versions="23.2.0,23.2.1,23.3.0,23.4.0,24.0.0-alpha1,24.0.0-alpha2,24.0.0-beta1,24.0.0,24.1.0,24.1.1,24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <recyclerview-v7 versions="21.0.0,21.0.2,21.0.3,22.0.0,22.1.0,22.1.1,22.2.0,22.2.1,23.0.0,23.0.1,23.1.0,23.1.1,23.2.0,23.2.1,23.3.0,23.4.0,24.0.0-alpha1,24.0.0-alpha2,24.0.0-beta1,24.0.0,24.1.0,24.1.1,24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <preference-leanback-v17 versions="23.0.0,23.0.1,23.1.0,23.1.1,23.2.0,23.2.1,23.3.0,23.4.0,24.0.0-alpha1,24.0.0-alpha2,24.0.0-beta1,24.0.0,24.1.0,24.1.1,24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <preference-v14 versions="23.0.0,23.0.1,23.1.0,23.1.1,23.2.0,23.2.1,23.3.0,23.4.0,24.0.0-alpha1,24.0.0-alpha2,24.0.0-beta1,24.0.0,24.1.0,24.1.1,24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <percent versions="22.2.0,23.0.0,23.0.1,23.1.0,23.1.1,23.2.0,23.2.1,23.3.0,23.4.0,24.0.0-alpha1,24.0.0-alpha2,24.0.0-beta1,24.0.0,24.1.0,24.1.1,24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <support-media-compat versions="24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <cardview-v7 versions="21.0.0,21.0.2,21.0.3,22.0.0,22.1.0,22.1.1,22.2.0,22.2.1,23.0.0,23.0.1,23.1.0,23.1.1,23.2.0,23.2.1,23.3.0,23.4.0,24.0.0-alpha1,24.0.0-alpha2,24.0.0-beta1,24.0.0,24.1.0,24.1.1,24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <wearable versions="26.0.0-alpha1"/>  
 <exifinterface versions="25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <support-annotations versions="19.1.0,20.0.0,21.0.0,21.0.2,21.0.3,22.0.0,22.1.0,22.1.1,22.2.0,22.2.1,23.0.0,23.0.1,23.1.0,23.1.1,23.2.0,23.2.1,23.3.0,23.4.0,24.0.0-alpha1,24.0.0-alpha2,24.0.0-beta1,24.0.0,24.1.0,24.1.1,24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <appcompat-v7 versions="18.0.0,19.0.0,19.0.1,19.1.0,20.0.0,21.0.0,21.0.2,21.0.3,22.0.0,22.1.0,22.1.1,22.2.0,22.2.1,23.0.0,23.0.1,23.1.0,23.1.1,23.2.0,23.2.1,23.3.0,23.4.0,24.0.0-alpha1,24.0.0-alpha2,24.0.0-beta1,24.0.0,24.1.0,24.1.1,24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <palette-v7 versions="21.0.0,21.0.2,21.0.3,22.0.0,22.1.0,22.1.1,22.2.0,22.2.1,23.0.0,23.0.1,23.1.0,23.1.1,23.2.0,23.2.1,23.3.0,23.4.0,24.0.0-alpha1,24.0.0-alpha2,24.0.0-beta1,24.0.0,24.1.0,24.1.1,24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <multidex-instrumentation versions="1.0.0,1.0.1,1.0.2"/>  
 <multidex versions="1.0.0,1.0.1,1.0.2"/>  
 <mediarouter-v7 versions="18.0.0,19.0.0,19.0.1,19.1.0,20.0.0,21.0.0,21.0.2,21.0.3,22.0.0,22.1.0,22.1.1,22.2.0,22.2.1,23.0.0,23.0.1,23.1.0,23.1.1,23.2.0,23.2.1,23.3.0,23.4.0,24.0.0-alpha1,24.0.0-alpha2,24.0.0-beta1,24.0.0,24.1.0,24.1.1,24.2.0,24.2.1,25.0.0,25.0.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <preference-v7 versions="23.0.0,23.0.1,23.1.0,23.1.1,23.2.0,23.2.1,23.3.0,23.4.0,24.0.0-alpha1,24.0.0-alpha2,24.0.0-beta1,24.0.0,24.1.0,24.1.1,24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <support-dynamic-animation versions="25.3.0,25.3.1,25.4.0,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <support-fragment versions="24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <design versions="22.2.0,22.2.1,23.0.0,23.0.1,23.1.0,23.1.1,23.2.0,23.2.1,23.3.0,23.4.0,24.0.0-alpha1,24.0.0-alpha2,24.0.0-beta1,24.0.0,24.1.0,24.1.1,24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <transition versions="24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <customtabs versions="23.0.0,23.0.1,23.1.0,23.1.1,23.2.0,23.2.1,23.3.0,23.4.0,24.0.0-alpha1,24.0.0-alpha2,24.0.0-beta1,24.0.0,24.1.0,24.1.1,24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <support-core-ui versions="24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <gridlayout-v7 versions="13.0.0,18.0.0,19.0.0,19.0.1,19.1.0,20.0.0,21.0.0,21.0.2,21.0.3,22.0.0,22.1.0,22.1.1,22.2.0,22.2.1,23.0.0,23.0.1,23.1.0,23.1.1,23.2.0,23.2.1,23.3.0,23.4.0,24.0.0-alpha1,24.0.0-alpha2,24.0.0-beta1,24.0.0,24.1.0,24.1.1,24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <animated-vector-drawable versions="23.2.0,23.2.1,23.3.0,23.4.0,24.0.0-alpha1,24.0.0-alpha2,24.0.0-beta1,24.0.0,24.1.0,24.1.1,24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <support-core-utils versions="24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <support-v13 versions="13.0.0,18.0.0,19.0.0,19.0.1,19.1.0,20.0.0,21.0.0,21.0.2,21.0.3,22.0.0,22.1.0,22.1.1,22.2.0,22.2.1,23.0.0,23.0.1,23.1.0,23.1.1,23.2.0,23.2.1,23.3.0,23.4.0,24.0.0-alpha1,24.0.0-alpha2,24.0.0-beta1,24.0.0,24.1.0,24.1.1,24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <instantvideo versions="26.0.0-alpha1"/>  
 <support-v4 versions="13.0.0,18.0.0,19.0.0,19.0.1,19.1.0,20.0.0,21.0.0,21.0.2,21.0.3,22.0.0,22.1.0,22.1.1,22.2.0,22.2.1,23.0.0,23.0.1,23.1.0,23.1.1,23.2.0,23.2.1,23.3.0,23.4.0,24.0.0-alpha1,24.0.0-alpha2,24.0.0-beta1,24.0.0,24.1.0,24.1.1,24.2.0,24.2.1,25.0.0,25.0.1,25.1.0,25.1.1,25.2.0,25.3.0,25.3.1,25.4.0,26.0.0-alpha1,26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <support-emoji versions="26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <wear versions="26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <support-emoji-appcompat versions="26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <support-emoji-bundled versions="26.0.0-beta1,26.0.0-beta2,26.0.0,26.0.1,26.0.2,26.1.0,27.0.0,27.0.1,27.0.2"/>  
 <support-content versions="27.0.0,27.0.1,27.0.2"/>  
</com.android.support>