CS260 Intro to Java & Android

Fall 2011

Creating Menus in xml

- Last class we created a menu using Java code
- Menus can also be created using xml and Java
- Grab the Game project from CS260-01 Public

Create menu.xml

- In the res folder, create a new folder called menu
- Add a string called Game Settings with a name game_settings
- Add an item to the menu
  - Id is @+id/game_settings
  - Title is @string/game_settings

Menu Inflater Code

```
public boolean onCreateOptionsMenu (Menu menu)
 {
   super.onCreateOptionsMenu (menu);
   MenuInflater inflater = getMenuInflater ();
   inflater.inflate (R.menu.menu, menu);
   return true;
 }
```

Processing Menu Code

```java
public boolean onOptionsItemSelected (MenuItem item)
  {
    int itemID = item.getItemId ();

    if (0 == itemID) // 0 is Game Setting selection
    {
      Intent intent = new Intent (this, GameSettngs.class);
      startActivity (intent);
      return true;
    }
    else if (1 == itemID) // Additional menu options if necessary
    {
      return true;
    }
    return true;
  }
```

Difficulty Level

- Most games have a difficulty level
- Add strings with Name and Value
  - difficulty_easy is Easy
  - difficulty_medium is Medium
  - difficulty_hard is Hard

xml Arrays

- Add an array of strings to strings.xml

```
<resources>
    <string name="hello">Game Title</string>
    <string name="app_name">Game</string>
    <string name="game_settings">Game Settings</string>
    <string-array name="game_level">
        <item>Easy</item>
        <item>Medium</item>
        <item>Hard</item>
    </string-array>
</resources>
```

Create Ask Difficulty Dialog

```
private void newGameDialog ()
  {
    new AlertDialog.Builder (this)
    .setTitle (R.string.game_settings)
    .setItems (R.array.game_difficulty, new
DialogInterface.OnClickListener()
    {
      public void onClick (DialogInterface dialog, int
difficultyLevel)
      {
        startGame (difficultyLevel);
      }
    }).show ();
  }
```

Start the game

- Write a private method startGame that accepts the difficulty level. The method is to use the Log.d method to display the difficulty level in the LogCat window based on the difficulty level the user selected.
- Hook up the button to start the game when the user presses New Game button.

Graphics

- We will explore still graphics and then graphic animation
- Android provides libraries to perform 2D and 3D graphics
- android.graphics provides low-level tools such as
  - canvas
  - color filters
  - points
  - rectangles

Android colors

- Represented with four 8-bit numbers (ARGB)
  - alpha - measures transparency where 0 is completely transparent and 255 is completely opaque
  - red
  - green
  - blue

```
int color = Color.BLACK;
int color = Color.argb (127, 0, 255, 255);
```

Accessing Color Information

- It is best to define your colors in an xml file

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
<color name="steelblue">#4682b4</color>
<color name="navy">#000080</color>
</resources>
```

- Note: Four values is ARGB while three values is RGB where A is fully opaque
- Colors can be accessed by `int color = getResources ().getColor (R.color.navy);`

Paint

- The Paint class holds information about
  - style and color
  - how to draw geometries, text, and bitmaps
- Before drawing a color, it is common to set the color with the method setColor (Color.BLUE); which is part of the Paint class

Canvas

- The canvas is a surface do draw on
- Android framework APIs provide 2D drawing APIs to:
  a) render your own graphics on a canvas - need to call onDraw () method passing a Canvas
  b) modify (customizing) existing Views - draw graphics or animations into an existing View
- a) is best for simple graphics with no animation
- b) is best for video games with complex animation and frequent redraws

Basic Display

- An Activity provides the UI for the display screen
- An Activity hosts a View
- A View hosts a Canvas
- The onDraw () method can be overridden to perform custom drawing on the Canvas

Custom View

```
public class CustomView extends View
{
  private ShapeDrawable mDrawable;
  public CustomView (Context context)
  {
    super (context);
    int x = 10, y = 10, width = 300, height = 50;

    mDrawable = new ShapeDrawable (new OvalShape());
    mDrawable.getPaint ().setColor (0xff74AC23);
    mDrawable.setBounds (x, y, x + width, y + height);
  }
  protected void onDraw (Canvas canvas)
  {
    mDrawable.draw (canvas);
  }
} // http://developer.android.com/guide/topics/graphics/2d-graphics.html
```

Drawing Custom View

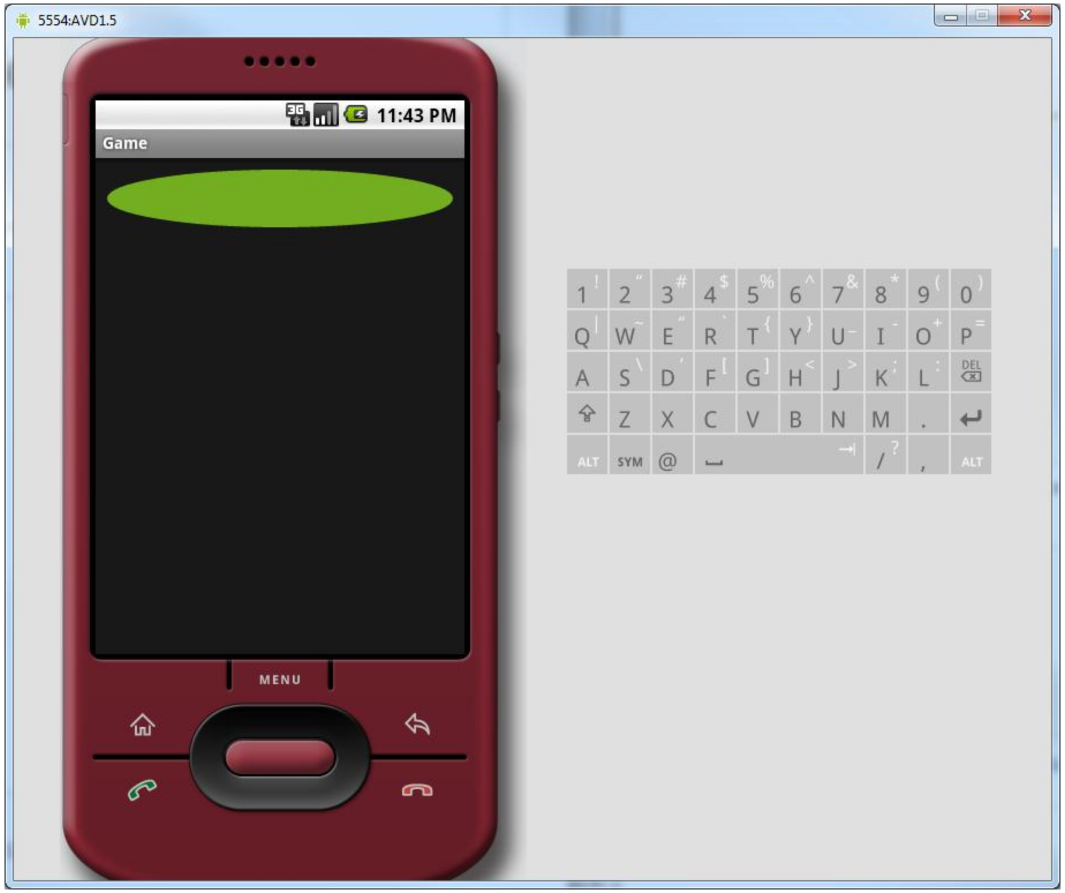
```
CustomDrawableView mCustomDrawableView;

protected void onCreate (Bundle savedInstanceState)
{
  super.onCreate (savedInstanceState);
  mCustomDrawableView = new CustomDrawableView (this);

  setContentView (mCustomDrawableView);
} // http://developer.android.com/guide/topics/graphics/2d-graphics.html
```

Tic-Tac-Toe

- Let’s assume we are going to create a tic-tac-toe game
- With what you know about Android, how would you design the game?

Wire up Tic-Tac-Toe

- Create a TicTacToe class that accepts the game difficulty from the GameActivity activity.
- The TicTacToe activity is to show the difficulty level in a Log.d call
- Then create a new CustomView and setContentView to your custom view instance variable
- The results are on the next slide


5554:AVD1.5
11:43 PM
Game
MENU

CustomView Modifications

- You can do any kind of drawing in the onDraw of a custom view.
- The following slides allow you to draw the lines of a Tic Tac Toe board
- See if you can get this to work

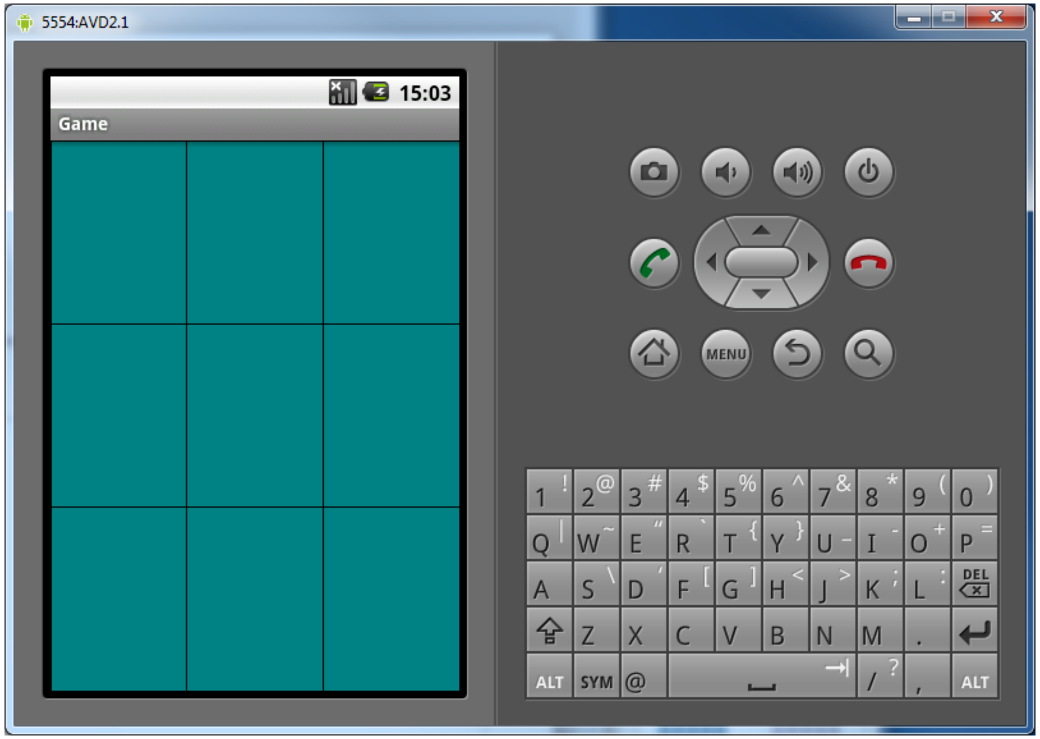
Add Instance Variables

```
private float mWidth;    // width of one tile
 private float mHeight;   // height of one tile
 private int mSelectX;       // X index of selection
 private int mSelectY;       // Y index of selection
 private final Rect mSelectRect = new Rect();
```

Add onDraw Logic

```
Paint background = new Paint ();
   background.setColor (getResources ().getColor (R.color.Teal));
   canvas.drawRect (0, 0, getWidth (), getHeight (), background);

   Paint darkLines = new Paint ();
   darkLines.setColor (Color.BLACK);

   // Draw the  grid lines
   for (int i = 0; i < (int) mNUMBER_OF_RECTANGLES; i++)
   {
        canvas.drawLine (0, i * mRectangleHeight, getWidth(),
                         i * mRectangleHeight, darkLines);
        canvas.drawLine (i * mRectangleWidth, 0,
                         i * mRectangleWidth, getHeight(), darkLines);
   }
```

Result


5554:AVD2.1
15:03
Game
MENU
DEL
ALT
SYM
ALT