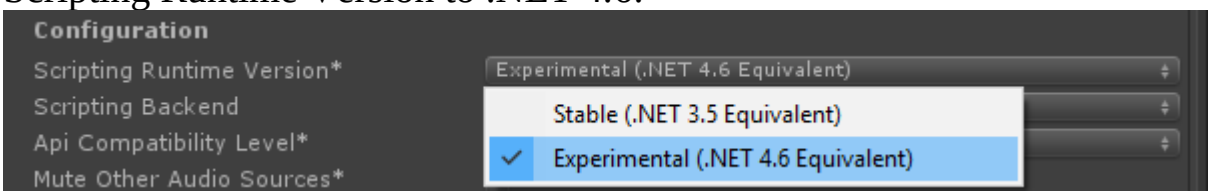
Hi!

If you read this document so you bought my asset! In this

game all is simple. All code has summaries.

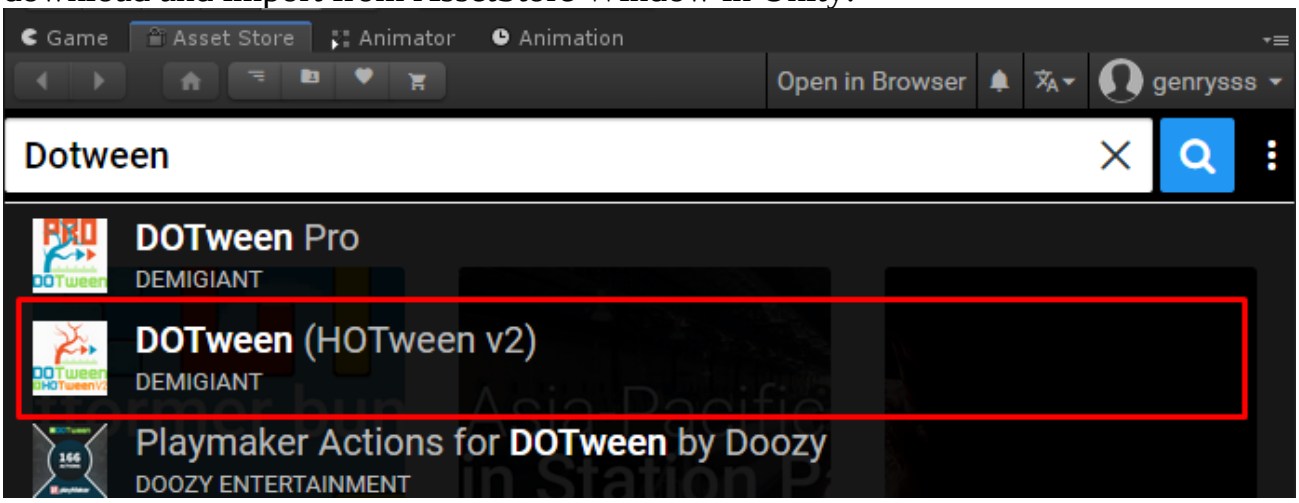
1. Open File-> Build settings -> Player settings -> Others Settings and change Scripting Runtime Version to .NET 4.6:



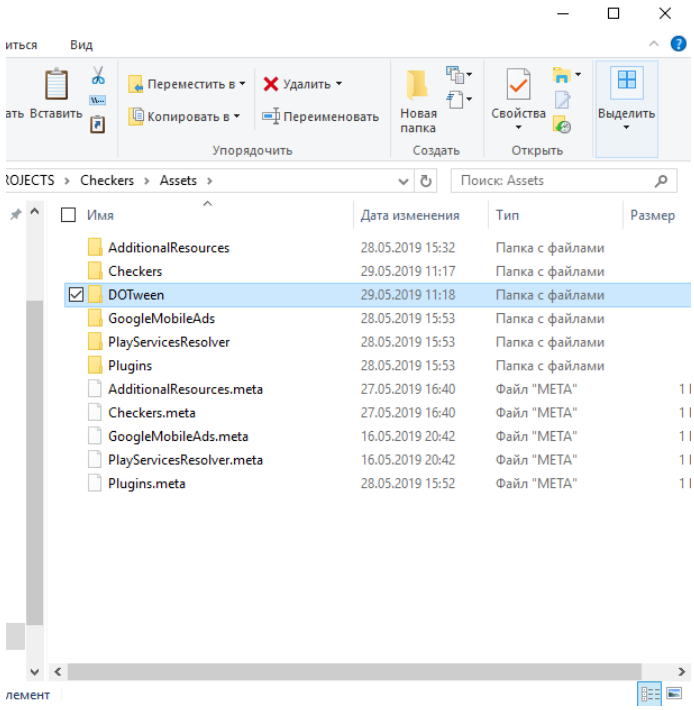
2. Before building this game to your phone you need to download the latest version of Admob plugin for Unity from there: https://github.com/googleads/googleads-mobile-unity/releases/tag/v5.4.0

3. After downloading complete import **GoogleMobileAds.unitypackage** to project.
4. Import last version of additional plugin for animations DOTween from official site: http://dotween.demigiant.com/download.php

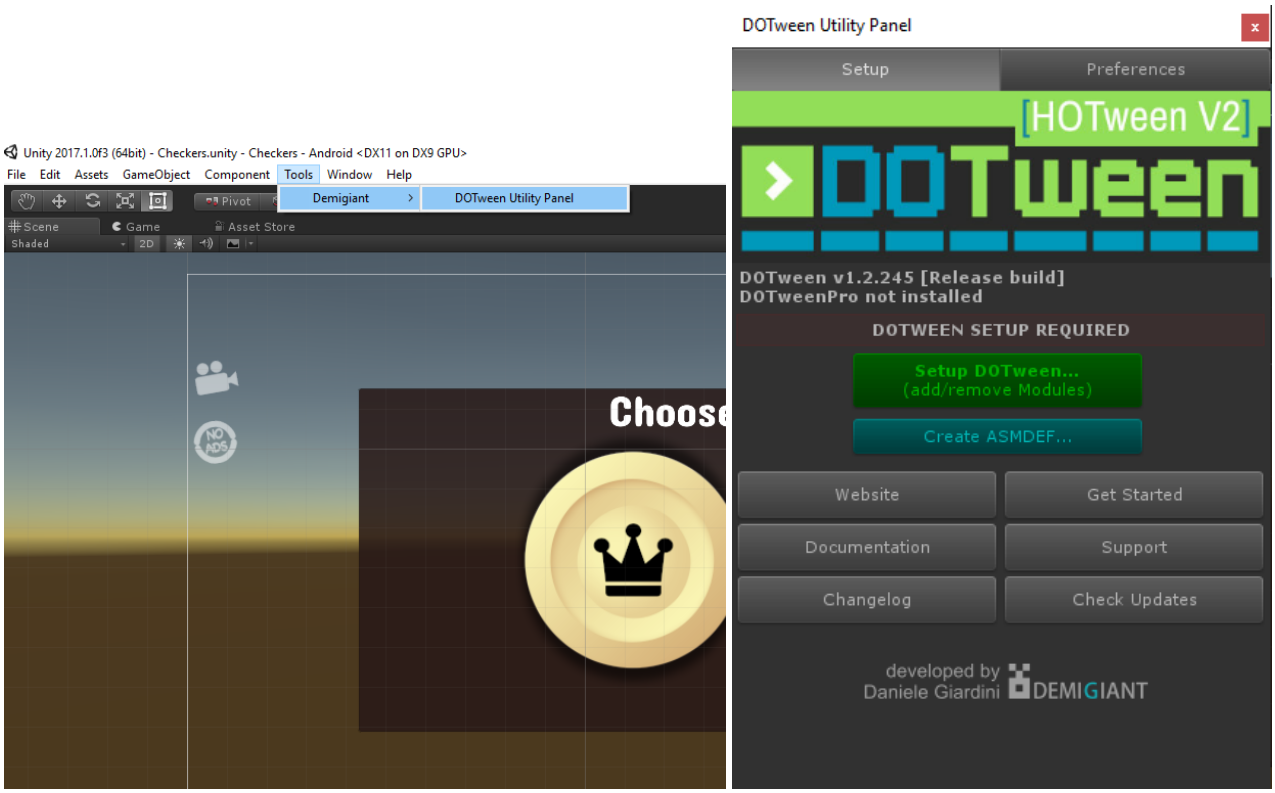
   Or download and import from AssetStore Window in Unity:



5. Unzip downloaded file and extract it to Assets folder(If you have downloaded DOTween plugin from site):

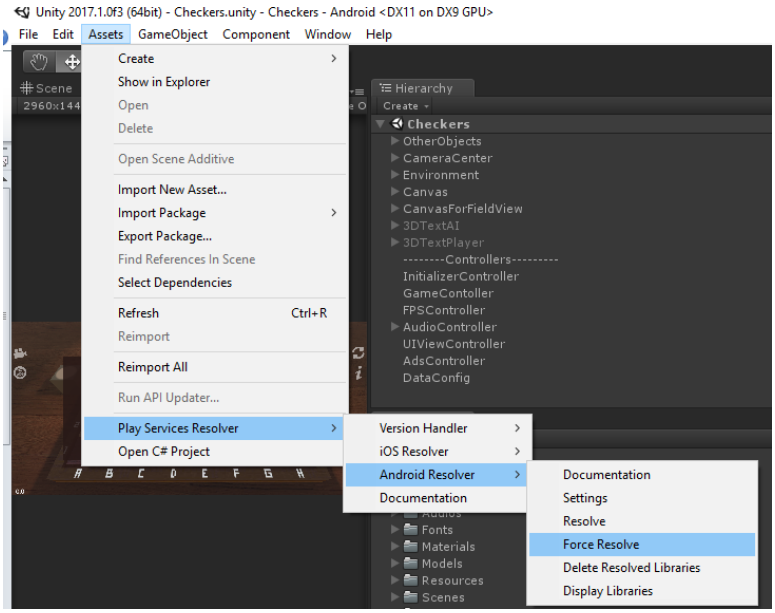


6. Open Unity and setup DOTween from opened panel.



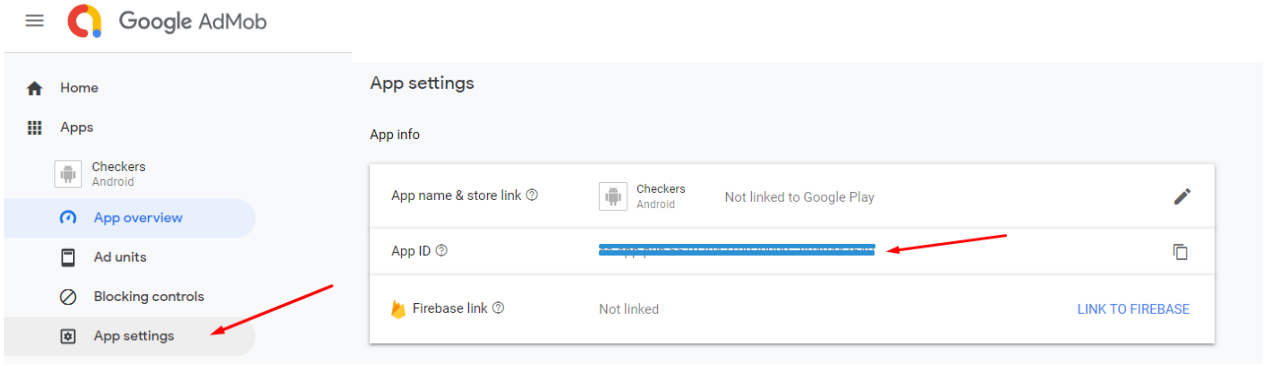
7. DOTween successfully setuped.

8. In the Unity editor click Menu→Assets→Play Services Resolver→Android→Force Resolve. If the menu item is not appear Menu→Assets→Play Services Resolver→Version Handler→Update.

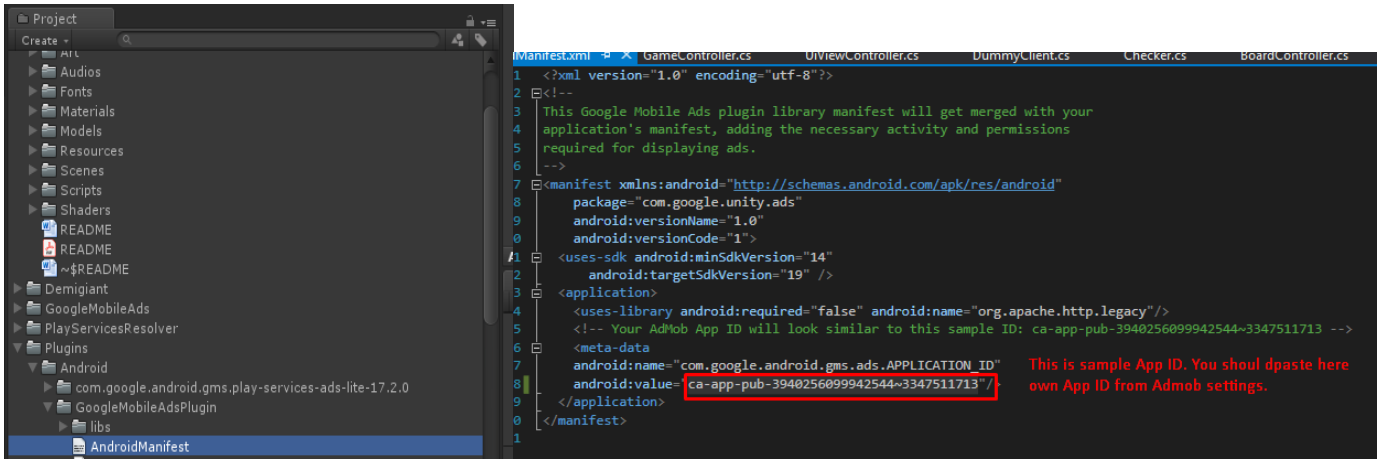


9. Update android manifest file from Assets-> Plugins folder:
   a). You should open your Admob account. Go to you App -> App settings an copy App ID.
   b). After this steps you should paste Id to manifest.

a)

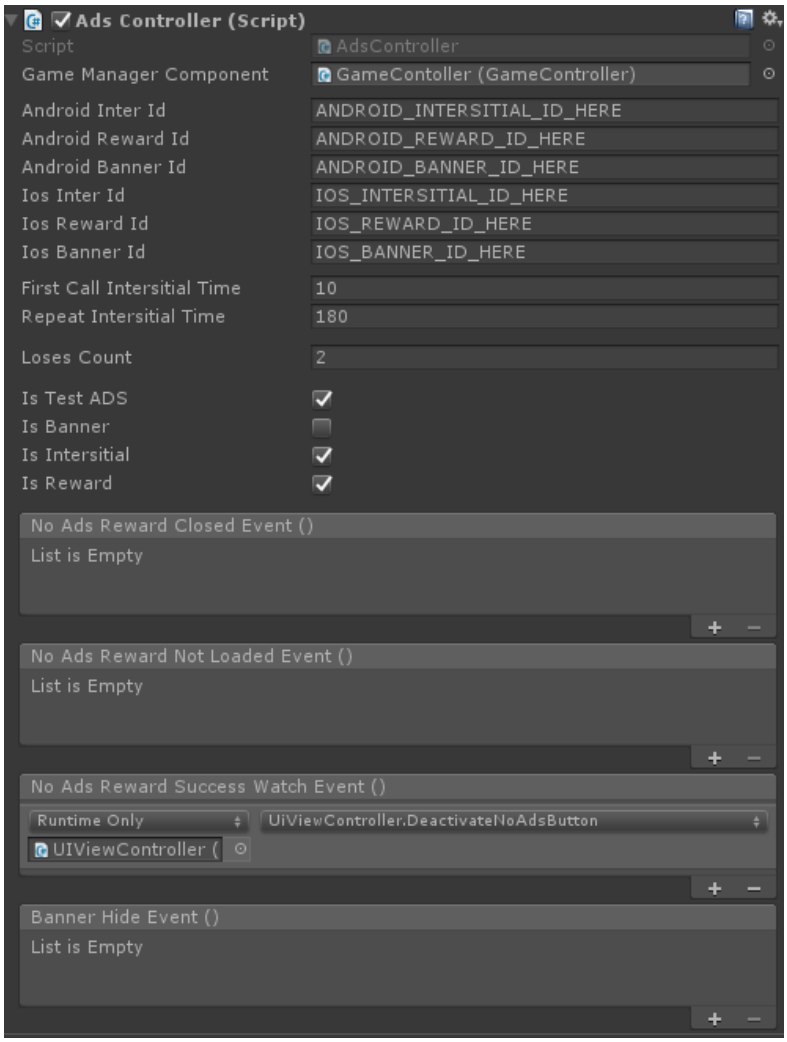


b)

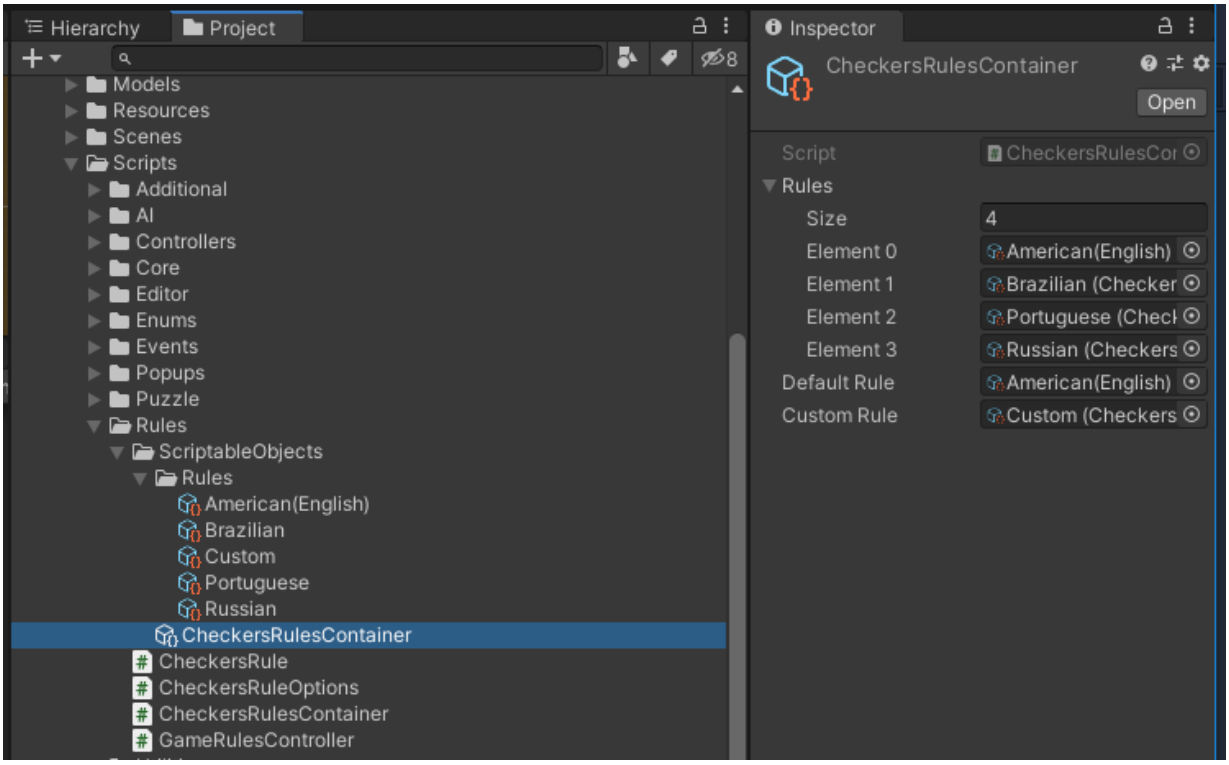


10. Then to show advertisement in your build you need to change advertisement id:

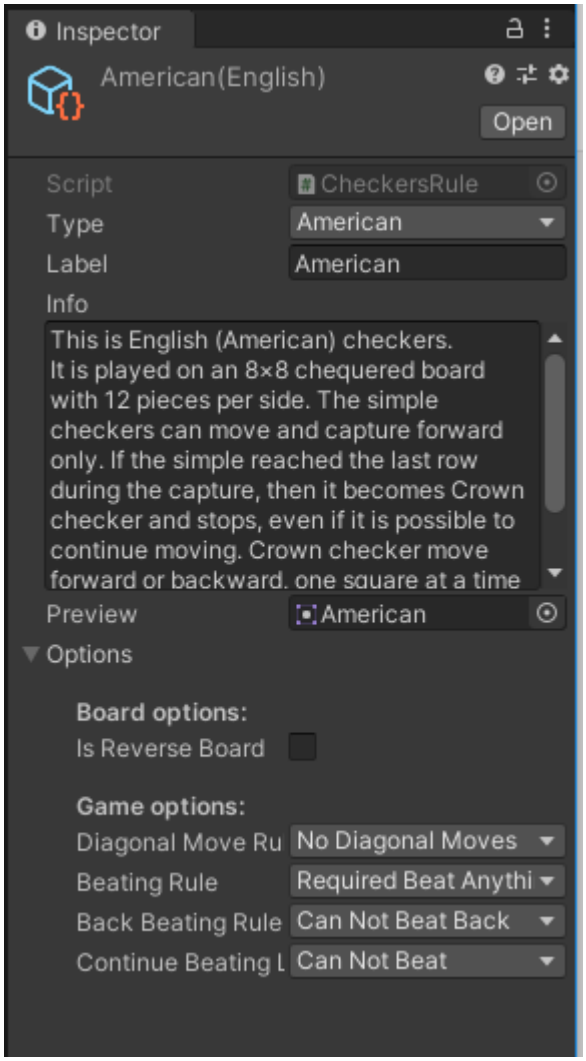
1) Open **AdsController**.cs script in the scene, (It was added to object **AdsController**) and change it there.
2) For activating test ADS you should set **IsTestADS** variable to **TRUE** before play (like on screen). For deactivating test ADS you should set **IsTestADS** variable to **FALSE** before play.
3) **LosesCount** field responsible for control user loses. When ads counter equals 0 user will be watch Ads Reward video.
4) Also this controller has **UnityEvents**. The names of events describe their place of call.



11. There is container with checkers rules.

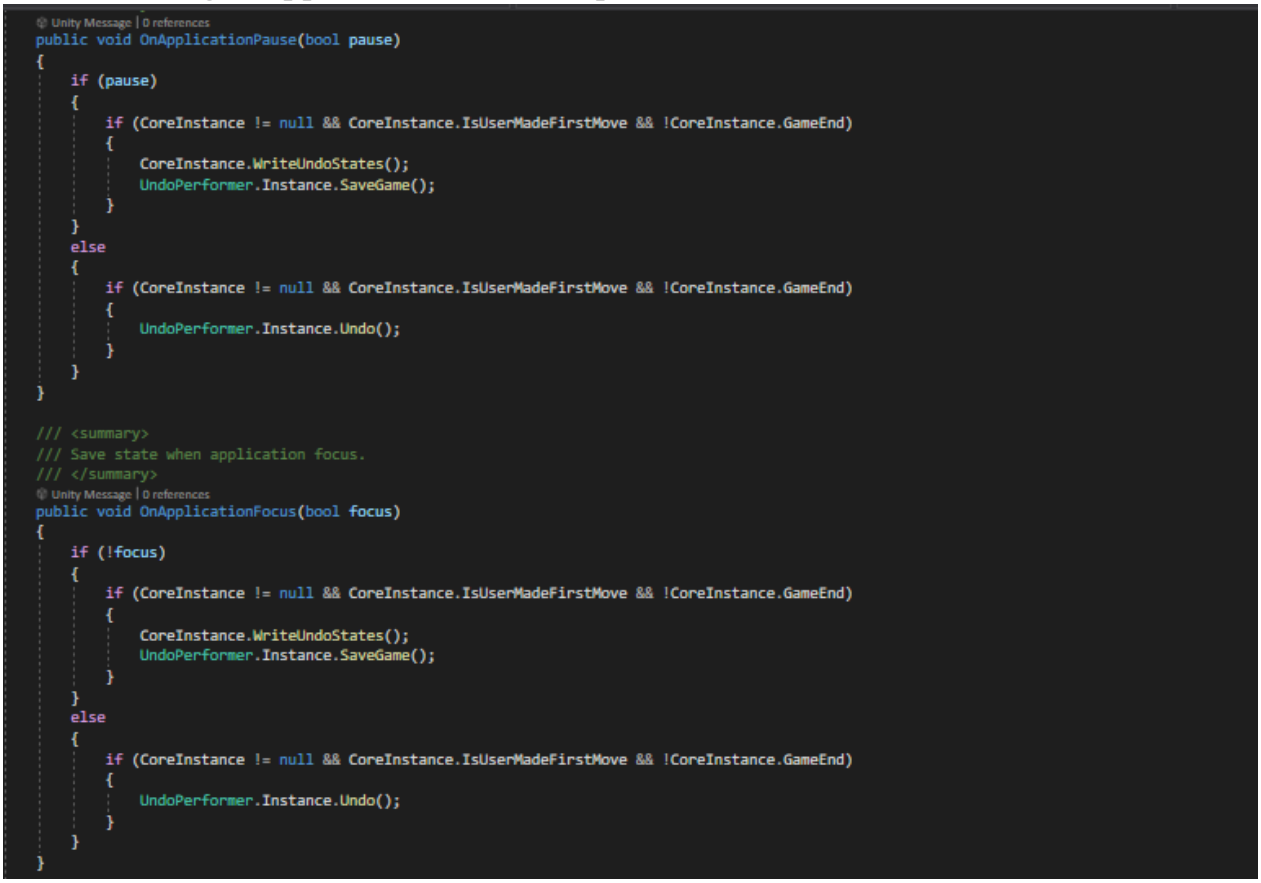


Each rule has options:



You can modifying it by your own using pre-setuped examples.

12. Game has undo feature and continue last game session.
State is saving in application focus and pause callbacks:



```
Unity Message | 0 references
public void OnApplicationPause(bool pause)
{
    if (pause)
    {
        if (CoreInstance != null && CoreInstance.IsUserMadeFirstMove && !CoreInstance.GameEnd)
        {
            CoreInstance.WriteUndoStates();
            UndoPerformer.Instance.SaveGame();
        }
    }
    else
    {
        if (CoreInstance != null && CoreInstance.IsUserMadeFirstMove && !CoreInstance.GameEnd)
        {
            UndoPerformer.Instance.Undo();
        }
    }
}

/// <summary>
/// Save state when application focus.
/// </summary>
Unity Message | 0 references
public void OnApplicationFocus(bool focus)
{
    if (!focus)
    {
        if (CoreInstance != null && CoreInstance.IsUserMadeFirstMove && !CoreInstance.GameEnd)
        {
            CoreInstance.WriteUndoStates();
            UndoPerformer.Instance.SaveGame();
        }
    }
    else
    {
        if (CoreInstance != null && CoreInstance.IsUserMadeFirstMove && !CoreInstance.GameEnd)
        {
            UndoPerformer.Instance.Undo();
        }
    }
}
```

All other information will be provided in video-reviews.
Play for fun ;) And thank you very much for buying my assets.

E-mail: support@selvassets.ltd.ua
Website: http://selvassets.ltd.ua
See others my assets: https://assetstore.unity.com/publishers/34779