

Beginner Android Dev Workshop Spring 2017

Getting started with Android Studio

before we begin

- it helps if you are familiar with:
 - Java
 - using an IDE
 - graphics
 - Git
- pair up with a partner or 2!
- **all users:** have the latest Android Studio downloaded (standard configuration)!

intro

What is Android Studio?

“Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA . On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps...”

Excerpt From: <https://developer.android.com/studio>

what to expect

you will be touching upon the basics of

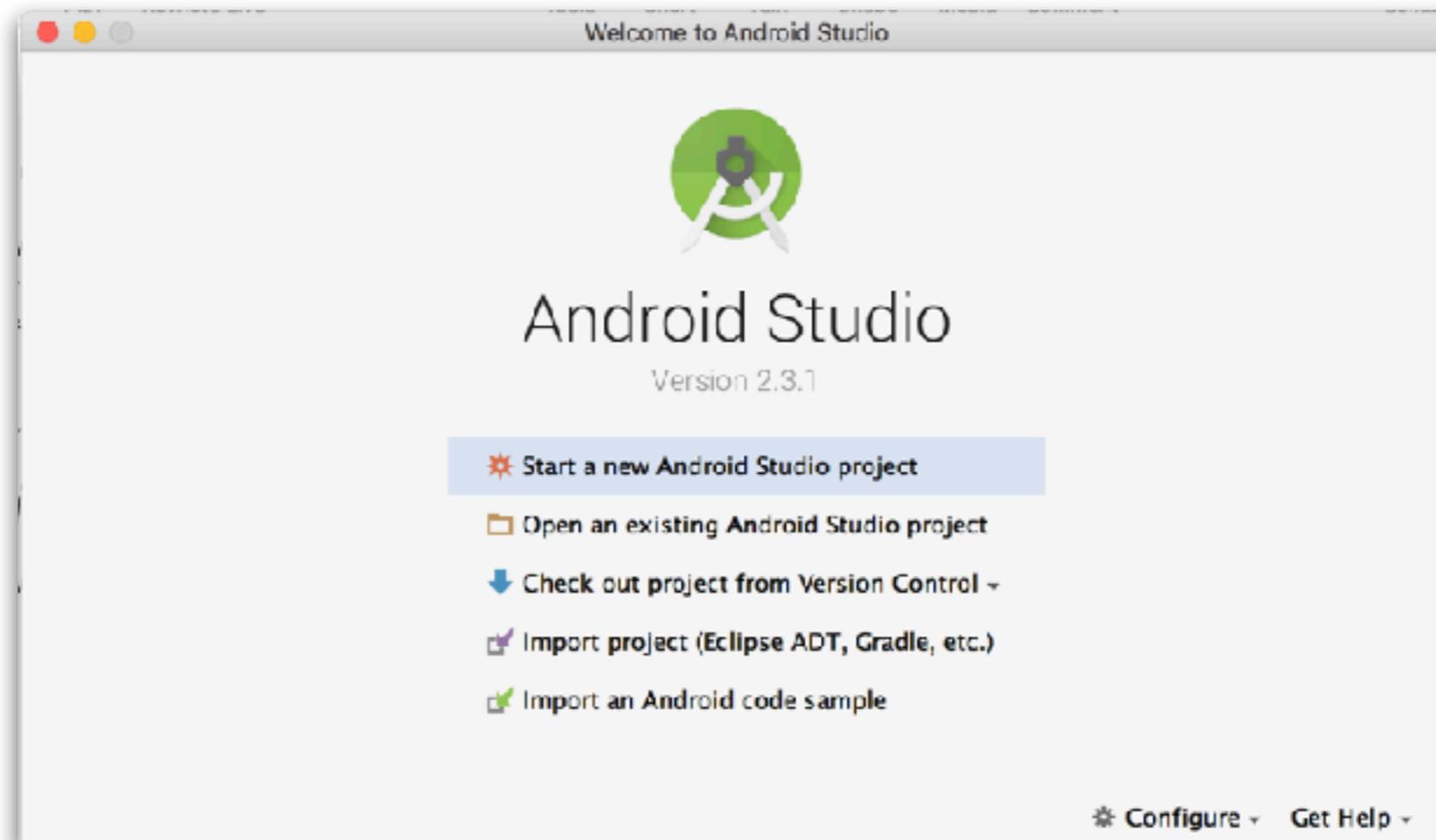
- [Android Studio](#)
- configuring Android projects
- Android project structure
- setting up an alternative emulator
- how graphics come into play with games

let's finish rocket guardian!

Rocket Guardian is an open-source game similar to Flappy Bird. During this workshop, we'll get to setup the environment and fix a few lines to complete the game.

1. let's check out rocket guardian! if you have an android phone, download **rocket guardian** from the play store
2. you need to download, install, and properly configure the latest **Android Studio, Java JDK, Genymotion, and VirtualBox 5.0.28**.
follow slides 4-19 of this presentation: <https://goo.gl/QtTF7C>
3. while your packages are downloading,
 - 3.1. **skim through the first page this resource** provided by [CodePath](#) on how to get started with general Android dev <https://goo.gl/msmNZH>
 - 3.2. **read the intro** for Android Studio at developer.android.com/studio/intro

4. once everything is properly installed and configured, download or clone the project located at <https://goo.gl/52ENTz> to your **AndroidStudioProjects** folder
5. from the Android Studio welcome screen, select *open project* and find the rocket guardian project you just downloaded in your computer



6. get familiarized with the IDE

6.1. notice the file structure to the left. you have several folders which contain configuration files, assets, and source files

6.2. with this IDE, you can even select a file and look at a breakdown of its own structure (variables, methods, etc)

The screenshot displays an IDE interface for an Android project named 'rocket-guardian'. The left sidebar shows the project structure with the following file tree:

- rocket-guardian
 - core
 - src
 - atm
 - rocketguardian
 - entities
 - Adult (selected)

The main editor window shows the code for `Adult.java` in the `atm.rocketguardian.entities` package. The code includes various imports and a class definition:

```
1 package atm.rocketguardian.entities;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 import com.badlogic.gdx.graphics.g2d.Animation;
7 import com.badlogic.gdx.graphics.g2d.Batch;
8 import com.badlogic.gdx.graphics.g2d.Sprite;
9 import com.badlogic.gdx.graphics.g2d.TextureAtlas;
10 import com.badlogic.gdx.math.MathUtils;
11 import com.badlogic.gdx.math.Vector2;
12 import com.badlogic.gdx.physics.box2d.Body;
13 import com.badlogic.gdx.physics.box2d.BodyDef;
14 import com.badlogic.gdx.physics.box2d.BodyDef.BodyType;
15 import com.badlogic.gdx.physics.box2d.Fixture;
16 import com.badlogic.gdx.physics.box2d.PolygonShape;
17 import com.badlogic.gdx.physics.box2d.World;
18 import com.badlogic.gdx.scenes.scene2d.Actor;
19
20 import atm.rocketguardian.helpers.AssetLoader;
21 import atm.rocketguardian.helpers.Constants;
22
23 public class Adult extends Actor implements Human {
24     private World world;
25     private Body body;
26     private BodyDef bodyDef;
27     private Fixture fixture;
28     private Sprite sprite;
29     private float stateTime;
30     private Boolean isAlive;
31     private List<Animation<TextureAtlas.AtlasRegion>> animationsList;
32     private int currentAnimation;
33     private Boolean isOnFloor;
34
35
36     public Adult(World world, Vector2 position) {
37
38         // Create the list with the available animations
39         animationsList = createAnimationsList();
40         currentAnimation = MathUtils.random(animationsList.size() - 1);
41         // Assign world and other properties
```

The bottom panel shows the Event Log with the following entries:

- 4/7/17 11:13 Gradle sync started
- 11:13 Gradle sync completed
- 11:13 Executing tasks: [:android:generateDebugSources, :android:generateDebugAndroidTestSources, :android:mockableAndroidJar, :android:prepareDebugUnitTestDependencies]
- 11:13 Gradle build finished in 1s 907ms

The status bar at the bottom indicates: 'Crade build finished in 1s 907ms (2 minutes ago)'. The system tray shows '7:24 LP+ UTF-8 CI: master Context: <no context>'.

for example, the `Adult.java` file structure under the **core** folder

break

7. try to run rocket guardian for the first time
 - 7.1. open Genymotion
 - 7.2. start the Nexus phone emulator in Genymotion
 - 7.3. hit play ▶ on in Android Studio and select the Nexus phone under Connected Devices
 - 7.4. exit the emulator and go back to the project in Android Studio

8. let's fix rocket guardian!

9. let's change the # of lives we have.
we only have 1

9.1.1. find the **RocketGuard.java** file under
core > src > entities and on Line 50
change it from "1" to "5"

9.1.2. the HUD for the game needs to keep up
with the current # of lives too! find the
Hud.java file under **core > src > screens**
and on Line 33 from "1" to "5"

10. some sounds are not loading properly

10.1. fix the path in the **AssetsLoader.java** file under **core > src > helpers** and on Line 59 fix the file name from **actionmusic** to **action_music**

10.2. uncomment Line 59 and search through the entire project for “**gameMusic**” (**cmd + shift + f** on mac).

if anything with gameMusic is commented out, remove the comment lines //

10.3. set the looping for **menuMusic** and **gameMusic** on Lines 60-61 to **true**

11. “how to play” is too wordy for me. let’s switch up the welcome screen buttons a bit

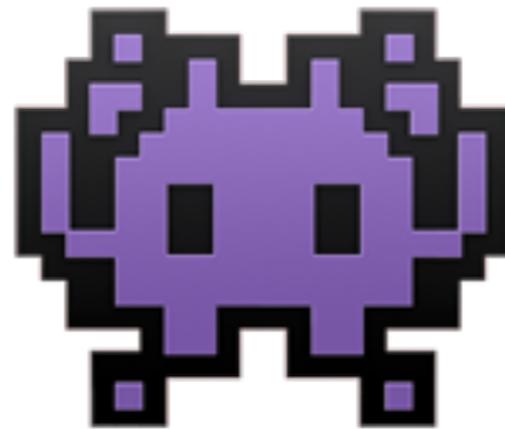
11.1. change **“How to Play”** to **“Instructions”** in Line 66 of **MainMenuScreen.java** (still under the screens folder of the core)

12. the HUD isn't perfect. we need to increase the padding on the score so that it doesn't overlap the Player label

12.1. change the value of **padTop()** to **5** instead of 0 on Lines 72 & 73

14. save your project and hit ▶

15. test it out!



resources

- Rocket Guardian project source <https://goo.gl/Sw2NOV>
- project git: <https://goo.gl/52ENtz>

contact

alinal LeBron.me

alina@timelesstechmiami.com

thank you!

