

Hi!

If you read this document so you bought my asset!

In this asset everything are simple. Code is easy and has summaries.

- 1) In the version of asset **5.0.0** and **earlier** we should install **Google Admob package** downloading the latest version of Admob plugin for Unity from here <https://github.com/googleads/googleads-mobile-unity/releases/tag/v8.5.3> and open asset in **Unity 2021** and upper (because of it is a requirement for Admob plugin) then we should complete import **GoogleMobileAds.unitypackage** to project. And finish steps from guide here:

<https://developers.google.com/admob/unity/start>

- 2) Starting from version **5.1.0** I register dependencies to external packages into manifest:

- **"com.google.ads.mobile"**
- **"com.unity.nuget.newtonsoft-json"**
- **"com.google.external-dependency-manager"**

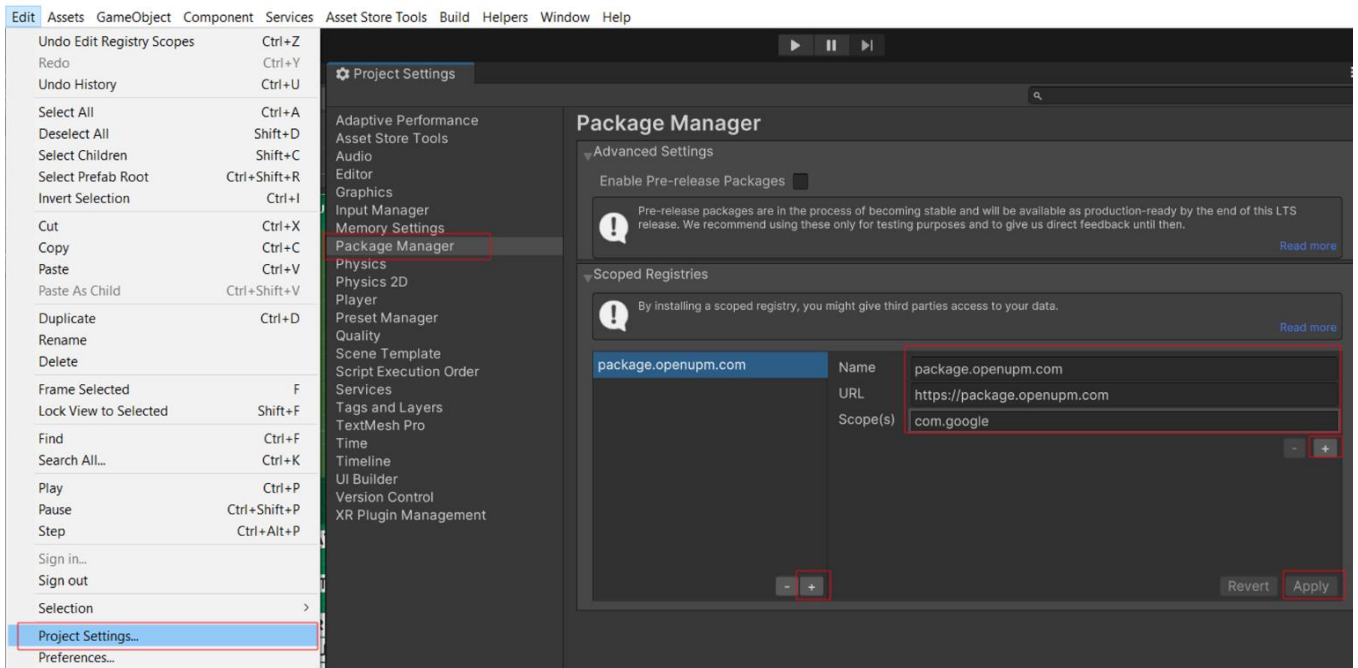
So if you have already imported Json .Net or Google Mobile Ads package please **remove** it from your project.

- 3) Go to **Edit** → **Project Settings** → **Package Manager** → **Create new registry** for **openupm** and add scopes:

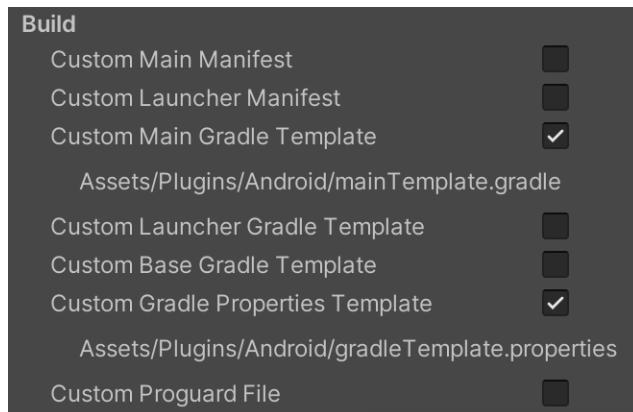
Name: doesn't matter

URL: **https://package.openupm.com**

Scopes: **com.google**



- 4) Also we should upgrade Unity to **2022 LTS** to completely build project for mobile.
- 5) Go to Project Settings > Player > Android > Publishing Settings > Build and select:



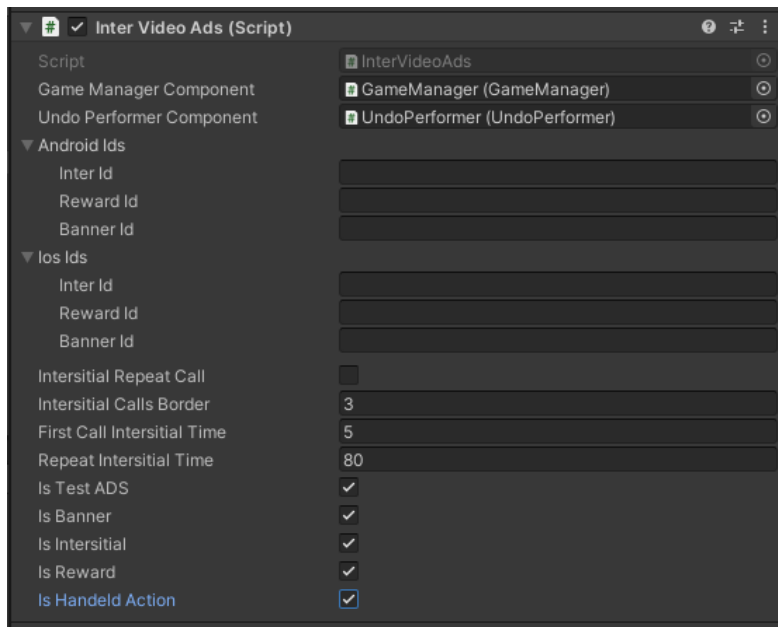
- 6) In the Unity editor click for **Android**:

Menu→Assets→External Dependency Manager→Android Resolver→Force Resolve.

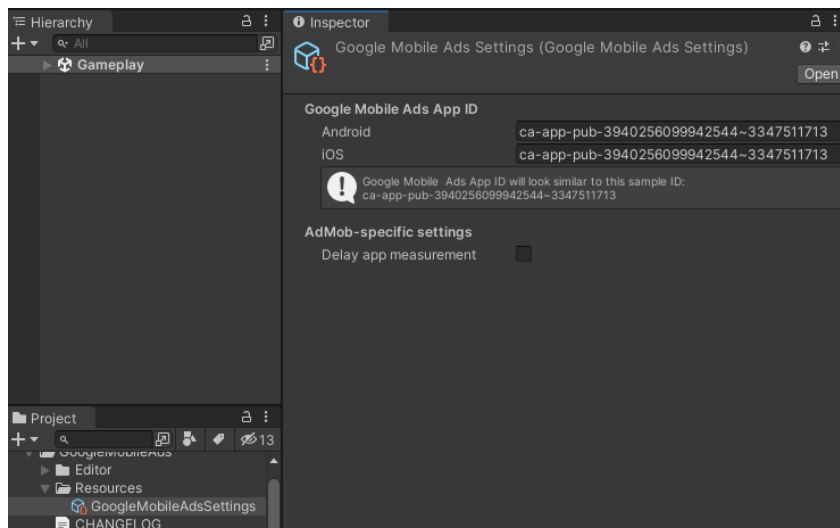
For iOS:

Menu→Assets→External Dependency Manager→ iOS Resolver → Install Cocoapods

- 7) Then to show advertisement in your build you need to change advertisement id. Open **InterVideoAds.cs** script in the scene, (It was added to object **AdsManager**) and change it there.



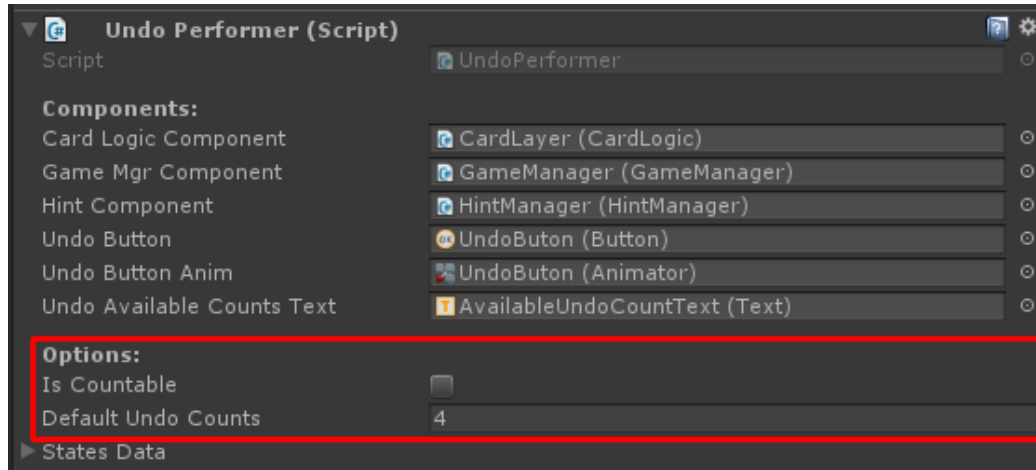
And *GoogleMobileAdsSettings.asset*



- 8) For activating test ADS you should set **IsTestADS** variable to **TRUE** before play (like on screen). For deactivating test ADS you should set **IsTestADS** variable to **FALSE** before play.

Undo

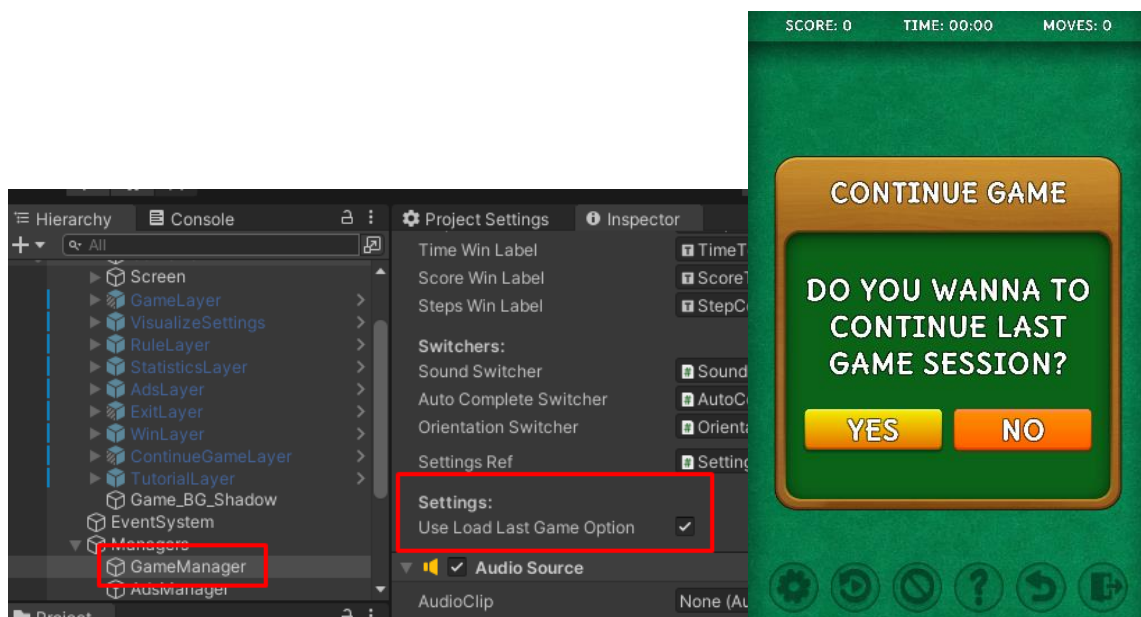
Undo logic has option for counting available uses. When counter equals 0 for getting additional uses user should watch Rewarded video. More information in videos at asset page.



Last game option

You can enable/disable it at **GameManager** component.

Principle of work: User open new game session. When user close application the system save last game state(decks, cards, score, time, steps). When user open game again the system show popup:



Hand orientation mode

Include two modes: **Left-handed** and **Right-handed**

You can enable/disable this feature in VisualizeSettings window.

Auto Complete

Include two modes: **FullGameSession** and **OnlyWhenAllDecksClear**.

FullGameSession mode appear auto complete button every time when user has available move during the game session

OnlyWhenAllDecksClear mode appear auto complete button only in the end of game when all cards is can drag by user just for win game.

You can enable/disable this feature in VisualizeSettings window.



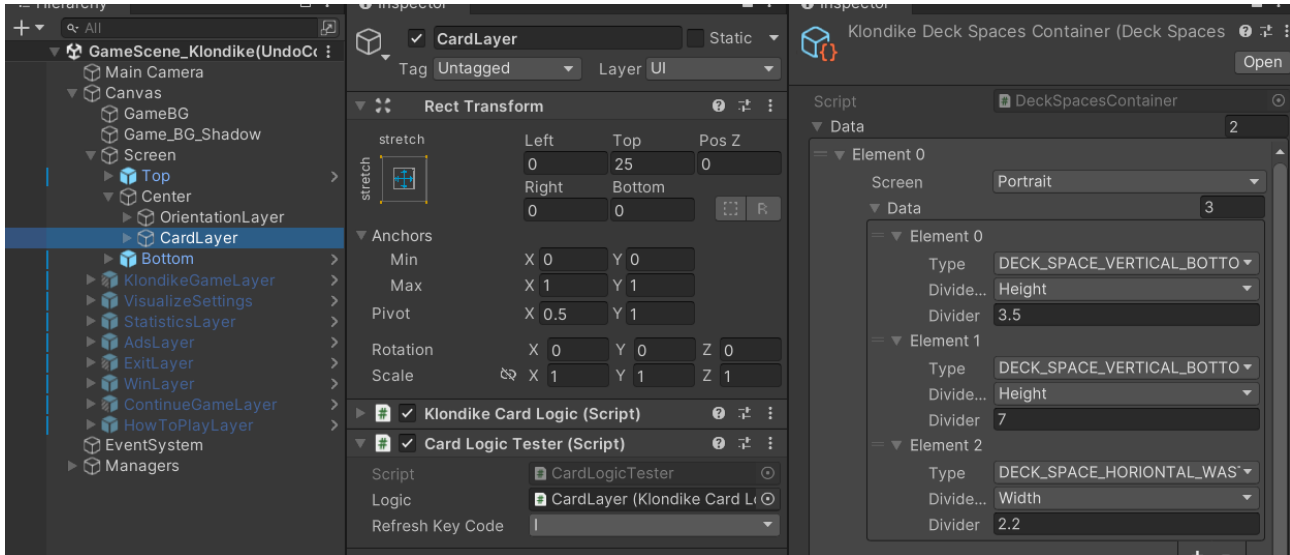
Draw rules



You can open game Window from this window and setup necessary rules by activating specific toggle and click **New** or **Replay** buttons to apply choose.

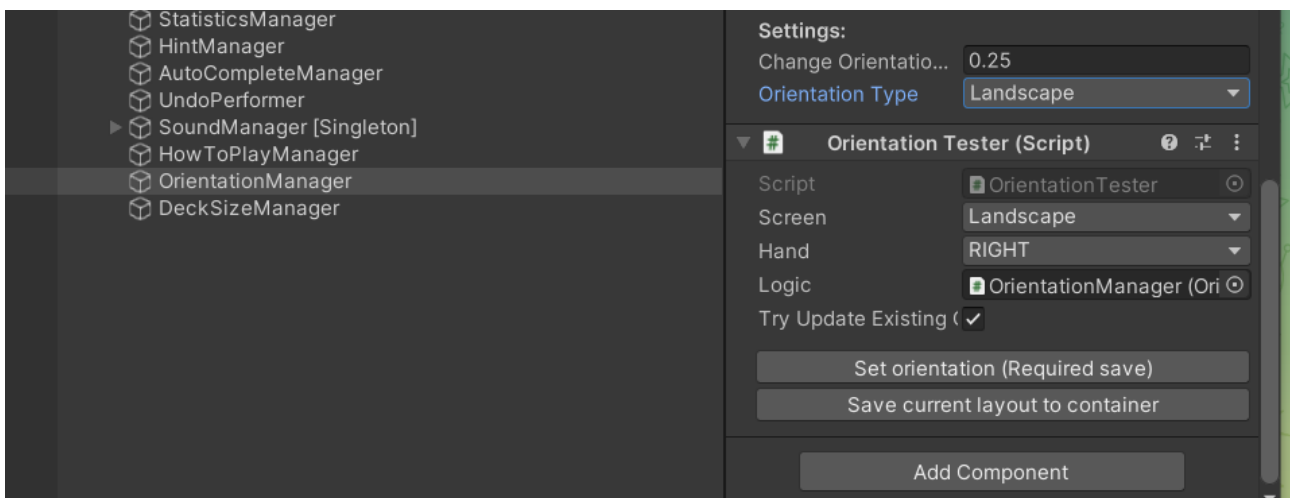
Deck Spaces

CardLayer contains a **CardLogicTester** component, which can help you modify the spaces used in the logic of **Deck** positioning. You can easily change existing values or, via this container, provide additional values to the code.



Orientation

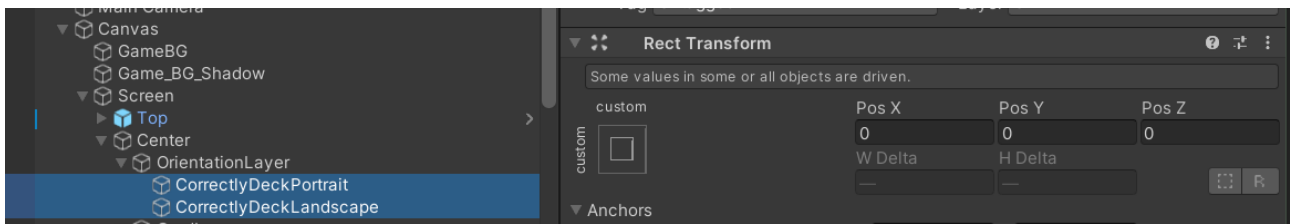
By default, in **Gameplay** scene, setup **Portrait** orientation. If you wanna change it to **Landscape**, just find **OrientationManager** component in scene, then set **OrientationType** to **Landscape**, then using **OrientationTester** set **Screen** to **Landscape** and **Hand** to **Right**. Don't forget to save changes.



During **WebGL limitations** we can use only one Screen orientation, Landscape or Portrait, so the button which change orientation in settings was disabled by platform.

If you want to change any orientation, you need to use the **OrientationTester** in the scene. Set the specific orientation and hand, then adjust the rect of the **decks** and press **Save Current Layout to Container**. Other items are changed using **OrientationObject** component inheritors, which are applied to different objects in the scene, with the setting provided in components.

Solitaire scene contains predefined **CorrectlyDeck** objects with copy of **Deck** from orientation. If you change orientation on your own, please be sure that you updated values of each of this object with new rect values from orientation.



If you find some issues or bugs please let me know via email below or join me in Skype.

Online documentation: [link](#)

For other questions write me an email: support@selvassets.ltd.ua

Skype: <https://join.skype.com/invite/EjSuvBELUYzw>

Thank you in advance!