RA en Kotlin con ARCore

KOTLIN / EVERYWHERE - Chimbote 2019

Ing. Cip. Eder Alberto Albino Huertas

GDG Chimbote

+

ARCore

GDG Chimbote

Kotlin && Android

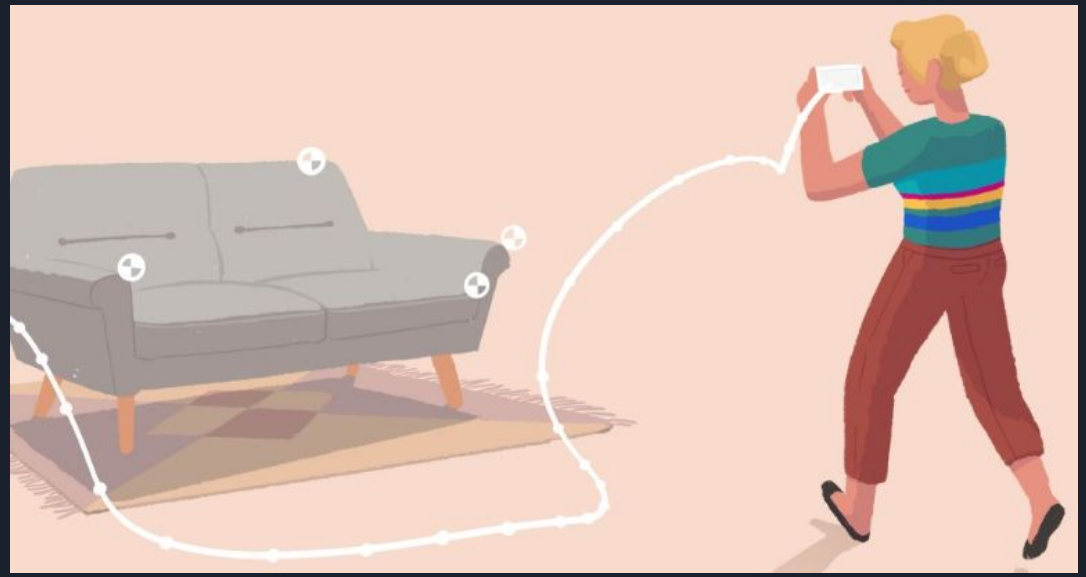
- Compatibilidad : Kotlin es totalmente compatible con JDK 6.
- Rendimiento : Estructura de código de bytes muy similar a JAVA.
- Interoperabilidad : Permite utilizar todas las bibliotecas de Android existentes.
- Curva de aprendizaje : Rápida.

GDG Chimbote

ARCore

Kit de desarrollo de software (SDK) de Google para desarrollar APPs de RA en Android, Unity, Unreal y IOS.

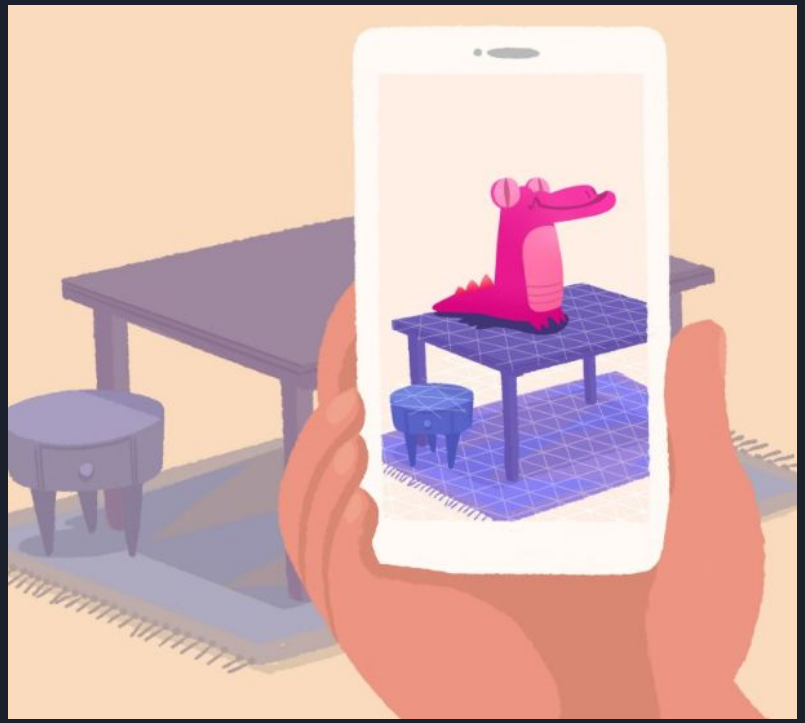
01 El seguimiento del movimiento.



GDG Chimbote

ARCore

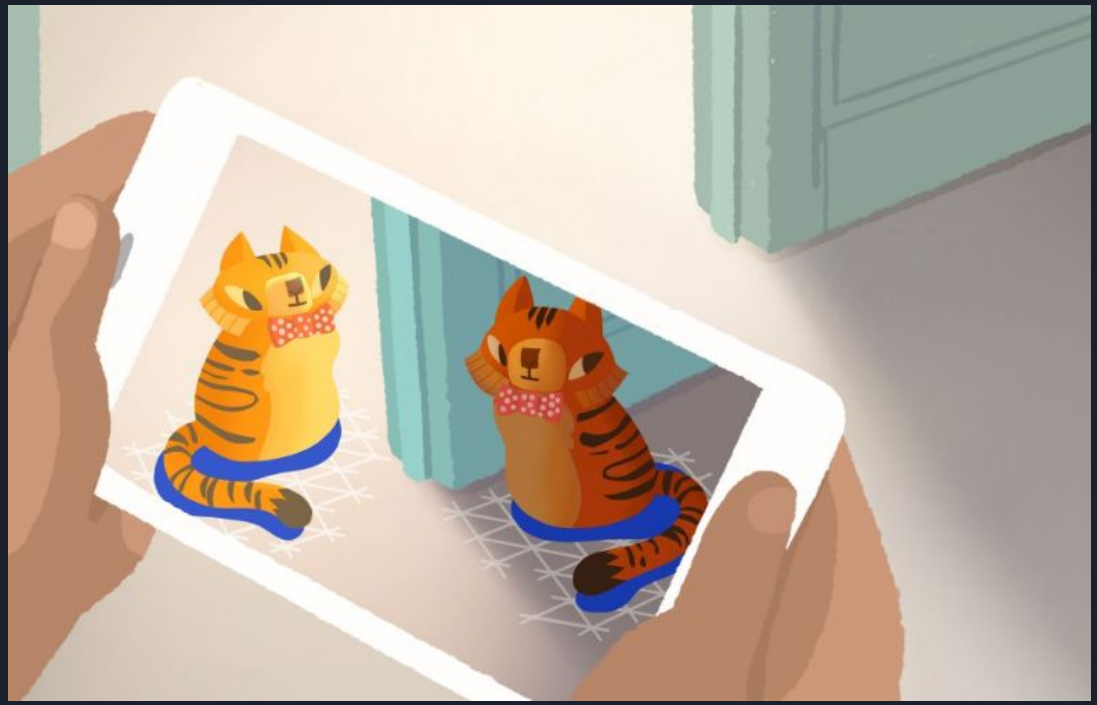
02 Entendimiento del ambiente



GDG Chimbote

ARCore

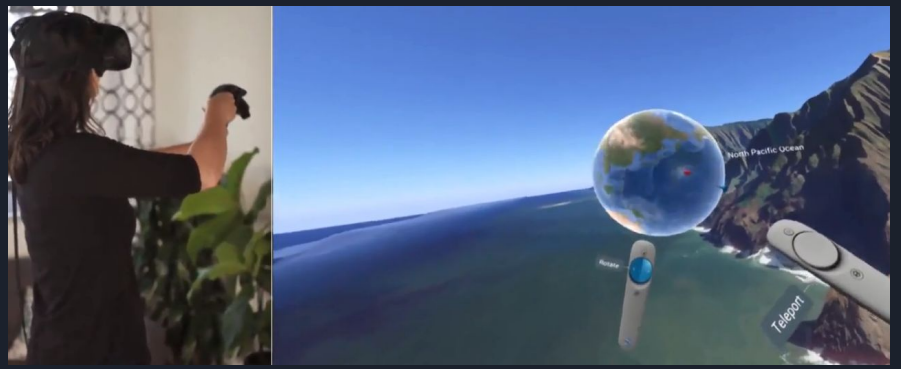
03 Estimación de luz



GDG Chimbote

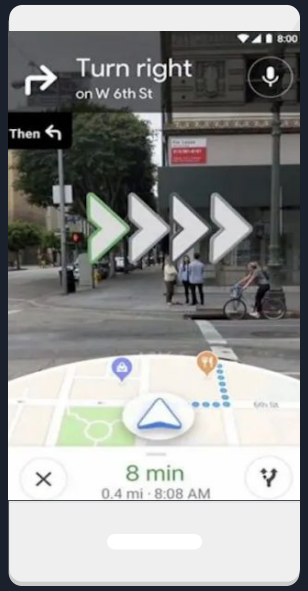
RA != RV

!=



GDG Chimbote

Aplicaciones de RA



GDG Chimbote

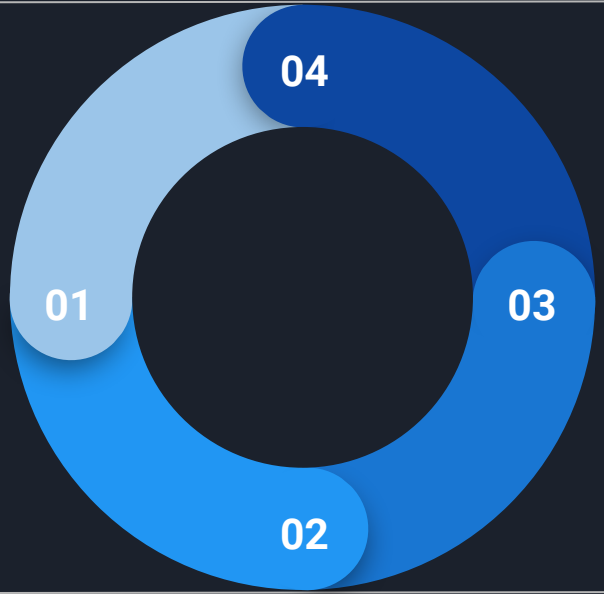
¿Por dónde empezamos?

Preparar dispositivo

Físico o emulador

Configurar IDE

Configurar Android Studio



Probar

Imprimir el target de ser necesarios

Programar

Tener listo los objetos y targets elegidos

GDG Chimbote

Preparar dispositivo

VERSIÓN ANDROID >= 7 Nougat

FÍSICO :

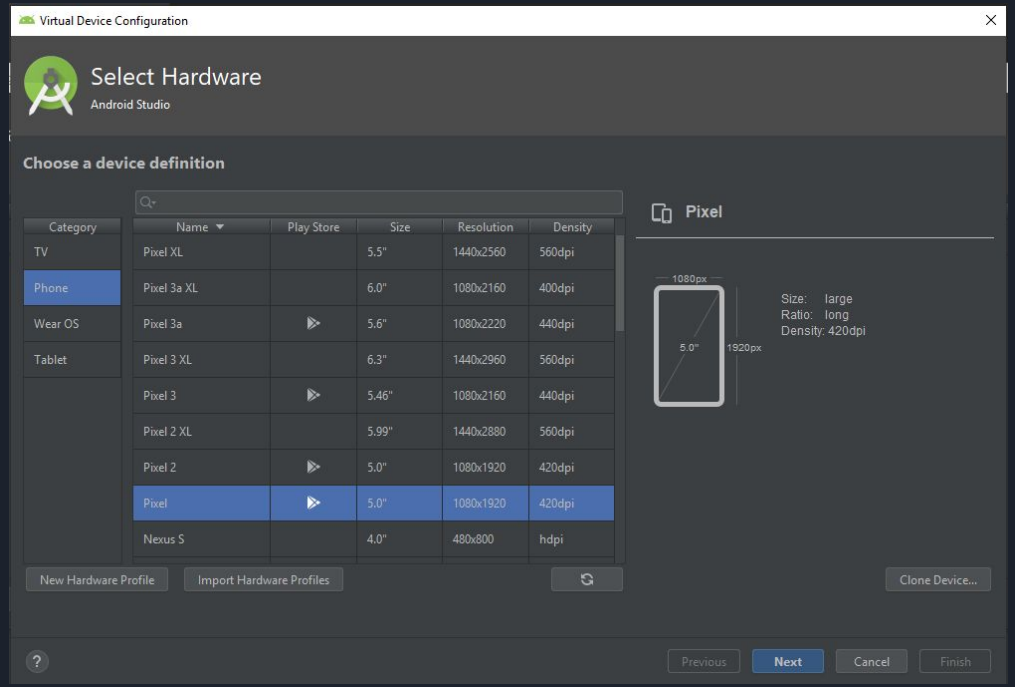
- Verificar compatibilidad.

- Instalar ARCore-Services.

GDG Chimbote

Preparar dispositivo

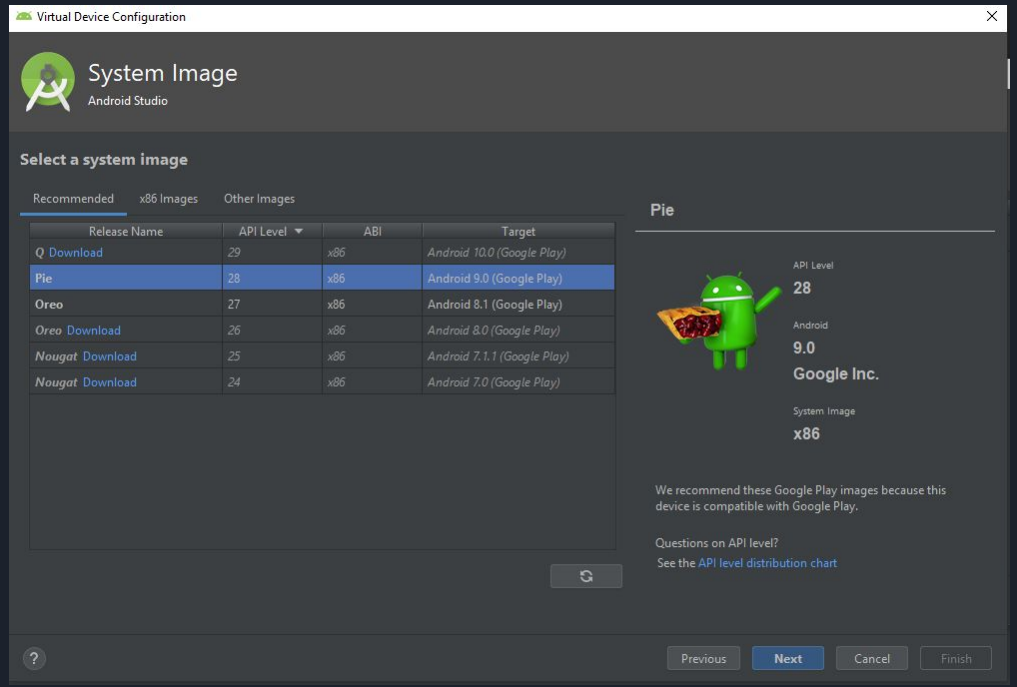
EMULADOR: Dispositivo



GDG Chimbote

Preparar dispositivo

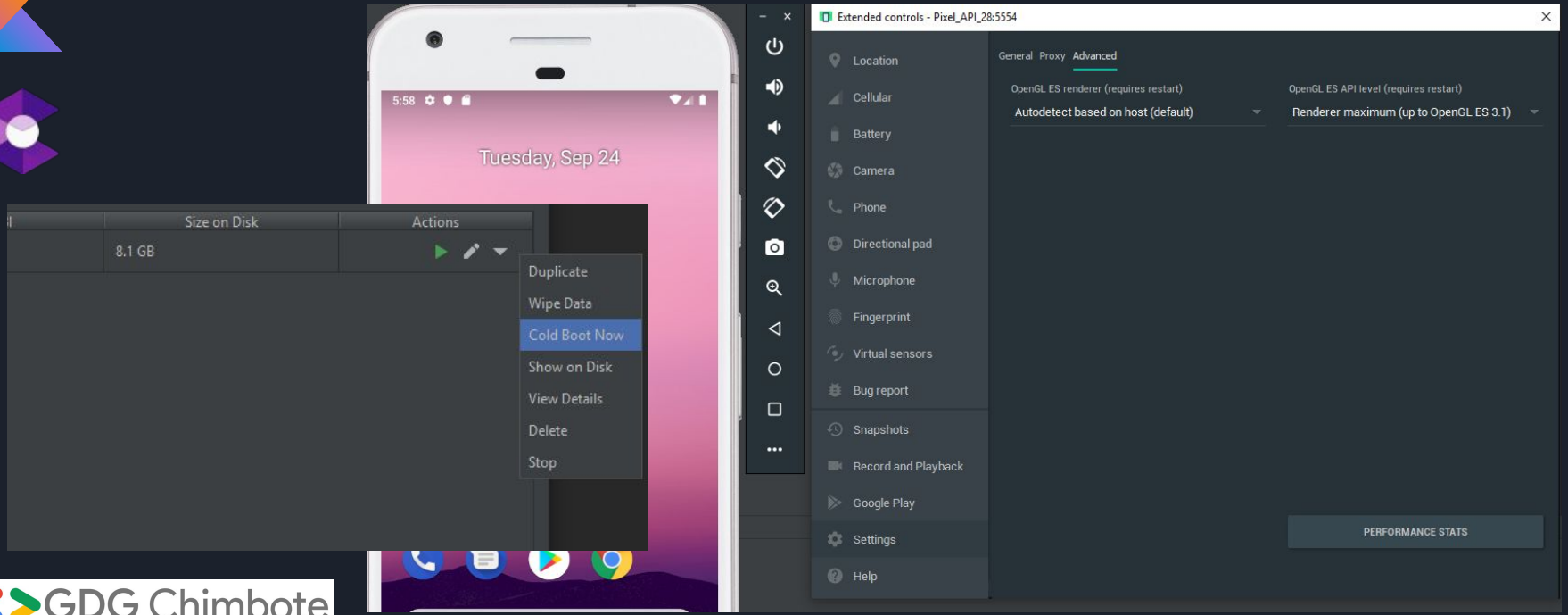
EMULADOR: SO



GDG Chimbote

Preparar dispositivo

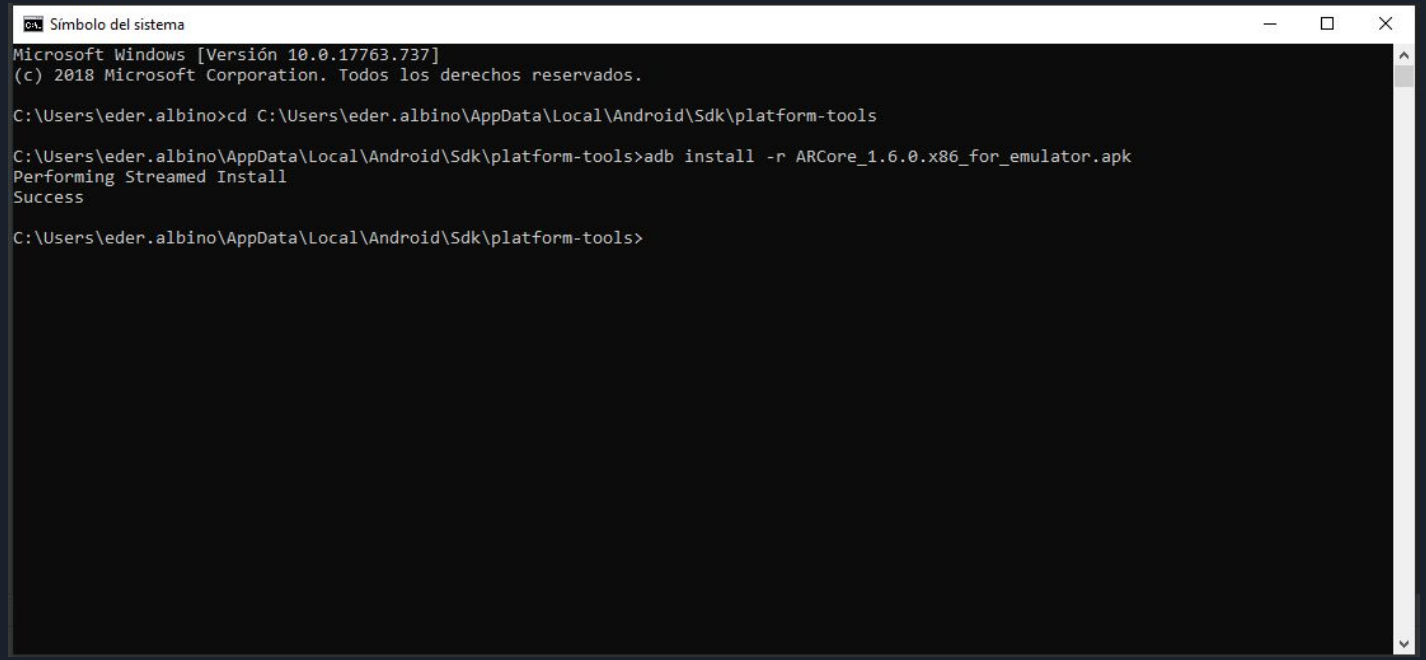
EMULADOR: OpenGL



GDG Chimbote

Preparar dispositivo

EMULADOR: Instalar ARCore-Services

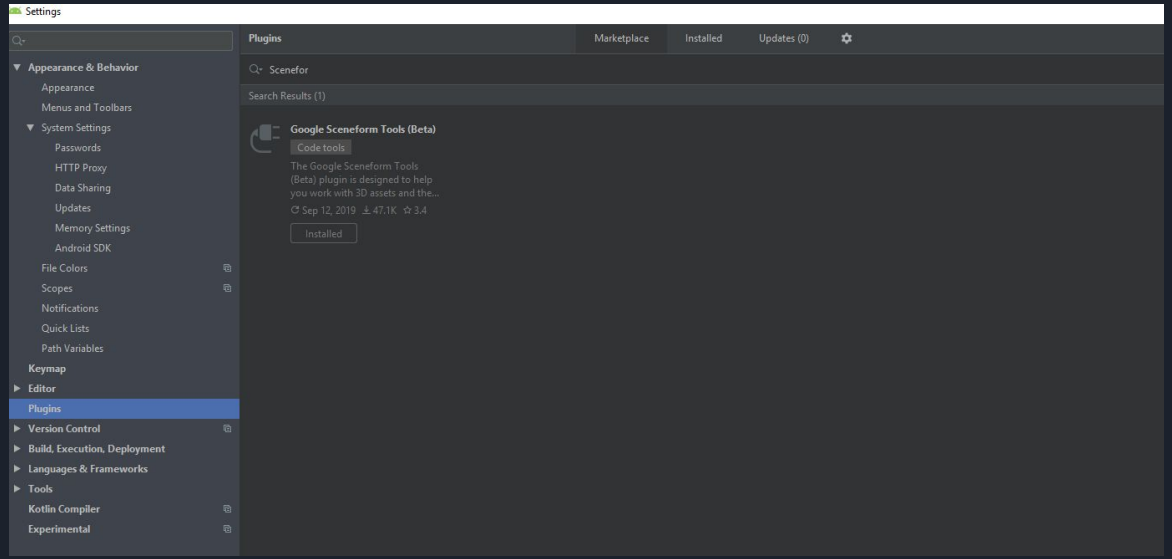


GDG Chimbote

Configurar IDE

Actualizar Android Studio >= 3.1

Instalar Google Sceneform tools(Beta)



GDG Chimbote

Programar

Permisos para usar Cámara

MainActivity.kt | AndroidManifest.xml | app | demoar-taller

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.demoar_taller">

    <uses-permission android:name="android.permission.CAMERA" />
    <uses-feature android:name="android.hardware.camera.ar" android:required="true"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="demoar-taller"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

        <meta-data android:name="com.google.ar.core" android:value="required" />

        <activity android:name=".MainActivity"...>
    </application>

</manifest>
```

AndroidManifest.xm

GDG Chimbote

Programar

Configuración de compilación y agregar dependencia de RA

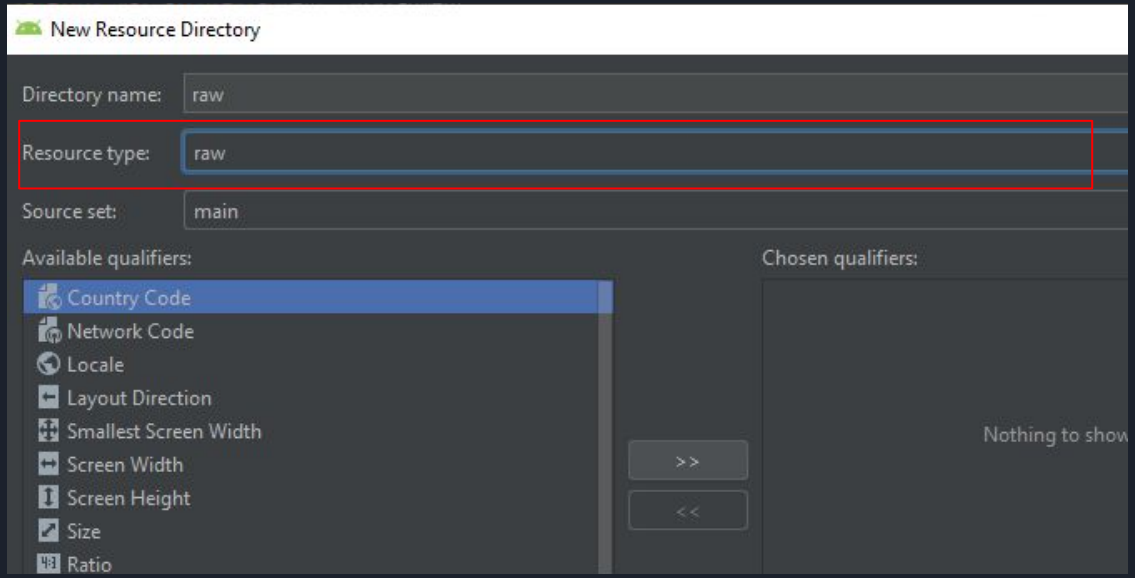
build.gradle (Module)

```
    buildToolsVersion "29.0.2"
    defaultConfig {
        applicationId "com.example.demoar_taller"
        minSdkVersion 27
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation"org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation 'androidx.core:core-ktx:1.1.0'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test:runner:1.2.0'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'

    implementation "com.google.ar.sceneform.ux:sceneform-ux:1.6.0"
}
```

GDG Chimbote

Programar

Crear un directorio de recurso de tipo “raw”



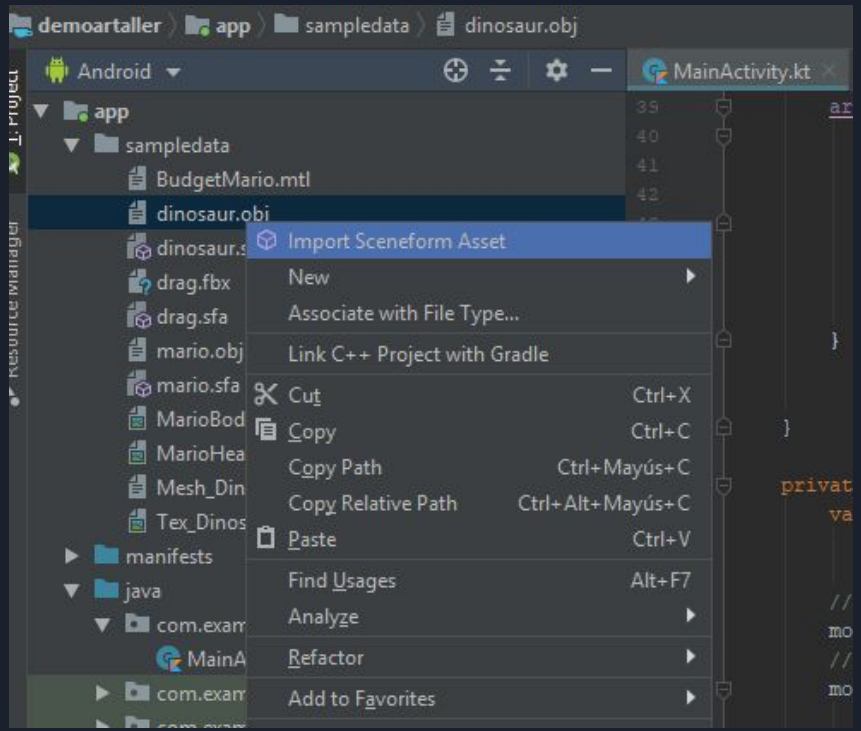
GDG Chimbote

Programar

- Crear directorio “sampledata” en app.

- Agregar Objetos (*.obj, *.fbx, *.gltf).

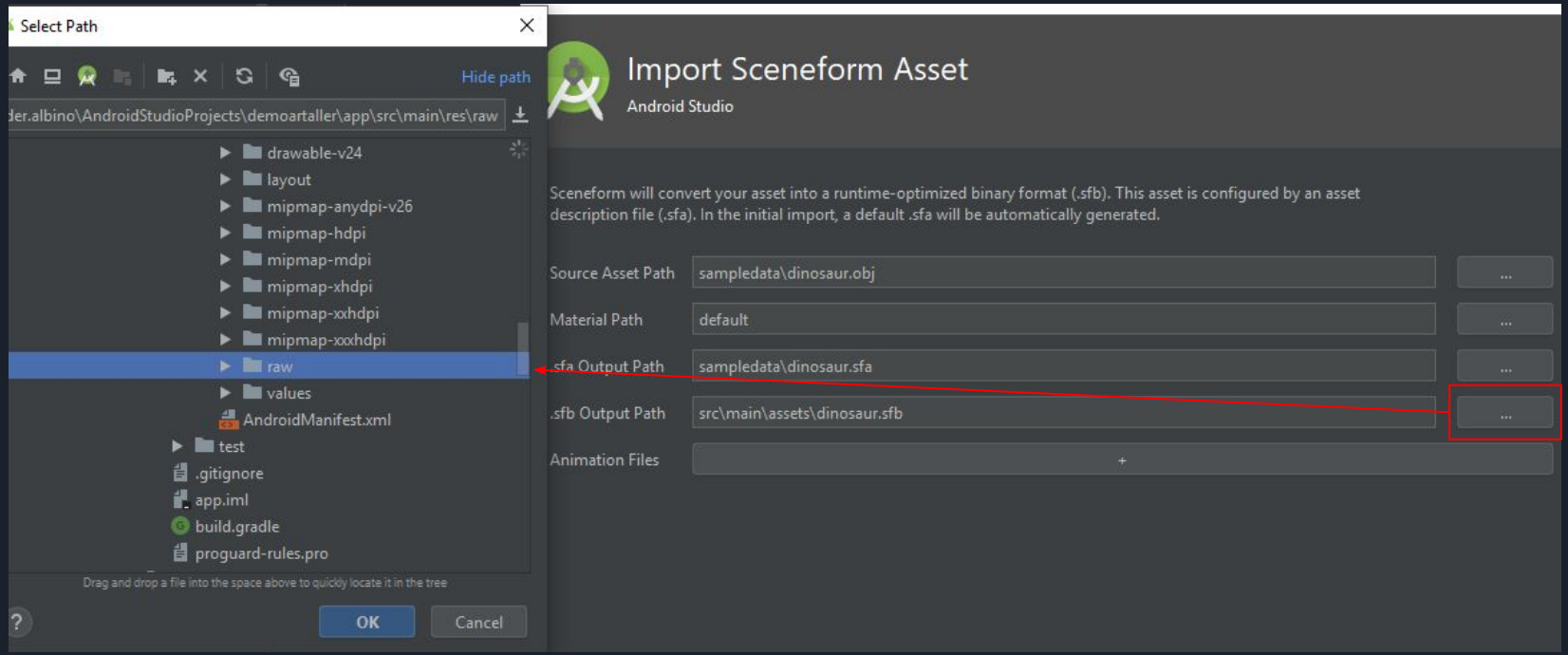
- Importar el Objeto al directorio de recurso “raw”



GDG Chimbote

Programar

Importar el Objeto al directorio de recurso “raw”



GDG Chimbote

Programar

1.- Instanciar el Objeto.

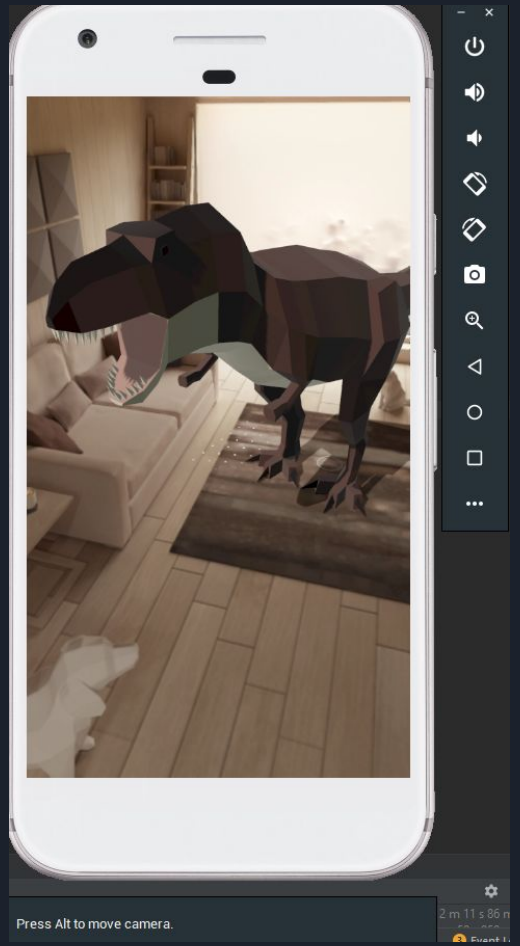
2.- Agregar el objeto instanciado al escenario.

```
private var TAG:String?="MiClassPrincipal"
private var MIN_OPENGL_VERSION:Double=3.0
private lateinit var arCoreFragment: ArFragment

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    if (!verificarDispositivo( activity: this)) {
        return
    }
    setContentView(R.layout.activity_main)

    arCoreFragment = supportFragmentManager.findFragmentById(R.id.main_fragment) as ArFragment
    arCoreFragment.setOnTapArPlaneListener{ hitResult, plane, MotionEvent ->
        if (plane.type != Plane.Type.HORIZONTAL_UPWARD_FACING) {
            //return for the callback
            return@setOnTapArPlaneListener
        }
        val anchor = hitResult.createAnchor()
        establecerObjeto(arCoreFragment,anchor)
    }
}

private fun establecerObjeto(fragment: ArFragment,anchor: Anchor){
    val modelRenderable = ModelRenderable.builder()
        .setSource(arCoreFragment.requireContext(), R.raw.dinosaur)
        .build()
    //when the model render is build add node to scene
    modelRenderable.thenAccept { renderableObject -> agregarNodo(fragment, anchor, renderableObject) }
    //handle error
    modelRenderable.exceptionally { it: Throwable!
        val toast = Toast.makeText(applicationContext, text: "Error", Toast.LENGTH_SHORT)
        toast.show()
        null ^exceptionally
    }
}

private fun agregarNodo(fragment: ArFragment, anchor: Anchor, renderableObject: Renderable) {
    val anchorNode = AnchorNode(anchor)
    val transformableNode = TransformableNode(fragment.transformationSystem)
    transformableNode.renderable = renderableObject
    transformableNode.setParent(anchorNode)
    fragment.arSceneView.scene.addChild(anchorNode)
    transformableNode.select()
}
```

GDG Chimbote

Probar

EMULADOR



GDG Chimbote

DEMO

https://github.com/ederalbino/demo-taller-ra/tree/master