

Android 传感设计——躲避小球（产品说明书）

项目介绍

这是一款基于 java 开发的移动端安卓传感器小游戏——躲避小球

项目原理

在一个容器内按时间的推移来增加小球（难度），系统生成的游戏会按照初始生成时的路径一直移动，并且不会消失，主要通过多加速度传感器来操控玩家小球去躲避系统生成的小球，两种小球碰撞之后游戏结束

项目设计

1. 我们需要一个容器，在里面放入我们玩家操控的小球
2. 在容器中生成系统小球，并随时间推移增加数量
3. 当玩家的小球碰到系统生成的小球时游戏结束

项目实施

代码实现

首先创建视图；然后获取传感器；接着创建玩家控制的小球与系统生成的小球；将玩家小球的移动与传感器的数据进行绑定；进行系统生成小球的移动控制（随机移动）；进行系统小球与玩家小球的碰撞判断（即游戏结束判断，失败则记录时间并回到重新开始的界面）

游戏视图的部分代码

```
public class Game extends ImageView {
    private static final String TAG = Game.class.getSimpleName();
    public static final int SPEED_DISTANCE = 8;
    public static final int DELAY_MILLIS = 30;
    public static final int DISTANCE = 20;
    private Paint mPaint;
    public int mWidth;
    public int mHeight;
    public int mNumber = 1;
    public int[] mPositionX;
    public int[] mPositionY;
    public int[] mDegree;
    public List<PointPosition> mPointPositions;
    public boolean isPlaying;
    public int mPlayerX;
    public int mPlayerY;
    public boolean failure = false;
    public long mTime;
    public long mSystemTime;
```

小球通过传感器控制移动后的位置判断的部分代码

```
public class PointPosition {
    public int mNumber;
    public int[] mPositionX;
    public int[] mPositionY;
    public int[] mDegree;

    public PointPosition(int number, int[] positionX, int[] positionY, int[] degree) {
        mNumber = number;
        mPositionX = positionX;
        mPositionY = positionY;
        mDegree = degree;
    }

    public int getPositionX(int number) {
        return mPositionX[number];
    }

    public void setPositionX(int number, int valueChanged) {
        mPositionX[number] += valueChanged;
    }

    public int getPositionY(int number) {
        return mPositionY[number];
    }

    public void setPositionY(int number, int valueChanged) {
        mPositionY[number] += valueChanged;
    }

    public int getDegree(int number) {
        return mDegree[number];
    }

    public void setDegree(int number, int valueChanged) {
        mDegree[number] += valueChanged;
    }
}
```

游戏结束的判断的部分代码

```
public void onSensorChanged(SensorEvent event) {
    if (mGame != null && !mGame.failure) {
        x -= event.values[0] / 15;
        y += event.values[1] / 15;
        if (mGame.mPlayerX + x >= mGame.mWidth) {
            mGame.mPlayerX = mGame.mWidth;
        } else if (mGame.mPlayerX + x <= 0) {
            mGame.mPlayerX = 0;
        } else {
            mGame.mPlayerX += x;
        }
        if (mGame.mPlayerY + y >= mGame.mHeight) {
            mGame.mPlayerY = mGame.mHeight;
        } else if (mGame.mPlayerY + y <= 0) {
            mGame.mPlayerY = 0;
        } else {
            mGame.mPlayerY += y;
        }
        String s = String.valueOf(mGame.mTime / 1000);
        mTextView.setText(s);
    } else if (mGame != null && mGame.isPlaying) {
        Toast.makeText(MainActivity.this, "你失败了,共坚持了" + mGame.mTime / 1000 + "秒", Toast.LENGTH_SHORT).show();
        if (mGame != null) {
            mGame.isPlaying = false;
            mGame.fresh();
        }
    }
}
```

游戏封面的设计

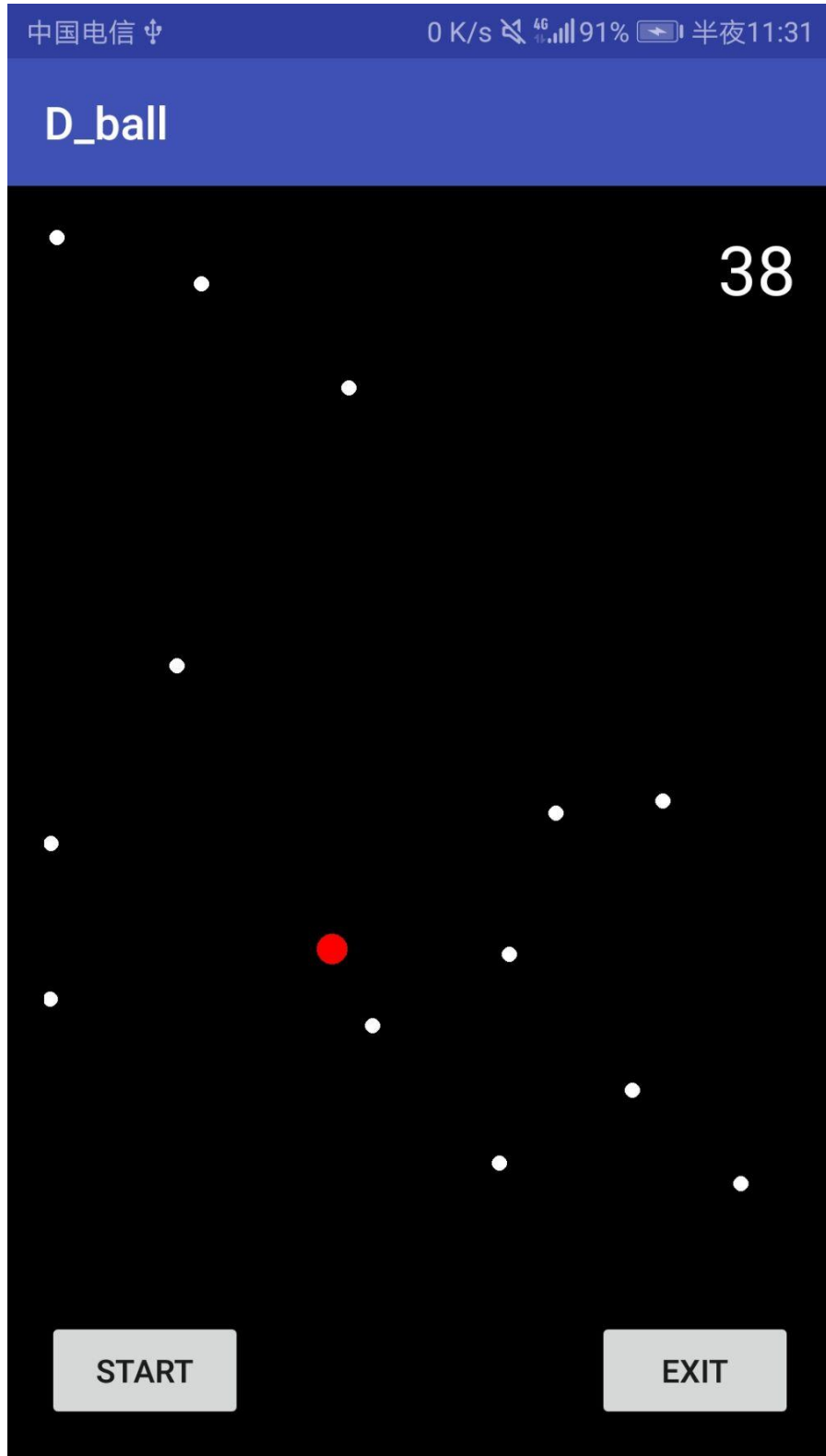
添加一张背景图片，写入游戏名字（躲避小球），再设置几个按钮，然后链接开始，通过开始按钮跳转到游戏主体里面。

游戏主体中点击 start 开始游戏，点击 exit 退出游戏

游戏逻辑代码主要分为 main，game，PointPosition 三部分，详情看详细代码

项目效果展示

通过躲避白色小球来进行时间的累积



D_ball

躲避小球

是男人就坚持100秒

开始

排行

设置

关于