



MOBILE PUSH NOTIFICATIONS



1. WHY DO YOU NEED TO USE THIS PLUGIN

- Increase your game engagement by sending notifications.
- Schedule local notifications with a single line of code.
- Click callback with custom message for each notification to track app sessions started by notification press.
- Custom notification icons
- Custom notification text.
- Device restart support
- Works for Android and iOS without any changes.
- Full code and demo scene included.
- Works with Unity 2019 and above with Free, Plus or Pro license.
- Requires Mobile Notifications package from Unity.



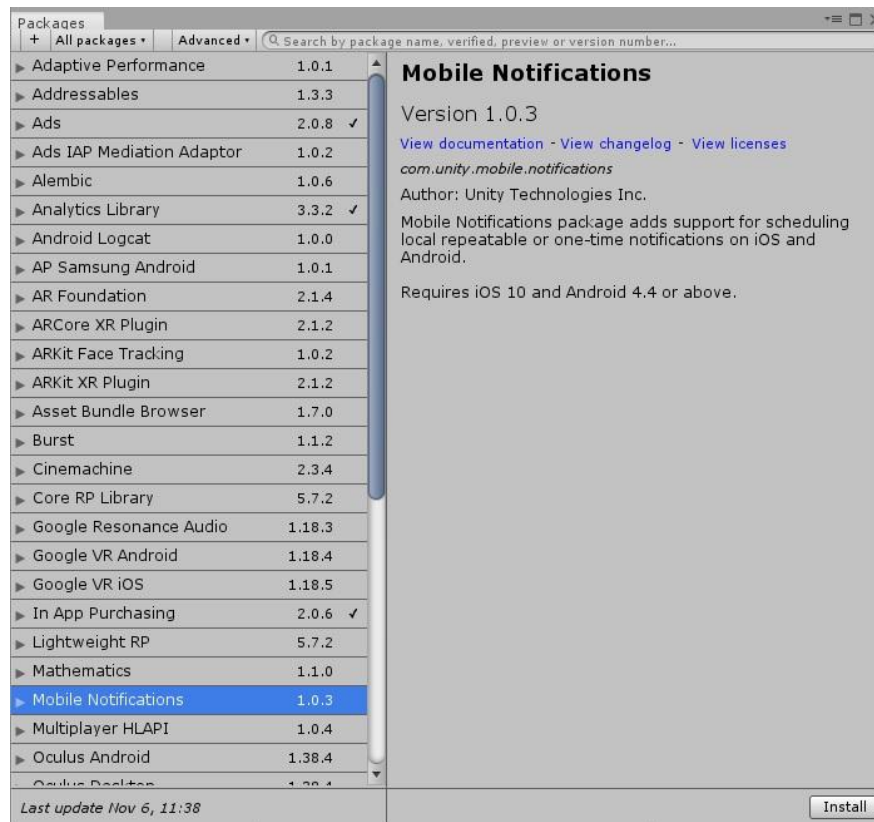
2. CURRENTLY SUPPORTED PLATFORMS

- **Android**
- **iOS**



3. INSTALL MOBILE NOTIFICATIONS

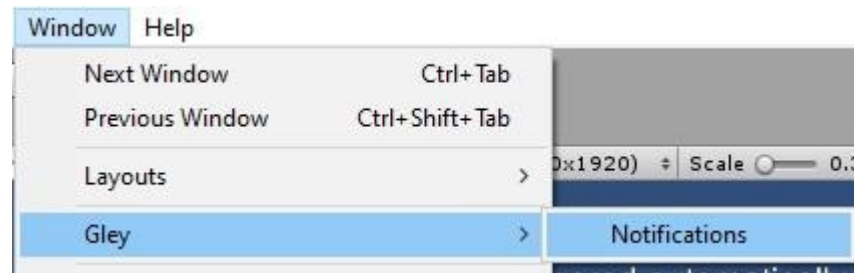
Go to **Window->Package Manager** select **All Packages** and install **Mobile Notifications**.



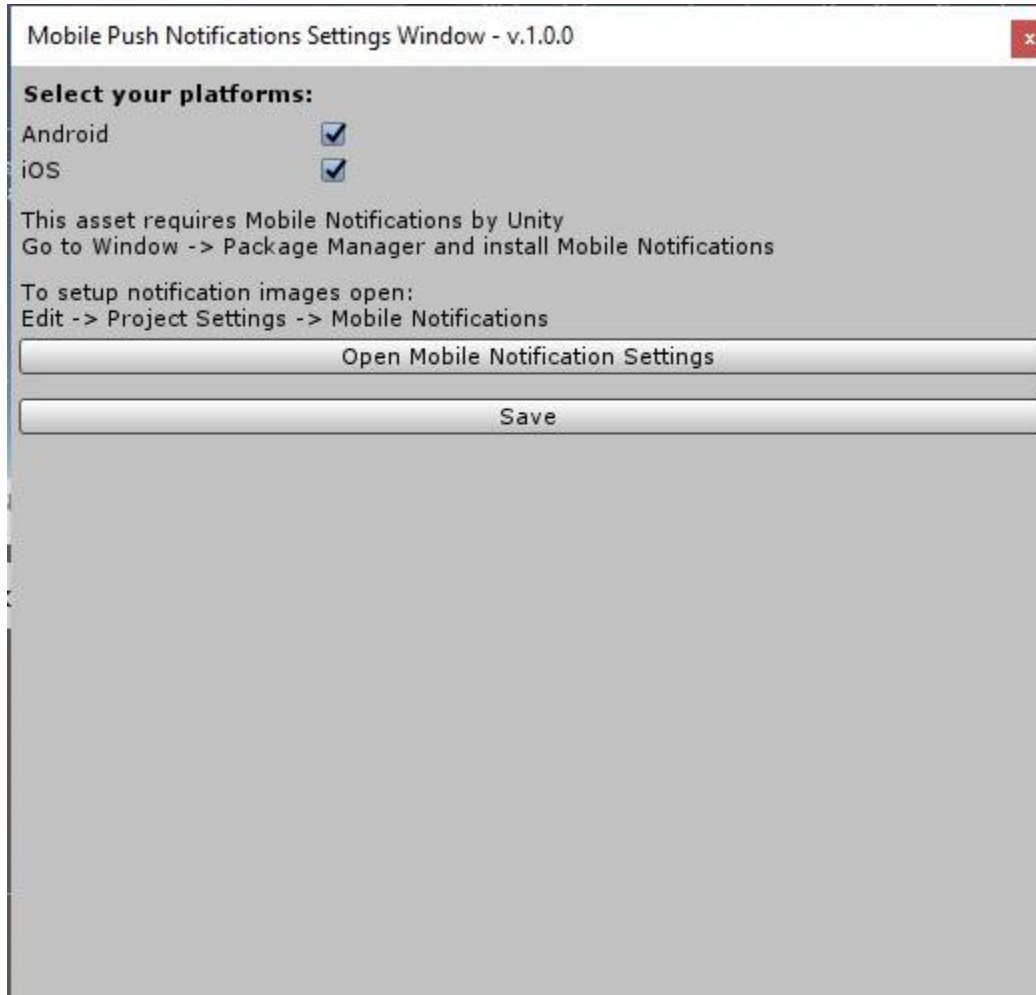


4. SETUP GUIDE

- Import **Gley Mobile Push Notifications Plugin** into Unity.
- Go to **Window->Gley->Notifications** to open the Settings Window.



- Settings Window will open



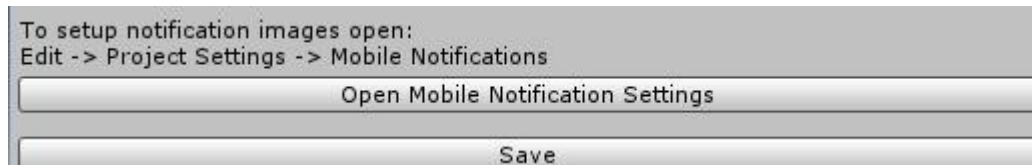


Notification Setup

- Select Platforms:



- Open Unity Mobile Notifications Settings from:
- **Edit -> Project Settings -> Mobile Notifications**
- or press the **Open Mobile Notification Settings** from Settings Window





Notification Setup Android

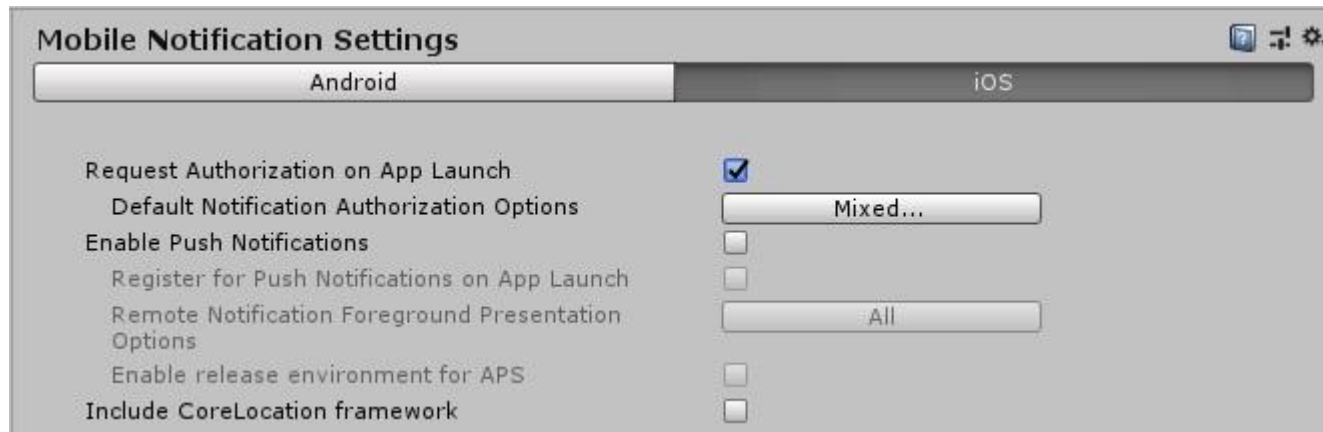
- Enable **Reschedule Notifications on Device Restart** to be able to send notifications even after device restart
- Select custom icons, small and large. If no custom icons are selected, app icon will be used

The screenshot shows the 'Mobile Notification Settings' window for an Android application. The 'Android' tab is selected. The 'Reschedule Notifications on Device Restart' checkbox is checked. The 'Use Custom AndroidActivity' checkbox is unchecked, and the 'Custom Android Activity Name' field contains the text 'com.unity3d.player.UnityPlayerActivity'. A text box provides instructions: 'Only icons added to this list or manually added to the `res/drawable` folder can be used by notifications. Small icons can only be composed simply of white pixels on a transparent backdrop and must be at least 48x48 pixels. Large icons can contain any colors but must be not smaller than 192x192 pixels.' Below this, the 'Notification icons' section shows two entries. The first entry has an identifier 'icon_0' and type 'Small Ico+', with a preview of a black square containing a white 'G' and a 'Select' button. The second entry has an identifier 'icon_1' and type 'Large Ico+', with a preview of a red square containing a white bell icon and a 'Select' button. There are also '+' and '-' buttons at the bottom right of the icon list.



Notification Setup iOS

- Enable **Request Authorization on App Launch** to request notification permission





5. USER GUIDE

- **GleyNotifications.Initialize();**
This method will create a notification channel and will cancel all pending notifications. It should be called every time user launches the app.
- **GleyNotifications.Initialize(false);**
This method will not cancel any pending notifications. They will be shown even if user is inside the app.


```
//title > Title of the notification  
//text > Content of the notification  
//timeDelayFromNow > delay to display the notification, this delay will be added to current time  
//smallIcon > name of the custom small icon from Mobile Notification Settings  
//largeIcon > name of the custom large icon from Mobile Notification Settings  
//customData > this data can be retrieved if the users opens app from notification
```
- **GleyNotifications.SendNotification(string title, string text, System.TimeSpan timeDelayFromNow, string smallIcon = null, string largeIcon = null, string customData = "")**
This method is used to schedule a notification.



5. USER GUIDE

//title > Title of the notification

//text > Content of the notification

//timeDelayFromNow > delay to display the notification, this delay will be added to current time

//repeatInterval > time until the next notifications will be sent.

//smallIcon > name of the custom small icon from Mobile Notification Settings

//largeIcon > name of the custom large icon from Mobile Notification Settings

//customData > this data can be retrieved if the users opens app from notification

- **GleyNotifications.SendRepeatNotification(string title, string text, System.TimeSpan timeDelayFromNow, System.TimeSpan repeatInterval, string smallIcon = null, string largeIcon = null, string customData = "")**

This method is used to schedule a notification.

// returns > the custom data sent to notification or null if the app was not opened from notification

- **string GleyNotifications.AppWasOpenFromNotification()**

Check if current session was opened from notification tap.



6. PLAYMAKER SUPPORT

- **Supported Playmaker Actions:**
 - **InitializeNotifications**
 - **AppWasOpenFromNotification**
 - **SendNotification**

The above Playmaker actions behavior is equivalent with corresponding methods from Section 5 - User Guide.



7. BOLT SUPPORT

- **Supported Bolt Actions:**
 - **InitializeNotifications**
 - **AppWasOpenFromNotification**
 - **SendNotification**

The above Bolt actions behavior is equivalent with corresponding methods from Section 5 - User Guide.



8. GAME FLOW SUPPORT

- **Supported Game Flow Actions:**
 - **InitializeNotifications**
 - **AppWasOpenFromNotification**
 - **SendNotification**

The above Game Flow actions behavior is equivalent with corresponding methods from Section 5 - User Guide.



9. EXAMPLE

You can find the example test scene here:

Assets/GleyPlugins/Notifications/Example/TestNotifications.unity

How to use the scene:

- Enter a time in minutes, press Send Notification and a notification will be displayed after time expires.
- If you close the app a notification will be sent after 1 minute
- When you open the app from notification a custom message will be displayed on screen.
- Check TestNotifications.cs for details.

When you minimize this app a notification will be triggered automatically after 1 minute

Enter time in minutes

Send Notification



Version 1.2.0 / 2021