LIBGDX Empaquetado distribución.

UNIDADE 2: Empaquetado e distribución

Sumario

- 1 Introdución
- 2 Desktop
- 3 Android
    - 3.1 Graficamente
    - 3.2 Dende consola/terminal
        - 3.2.1 Sen firmar
        - 3.2.2 Firmada
            - 3.2.2.1 Opción a)
            - 3.2.2.2 Opción b)
- 4 IOS
- 5 HTML
- 6 Nova ferramenta de empaquetado

Introdución

Nesta sección imos ver como podemos crear o arquivo que vai ser descargado polos xogadores para instalar o xogo no seu computador / tablet.

Información na wiki: https://github.com/libgdx/libgdx/wiki/Gradle-on-the-Commandline#packaging-the-project

Vídeo explicativo: https://www.youtube.com/watch?v=shxHON23qNo

O proceso é moi sinxelo. Temos que ir ó cartafol onde se atopa o noso proxecto dende a consula de ms-dos (Windows) ou dende un terminal en Linux.

Unha vez nel, e dependendo da plataforma, faremos:

Desktop

Windows:

```
gradlew desktop:dist
```

Linux:

```
./gradlew desktop:dist
```

Unha vez xerado teremos o arquivo coa extensión jar no cartafol **desktop/build/libs**/ do proxecto.

Android

As aplicación Android poder xerarse firmadas ou non. A firma consiste nun proceso polo que 'asociamos' un certificado cunha información (persoa ou empresa que publica a aplicación / xogo, data de validez do certificado,....) ó xogo desenvolto.

Se quixeramos levar o xogo o Market de Android teríamos que firmar a aplicación. Dito certificado serviríanos despois para subir novas versións do xogo ó Market.

Graficamente

O podemos facer dende o entorno do eclipse. Simplemente nos situamos sobre o proxecto versión de Android. Neste caso imos explicar o proceso para xerar unha versión firmada, tendo que escoller a opción Export UnSigned Application Package se queremos xerar a non firmada.

- Xeración do xogo con firma
-

  Prememos botón dereito e escollemos a opción Android Tools => 'Export Signed Application Package':.
-

  Aparece un asistente.Prememos next.
-

  Agora debemos indicar se usamos un keystore previamente creado ou creamos un novo. Escollemos esta segunda opción e indicamos onde se vai gardar o keystore e a contrasinal do almacén de claves. Prememos next.
-

Cubrimos os datos do certificado.

-

  Indicamos o lugar onde se vai xerar o apk xa coa firma.

Nota: Os datos do certificado son:

- Alias: Un alias para o keystore. Pode ser o mesmo que o nome ou unha abreviación do mesmo.
- Password: Novamente asinámoslle unha contrasinal. Ten que ter o lo menos 6 caracteres. Esta vai ser a contrasinal da key,
- Validity (years): Aquí indicamos o tempo que vai ser válida a nosa keystore en anos.
- Os seguintes campos fan referencia a información persoal e da organización. O campo de Country Code, se pode consultar no listado da ISO 3166-1. No noso caso é **ES**.

Dende consola/terminal

Sen firmar

Windows:

```
gradlew android:assembleRelease
```

Linux:

```
./gradlew android:assembleRelease
```

Unha vez xerado teremos o arquivo coa extensión apk no cartafol **android/build/outputs/apk** do proxecto.

Se agora intentades instalar dita aplicación nun dispositivo móbil vos avisará que tedes que permitir instalar aplicación dende fontes descoñecidas. Isto é así xa que a aplicación que se xera non está firmada.

Nota: Parece ser que dende que se cambiou a Gradle a xeración de proxectos Android sen firmar leva consigo que cando intentedes instalalo nun dispositivo móbil vos dea o erro de **Aplicación no instalada**. Se é ese o caso probade a firmar a aplicación.

Firmada

- Primeiro temos que ir cunha consola / terminal ó cartafol onde se atope o JDK ou JRE instalado no equipo. Ó cartafol bin (jre/bin ou jdk/bin). Se a ruta xa está engadida ó path non faría falla facer este paso.

- Despois temos que teclear o seguinte:

  ```
  keytool -genkey -v -keystore almacen.keystore -alias xogoplatega -keyalg RSA -keysize 2048 -validity 10000
  ```

  Isto vai crear unha clave (xogoplatega) que vai ser gardada nun almacén de claves de nome almacen.keystore (un almacén pode ter moitas claves gardadas).

  Os datos a pedir serán:

  ◊ Contrasinal do almacén de datos (dúas veces).
  ◊ Nome e apelidos.
  ◊ Nome da unidade da organización.

◊ Nome da organización.
◊ Cidade.
◊ Provincia.
◊ Código do país: ES

NOTA: Coidado en Windows 7 xa que se abrides unha consola de ms-dos en modo non administrador non terá permiso para crear o almacén. Podedes ir a outro cartafol con permisos de escritura e referenciar o keytool coa ruta completa: c:\archivos de programa\java\jreXXXX\bin\keytool ....

Confirmaremos e xa teremos xerado a clave coa que imos 'firmar' o noso xogo.

Podemos verificar que o almacén e a clave son correctos:

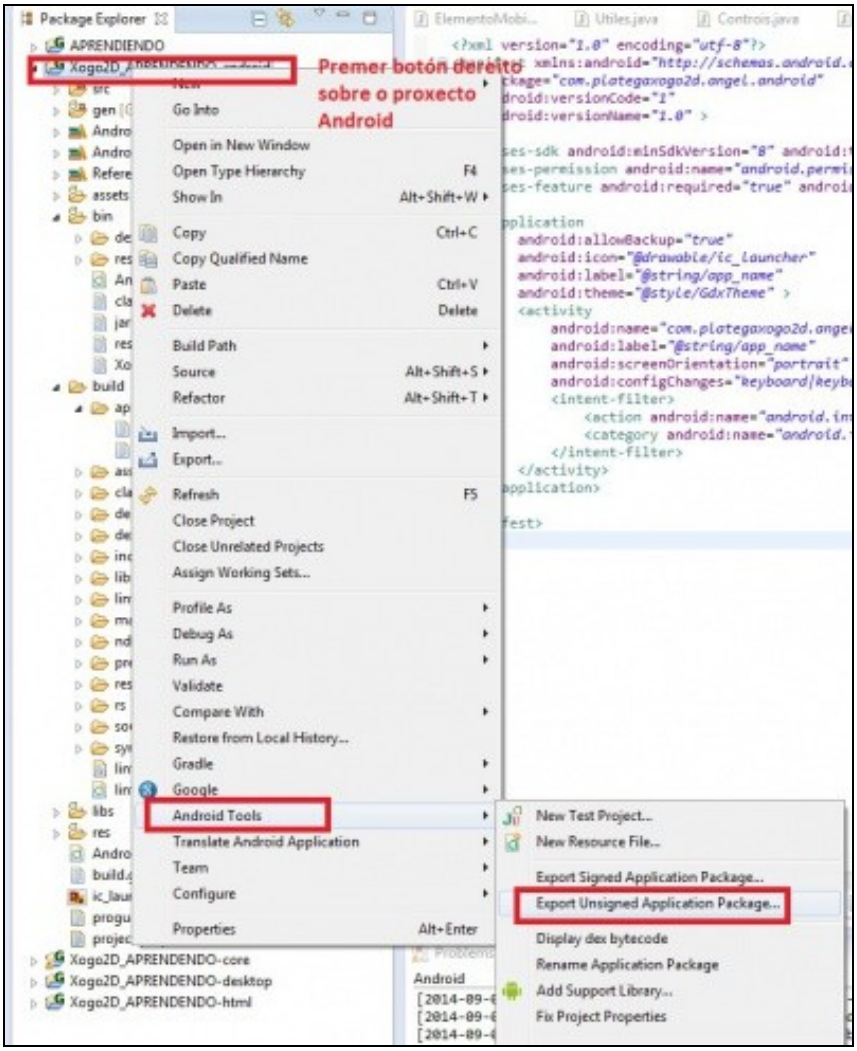
```
keytool -list -v -keystore almacen.keystore
```

A partir de aquí dispoñemos de dúas opcións:

Opción a)

Firmar a aplicación de forma manual.

Primeiro debemos xerar o apk sen firmar. Podemos facelo como está indicado nun punto anterior, dende a consola/terminal ou outra forma de facelo é graficamente:



Despois debemos executar esta orde dende un terminal:

```
jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore ruta_e_arquivo_almacen_datos omeuxogo.apk nome_do_alias
```

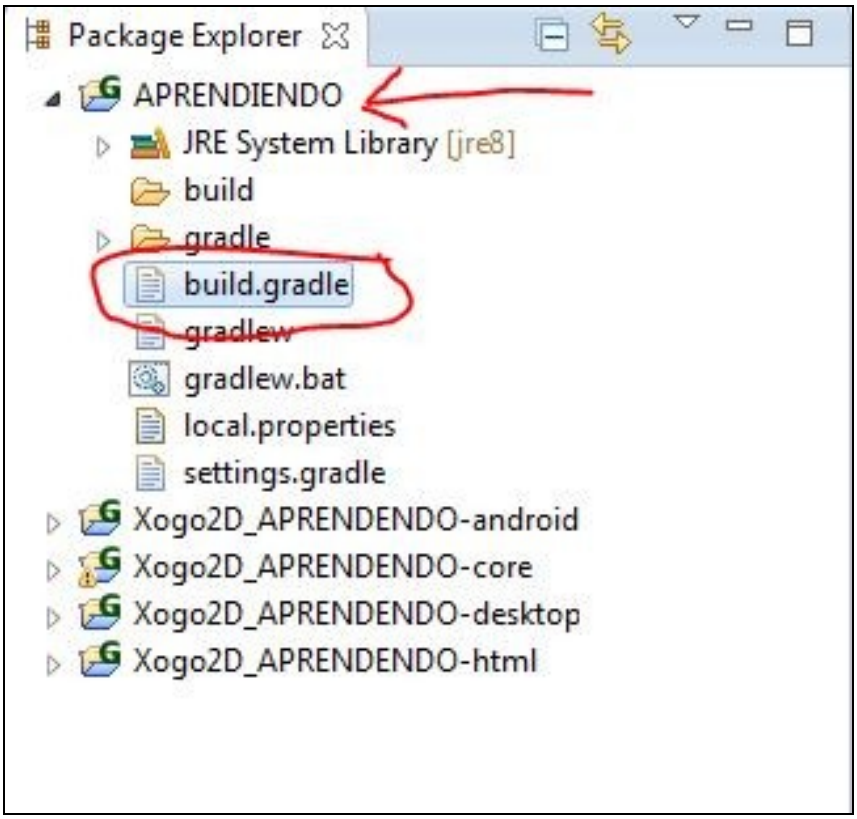
Nota: O executable jarsigner se atopa no cartafol bin do SDK instalado.

Para verificar se todo está correcto:

```
jarsigner -verify -verbose -certs nome_xogo.apk
```

Opción b)

- Agora debemos modificar o arquivo **build.gradle** que se atopa no raíz do proxecto. O podemos facer dende o propio eclipse se prememos dúas veces sobre o arquivo.



Debemos ir ás sección proyect(":android") e escribir o marcado:

```
project(":android") {
    apply plugin: "android"

    configurations { natives }

    dependencies {
        compile project(":core")
        compile "com.badlogicgames.gdx:gdx-backend-android:$gdxVersion"
        natives "com.badlogicgames.gdx:gdx-platform:$gdxVersion:natives-armeabi"
        natives "com.badlogicgames.gdx:gdx-platform:$gdxVersion:natives-armeabi-v7a"
        natives "com.badlogicgames.gdx:gdx-platform:$gdxVersion:natives-x86"
    }
    android {
        //...
        signingConfigs {
            releaseSigning {
                storeFile file("willBeReplaced")
                storePassword ""
                keyAlias ""
                keyPassword ""
            }
        }
        buildTypes {
            release {
                signingConfig signingConfigs.releaseSigning
            }
        }
    }
}
```

- Agora debemos de poñer este código o **comezo do arquivo**:

```
import groovy.swing.SwingBuilder
```

- E este código **ó final do arquivo**.

```
gradle.taskGraph.whenReady { taskGraph ->
    if(taskGraph.hasTask(':android:assembleRelease')) {
        def _store = ''
        def _storePassword = ''
        def _keyPassword = ''
        def _keyAlias = ''
        if(System.console() == null) {
            new SwingBuilder().edt {
                def inputStore
                def inputStorePass
                def inputKey
                def inputKeyPass
                dialog(modal: true,
                        title: 'Key Store',
                        alwaysOnTop: true,
                        resizable: false,
                        locationRelativeTo: null,
                        pack: true,
                        show: true
                ) {
                    vbox {
                        label(text: "Key store:")
                        inputStore = textField(text: "myKeyStore.keystore")
                        label(text: "Key store password:")
                        inputStorePass = passwordField()
                        label(text: "Key alias:")
                        inputKey = textField(text: "myAlias")
                        label(text: "Key password:")
                        inputKeyPass = passwordField()
```

```
                        button(defaultButton: true, text: 'OK', actionPerformed: {
                            _store = inputStore.text
                            _storePassword =  new String(inputStorePass.password);
                            _keyAlias =  inputKey.text
                            _keyPassword =  new String(inputKeyPass.password)
                            dispose();
                        })
                    }
                }
            }
        } else {
            _store = System.console().readLine("\nLocation of the key store: ")
            _storePassword = System.console().readPassword("\nKey store password: ")
            _storePassword = new String(_storePassword)
            _keyAlias = System.console().readLine("\nKey alias: ")
            _keyPassword = System.console().readPassword("\nKey password: ")
            _keyPassword = new String(_keyPassword)
        }
        if(_storePassword.size() <= 0 || _keyPassword.size() <= 0 ||
                _store.size() <= 0 || _keyAlias.size() <= 0) {
            throw new InvalidUserDataException("You must enter a key store password and key password.")
        }
        if(!file(_store).exists()){
            throw new InvalidUserDataException("Could not find key store ["+_store+"]")
        }
        //set signing values to android project
        project("android").android.signingConfigs.releaseSigning.storeFile = file(_store)
        project("android").android.signingConfigs.releaseSigning.storePassword = _storePassword
        project("android").android.signingConfigs.releaseSigning.keyAlias = _keyAlias
        project("android").android.signingConfigs.releaseSigning.keyPassword = _keyPassword
    }
}
```

- Agora só queda xerar a apk firmada, como fixemos antes:

Windows:

```
gradlew android:assembleRelease
```

Linux:

```
./gradlew android:assembleRelease
```

Durante o proceso se nos pedirá:

◊ A ruta e nome onde se atopa o almacén de datos. **Debemos indicar a ruta xunto có nome do almacén (incluída extensión).**
◊ O password do almacén.
◊ A nome da clave gardada (o alias).
◊ O password da clave.

Unha vez xerado teremos o arquivo coa extensión apk no cartafol **android/build/apk** do proxecto e xa poderíamos subilo ó Appstore.

TAREFA 2.15 A FACER: Esta parte está asociada á realización dunha tarefa.

IOS

Windows:

```
gradlew ios:createIPA
```

Linux:

```
./gradlew ios:createIPA
```

Unha vez xerado teremos o arquivo coa extensión IPA no cartafol **ios**/**build**/**robovm** do proxecto.

HTML

Windows:

```
gradlew html:dist
```

Linux:

```
./gradlew html:dist
```

Unha vez xerado teremos os arquivos HTML, javascript e arquivos do cartafol assets no cartafol **html**/**build**/**dist**/ do proxecto. Este é o contido que hai que subir ó servidor web (por exemplo Apache).

Nova ferramenta de empaquetado

A partires de Maio do 2014 é posible distribuír o xogo coa máquina virtual de java 'incrustada' dentro do arquivo, de tal forma que podemos xogar en computadores onde o software de java non estea instalado.

O tamaño do arquivo aumenta considerablemente (sobre 12MB, pero depende do xogo).

Enlace á nova: http://www.badlogicgames.com/wordpress/?p=3428 Enlace como usala: https://github.com/libgdx/packr

-- Ángel D. Fernández González -- (2014).