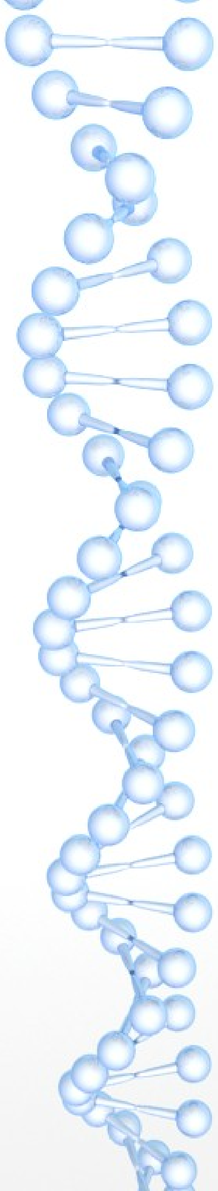
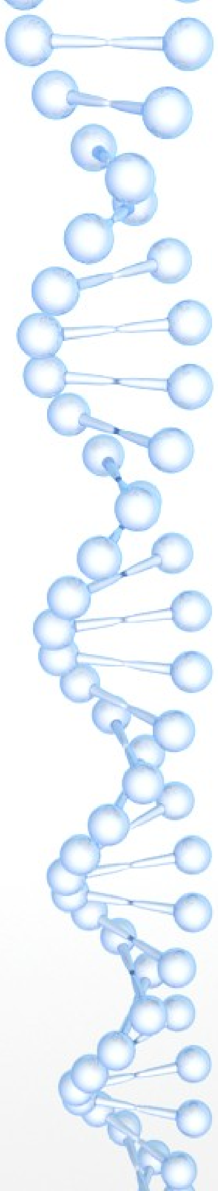


Dou Shou Qi (animal chess) assignment (+ Java + Android Studio + libgdx)

individual project

獸穴
DOU SHOU QI
8
3
5
1
2
4
6
7



What you will need:

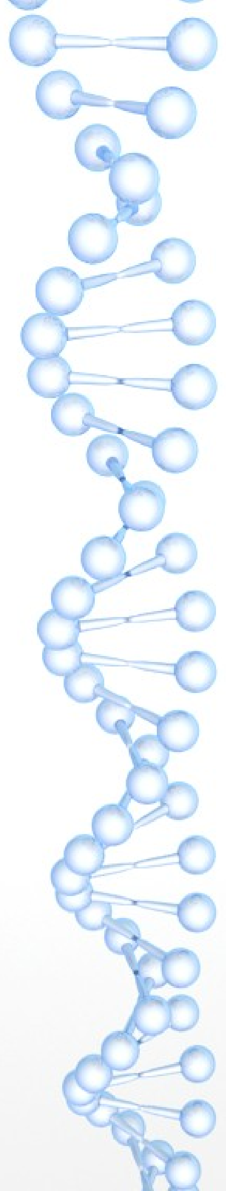
1) a Java JDK
2) Android Studio (based on IntelliJ)
3) libgdx
4) starter code

1) A Java JDK

- You probably already have one installed.
  - Android Studio often will install it for you. So you can skip this step.
  - If it (AS) doesn’t install a JDK, visit https://www.oracle.com/java/technologies/downloads/.

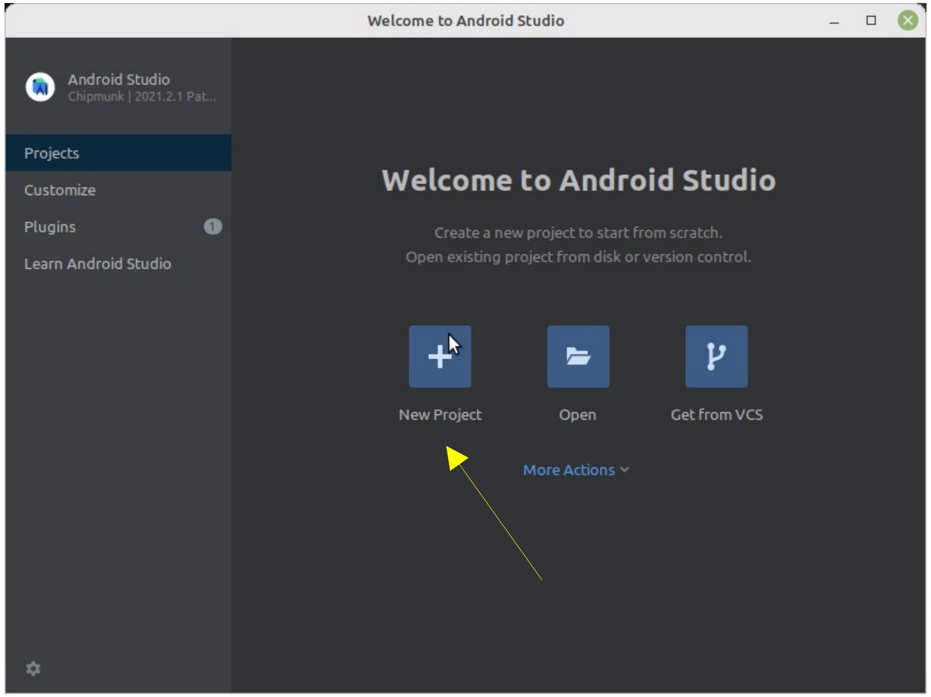
2) Android Studio

- Based on IntelliJ (but install Android Studio anyway).
- Visit https://developer.android.com/studio/install.
- Why Android Studio?
  - It's Google's official develop platform for Android.
  - It can be used to develop mobile apps for both Android and IOS (if one uses Dart + Flutter).
    - Android has 87% of the world market share; IOS has 22% (https://leftronic.com/blog/android-vs-ios-market-share/).
  - **It can be used to develop game apps for Android and IOS as well as Mac, Windows, and Linux desktops (if one uses Java + libgdx).**



After installing, let’s get Android Studio up and running for the first time.

Let’s check the installation.



Welcome to Android Studio

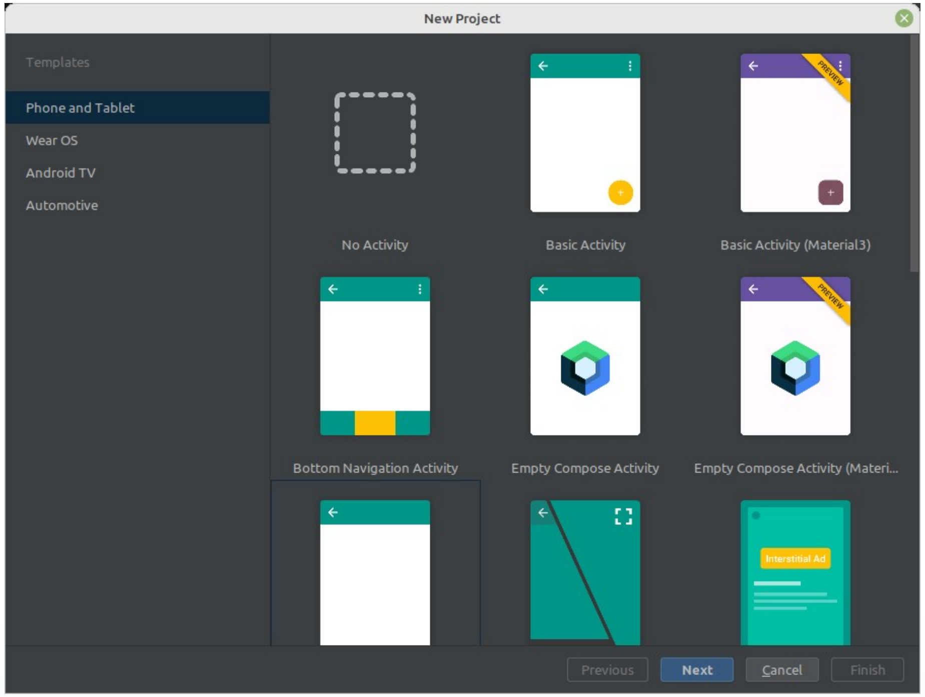
Android Studio
Chipmunk | 2021.2.1 Pat...

Projects
Customize
Plugins 1
Learn Android Studio

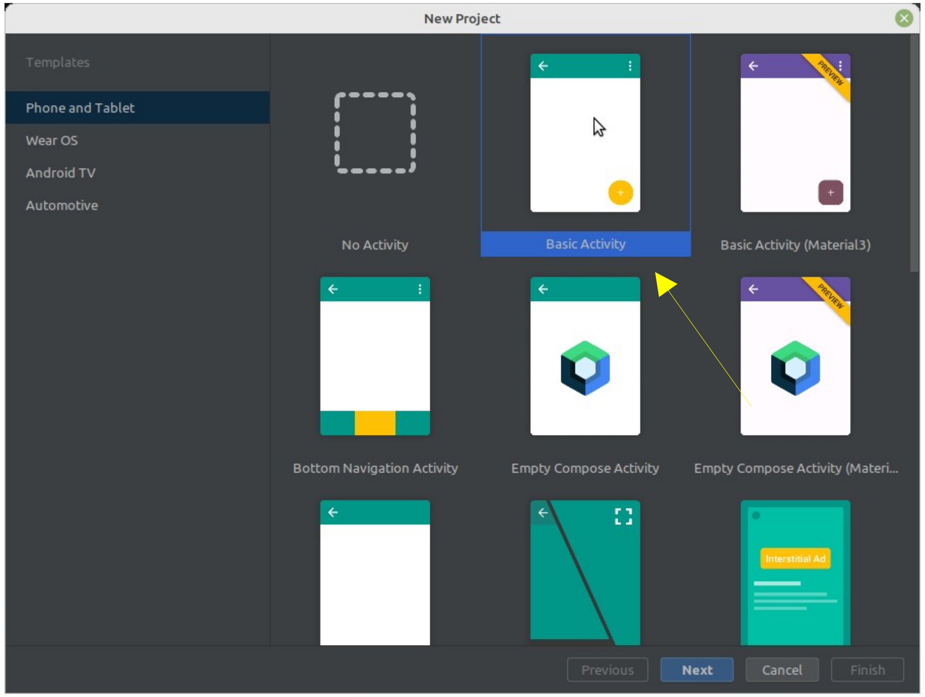
Welcome to Android Studio

Create a new project to start from scratch.
Open existing project from disk or version control.

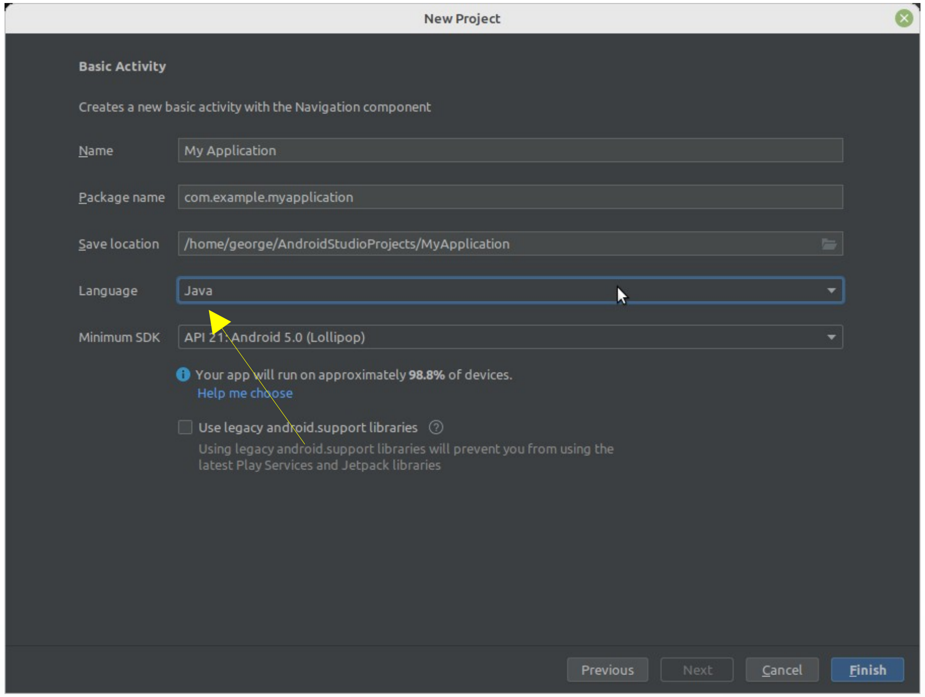
Many different project types are supported.



Choose a simple one.



Change Kotlin to Java.


New Project

Basic Activity

Creates a new basic activity with the Navigation component

Name: My Application

Package name: com.example.myapplication

Save location: /home/george/AndroidStudioProjects/MyApplication

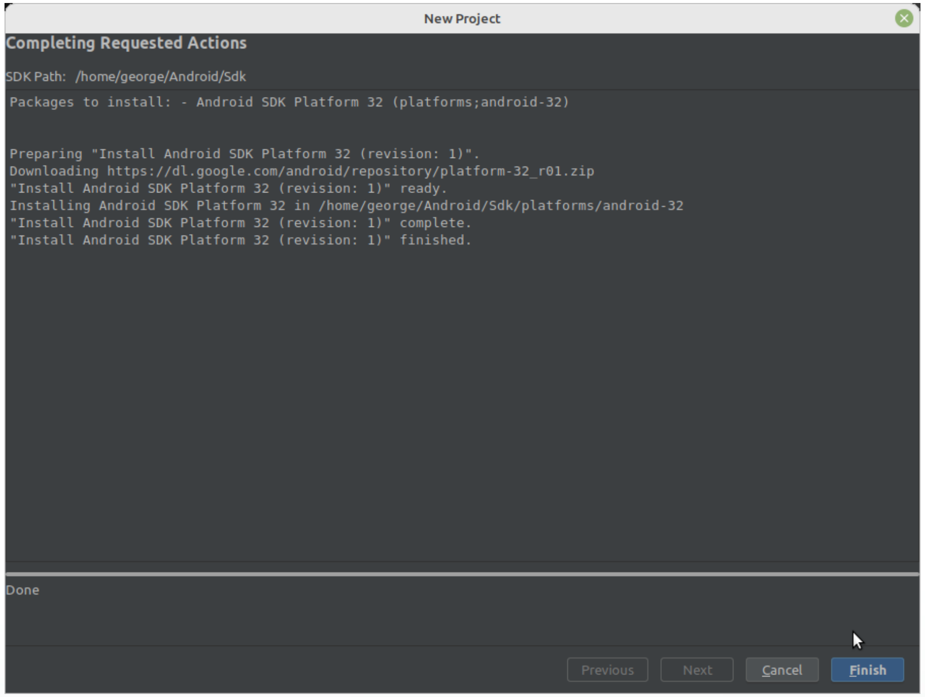
Language: Java

Minimum SDK: API 21: Android 5.0 (Lollipop)

Your app will run on approximately **98.8%** of devices.
Help me choose

☐ Use legacy android.support libraries
Using legacy android.support libraries will prevent you from using the latest Play Services and Jetpack libraries

An Android SDK will be installed (one time only).

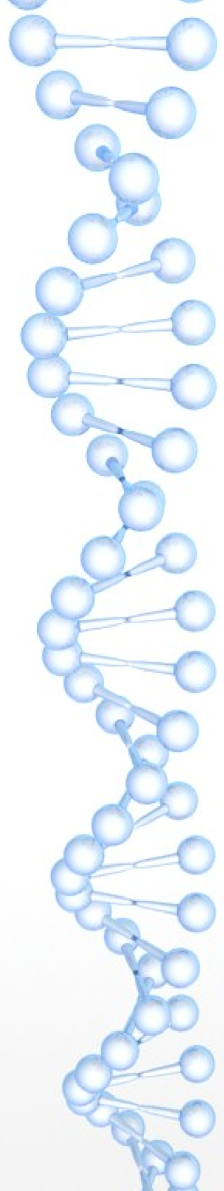


Completing Requested Actions

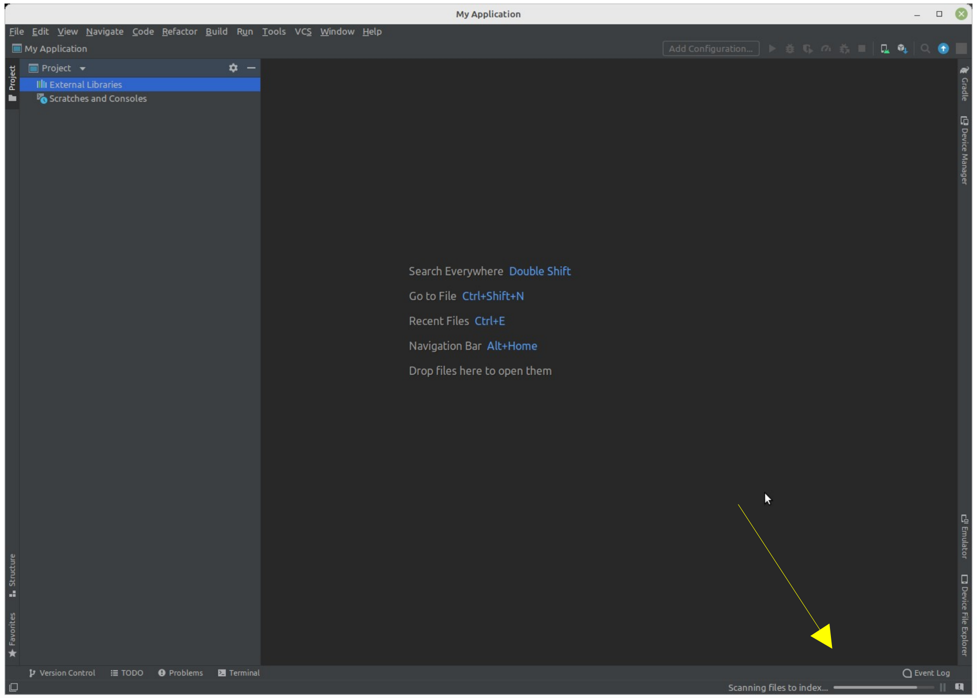
SDK Path: /home/george/Android/Sdk

```
Packages to install: - Android SDK Platform 32 (platforms;android-32)


Preparing "Install Android SDK Platform 32 (revision: 1)".
Downloading https://dl.google.com/android/repository/platform-32_r01.zip
"Install Android SDK Platform 32 (revision: 1)" ready.
Installing Android SDK Platform 32 in /home/george/Android/Sdk/platforms/android-32
"Install Android SDK Platform 32 (revision: 1)" complete.
"Install Android SDK Platform 32 (revision: 1)" finished.
```

Done

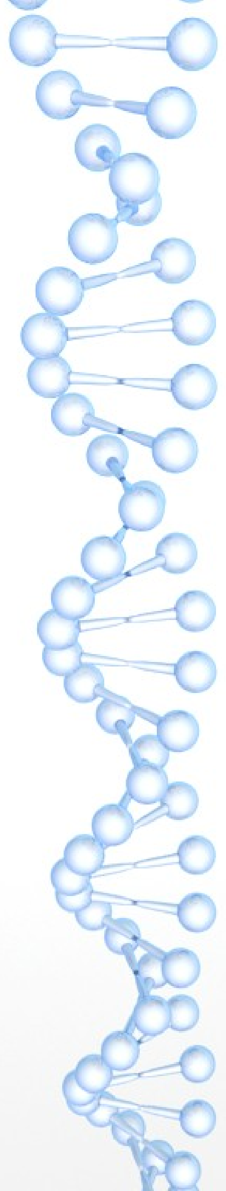


Busy.

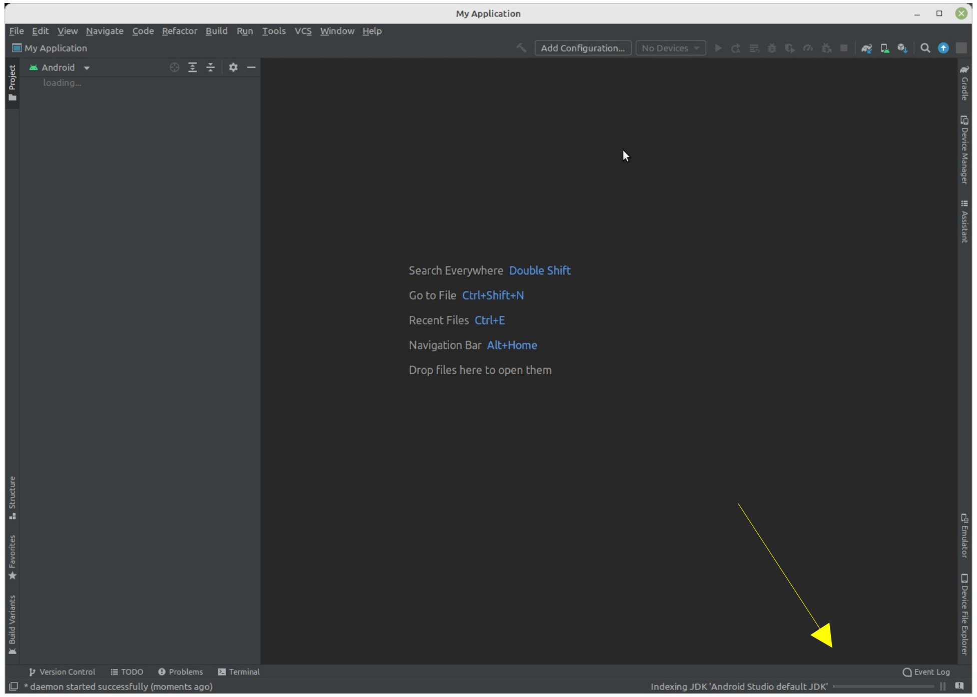


Search Everywhere Double Shift
Go to File Ctrl+Shift+N
Recent Files Ctrl+E
Navigation Bar Alt+Home
Drop files here to open them

Scanning files to index...



Busy.



loading...

Search Everywhere Double Shift

Go to File Ctrl+Shift+N

Recent Files Ctrl+E

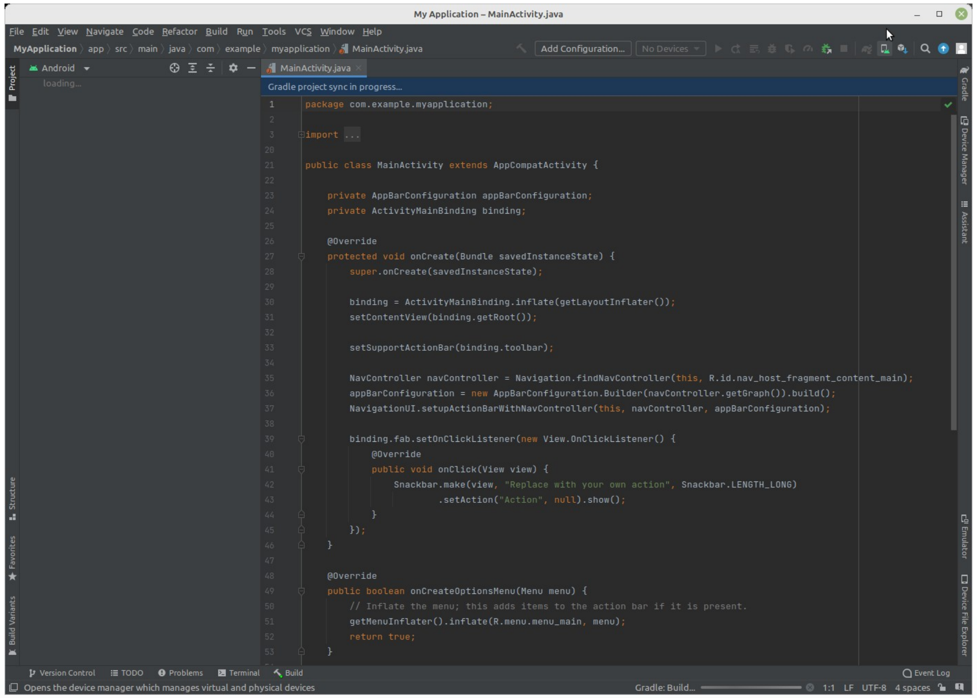
Navigation Bar Alt+Home

Drop files here to open them

* daemon started successfully (moments ago)

Indexing JDK 'Android Studio default JDK'

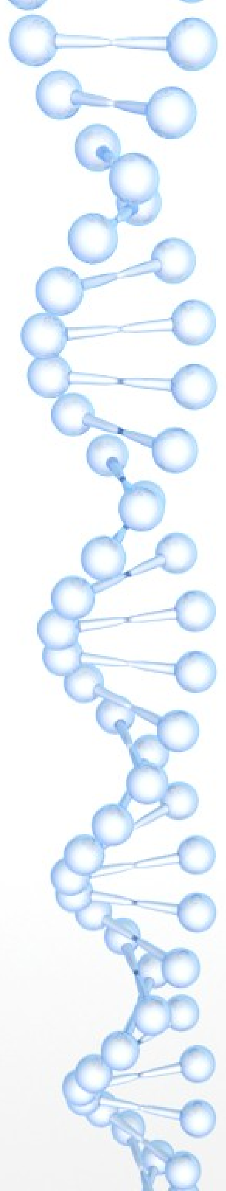
It created a shell of an Android app for us.



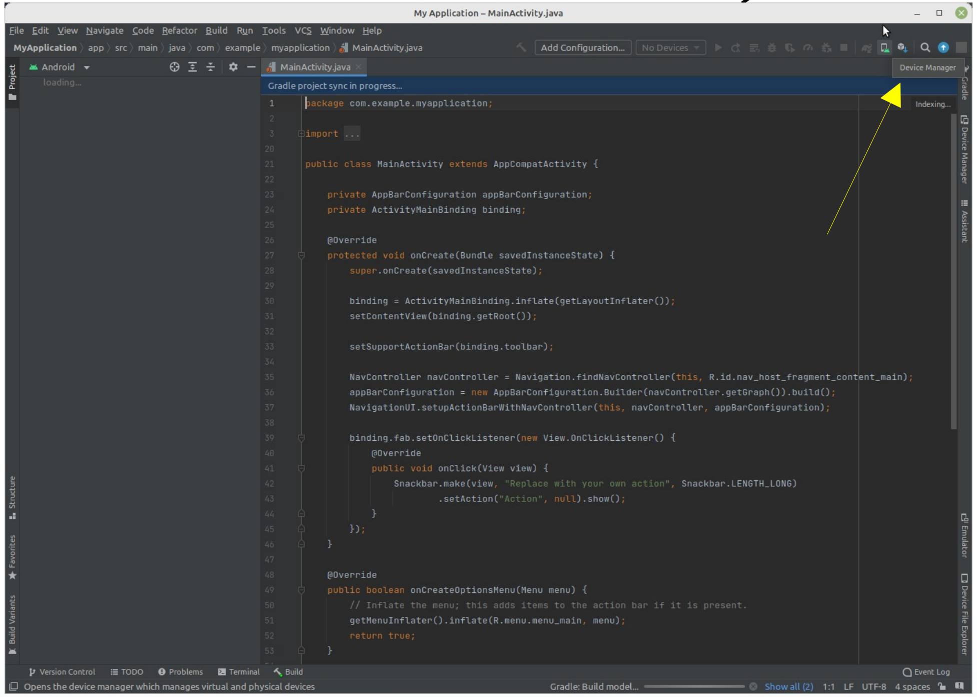
Gradle project sync in progress...

```java
package com.example.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

    private AppBarConfiguration appBarConfiguration;
    private ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        setSupportActionBar(binding.toolbar);

        NavController navController = Navigation.findNavController(this, R.id.nav_host_fragment_content_main);
        appBarConfiguration = new AppBarConfiguration.Builder(navController.getGraph()).build();
        NavigationUI.setupActionBarWithNavController(this, navController, appBarConfiguration);

        binding.fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }
```

Opens the device manager which manages virtual and physical devices   Gradle: Build...   1:1 LF UTF-8 4 spaces



Before we can run it, we need to create an Android emulator (or plug in a real Android device via USB).

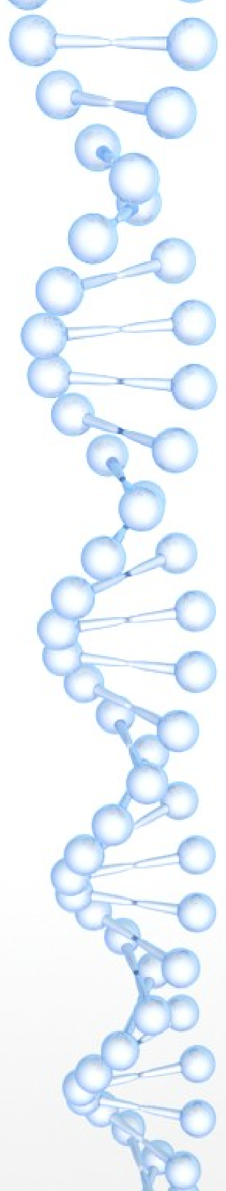


Gradle project sync in progress...

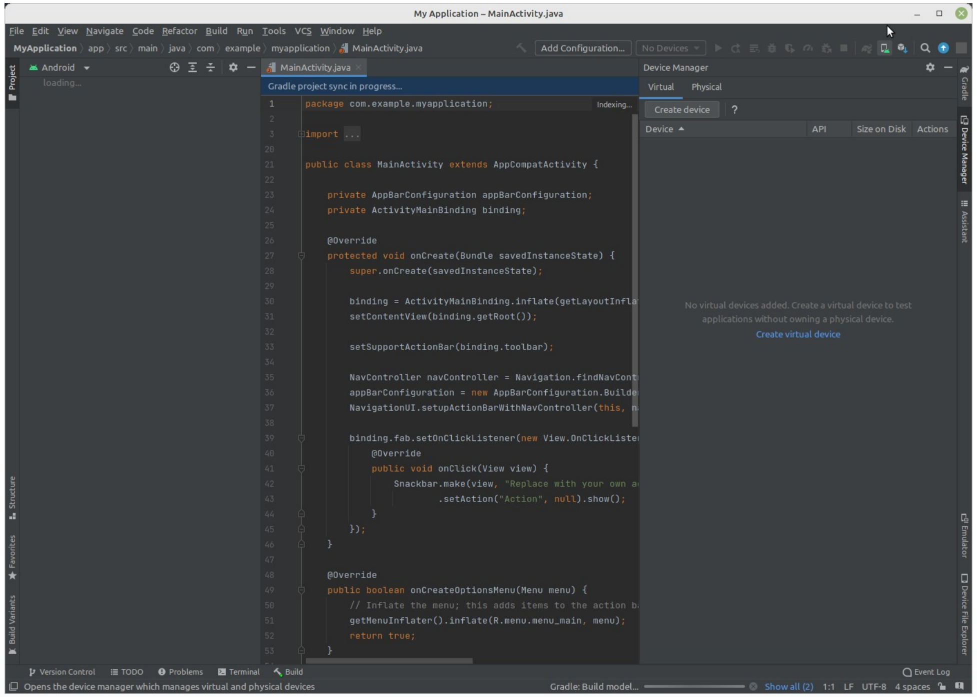
```
package com.example.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

    private AppBarConfiguration appBarConfiguration;
    private ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        setSupportActionBar(binding.toolbar);

        NavController navController = Navigation.findNavController(this, R.id.nav_host_fragment_content_main);
        appBarConfiguration = new AppBarConfiguration.Builder(navController.getGraph()).build();
        NavigationUI.setupActionBarWithNavController(this, navController, appBarConfiguration);

        binding.fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }
```

Opens the device manager which manages virtual and physical devices

Gradle: Build model... Show all (2) 1:1 LF UTF-8 4 spaces



Create a virtual device (emulator).



Gradle project sync in progress...

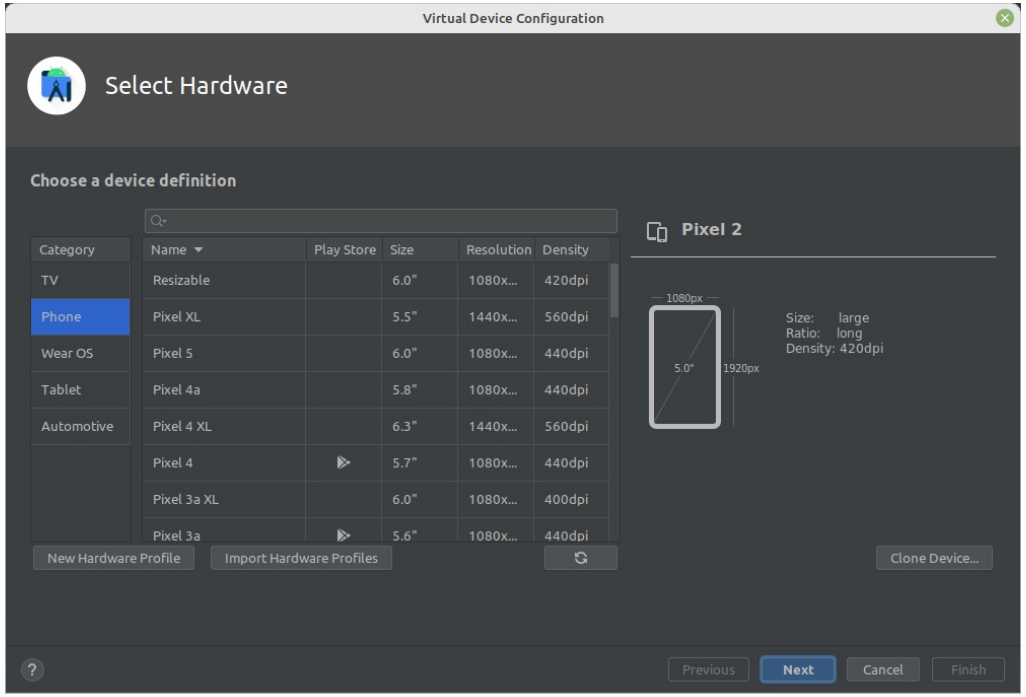
Indexing...

```
package com.example.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

    private AppBarConfiguration appBarConfiguration;
    private ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = ActivityMainBinding.inflate(getLayoutInfla
        setContentView(binding.getRoot());

        setSupportActionBar(binding.toolbar);

        NavController navController = Navigation.findNavCont
        appBarConfiguration = new AppBarConfiguration.Builde
        NavigationUI.setupActionBarWithNavController(this, n

        binding.fab.setOnClickListener(new View.OnClickListe
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own a
                        .setAction("Action", null).show();
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action b
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }
```

No virtual devices added. Create a virtual device to test applications without owning a physical device.
Create virtual device

Opens the device manager which manages virtual and physical devices

Choose a specific one to emulate.



Virtual Device Configuration

Select Hardware

Choose a device definition

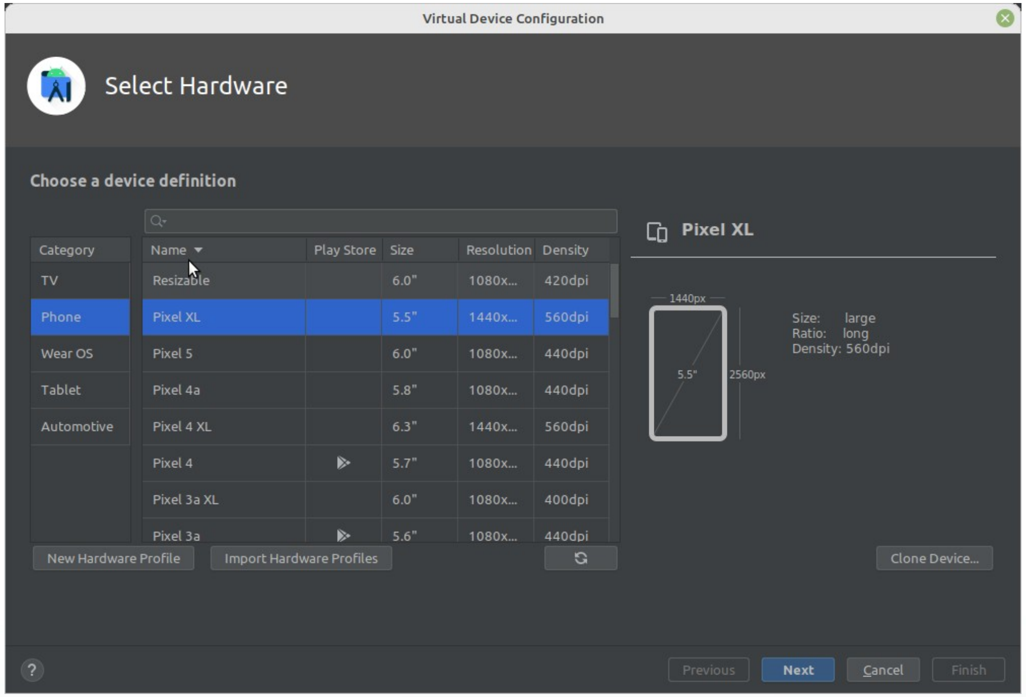
Category
TV
Phone
Wear OS
Tablet
Automotive

Name	Play Store	Size	Resolution	Density
Resizable		6.0"	1080x...	420dpi
Pixel XL		5.5"	1440x...	560dpi
Pixel 5		6.0"	1080x...	440dpi
Pixel 4a		5.8"	1080x...	440dpi
Pixel 4 XL		6.3"	1440x...	560dpi
Pixel 4	▶	5.7"	1080x...	440dpi
Pixel 3a XL		6.0"	1080x...	400dpi
Pixel 3a	▶	5.6"	1080x...	440dpi

Pixel 2

1080px
5.0"
1920px

Size: large
Ratio: long
Density: 420dpi


Virtual Device Configuration

Select Hardware

Choose a device definition

Category
TV
Phone
Wear OS
Tablet
Automotive

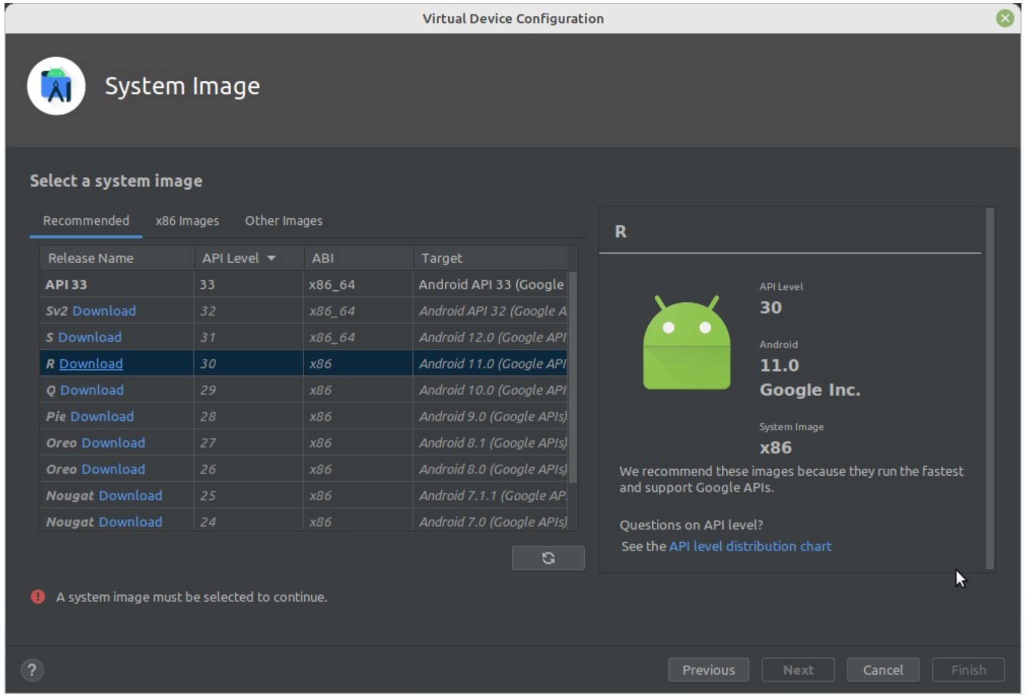
Name	Play Store	Size	Resolution	Density
Resizable		6.0"	1080x...	420dpi
Pixel XL		5.5"	1440x...	560dpi
Pixel 5		6.0"	1080x...	440dpi
Pixel 4a		5.8"	1080x...	440dpi
Pixel 4 XL		6.3"	1440x...	560dpi
Pixel 4	▶	5.7"	1080x...	440dpi
Pixel 3a XL		6.0"	1080x...	400dpi
Pixel 3a	▶	5.6"	1080x...	440dpi

Pixel XL

1440px
5.5"
2560px

Size: large
Ratio: long
Density: 560dpi

Choose a version of the Android API to emulate.


Virtual Device Configuration

System Image

Select a system image

Recommended | x86 Images | Other Images

Release Name	API Level	ABI	Target
API 33	33	x86_64	Android API 33 (Google
Sv2 Download	32	x86_64	Android API 32 (Google A
S Download	31	x86_64	Android 12.0 (Google API
R Download	30	x86	Android 11.0 (Google API
Q Download	29	x86	Android 10.0 (Google API
Pie Download	28	x86	Android 9.0 (Google APIs)
Oreo Download	27	x86	Android 8.1 (Google APIs)
Oreo Download	26	x86	Android 8.0 (Google APIs)
Nougat Download	25	x86	Android 7.1.1 (Google AP
Nougat Download	24	x86	Android 7.0 (Google APIs)

R

API Level
30

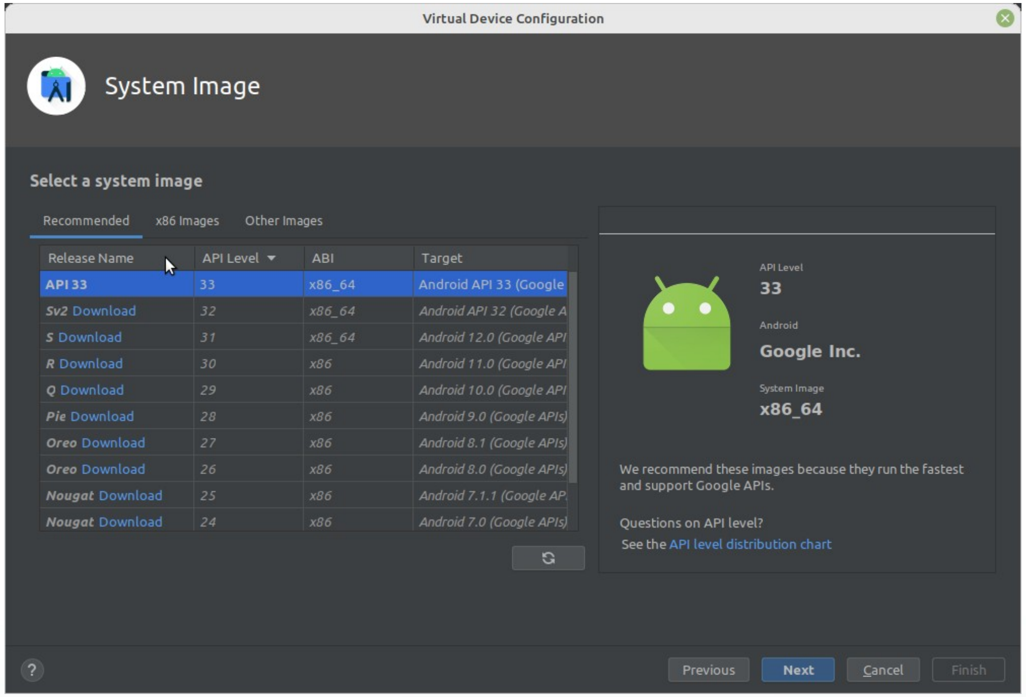
Android
11.0
Google Inc.

System Image
x86

We recommend these images because they run the fastest and support Google APIs.

Questions on API level?
See the API level distribution chart

A system image must be selected to continue.


Virtual Device Configuration

System Image

Select a system image

Release Name	API Level	ABI	Target
API 33	33	x86_64	Android API 33 (Google
Sv2 Download	32	x86_64	Android API 32 (Google A
S Download	31	x86_64	Android 12.0 (Google API
R Download	30	x86	Android 11.0 (Google API
Q Download	29	x86	Android 10.0 (Google API
Pie Download	28	x86	Android 9.0 (Google APIs)
Oreo Download	27	x86	Android 8.1 (Google APIs)
Oreo Download	26	x86	Android 8.0 (Google APIs)
Nougat Download	25	x86	Android 7.1.1 (Google AP
Nougat Download	24	x86	Android 7.0 (Google APIs)

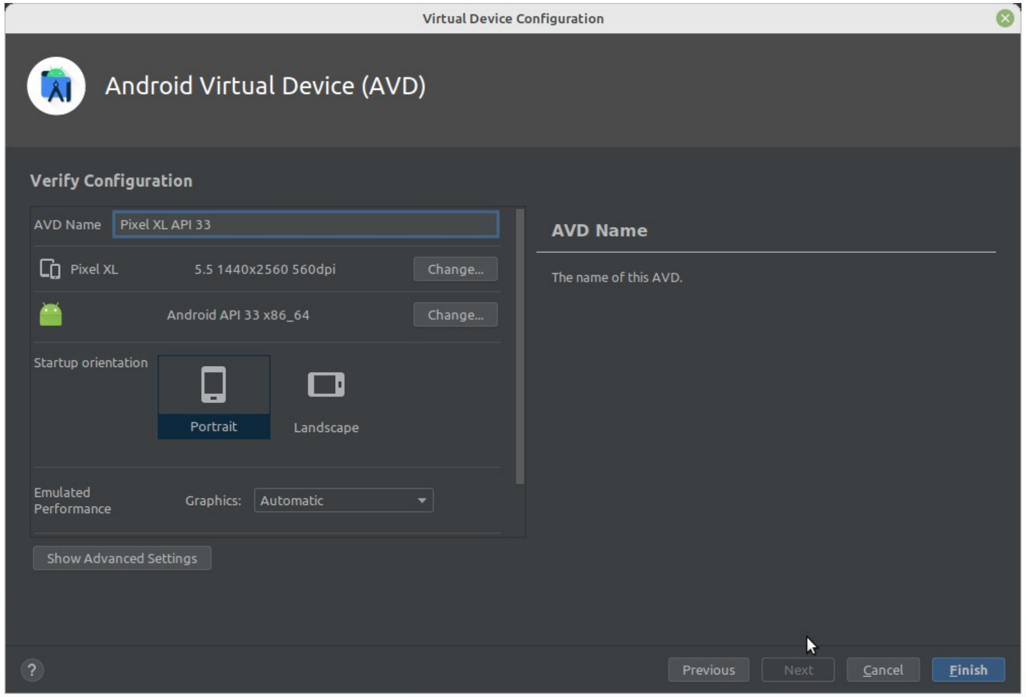
API Level
33

Android
Google Inc.

System Image
x86_64

We recommend these images because they run the fastest and support Google APIs.

Questions on API level?
See the API level distribution chart


Virtual Device Configuration

Android Virtual Device (AVD)

Verify Configuration

AVD Name: Pixel XL API 33

Pixel XL 5.5 1440x2560 560dpi Change...

Android API 33 x86_64 Change...

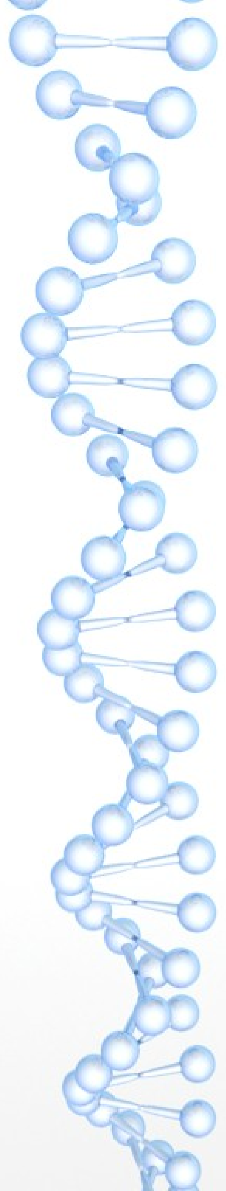
Startup orientation: [✓ Portrait] Landscape

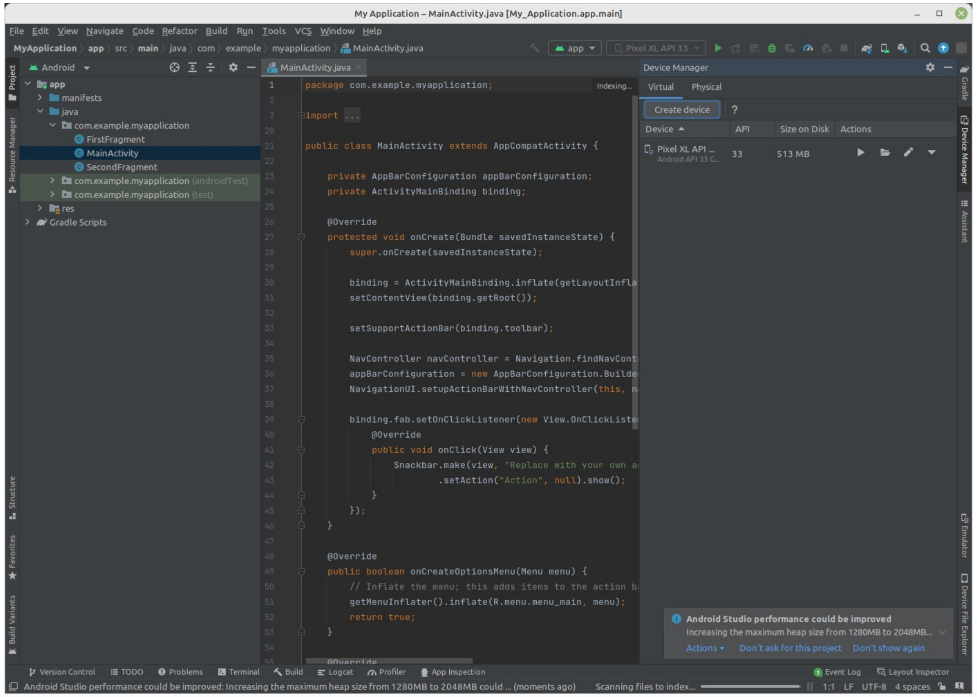
Emulated Performance Graphics: Automatic

Show Advanced Settings

AVD Name

The name of this AVD.



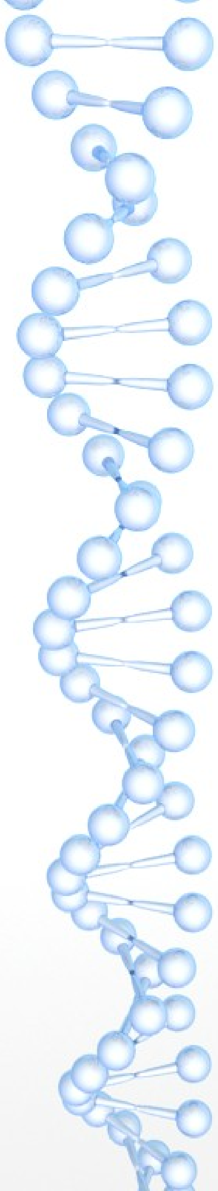

My Application – MainActivity.java [My_Application.app.main]
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
MyApplication 〉 app 〉 src 〉 main 〉 java 〉 com 〉 example 〉 myapplication 〉 MainActivity.java
app
Pixel XL API 33
Android
app
manifests
java
com.example.myapplication
FirstFragment
MainActivity
SecondFragment
com.example.myapplication (androidTest)
com.example.myapplication (test)
res
Gradle Scripts
MainActivity.java
Indexing...
Device Manager
Virtual
Physical
Create device
?
Version Control
TODO
Problems
Terminal
Build
Logcat
Profiler
App Inspection
Event Log
Layout Inspector

```
package com.example.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

    private AppBarConfiguration appBarConfiguration;
    private ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = ActivityMainBinding.inflate(getLayoutInfla
        setContentView(binding.getRoot());

        setSupportActionBar(binding.toolbar);

        NavController navController = Navigation.findNavCont
        appBarConfiguration = new AppBarConfiguration.Builde
        NavigationUI.setupActionBarWithNavController(this, n

        binding.fab.setOnClickListener(new View.OnClickListe
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own a
                        .setAction("Action", null).show();
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action b
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
```

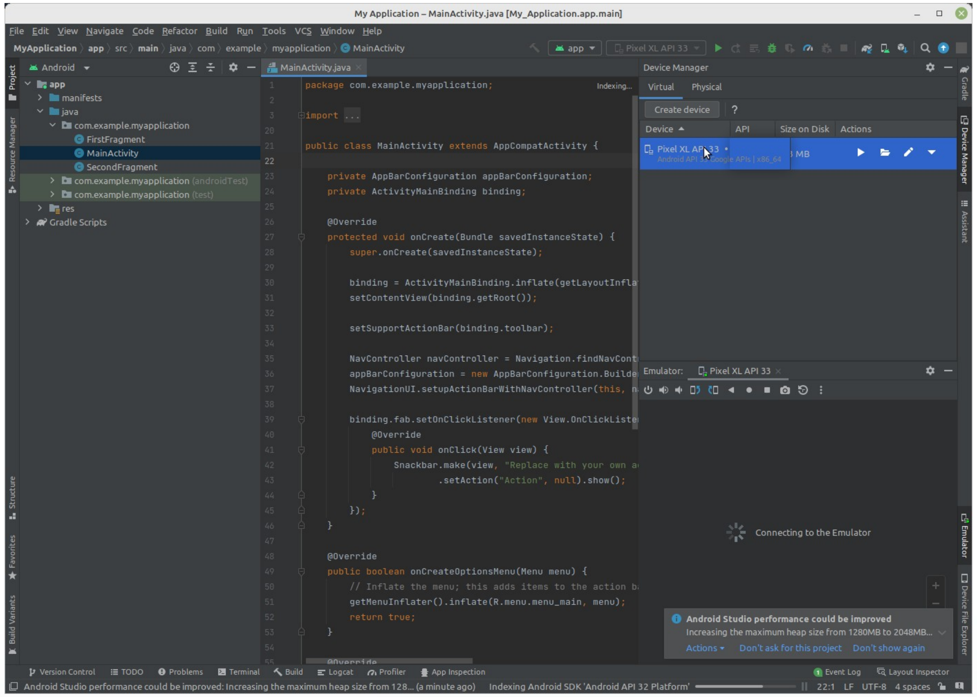
Device	API	Size on Disk	Actions
Pixel XL API ... Android API 33 G...	33	513 MB	

Android Studio performance could be improved
Increasing the maximum heap size from 1280MB to 2048MB...
Actions Don't ask for this project Don't show again

Android Studio performance could be improved: Increasing the maximum heap size from 1280MB to 2048MB could ... (moments ago)   Scanning files to index...   1:1 LF UTF-8 4 spaces



Run the emulator.



```
package com.example.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

    private AppBarConfiguration appBarConfiguration;
    private ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = ActivityMainBinding.inflate(getLayoutInfla
        setContentView(binding.getRoot());

        setSupportActionBar(binding.toolbar);

        NavController navController = Navigation.findNavCont
        appBarConfiguration = new AppBarConfiguration.Builde
        NavigationUI.setupActionBarWithNavController(this, n

        binding.fab.setOnClickListener(new View.OnClickListe
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own a
                        .setAction("Action", null).show();
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action b
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
```

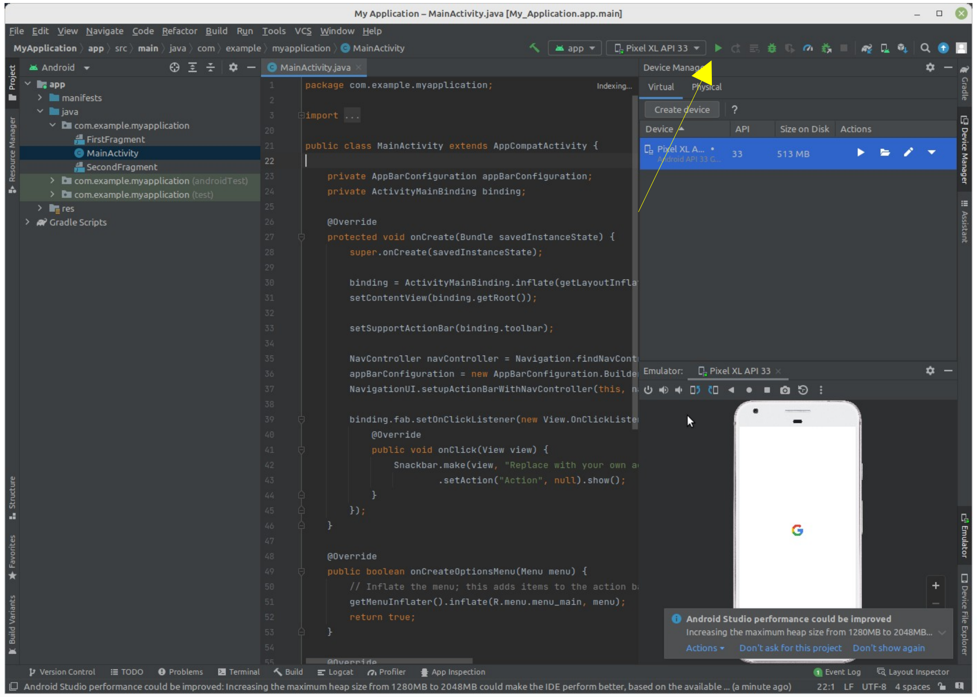
Pixel XL API 33 · Android API 33 Google APIs | x86_64 · MB

Connecting to the Emulator

Android Studio performance could be improved
Increasing the maximum heap size from 1280MB to 2048MB...
Actions · Don't ask for this project · Don't show again

Android Studio performance could be improved: Increasing the maximum heap size from 128... (a minute ago)   Indexing Android SDK 'Android API 32 Platform'   22:1 LF UTF-8 4 spaces

Run our app.



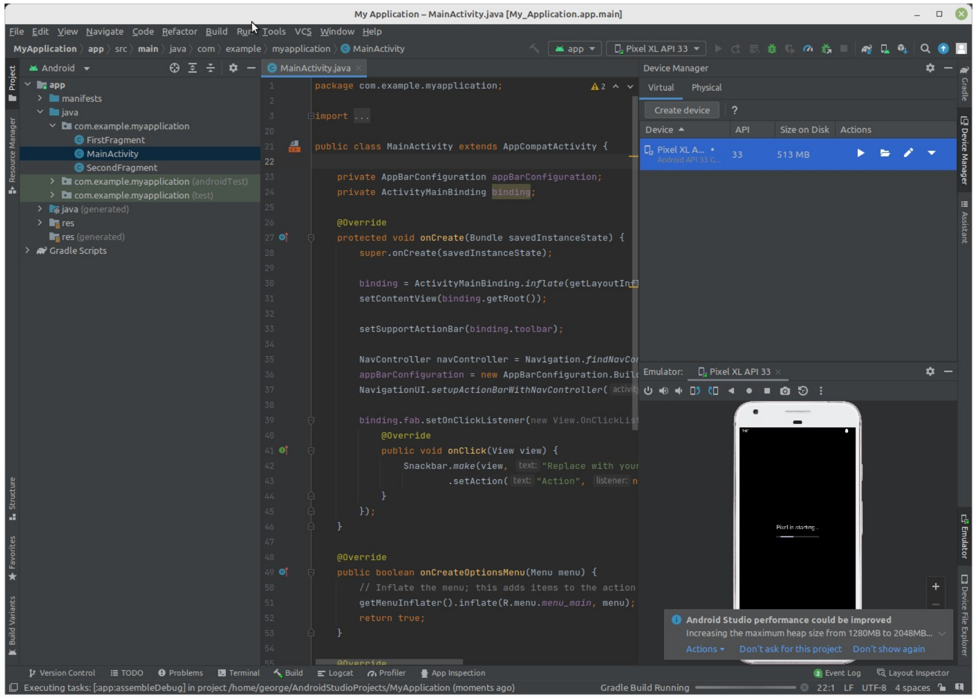
```
package com.example.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

    private AppBarConfiguration appBarConfiguration;
    private ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = ActivityMainBinding.inflate(getLayoutInfla
        setContentView(binding.getRoot());

        setSupportActionBar(binding.toolbar);

        NavController navController = Navigation.findNavCont
        appBarConfiguration = new AppBarConfiguration.Builde
        NavigationUI.setupActionBarWithNavController(this, n

        binding.fab.setOnClickListener(new View.OnClickListe
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own a
                        .setAction("Action", null).show();
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action b
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
```

Device	API	Size on Disk	Actions
Pixel XL A...	33	513 MB	

Android Studio performance could be improved
Increasing the maximum heap size from 1280MB to 2048MB...
Actions Don't ask for this project Don't show again

Android Studio performance could be improved: Increasing the maximum heap size from 1280MB to 2048MB could make the IDE perform better, based on the available ... (a minute ago)

Busy.



My Application – MainActivity.java [My_Application.app.main]

```
package com.example.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

    private AppBarConfiguration appBarConfiguration;
    private ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = ActivityMainBinding.inflate(getLayoutInf
        setContentView(binding.getRoot());

        setSupportActionBar(binding.toolbar);

        NavController navController = Navigation.findNavCo
        appBarConfiguration = new AppBarConfiguration.Buil
        NavigationUI.setupActionBarWithNavController( activi

        binding.fab.setOnClickListener(new View.OnClickLis
            @Override
            public void onClick(View view) {
                Snackbar.make(view, text: "Replace with your
                        .setAction( text: "Action", listener: n
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
```

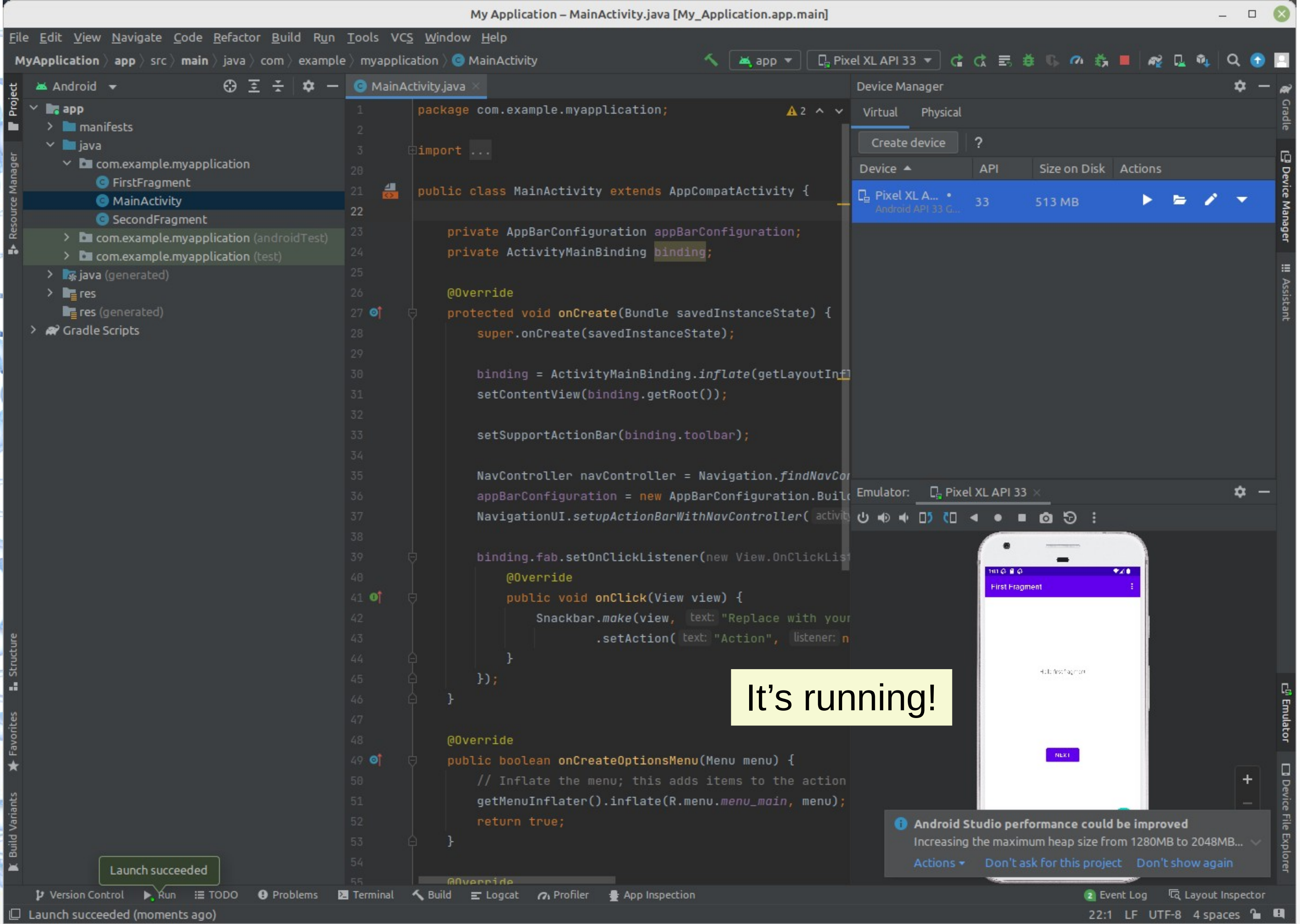
Device Manager

Device	API	Size on Disk	Actions
Pixel XL A...	33	513 MB	

Emulator: Pixel API 33

Android Studio performance could be improved
Increasing the maximum heap size from 1280MB to 2048MB...
Actions · Don't ask for this project · Don't show again

```
Executing tasks: [:app:assembleDebug] in project /home/george/AndroidStudioProjects/MyApplication (moments ago)    Gradle Build Running    22:1  LF  UTF-8  4 spaces
```



My Application – MainActivity.java [My_Application.app.main]

```
package com.example.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

    private AppBarConfiguration appBarConfiguration;
    private ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = ActivityMainBinding.inflate(getLayoutInf
        setContentView(binding.getRoot());

        setSupportActionBar(binding.toolbar);

        NavController navController = Navigation.findNavCo
        appBarConfiguration = new AppBarConfiguration.Buil
        NavigationUI.setupActionBarWithNavController( activ

        binding.fab.setOnClickListener(new View.OnClickLis
            @Override
            public void onClick(View view) {
                Snackbar.make(view, text: "Replace with your
                        .setAction( text: "Action", listener: n
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
```

Device Manager

Device	API	Size on Disk	Actions
Pixel XL A... Android API 33 G...	33	513 MB	

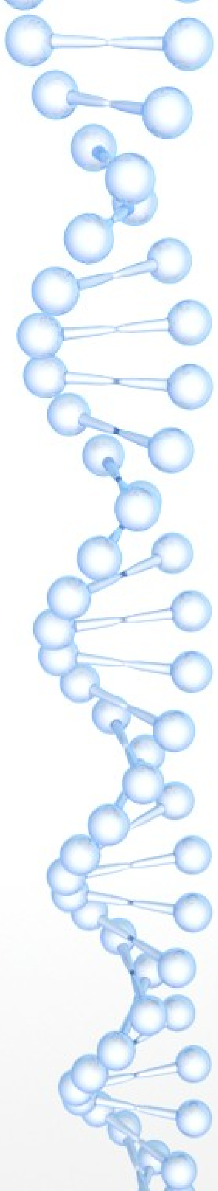
Emulator: Pixel XL API 33

Android Studio performance could be improved
Increasing the maximum heap size from 1280MB to 2048MB...
Actions ▾ Don't ask for this project Don't show again

Launch succeeded

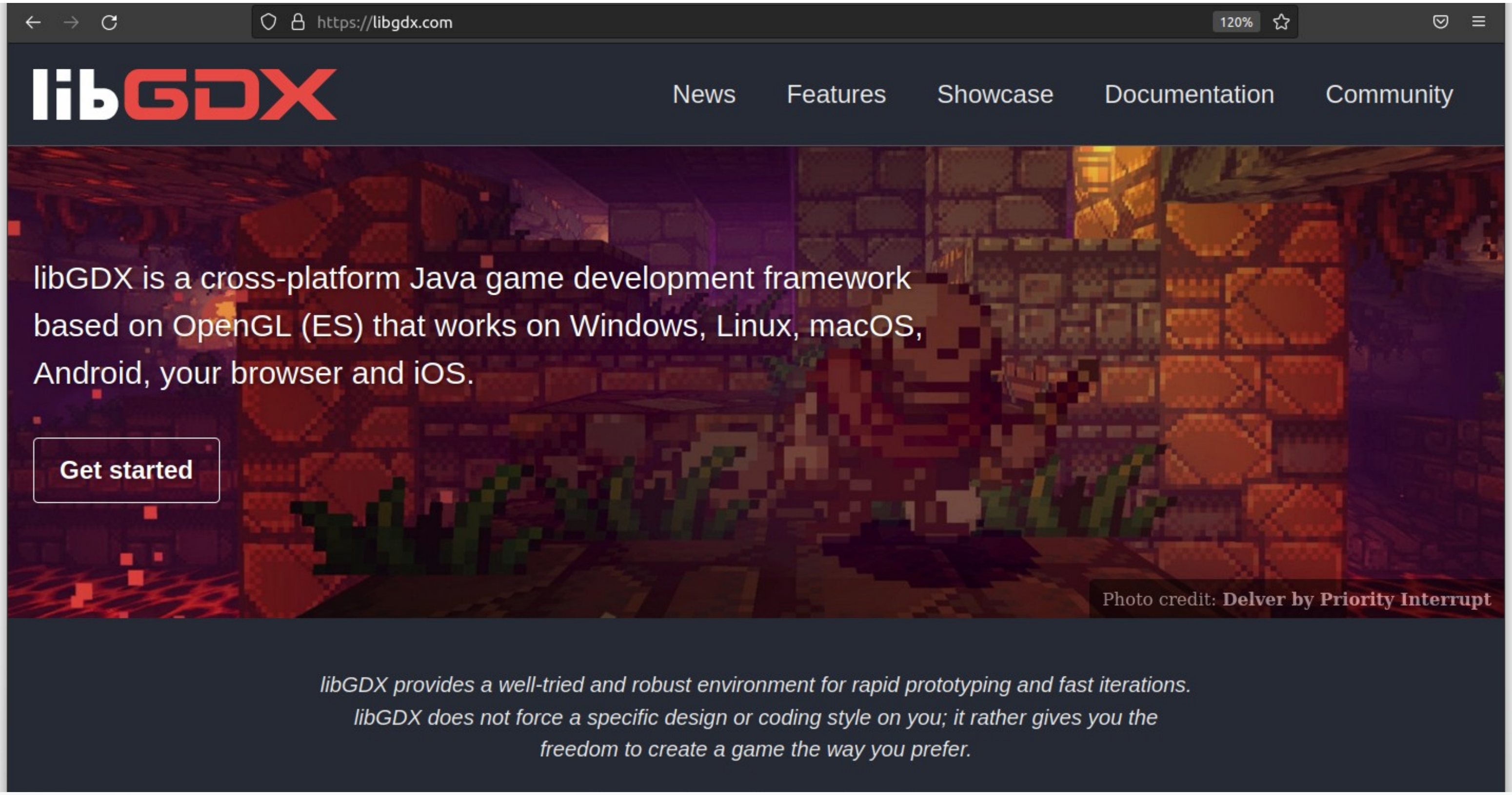
Launch succeeded (moments ago)

It’s running!



3) libGDX

What is libgdx?



libGDX is a cross-platform Java game development framework based on OpenGL (ES) that works on Windows, Linux, macOS, Android, your browser and iOS.

Photo credit: **Delver by Priority Interrupt**

libGDX provides a well-tried and robust environment for rapid prototyping and fast iterations. libGDX does not force a specific design or coding style on you; it rather gives you the freedom to create a game the way you prefer.

Why libgdx?

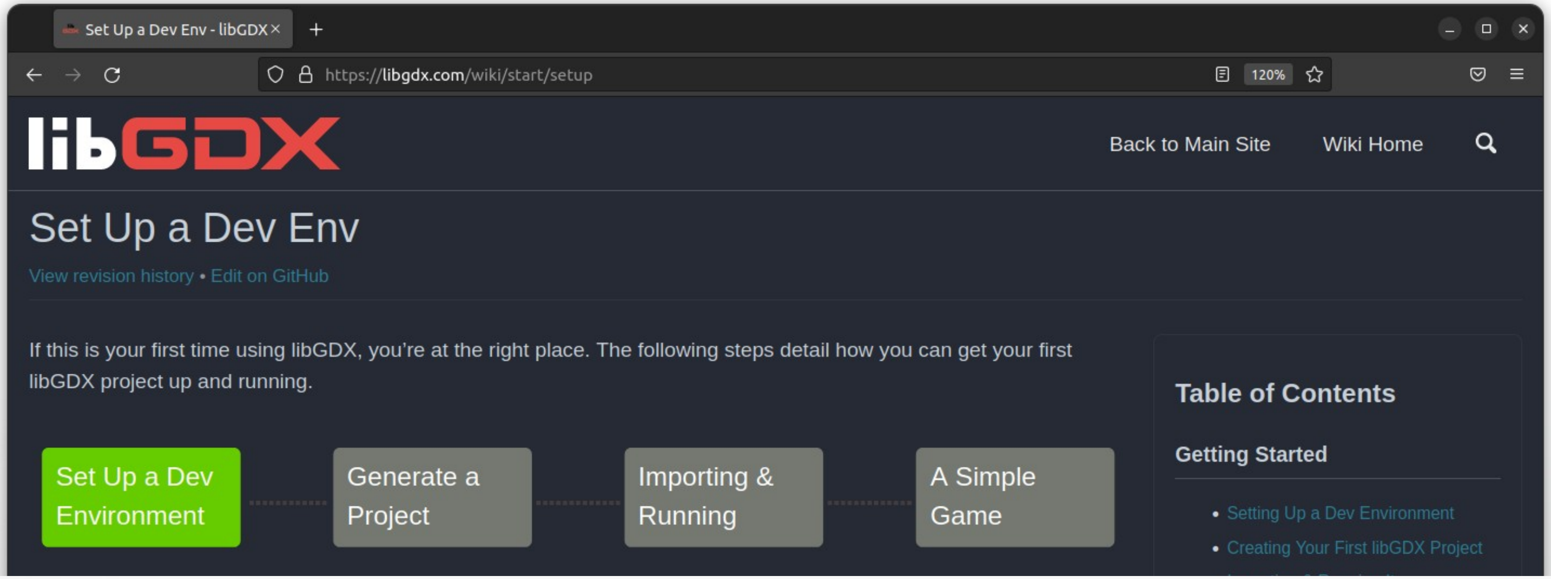
How many other cross-platform APIs also support 2D and 3D graphics?

- Dart + Flutter comes close but only has 2D graphics.
    - “Can I build 3D (OpenGL) apps with Flutter? Today we don’t support for 3D via OpenGL ES or similar. We have long-term plans to expose an optimized 3D API, but right now we’re focused on 2D.” - https://docs.flutter.dev/resources/faq
- Note: OpenGL has been around since 1992. Vulkan (glNext) has been around since 2016.

libgdx install

Follow these steps: https://libgdx.com/wiki/start/setup.

- You already did the first step below.



Set Up a Dev Env - libGDX

https://libgdx.com/wiki/start/setup

Set Up a Dev Env

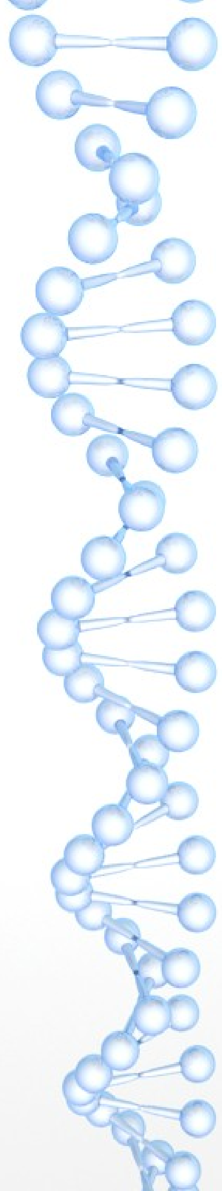
View revision history • Edit on GitHub

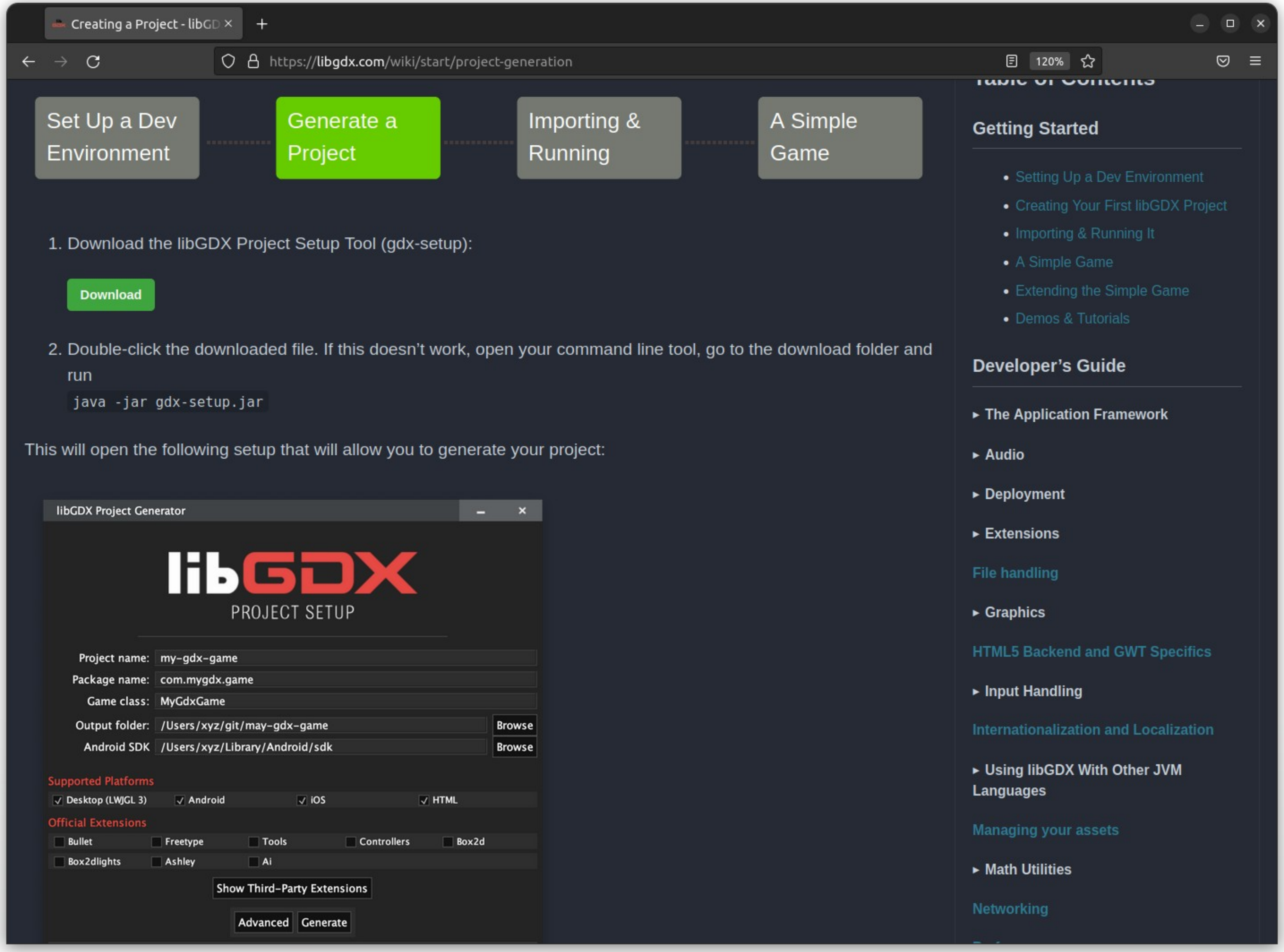
If this is your first time using libGDX, you're at the right place. The following steps detail how you can get your first libGDX project up and running.

Table of Contents

Getting Started

- Setting Up a Dev Environment
- Creating Your First libGDX Project





Creating a Project - libGD

https://libgdx.com/wiki/start/project-generation

Set Up a Dev Environment | Generate a Project | Importing & Running | A Simple Game

1. Download the libGDX Project Setup Tool (gdx-setup):

Download

2. Double-click the downloaded file. If this doesn't work, open your command line tool, go to the download folder and run

```
java -jar gdx-setup.jar
```

This will open the following setup that will allow you to generate your project:

libGDX Project Generator

libGDX PROJECT SETUP

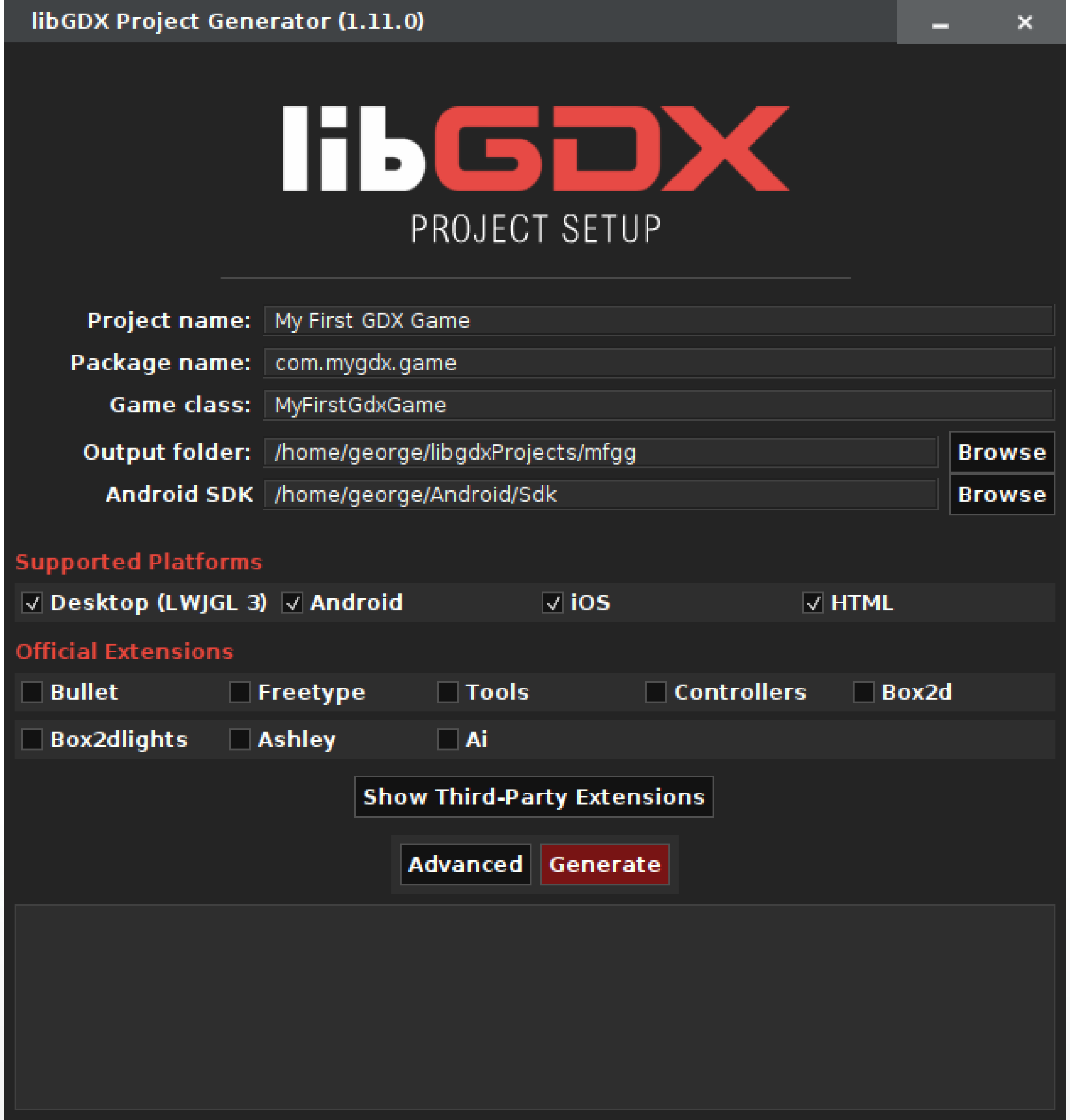
Project name: my-gdx-game
Package name: com.mygdx.game
Game class: MyGdxGame
Output folder: /Users/xyz/git/may-gdx-game Browse
Android SDK /Users/xyz/Library/Android/sdk Browse

Supported Platforms
[✓] Desktop (LWJGL 3) [✓] Android [✓] iOS [✓] HTML

Official Extensions
[ ] Bullet [ ] Freetype [ ] Tools [ ] Controllers [ ] Box2d
[ ] Box2dlights [ ] Ashley [ ] Ai

Show Third-Party Extensions

Advanced Generate


libGDX Project Generator (1.11.0)

libGDX PROJECT SETUP

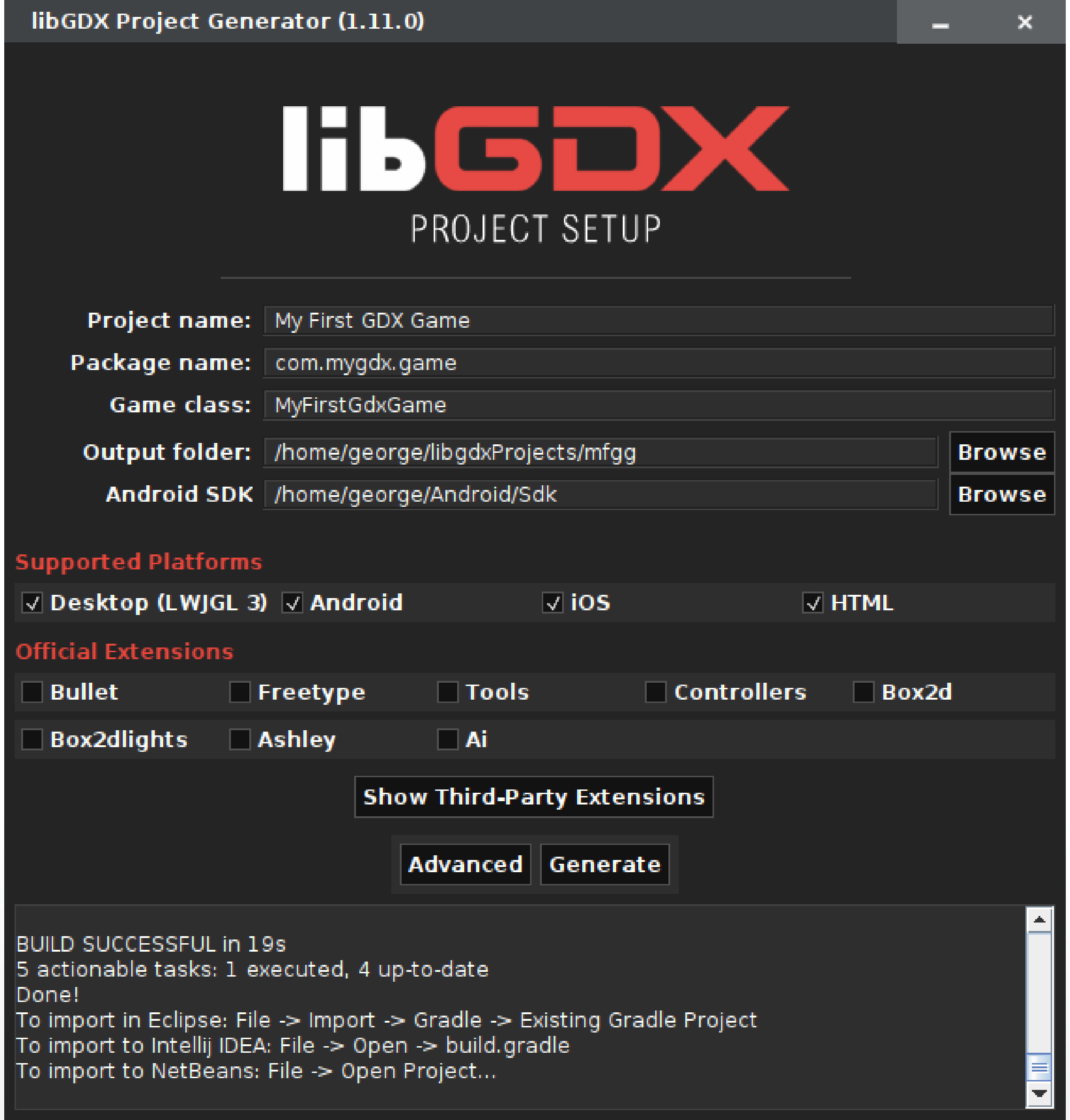
Project name: My First GDX Game
Package name: com.mygdx.game
Game class: MyFirstGdxGame
Output folder: /home/george/libgdxProjects/mfgg
Android SDK /home/george/Android/Sdk

Supported Platforms

✓ Desktop (LWJGL 3) ✓ Android ✓ iOS ✓ HTML

Official Extensions

☐ Bullet ☐ Freetype ☐ Tools ☐ Controllers ☐ Box2d
☐ Box2dlights ☐ Ashley ☐ Ai


libGDX Project Generator (1.11.0)
libGDX
PROJECT SETUP

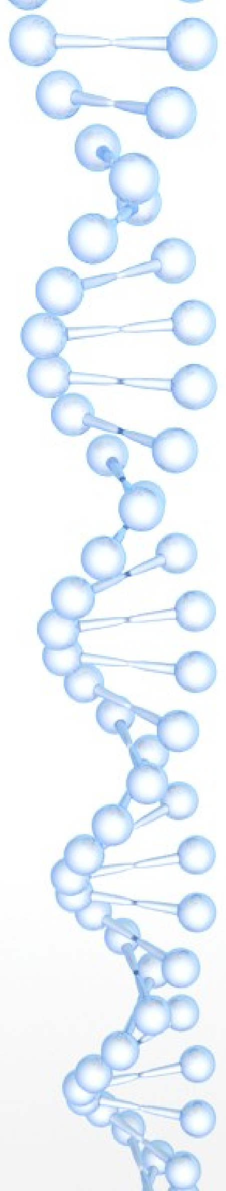
Project name: My First GDX Game
Package name: com.mygdx.game
Game class: MyFirstGdxGame
Output folder: /home/george/libgdxProjects/mfgg
Android SDK /home/george/Android/Sdk

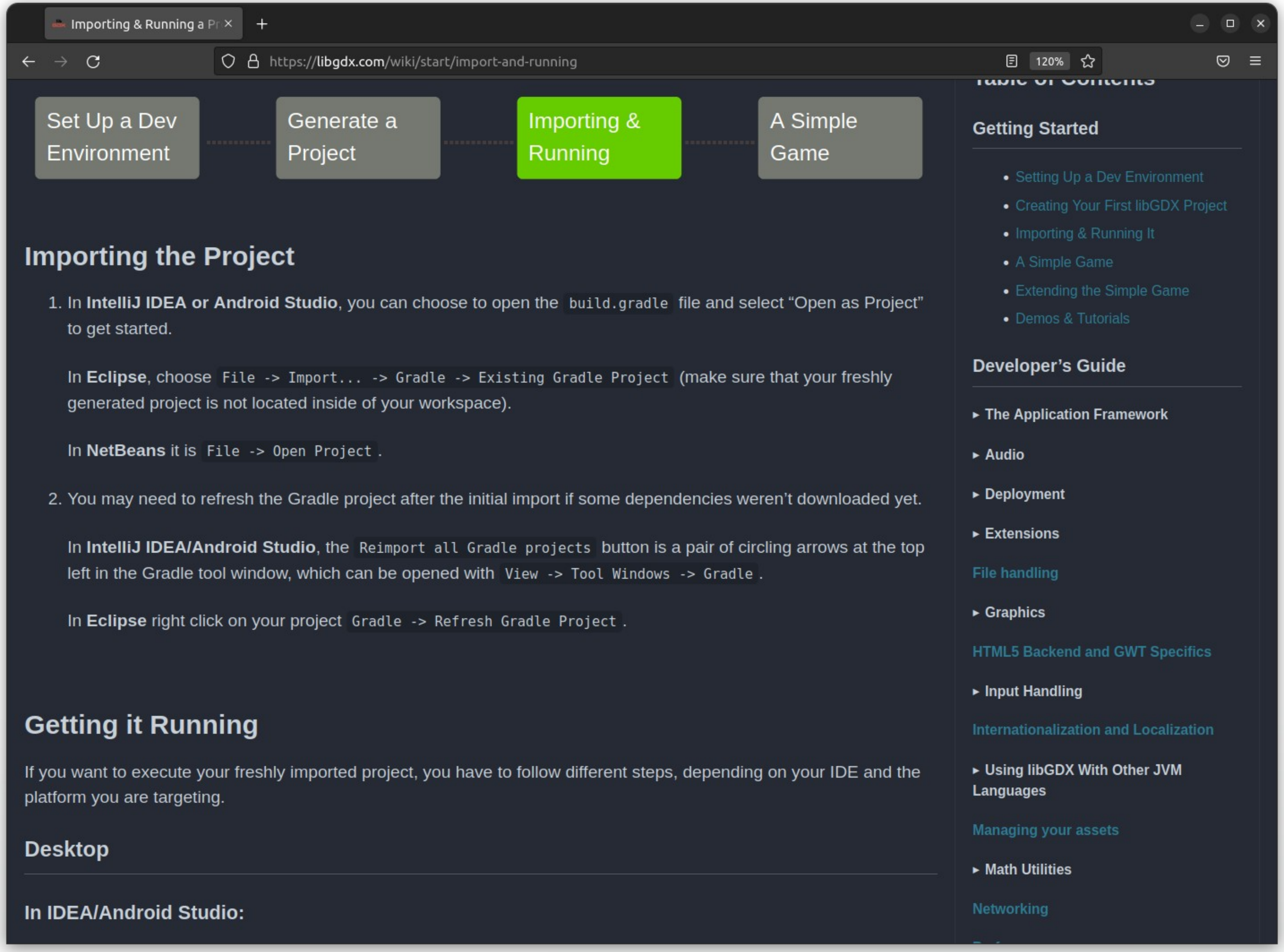
Supported Platforms
✓ Desktop (LWJGL 3) ✓ Android ✓ iOS ✓ HTML

Official Extensions
☐ Bullet ☐ Freetype ☐ Tools ☐ Controllers ☐ Box2d
☐ Box2dlights ☐ Ashley ☐ Ai

Browse
Browse
Show Third-Party Extensions
Advanced
Generate

```
BUILD SUCCESSFUL in 19s
5 actionable tasks: 1 executed, 4 up-to-date
Done!
To import in Eclipse: File -> Import -> Gradle -> Existing Gradle Project
To import to Intellij IDEA: File -> Open -> build.gradle
To import to NetBeans: File -> Open Project...
```





Set Up a Dev Environment

Generate a Project

Importing & Running

A Simple Game

Importing the Project

1. In **IntelliJ IDEA or Android Studio**, you can choose to open the `build.gradle` file and select "Open as Project" to get started.

   In **Eclipse**, choose `File -> Import... -> Gradle -> Existing Gradle Project` (make sure that your freshly generated project is not located inside of your workspace).

   In **NetBeans** it is `File -> Open Project`.

2. You may need to refresh the Gradle project after the initial import if some dependencies weren't downloaded yet.

   In **IntelliJ IDEA/Android Studio**, the `Reimport all Gradle projects` button is a pair of circling arrows at the top left in the Gradle tool window, which can be opened with `View -> Tool Windows -> Gradle`.

   In **Eclipse** right click on your project `Gradle -> Refresh Gradle Project`.

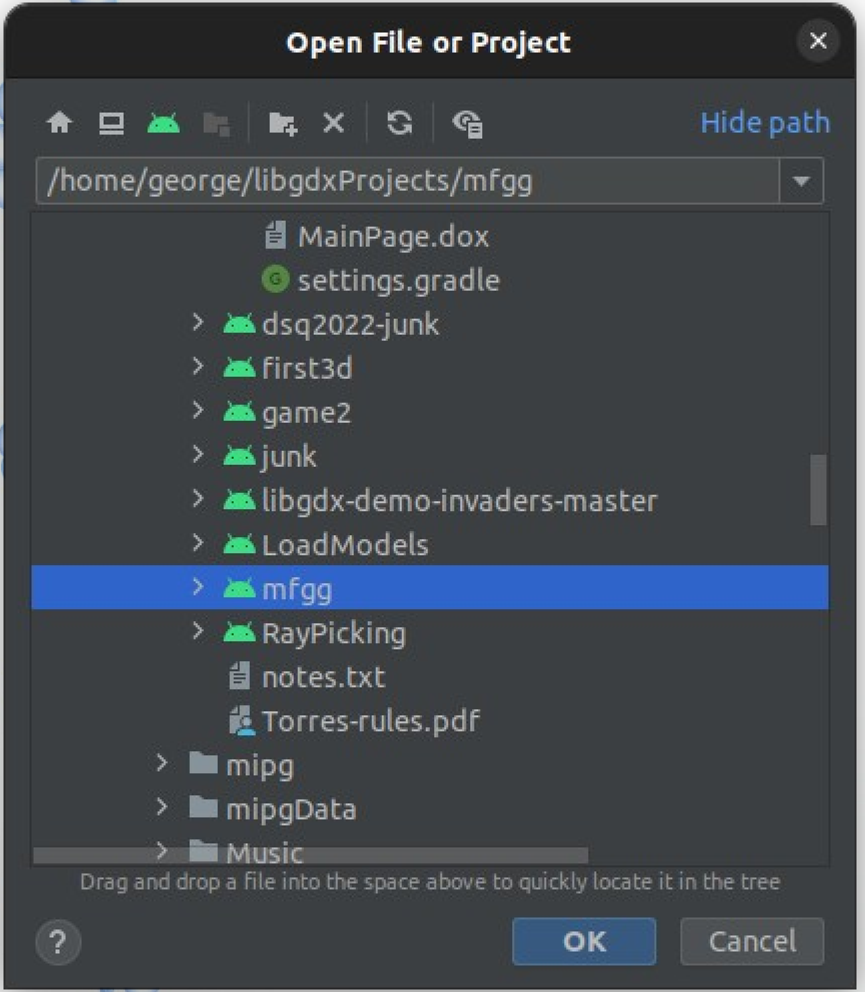
Getting it Running

If you want to execute your freshly imported project, you have to follow different steps, depending on your IDE and the platform you are targeting.

Desktop

In IDEA/Android Studio:

Run AS and open it.

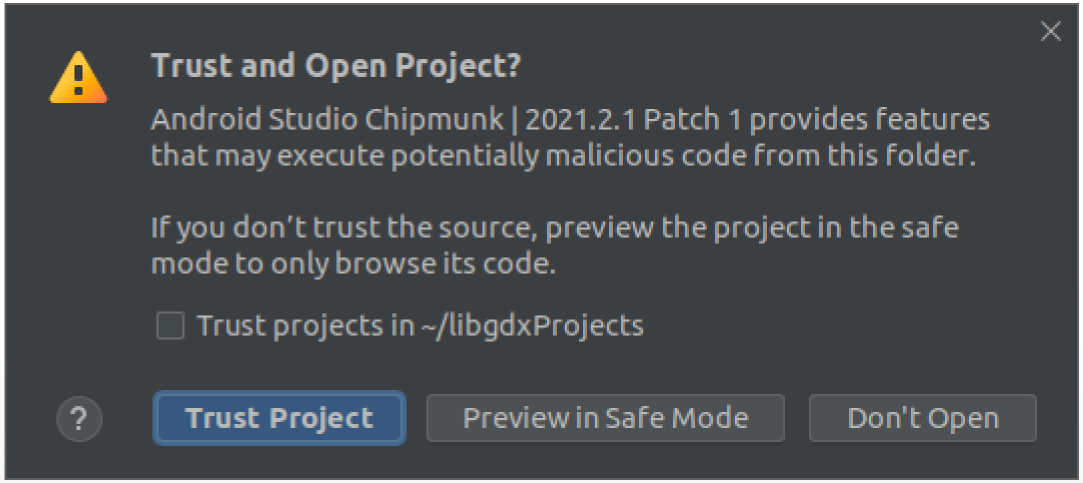

Open File or Project

Hide path

/home/george/libgdxProjects/mfgg

MainPage.dox
settings.gradle
dsq2022-junk
first3d
game2
junk
libgdx-demo-invaders-master
LoadModels
mfgg
RayPicking
notes.txt
Torres-rules.pdf
mipg
mipgData
Music

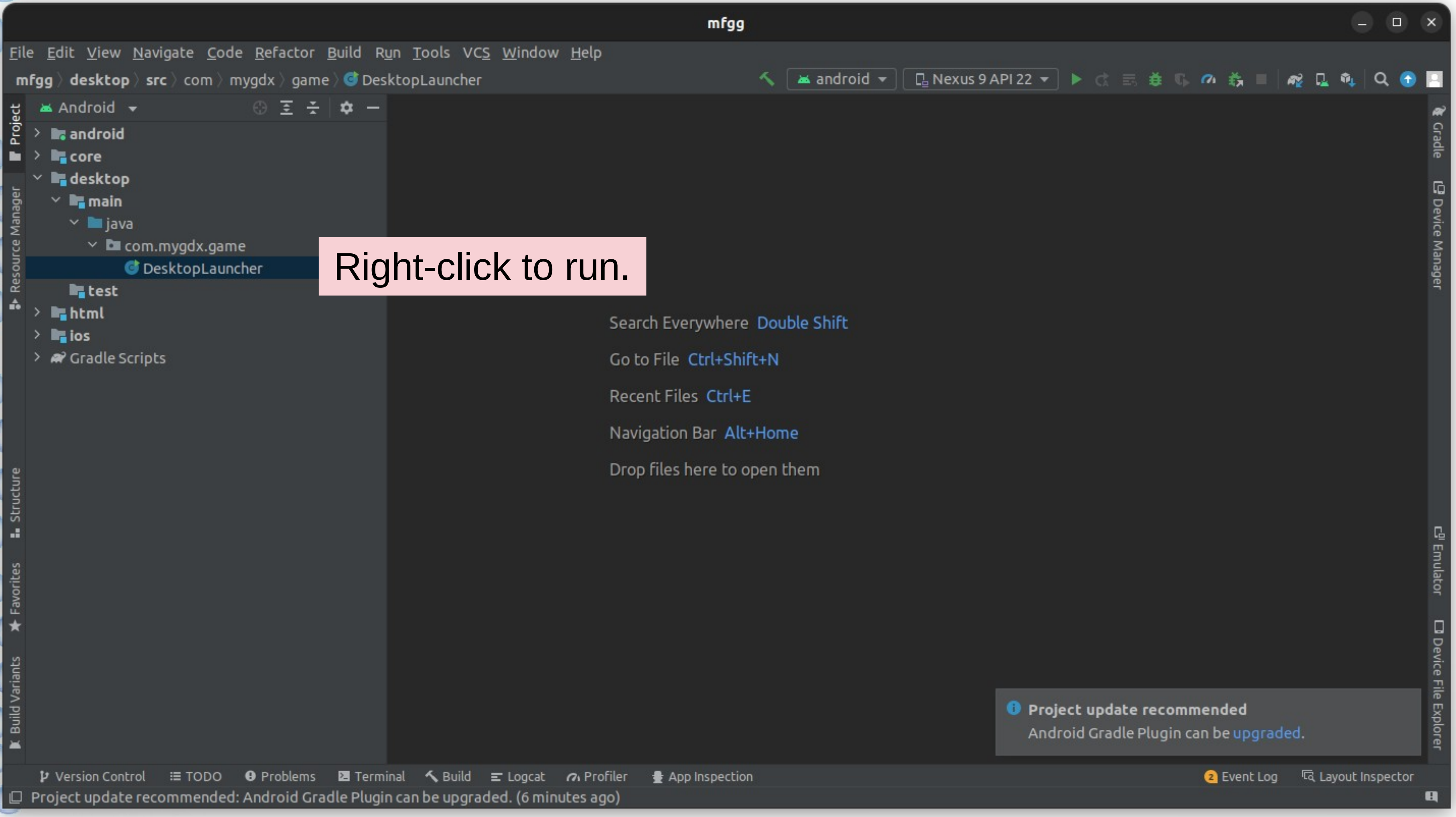
Drag and drop a file into the space above to quickly locate it in the tree


Trust and Open Project?

Android Studio Chipmunk | 2021.2.1 Patch 1 provides features that may execute potentially malicious code from this folder.

If you don't trust the source, preview the project in the safe mode to only browse its code.

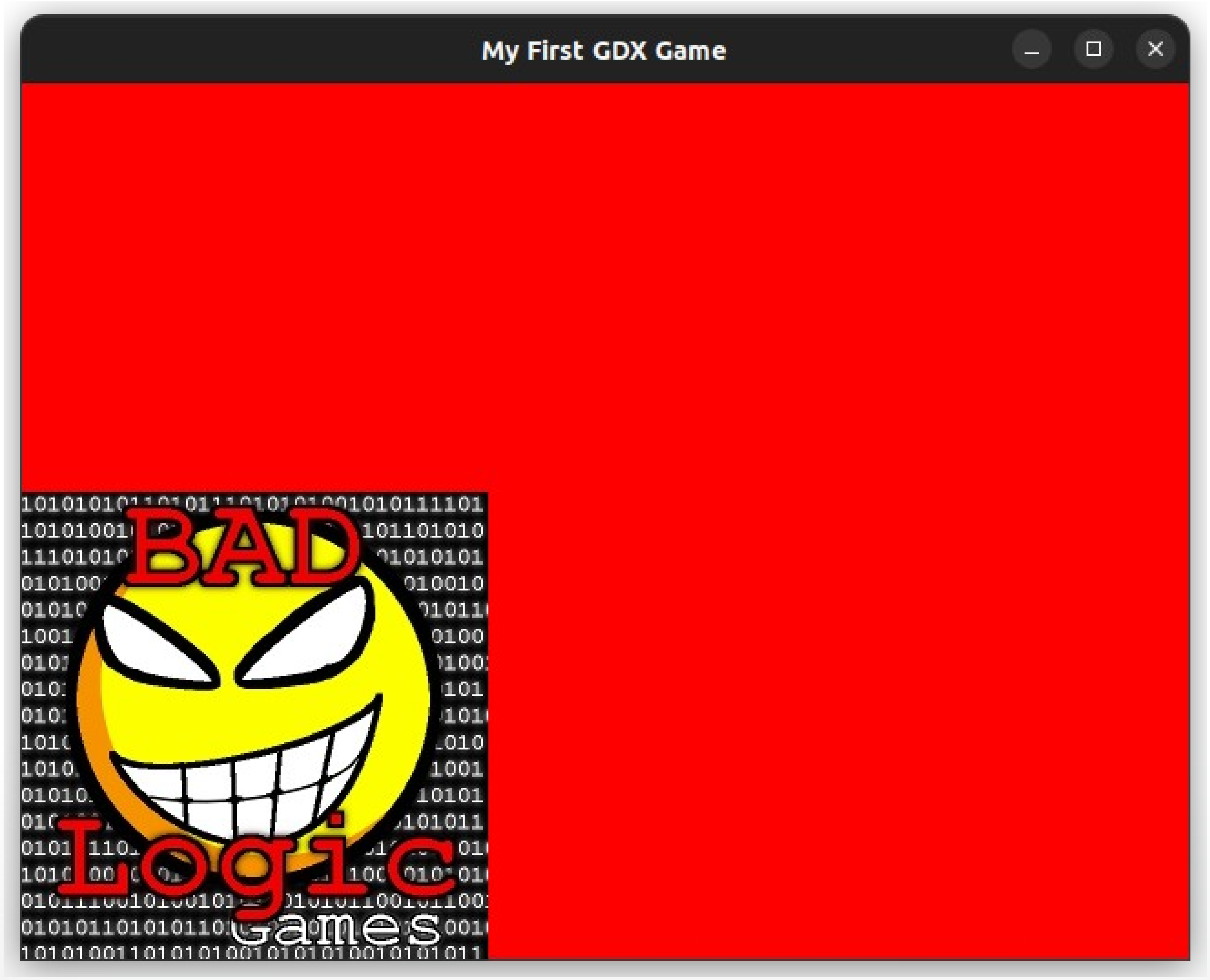
☐ Trust projects in ~/libgdxProjects

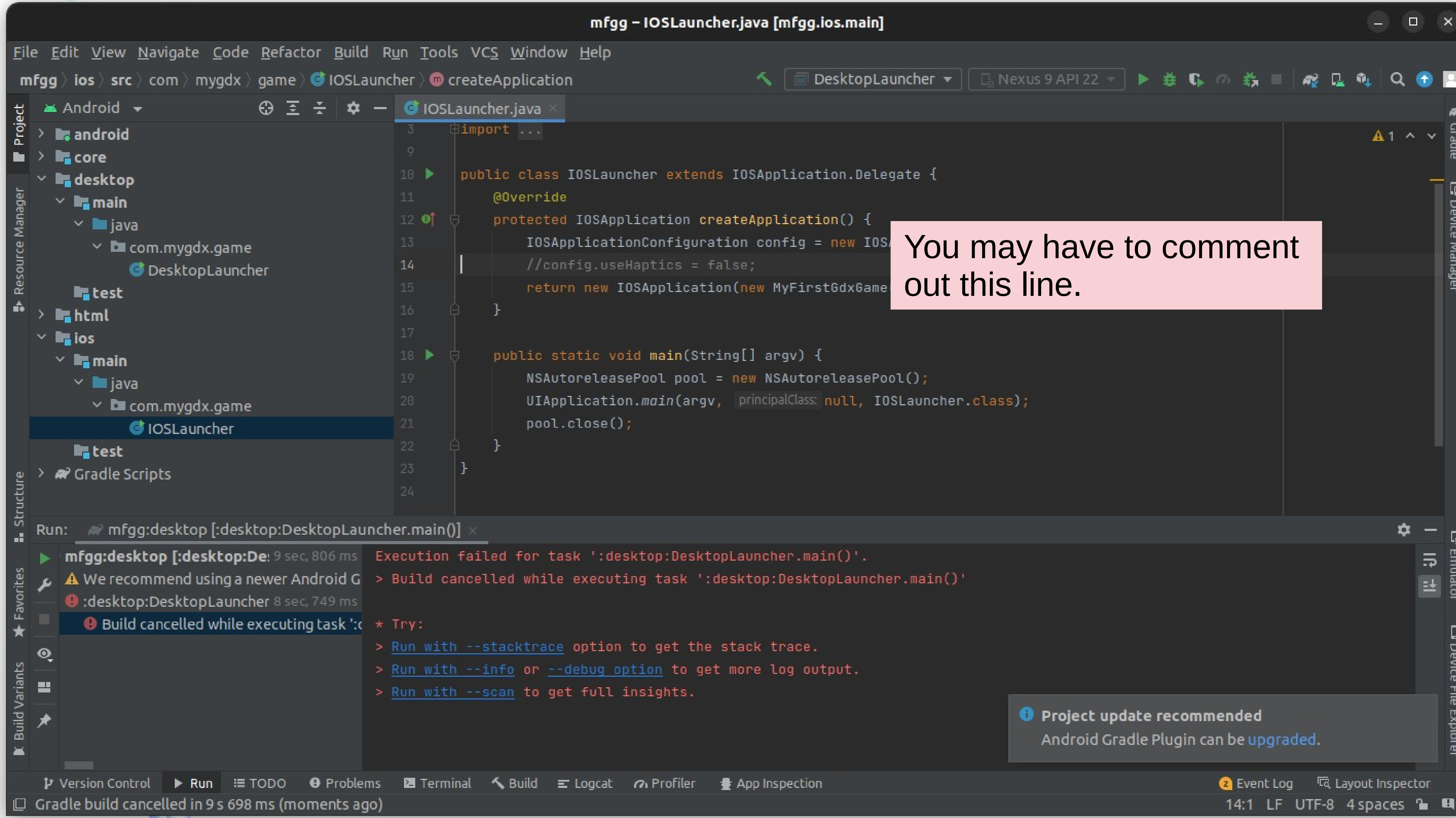


Right-click to run.

Project update recommended
Android Gradle Plugin can be upgraded.

Project update recommended: Android Gradle Plugin can be upgraded. (6 minutes ago)


My First GDX Game
BAD
Logic
Games



You may have to comment out this line.

mfgg – IOSLauncher.java [mfgg.ios.main]

```
import ...

public class IOSLauncher extends IOSApplication.Delegate {
    @Override
    protected IOSApplication createApplication() {
        IOSApplicationConfiguration config = new IOS
        //config.useHaptics = false;
        return new IOSApplication(new MyFirstGdxGame
    }

    public static void main(String[] argv) {
        NSAutoreleasePool pool = new NSAutoreleasePool();
        UIApplication.main(argv, principalClass: null, IOSLauncher.class);
        pool.close();
    }
}
```

Run: mfgg:desktop [:desktop:DesktopLauncher.main()]

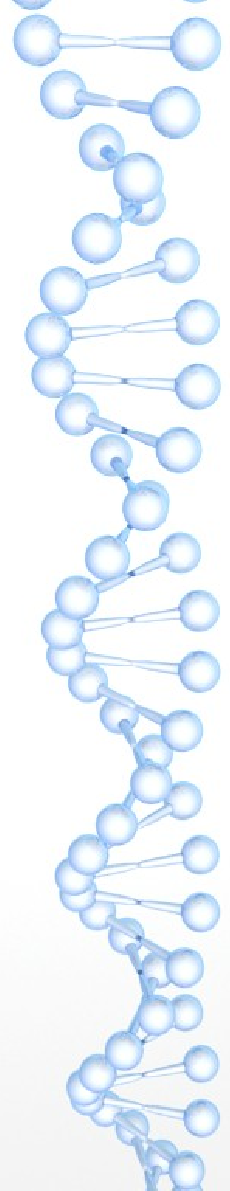
```
mfgg:desktop [:desktop:De: 9 sec, 806 ms
We recommend using a newer Android G
:desktop:DesktopLauncher 8 sec, 749 ms
Build cancelled while executing task ':
```

```
Execution failed for task ':desktop:DesktopLauncher.main()'.
> Build cancelled while executing task ':desktop:DesktopLauncher.main()'

* Try:
> Run with --stacktrace option to get the stack trace.
> Run with --info or --debug option to get more log output.
> Run with --scan to get full insights.
```

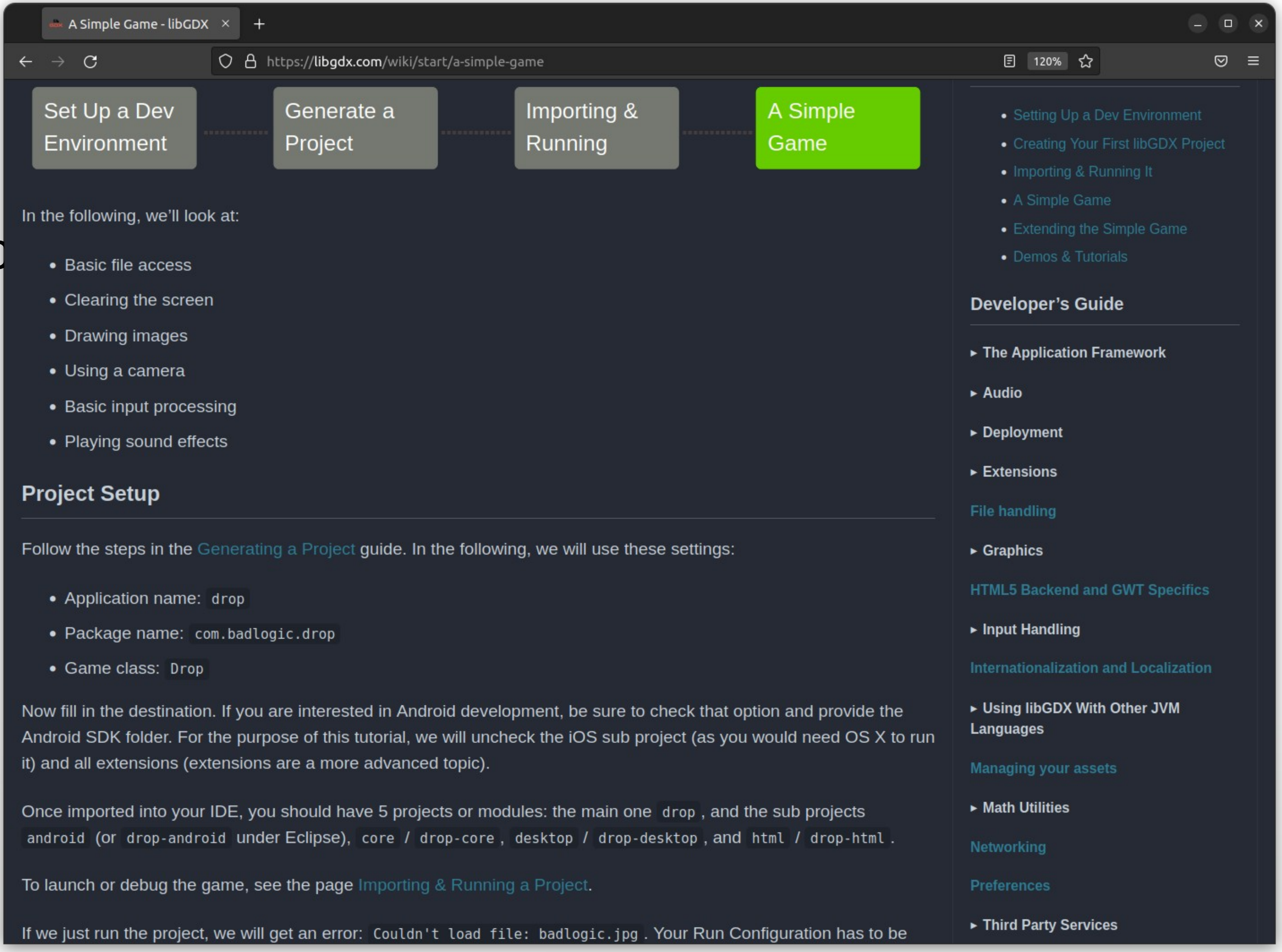
Project update recommended
Android Gradle Plugin can be upgraded.

Gradle build cancelled in 9 s 698 ms (moments ago)

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
mfgg ios src com mygdx game IOSLauncher createApplication
DesktopLauncher
Nexus 9 API 22
Android
android
core
desktop
main
java
com.mygdx.game
DesktopLauncher
test
html
ios
main
java
com.mygdx.game
IOSLauncher
test
Gradle Scripts
IOSLauncher.java
Version Control Run TODO Problems Terminal Build Logcat Profiler App Inspection Event Log Layout Inspector
14:1 LF UTF-8 4 spaces



Optic



In the following, we'll look at:

- Basic file access
- Clearing the screen
- Drawing images
- Using a camera
- Basic input processing
- Playing sound effects

Project Setup

Follow the steps in the Generating a Project guide. In the following, we will use these settings:

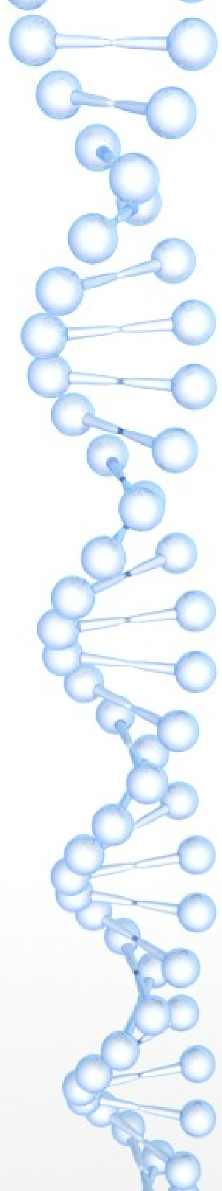
- Application name: `drop`
- Package name: `com.badlogic.drop`
- Game class: `Drop`

Now fill in the destination. If you are interested in Android development, be sure to check that option and provide the Android SDK folder. For the purpose of this tutorial, we will uncheck the iOS sub project (as you would need OS X to run it) and all extensions (extensions are a more advanced topic).

Once imported into your IDE, you should have 5 projects or modules: the main one `drop`, and the sub projects `android` (or `drop-android` under Eclipse), `core` / `drop-core`, `desktop` / `drop-desktop`, and `html` / `drop-html`.

To launch or debug the game, see the page Importing & Running a Project.

If we just run the project, we will get an error: `Couldn't load file: badlogic.jpg`. Your Run Configuration has to be



4) assignment starter code

To be continued ...