

# 移动平台开发作品展示

---

周芳 张文杰 谢殊凡

# One touch Drawing



Draw everything  
with only "One touch"



# 作品简介

# 一、作品简介

- 应用名称：一笔画
- 功能：随机生成一笔画图案。玩家可以在屏幕上描绘图案，程序会判定绘制成功或者失败。
- 设计目的：制作一个休闲益智类的小游戏



# 设计思路

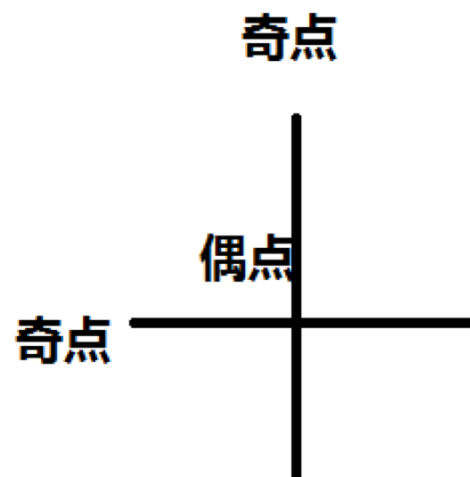
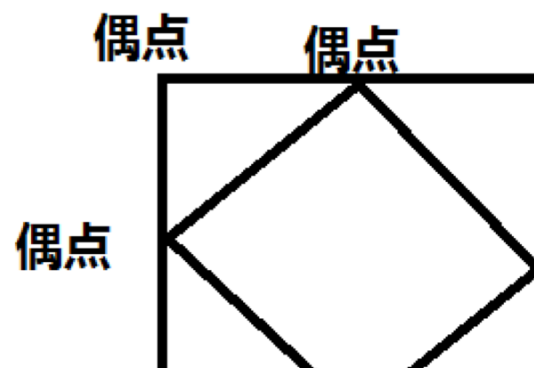
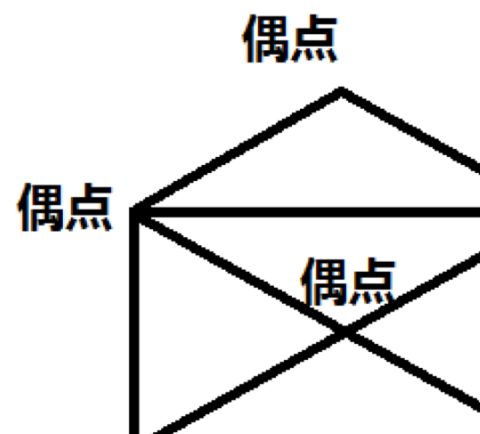
## 二、设计思路



### ➤ 1.自动生成随机的一笔画

奇顶点：指数为奇数的顶点。

偶顶点：指数为偶数的顶点。

一笔画图形的必要条件是：奇顶点数目是0或者2。



- 
- 利用算法生成图案，图案由一些点和这些点之间的连线构成，生成图案需有0或2个奇点，以保证能够一笔画出。
  - 将这些点和连线记录下来，从触摸屏幕到松开屏幕的过程中，如果图案中的所有连线都出现且仅出现过一次，则判定为过关，否则失败。
  - 利用Canvas类和Paint类将图形显示到屏幕上。
- 

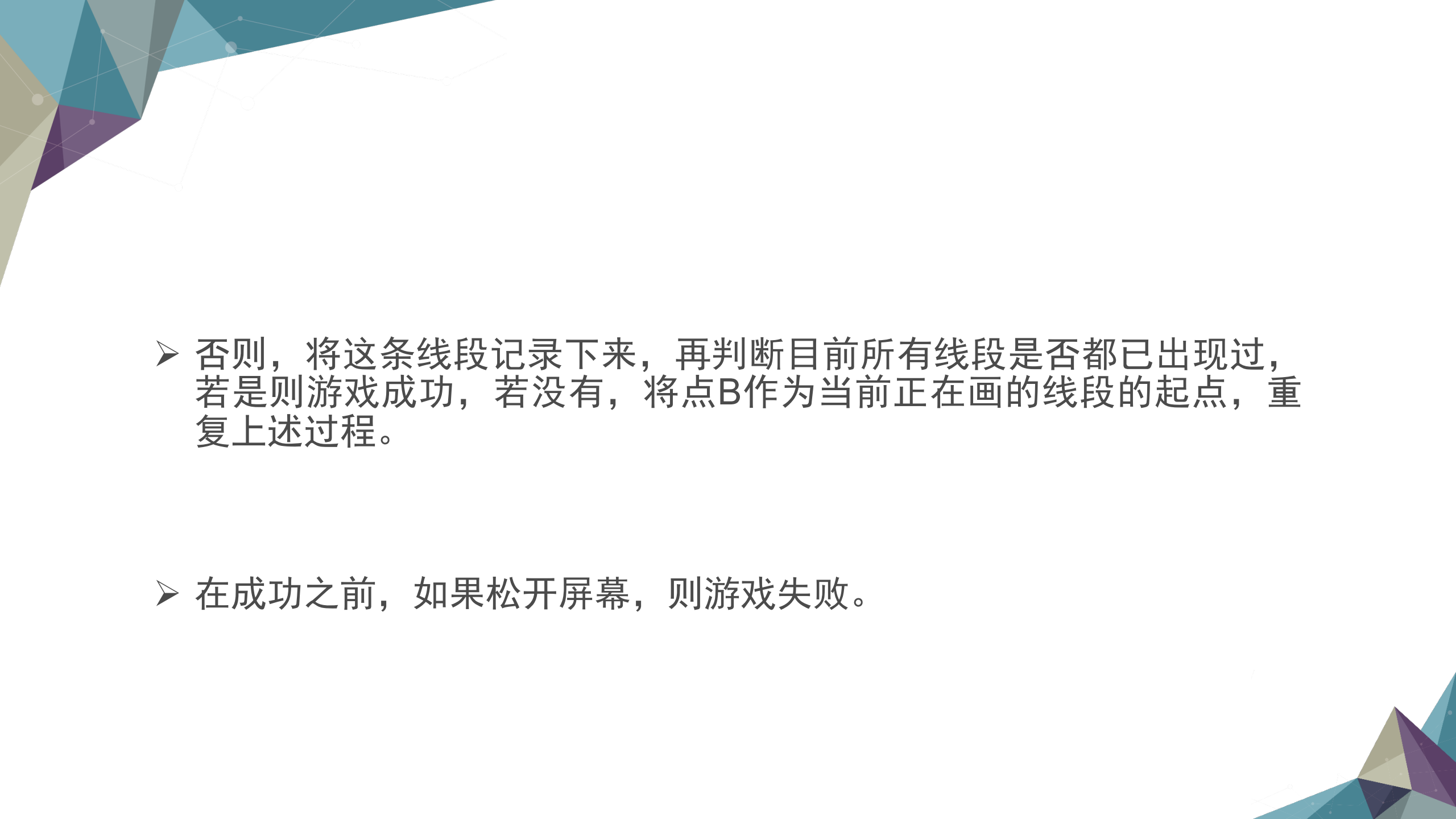


## 二、设计思路

### ➤ 2.跟踪玩家绘制过程并判断正误

当触点位于一点A附近一定大小的区域时，认为玩家正在触碰该点，该点作为当前正在画的线段的起点；

接下来，当玩家触碰下一点B时，认为玩家已画完一条线段，此时判断这条线段是否：（1）正确；（2）未出现过；如果有一条不满足，就认为游戏失败。

- 
- 否则，将这条线段记录下来，再判断目前所有线段是否都已出现过，若是则游戏成功，若没有，将点B作为当前正在画的线段的起点，重复上述过程。
  - 在成功之前，如果松开屏幕，则游戏失败。




# 详细设计

# 三、详细设计

## ➤ 1.自动生成一笔画

(1) 随机生成点，将点的编号和坐标记录在数组中（新生成的点和之前的点间距离不能过近）；

(2) 最初每个点都需连入两条线段( $n=2$ )：每个点与它的下一点和相隔一位的点相连。此时图案中无奇点，可一笔画出。也可在此基础上任选两点再连一条线段，此时图案中有两奇点，也可一笔画出。



(3) 根据刚才记录的各点的坐标和点与点间的连接情况，将图案显示到屏幕上。

方法：利用Canvas类和Paint类。



# 三、详细设计

## ➤ 2.跟踪玩家绘制过程并判断正误

设置触点监听事件：

- (1) 接触屏幕时，游戏开始，对变量进行初始化；
- (2) 接触并移动过程中，动态地判断是否连入新的点；如果连入新的点，则认为连了一条新线段；判断当前线段是否重复或者错误，如果是则游戏结束；否则将这条线段记录下来，并判断是否成功，若成功则游戏结束，否则继续；
- (3) 松开屏幕时，判断是否成功，若不是则失败。

```
public boolean  
onTouchEvent (MotionEvent event) {  
  
    //获得触摸的坐标  
    float x = event.getX();  
    float y = event.getY();  
    switch (event.getAction())  
    {  
        //触摸屏幕时刻  
        case MotionEvent.ACTION_DOWN:  
            ...  
            break;
```

```
        //触摸并移动时刻  
        case MotionEvent.ACTION_MOVE:  
            ...  
            break;  
        //终止触摸时刻  
        case MotionEvent.ACTION_UP:  
            ...  
            break;  
    }  
    return true;  
}
```



# 代码分析



# 页面设计-main activity

```
Button mode = (Button)findViewById(R.id.button1);
mode.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent =new Intent();

intent.setClass(MainActivity.this,SubActivity1.class);
        startActivity(intent);
    }
});
```



# 页面设计-subactivity

<Button

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="07"  
    android:id="@+id/game7"  
    android:layout_marginTop="46dp"  
    android:textSize="40dp"  
    android:layout_below="@+id/game4"  
    android:layout_alignLeft="@+id/game4"  
    android:layout_alignStart="@+id/game4" />
```



# 内容实现-subactivity1.java

```
Button but1 = (Button)findViewById(R.id.game1);
Button but2 = (Button)findViewById(R.id.game2);
Button but3 = (Button)findViewById(R.id.game3);
Button but4 = (Button)findViewById(R.id.game4);
Button but5 = (Button)findViewById(R.id.game5);
Button but6 = (Button)findViewById(R.id.game6);
Button but7 = (Button)findViewById(R.id.game7);
Button but8 = (Button)findViewById(R.id.game8);
Button but9 = (Button)findViewById(R.id.game9);

View.OnClickListener clickListener = new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Bundle bundle = new Bundle();
        if (view.getId()==R.id.game1)
        {
            bundle.putString("level","1");
        }
    }
}
```

# 内容实现-subactivity1.java

```
if (view.getId()==R.id.game2)
{
    bundle.putString("level","2");
}
if (view.getId()==R.id.game3)
{
    bundle.putString("level","3");
}
.....
Intent intent = new Intent();
intent.putExtras(bundle);
intent.setClass(SubActivity1.this,EasyLevel.class);
startActivity(intent);
```

## 内容实现-easylevel.java

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_easy_level);  
  
    .....  
  
    MyView my = new MyView(EasyLevel.this);  
    setContentView(my);  
  
}
```

# 内容实现-myview.java – 变量定义

```
private int n;           /点的数量

public int []locx=new int[30];    /记录图形中的点的横坐标
public int []locy=new int[30];    /
public int [][]four = new int[3][5]; /记录预先生成的四个点的横坐标
public int [][]six = new int[3][7]; /
public int [][]eight=new int[3][9]; /
public int [][]line = new int[30][30]; /记录电与点之间的连线的情况
public int lastdot=0;           /记录上一个被匹配中的点的序号
public int [][]paintline=new int[30][30];    /记录用户已经划过哪些线段
public boolean success=true;
DisplayMetrics dm2;
```

# 内容实现-myview.java – init

```
for (i=0;i<20;i++)
{
    locx[i]=0;
    locy[i]=0;
    for(j=0;j<20;j++)
    {
        line[i][j]=0;
    }
}

int first=0,second=0;
for (first=0;first<30;first++)
    for (second=0;second<30;second++)
        paintline[first][second]=0;

int height=(int)(dm2.heightPixels*0.8);
int width=(int)(dm2.widthPixels*0.8);
Log.i("height",""+height);
Log.i("width",""+width);
```

# 内容实现-myview.java – onDraw

```
if (n==4){  
    int a=0;  
    for (a=1;a<5;a++)  
    {  
        locx[a]=four[1][a];  
        locy[a]=four[2][a];  
        p.setStyle(Paint.Style.STROKE);  
        canvas.drawCircle(locx[a], locy[a], 20, p);// 小圆  
        p.setAntiAlias(true);// 设置画笔的锯齿效果。 true是去除，大  
        家一看效果就明白了  
    }  
}
```



# 内容实现-myview.java – onDraw

```
for(int index=n; index>0; index--) {  
    //从0到index处之间随机取一个值，跟index处的元素交换  
    int nextone=(int)(Math.random()*(n+n))%(index+1)+1;  
    int nx=locx[nextone];  
    int ny=locy[nextone];  
    locx[nextone]=locx[index];  
    locy[nextone]=locy[index];  
    locx[index]=nx;  
    locy[index]=ny;  
}
```

# 内容实现-myview.java – onDraw

```
for (i=1;i<=n;i++)
{
    int dot = (i+1)%n;
    if (dot==0)
        dot=1;
    line[i][dot]=1;
    line[dot][i]=1;

    Paint paint = new Paint();
    paint.setColor(Color.RED);
    canvas.drawLine(locx[i], locy[i], locx[dot], locy[dot], paint);// 画线
}
```

# 内容实现-myview.java – ontouchevent

```
float x = event.getX();
```

```
float y = event.getY();
```

```
int action = event.getAction();
```

# 内容实现-myview.java – ontouchevent

```
case MotionEvent.ACTION_MOVE://移动
for (i=1;i<=n;i++)
{
    if ((x-locx[i]<20&&x-locx[i]>-20)&&(y-locy[i]<20&&y-locy[i]>-20)){
        Log.i("drawdot",""+locx[i]+" "+locy[i]);
        Log.i("current",""+i+" "+lastdot);
        Log.i("line",""+line[i][lastdot] +" "+" +paintline[i][lastdot]);
        if (lastdot>0&&line[i][lastdot]==1&&paintline[i][lastdot]==0){
            Log.i("drawlime",""+locx[lastdot]+" "+locy[lastdot]);
            Log.i("drawline",""+locx[i]+" "+locy[i]);
            paintline[i][lastdot]=1;
            paintline[lastdot][i]=1;
        }
    }
}
```

# 内容实现-myview.java – ontouchevent

```
else if (lastdot>0&&line[i][lastdot]==1&&paintline[i][lastdot]==1){
    success=false;
}
else if (lastdot>0&&line[i][lastdot]==0&&lastdot!=i){
    success=false;
}
lastdot=i;
}
}
Log.i("touch", "ACTION_MOVE");
break;
}
```

# 内容实现-myview.java – ontouchevent

```
case MotionEvent.ACTION_UP://松开
```

```
{
```

```
int i=0,j=0;
```

```
for (i=0;i<30;i++)
```

```
    for (j=0;j<30;j++)
```

```
        if (line[i][j]!=paintline[i][j])
```

```
            success=false;
```

```
if (!success)
```

```
    Log.i("final", "fail");
```

```
else
```

```
    Log.i("final", "success");
```

```
Log.i("touch", "ACTION_UP");
```

```
break; }
```