School of Science

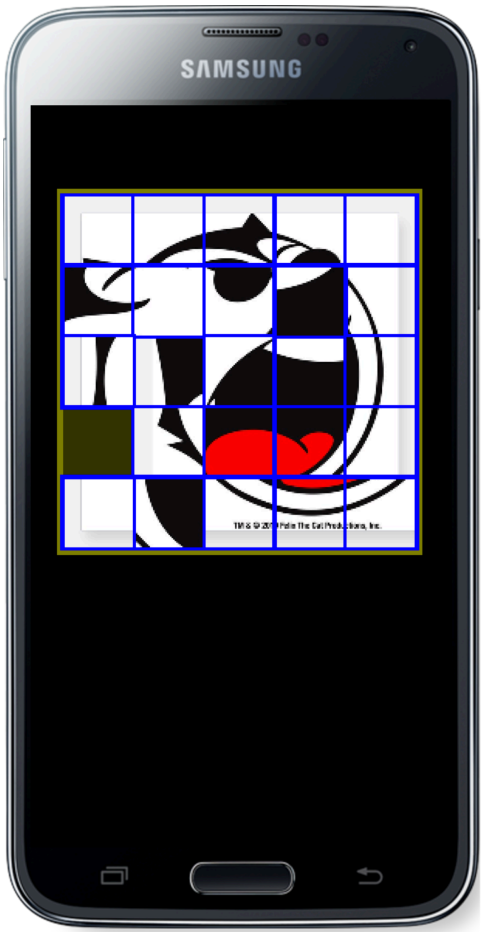
CSP2108 Introduction to Mobile Application Development

Workshop: Interaction (using Events and Listeners)

In this workshop you will complete a time-wasting puzzle app, based on a once-popular children's toy called a slider puzzle. As for last week, most of the code is provided, but this time the code that sets up events and listeners is missing. Your job is to add that in.

Instructions

Download Slider.zip from Blackboard and unzip it to a suitable location. This is the Corona project. Try it out. It looks like this:



There is an image there, but it has been broken up into a 5x5 grid of "tiles", the bottom-right tile has been removed, and then the pieces have been jumbled up. Notice that there is a blank space because of the tile that was removed (it's now in the first column of the 4th row). The player tries to restore the original picture by moving tiles around. Tiles can be moved by "sliding" one of the tiles next to the hold into the hole. As a short cut, several tiles can be slid together – e.g. imaging putting your finger on the top-left tile and sliding 3 tiles down one position, leaving the hole in the top-left.

The provided code has the right logic for making these moves, but no code has been added to allow the user of the app to make use of it.

Look at the code in sliderScene.lua. You can see that it creates and uses an array, *tiles*, of ImageRect objects.

doMovesForTile(tile, sound) in sliderScene.lua will carry out the right actions if passed one of the tiles, and a Boolean, sound, that determines whether to play a sound when the tiles are moved (set it to true when doing user actions).

Task 1: get the tiles moving

What you need to do here is to create a function listener (which will call doMovesForTile), and attach it to each tile.

Step 1: add a function listener with the header

```
local function tapTile( event )
```

This should be a short function – only one line is needed.

Step 2:

Add this function listener to each tile, and have it listen for "tap" events. You can do this in makeTile().

Hints:

1. The lecture slide "General Method 1" shows how to add a function listener.
2. If you do that correctly, then when the function is called, event.target will be the tile that was tapped.

Task 2: Add a "New Game" button.

There's a big empty space at the bottom of the display. That's where the button is meant to go. Add in a button, and attach a function listener that calls the method jumbleTiles().

Hint: The lecture slide "Widget example : simple button" shows how you could do this. You could also review last week's workshop code.

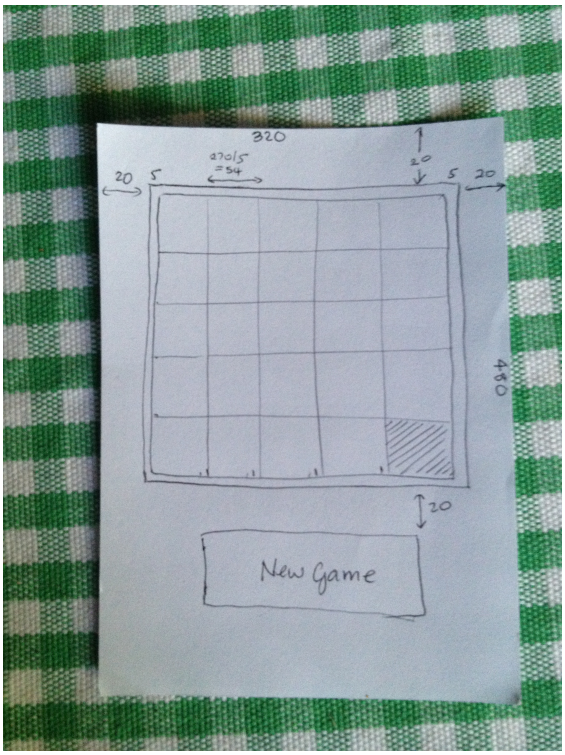
Task 3: Make the code nicer

At the moment the code contains a lot of hard coded locations and calculations for positioning things on the screen. Things like this, for example:

```
tiles[i].x = 25 + (tiles[i].column-1)*54
tiles[i].y = 25 + (tiles[i].row-1)*54
```

This is actually not very nice, and it would be better to replace all the hard-coded parts with calculations based on display.contentWidth and display.contentHeight, for example.

Here is the design sketch that was used:



If you do this well, the code will be much more maintainable and flexible. For example, you could change the grid size by changing the variable slider.SIZE in slider.lua, and the rest of the code would work correctly (try changing this to 3 and see what happens at the moment).