

Final Report and Summary
Project: Mini Spaceship
Name: Jianhua Wang

Final Report:

1. What I have done:

I implement an android base game call mini-spaceship. The whole idea of this game is to control your spaceship and avoid to collision with the other item as long as possible. The core feature of this game is the player no longer using any simulator controller, game pad or key pad anymore, and instead, player just need to draw the trajectory on the screen, and the spaceship will move to its own direction according to the curve you draw. The game has different levels, and each level has different number of items, the higher level the player select and the game will be more challenge to be played.

2. Technology:

The whole app is built on Android system environment using J2ME. This game's playing rule is use trajectory to control the plane and avoid to collision the item that generate on the screen. The game should cover different level by generate different numbers of items. As for the technology, the app should cover the path generate, path analysis, spaceship animation, and also collision detect. As for the item it should cover random algorithm and moving animation. As for the record, should use Timer task to record the time and change it dynamically. As for the Menu changing, should cover intern initiation. As for the whole game, the app should cover multi threads.

3. Detail Implementation

The Implementation should be divided into three core components, which are Layout, Animation, and Function Logic.

3.1 Layout

As for the Layout, the game has four layouts, covering Start Menu, Setting Screen, Gaming Screen, and Ending Screen. In Start Menu, I use four buttons there, however, I do not use the traditional onClick function, and instead, I use the collision between the space ship and button to active the button. I did this because I want the player to get familiar with the control of the space ship, and also showing our game core rule is collision and not clicking. In the Setting Menu, player is free to change the color of the trajectory to fits everyone's habit.

3.2 Animation

As for the Animation part, I need to implement the animation for both space ship and the item. I have two choices on implementing the animation. Use View Animation to do that or use Property Animation. After researching, I prefer and select the Property Animation. The View Animation has a “bug”, when it is animating, the real object is not actually move, and it always stays in the starting point (X & Y value). So the app can’t do collision detect due to the real object is always in the start point. Hence, I choose to use the Object Animator in Property Animation.

3.3 Function Logic

As for the function logic, cores are including path generating, path analysis, and collision detect. For the path generating, I create it by extends the View Class. I add onTouch events, move events and up events so that that application can know the user’s activity and hence can retrieve the coordinate that they interactive with the screen. After I get a group of points, I link these points using quadratic Bezier, and at last drawing it in a timer task. For the path analysis, I use Path Measure to separate the path into points, and use the associate delta of sequence point, to move the space ship. For the collision detect, I basically check the nearby objects and see if there distance less than their radius.

4. Problem I have met

4.1 View Animation is actually not moving

1) Problem: The View Animation has a “bug”, when it is animating, the real object is not actually move, and it always stays in the starting point (X & Y value).

Solution: In order to make the animation actually move instead of a fake vision, one have to add a function to modify its’ actually value by either add an Animation Listener or use Timer Task to modify it every time sequence.

Cleverer solution: Use Property Animation instead, however, this API is available after android version 11.

2) Problem: Create items every time sequence instead of pre-arrange it

Solution: One can use Timer Task to create a task to automatically random generate the items, but one has to be careful, when you are using a timer task, be sure to clear it if you use a Intent, otherwise it will not stop.

5. Reusable component

1. Draw trajectory

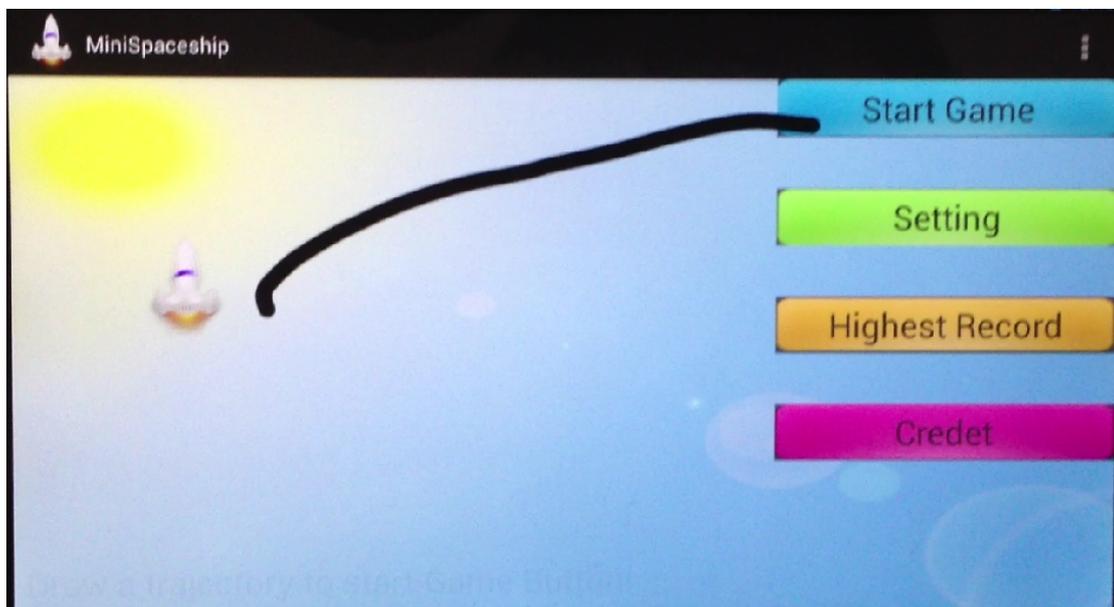
I have implemented a class just only for a user to draw a trajectory. The class is Trace.java in my source code. One can use it simply just introduce this code, call the class and add this class in a layout.

2. Randomly generate items

I have implemented a class just for randomly generate items from the layout's top, bottom, left, and right. This class is Generate.java in Source code. One can use it to generate it randomly without passing it any parameters, however, if you want to change the image resources, one have to do it by changing the resource id in the code.

6. Summary

Screenshot:



This application is a game application which use trajectory to control the character in the game instead of using simulate game pad, simulate controller or keyboard. Control you space ship to avoid the collision with other items. You're your trajectory to start control your space ship.