

# Export Love2D to APK

Ce guide vous montre étape par étape la méthode que j'emploie pour transformer un jeu développé avec Love2D en fichier .apk pour être joué sur un smartphone Android.

J'ai utilisé la méthode utilisant l'APKTool, expliquée sur le Wiki de Löve ( [https://love2d.org/wiki/Game\\_Distribution/APKTool](https://love2d.org/wiki/Game_Distribution/APKTool) )

## Récupération des outils

### Java SDK

Tout d'abord, il faut télécharger et installer le Java JDK. Il permet de lancer les fichiers .jar et contient 2 outils nécessaires à la procédure : keytool.exe et jarsigner.exe.

=> <https://www.oracle.com/java/technologies/javase-downloads.html>

Une fois installé, lancez une invite de commande ( appuyez sur les touches "windows + r" puis tapez "cmd"), et vérifiez que votre version de java est bien en 1.8 ou supérieur en tapant la commande :

```
java -version
```

### APK Tool

Il faut ensuite récupérer l'APKTool qui nous permet de créer le fichier .apk, ainsi que le script batch pour l'utiliser. (Source : <https://ibotpeaches.github.io/Apktool/install/> )

APK Tool => <https://bitbucket.org/iBotPeaches/apktool/downloads/>

Renommez le fichier en apktool.jar

apktool.bat (faites un clic droit / enregistrer la cible du lien sous... puis assurez-vous que le fichier soit bien apktool.bat, si c'est apktool.bat.txt, renommez-le.) =>

<https://raw.githubusercontent.com/iBotPeaches/Apktool/master/scripts/windows/apktool.bat>

Copiez ces deux fichiers (apktool.jar et apktool.bat) dans le répertoire de windows (C:\Windows\), puis lancez une invite de commande, et vérifiez si cela fonctionne en tapant la commande :

```
apktool
```

### Löve APK

Récupérez le fichier .apk de Löve à l'adresse suivante

=> <https://github.com/love2d/love/releases/download/11.3/love-11.3-android-embed.apk>

Créez un répertoire qui va vous servir à construire votre fichier .apk et mettez-y l'apk de Löve.

### Android Build Tools

Enfin, récupérez la boîte à outils Android contenant zipalign.exe.

=> [build-tools\\_r30-rc1-windows.zip](https://developer.android.com/studio/build)

## Création du trousseau de clés

Pour pouvoir signer votre fichier .apk et permettre son installation sur Android, vous devez vous créer un trousseau de clés. Pour se faire, il va falloir utiliser **keytool.exe** disponible dans le Java JDK. Ouvrez une invite de commande et rendez-vous dans le dossier **bin** du JDK ( C:\Program Files\Java\jdk-14\bin ), puis tapez la commande :

```
keytool -genkeypair -v -keystore keyring.keystore -alias keyname -keyalg RSA  
-keysize 2048 -validity 10000
```

**keyring.keystore** est le nom du trousseau de clé, et **keyname** est la clé contenue dans le trousseau. Vous pouvez modifier les nom si vous le désirez.

Cette commande va vous demander de renseigner plusieurs choses :

- Un mot de passe pour pouvoir utiliser la clé
- Votre nom et prénom
- Le nom de l'unité au sein de votre organisation (vous pouvez mettre le nom de l'organisation)
- Le nom de votre organisation
- Le nom de votre ville
- Le nom de votre département
- Les 2 lettres du code de votre pays

Vous validez en tapant **yes**.

Conservez le fichier .keystore qui a été créé, il va vous servir à **signer numériquement** vos applications.

## Création du fichier .apk

### Décompilier le fichier APK

D'abord, compressez les fichiers de votre jeu en **game.zip** puis renommez ce fichier en **game.love**. Il est **important** que le nom du fichier .love soit **game.love**.

Ouvrez une invite de commande, et rendez-vous dans le dossier contenant le fichier **love-11.3-android-embed.apk**, et tapez la commande **apktool d -s -o love\_decoded love-11.3-android-embed.apk**. Cela va créer un répertoire **love\_decoded** contenant les fichiers nécessaires pour la création du fichier .apk. Vous pouvez remplacer **love\_decoded** par le nom que vous souhaitez, il faudra remplacer **love\_decoded** par le nom que vous aurez choisi dans la suite de la procédure.

Dans ce répertoire **love\_decoded**, vous devez créer un répertoire **assets** dans lequel vous y placez votre fichier **game.love**.

Vous devez ensuite modifier le fichier AndroidManifest.xml en effaçant son contenu pour le remplacer par celui-ci (disponible également ici :

[https://love2d.org/wiki/Game\\_Distribution/APKTool](https://love2d.org/wiki/Game_Distribution/APKTool)) :

```

<?xml version="1.0" encoding="utf-8" standalone="no"?>
<manifest package="${GamePackageName}"
    android:versionCode="${GameVersionCode}"
    android:versionName="${GameVersionSemantic}"
    android:installLocation="auto"
xmlns:android="http://schemas.android.com/apk/res/android">
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.VIBRATE"/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"
android:maxSdkVersion="18"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
android:maxSdkVersion="18"/>
    <!--Uncomment line below if your game uses microphone capabilities. !-->
    <!--uses-permission android:name="android.permission.RECORD_AUDIO" /!-->
    <uses-feature android:glEsVersion="0x00020000"/>
    <application
        android:allowBackup="true"
        android:icon="@drawable/love"
        android:label="${GameName}"
        android:theme="@android:style/Theme.NoTitleBar.Fullscreen"
    >
        <activity
            android:configChanges="orientation|screenSize"
            android:label="${GameName}"
            android:launchMode="singleTask"
            android:name="org.love2d.android.GameActivity"
            android:screenOrientation="sensorLandscape"
        >
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
                <category android:name="tv.ouya.intent.category.GAME"/>
            </intent-filter>
        </activity>
    </application>
</manifest>

```

Remplacez les variables suivantes :

- `${GamePackageName}`
- `${GameVersionCode}`
- `${GameVersionSemantic}`
- `${GameName}`

`$(GamePackageName)` correspond au nom de votre fichier .apk, par exemple monJeu.apk.

`$(GameVersionCode)` correspond au code de version de votre APK, par exemple 1.

`$(GameVersionSemantic)` correspond à la version de votre APK, par exemple 1,0.

`$(GameName)` correspond au nom de votre jeu, par exemple MonJeu.

Exemple :

```

<manifest package="monJeu.apk"

    android:versionCode="1"
    android:versionName="1.0"

```

etc...

## Changer l'icône (optionnel)

Pour changer l'icône de votre application, vous devez remplacer les fichiers **love.png** des dossiers **drawable-mdpi**, **drawable-hdpi**, **drawable-xhdpi**, **drawable-xxhdpi** et **drawable-xxxhdpi** dans le dossier **res**, en gardant les **mêmes dimensions** que les fichiers que vous avez remplacés.

Les dimensions :

- drawable-mdpi : 48\*48
- drawable-hdpi : 72\*72
- drawable-xhdpi : 96\*96
- drawable-xxhdpi : 144\*144
- drawable-xxxhdpi : 192\*192

*Astuce : Créez un dossier (sur votre bureau par exemple) dans lequel vous créez les 5 dossiers drawable-[...], puis réalisez votre icône au format 192\*192 et exportez la sous le nom love.png dans le dossier drawable-xxxhdpi. Redimensionnez ensuite votre image avec la taille inférieure et exportez la dans le dossier correspondant, toujours sous le nom de love.png, et répétez l'opération pour chaque format. Votre image perdra moins en qualité si vous partez d'un grand format que vous rétrécissez, plutôt qu'en partant d'un petit format que vous agrandissez. Sélectionnez ensuite tous les dossiers drawable-[...], copiez les, et collez-les dans le répertoire **res**.*

## Compiler votre fichier APK

Retournez sur l'**invite de commande**, dans le dossier contenant le fichier **love-11.3-android-ambd.apk** et tapez la commande suivante :

```
apktool b -o monJeuUnsigned.apk love_decoded
```

Si vous avez choisi un autre nom de dossier que love\_decoded lors de la décompilation, mettez-le à la place de love\_decoded.

J'ai choisi d'appeler le fichier .apk **monJeuUnsigned.apk** pour voir qu'il n'a pas encore été signé numériquement, et parce que l'utilitaire zipalign.exe a besoin de créer un nouveau fichier lors de son utilisation, ce fichier sera le fichier .apk terminé, qui contiendra le nom voulu.

## Signer numériquement votre fichier APK

(Source : <https://docs.microsoft.com/fr-fr/xamarin/android/deploy-test/signing/manually-signing-the-apk> )

Pour permettre à Android d'installer votre fichier .apk, il faut le signer numériquement. Copiez votre fichier monJeuUnsigned.apk dans le répertoire du JDK de Java ( C:\Program Files\Java\jdk-14\bin ), ainsi que votre trousseau de clé **keyring.keystore**, puis rendez-vous dans ce répertoire via l'invite de commandes.

Tapez ensuite la commande :

```
jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore  
keyring.keystore monJeuUnsigned.apk keyname
```

**keyring.keystore** correspond à votre trousseau de clé, et **keyname** correspond à la clé dans le trousseau.

Le fichier **monJeuUnsigned.apk** sera alors modifié, vous devrez le copier dans le répertoire contenant les outils pour Android, où se situe **zipalign.exe** ( build-tools\_r30-rc1-windows\android-R )

Rendez-vous dans ce dossier via l'invite de commandes, et tapez la commande :

```
zipalign -f -v 4 monJeuUnsigned.apk monJeu.apk
```

Cela va créer un nouveau fichier **monJeu.apk** qui sera le fichier final de votre jeu.

Vous pouvez à présent copier votre fichier .apk sur votre smartphone, et lancer son installation en acceptant les **installations des sources inconnues**.