Notes for SE (Object passing/Deserializing)

Creating our database inside unity could be a simple translation to a game object

* We make an empty game object
* We build a c# script attached to it that holds player information

We also, could have java programs running at the same time as unity’s engine, and have the scripts call for results from functions we use in Java.

<https://docs.unity3d.com/Manual/PluginsForAndroid.html>

Unity:

AndroidJavaClass (for Java class references and static calls)

AndroidJavaObject (for references to objects).

<https://docs.unity3d.com/ScriptReference/JsonUtility.FromJson.html>

Java:

UnitySendMessage()

The last parameter in UnitySendMessage() is an optional string argument that you can use to pass JSON to Unity.

From StackOverflow:

Get classes.jar from one of the sub-folders from

<UnityInstallationDirectory>\Editor\Data\PlaybackEngines\AndroidPlayer\Variations\mono

or

<UnityInstallationDirectory>\Editor\Data\PlaybackEngines\AndroidPlayer\Variations\il2cpp

depending on if you are using mono or IL2CPP as Scripting Backend to build your Android Project.

Once you import the classes.jar module into your Android Java project, you can the use UnityPlayer.UnitySendMessage to call C# function from Java.

UnityPlayer.UnitySendMessage("Gameobject Name","Method","Message")

If you plan to extend UnityPlayerActivity then grab UnityPlayerActivity.java from

<UnityInstallationDirectory>\Editor\Data\PlaybackEngines\AndroidPlayer\Source\com\unity3d\player.

Finally, UnityPlayer.UnitySendMessage is what you are looking for. There are many tutorials out there on how to use it, if you are still confused.

Might be the best option!

Using Gradle to build an APK <https://docs.unity3d.com/Manual/android-BuildProcess.html>