

Develop the UI

The game view, in general, consists of two parts:

The main game interface.

The Start Button and Data display area.

The application use RelativeLayout as the whole layout.

And below is a LinearLayout.



Problem 1:

How to draw the pictures based on the game status.

Solution:

Develop a class "GameView" extends "View". The function of this GameView is to draw all pictures based on the game status.

In order to develop the "GameView", I created a class called "Piece". The object of "Piece" represents a block on the screen. The object encapsulates the picture on the block. Besides, it stands for which element the picture in the two-dimensional array is. Also, the object of Piece also encapsulates the X, Y coordinates of the picture's top left corner.

The X Y coordinates decide the picture's position on the screen. The class "GameView" draws all pictures based on the XY coordinates.

Problem 2:

How to deal with the connecting line between two pictures.

Solution:

Use a simple utility class-"LinkInfo" to encapsulate the link information between two blocks. Actually, it is a list, including the points which the line passed by.

Analysis:

The rule is no more than three lines, equal to no more than two points of inflection.

The situation can be divided into four cases:

1. Two blocks located at the same horizontal line. They can link directly.
2. Two blocks located at the same vertical line. They can link directly.
3. Two blocks linked by two lines, which is to say, they have one point of inflection.
4. Two blocks linked by three lines, which is to say, they have two points of inflection.

Let's just assume the piece has value form likePiece[X][Y]. And the function isBlock is Boolean type and it return if there are obstacles between the two pictures

Case 1: If two pieces had the same Y, then call function isBlock, if no obstacles, then eliminate

the two pieces.

Case 2: If two pieces had the same X, then call function isBlock, if no obstacles, then eliminate the two pieces.

Case 3: If there was one point of inflection, then the program need to find the point first. Then call function isBlock, if no obstacles, then eliminate the two pieces.

Case 4: This is the most complex step. The program needs to build a map which contains all the possible link lines between the two pictures, then select the shortest path line.

```
/**
 * Get the shortest connection information between p1 and p2
 *
 * @param p1
 * @param p2
 * @param turns map contains points of inflection
 * @param shortDistance
 * @return the shortest line information between p1 and p2
 */
private LinkInfo getShortcut(Point p1, Point p2, Map<Point, Point> turns,
    int shortDistance)
{
    List<LinkInfo> infos = new ArrayList<LinkInfo>();
    // Ergodic Map,
    for (Point point1 : turns.keySet())
    {
        Point point2 = turns.get(point1);
        // Package turning points and option points. Set them on the list
        infos.add(new LinkInfo(p1, point1, point2, p2));
    }
    return getShortcut(infos, shortDistance);
}

/**
 * Get the shortest linkinfo object from the list
 *
 * @param infos
 * @return the shortest linkinfo object information
 */
private LinkInfo getShortcut(List<LinkInfo> infos, int shortDistance)
{
    int temp1 = 0;
    LinkInfo result = null;
```

```

for (int i = 0; i < infos.size(); i++)
{
    LinkInfo info = infos.get(i);
    // Calculate the total distance of several points
    int distance = countAll(info.getLinkPoints());
    // Use temp1 to save the gap for the first cycle
    if (i == 0)
    {
        temp1 = distance - shortDistance;
        result = info;
    }
    // chose the smallest temp1
    if (distance - shortDistance < temp1)
    {
        temp1 = distance - shortDistance;
        result = info;
    }
}
return result;
}

/**
 * Calculate List < Point > all points total distance
 *
 * @param points the connection points need to be calculated
 * @return The total distance
 */
private int countAll(List<Point> points)
{
    int result = 0;
    for (int i = 0; i < points.size() - 1; i++)
    {
        // Get the I point
        Point point1 = points.get(i);
        // Get the i+1 point
        Point point2 = points.get(i + 1);
        // Calculate the distance between point1 and point2, then add them to result
        result += getDistance(point1, point2);
    }
    return result;
}

/**
 * Get the shortest distance between two points

```

```

*
* @param p1 The first point
* @param p2 The second point
* @return the sum of the distance between two points
*/
private int getDistance(Point p1, Point p2)
{
    int xDistance = Math.abs(p1.x - p2.x);
    int yDistance = Math.abs(p1.y - p2.y);
    return xDistance + yDistance;
}

```

Problem 3: The background music.

Solution: At first, the music can't be played while the whole application can be run normally. After debug, I found the reason is the size of the music is beyond the capacity. So I changed the size of the music, then problem solved.

Reusable classes: MusicServer

1. Add the following sentence in AndroidManifest.xml 's <application> label.

```

<service android:name=".MusicServer">
<intent-filter>
<action android:name="com.angel.Android.MUSIC"/>
<category android:name="android.intent.category.default" />
</intent-filter>
</service>

```

2. Build class--MusicServer.java

```

import android.app.Service;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.IBinder;

public class MusicServer extends Service {

private MediaPlayer mediaPlayer;

@Override
public IBinder onBind(Intent intent) {
// TODO Auto-generated method stub
return null;
}

```

```
@Override
public void onStart(Intent intent,int startId){
super.onStart(intent, startId);

if(mediaPlayer==null){

// R.raw.mmp is a source file, mp3 type
mediaPlayer = MediaPlayer.create(this, R.raw.abc);
mediaPlayer.setLooping(true);
mediaPlayer.start();

}
}
```

```
@Override
public void onDestroy() {
// TODO Auto-generated method stub
super.onDestroy();
mediaPlayer.stop();
}
}
```

3. Put the music under folder “raw”, name it “abc”

4. Add following code in Activity

```
private Intent intent = new Intent("com.angel.Android.MUSIC");
```

Add startService(intent) at onCreate method.