UniLWP.Droid Documentation

Release 0.0.2 preview.2

FinGameWorks (Haotian Zheng)

Nov 30, 2020

General

1.2 Features 2 1.3 Version Comparison 4 1.4 Demo 4 1.5 Legacy Documentation 4 1.5 Legacy Documentation 4 2 Structure 5 3 Android Plugin 7 3.1 Android Library 7 3.2 Android Module 5 4 Customize Look and Feel 11 4.1 I con, Name, Metadata 11 4.2 Advanced Behavior 12 4.3 Native User Interface 13 5.1 Simple (Default) 15 5.2 Advanced (Optional) 15 6 Preparation 17 7.1 Free Version 16 7.2 Store Version 24 8 Listen To Callbacks 33 8.1 Callbacks In Your Own Implementation 34 9 Behavior 35 10 Export 35 10.1 One-Click 37	1	Introduction	3
1.3 Version Comparison 4 1.4 Demo 4 1.5 Legacy Documentation 4 1.5 Legacy Documentation 4 2 Structure 5 3 Android Plugin 7 3.1 Android Library 7 3.2 Android Module 5 4 Customize Look and Feel 11 4.1 Icon, Name, Metadata 11 4.2 Advanced Behavior 12 4.3 Native User Interface 13 4.3 Native User Interface 15 5 Build Pipeline 15 5.1 Simple (Default) 15 5.2 Advanced (Optional) 15 6 Preparation 17 7.1 Free Version 19 7.2 Store Version 19 7.2 Store Version 31 8.1 Callbacks 31 8.2 Trigger Callbacks 31 8.2 Trigger Callbacks In Your Own Implementation 34		1.1 Requirements	3
1.4 Demo 4 1.5 Legacy Documentation 4 2 Structure 4 3 Android Plugin 7 3.1 Android Library 7 3.2 Android Module 7 3.2 Android Module 7 4.1 Icon, Name, Metadata 11 4.2 Advanced Behavior 12 4.3 Native User Interface 12 5 Build Pipeline 15 5.1 Simple (Default) 15 5.2 Advanced (Optional) 15 6 Preparation 17 7.1 Free Version 15 7.2 Store Version 24 8 Listen To Callbacks 31 8.1 Callbacks In Your Own Implementation 34 9 Behavior 35 10 Export 37 10.1 One-Click 37		1.2 Features	3
1.5 Legacy Documentation 4 2 Structure 5 3 Android Plugin 7 3.1 Android Library 7 3.2 Android Module 7 3.2 Android Module 7 3.2 Android Module 7 4 Customize Look and Feel 11 4.1 Icon, Name, Metadata 11 4.2 Advanced Behavior 12 4.3 Native User Interface 13 5 Build Pipeline 14 5.1 Simple (Default) 15 5.2 Advanced (Optional) 15 6 Preparation 17 7.1 Free Version 19 7.2 Store Version 19 7.1 Free Version 31 8.1 Callbacks 33 8.1 Callbacks Types And Registration 31 8.2 Trigger Callbacks In Your Own Implementation 34 9 Behavior 35 10 Export 37 10		1	4
2 Structure 5 3 Android Plugin 7 3.1 Android Library 7 3.2 Android Module 7 3.2 Android Module 7 3.2 Android Module 7 4 Customize Look and Feel 11 4.1 Icon, Name, Metadata 11 4.2 Advanced Behavior 13 4.3 Native User Interface 13 5 Build Pipeline 15 5.1 Simple (Default) 15 5.2 Advanced (Optional) 15 6 Preparation 17 7.1 Free Version 15 7.2 Store Version 15 8 Listen To Callbacks 31 8.1 Callback Types And Registration 31 8.2 Trigger Callbacks In Your Own Implementation 34 9 Behavior 35 10 Export 35 10.1 One-Click 37		1.4 Demo	4
3 Android Plugin 7 3.1 Android Library 7 3.2 Android Module 7 4 Customize Look and Feel 11 4.1 Icon, Name, Metadata 11 4.2 Advanced Behavior 12 4.3 Native User Interface 13 5 Build Pipeline 15 5.1 Simple (Default) 15 5.2 Advanced (Optional) 15 6 Preparation 17 7 Import & Setup 19 7.1 Free Version 19 7.2 Store Version 24 8 Listen To Callbacks 31 8.1 Callback Types And Registration 31 8.2 Trigger Callbacks In Your Own Implementation 32 9 Behavior 33 310 Export 33 10.1 One-Click 37		1.5 Legacy Documentation	4
3.1 Android Library 7 3.2 Android Module 7 3.2 Android Module 9 4 Customize Look and Feel 11 4.1 Icon, Name, Metadata 11 4.2 Advanced Behavior 12 4.3 Native User Interface 12 4.3 Native User Interface 13 5 Build Pipeline 15 5.1 Simple (Default) 15 5.2 Advanced (Optional) 15 6 Preparation 17 7 Import & Setup 19 7.1 Free Version 19 7.2 Store Version 24 8 Listen To Callbacks 31 8.1 Callback Types And Registration 31 8.2 Trigger Callbacks In Your Own Implementation 34 9 Behavior 35 10 Export 37 10.1 One-Click 37	2	Structure	5
3.2 Android Module 9 4 Customize Look and Feel 11 4.1 Icon, Name, Metadata 11 4.2 Advanced Behavior 12 4.3 Native User Interface 13 5 Build Pipeline 15 5.1 Simple (Default) 15 5.2 Advanced (Optional) 15 6 Preparation 17 7 Import & Setup 19 7.1 Free Version 19 7.2 Store Version 31 8.1 Callbacks 31 8.1 Callbacks In Your Own Implementation 34 9 Behavior 35 10 Export 37 10.1 One-Click 37	3	Android Plugin	7
4 Customize Look and Feel 11 4.1 Icon, Name, Metadata 11 4.2 Advanced Behavior 12 4.3 Native User Interface 13 5 Build Pipeline 15 5.1 Simple (Default) 15 5.2 Advanced (Optional) 15 6 Preparation 17 7 Import & Setup 19 7.1 Free Version 19 7.2 Store Version 24 8 Listen To Callbacks 31 8.1 Callbacks In Your Own Implementation 34 9 Behavior 35 10 Export 37 10.1 One-Click 37		3.1 Android Library	7
4.1 Icon, Name, Metadata 11 4.2 Advanced Behavior 13 4.3 Native User Interface 13 5 Build Pipeline 15 5.1 Simple (Default) 15 5.2 Advanced (Optional) 15 6 Preparation 17 7 Import & Setup 19 7.1 Free Version 19 7.2 Store Version 24 8 Listen To Callbacks 31 8.1 Callback Types And Registration 31 8.2 Trigger Callbacks In Your Own Implementation 34 9 Behavior 35 10 Export 37 10.1 One-Click 37		3.2 Android Module	9
4.1 Icon, Name, Metadata 11 4.2 Advanced Behavior 13 4.3 Native User Interface 13 5 Build Pipeline 15 5.1 Simple (Default) 15 5.2 Advanced (Optional) 15 6 Preparation 17 7 Import & Setup 19 7.1 Free Version 19 7.2 Store Version 24 8 Listen To Callbacks 31 8.1 Callback Types And Registration 31 8.2 Trigger Callbacks In Your Own Implementation 34 9 Behavior 35 10 Export 37 10.1 One-Click 37	4	Customize Look and Feel	11
4.3 Native User Interface		4.1 Icon, Name, Metadata	11
5 Build Pipeline 15 5.1 Simple (Default) 15 5.2 Advanced (Optional) 15 6 Preparation 17 7 Import & Setup 19 7.1 Free Version 19 7.2 Store Version 24 8 Listen To Callbacks 31 8.1 Callback Types And Registration 31 8.2 Trigger Callbacks In Your Own Implementation 34 9 Behavior 35 10 Export 37 10.1 One-Click 37			13
5.1 Simple (Default) 15 5.2 Advanced (Optional) 15 6 Preparation 17 7 Import & Setup 19 7.1 Free Version 19 7.2 Store Version 24 8 Listen To Callbacks 31 8.1 Callback Types And Registration 31 8.2 Trigger Callbacks In Your Own Implementation 34 9 Behavior 35 10 Export 37 10.1 One-Click 37		4.3 Native User Interface	13
5.1 Simple (Default) 15 5.2 Advanced (Optional) 15 6 Preparation 17 7 Import & Setup 19 7.1 Free Version 19 7.2 Store Version 24 8 Listen To Callbacks 31 8.1 Callback Types And Registration 31 8.2 Trigger Callbacks In Your Own Implementation 34 9 Behavior 35 10 Export 37 10.1 One-Click 37	5	Build Pipeline	15
5.2 Advanced (Optional) 15 6 Preparation 17 7 Import & Setup 19 7.1 Free Version 19 7.2 Store Version 19 8 Listen To Callbacks 31 8.1 Callback Types And Registration 31 8.2 Trigger Callbacks In Your Own Implementation 34 9 Behavior 35 10 Export 37 10.1 One-Click 37		5.1 Simple (Default)	15
7 Import & Setup 19 7.1 Free Version 19 7.2 Store Version 24 8 Listen To Callbacks 31 8.1 Callback Types And Registration 31 8.2 Trigger Callbacks In Your Own Implementation 34 9 Behavior 35 10 Export 37 10.1 One-Click 37			15
7.1 Free Version 19 7.2 Store Version 24 8 Listen To Callbacks 31 8.1 Callback Types And Registration 31 8.2 Trigger Callbacks In Your Own Implementation 34 9 Behavior 35 10 Export 37 10.1 One-Click 37	6	Preparation	17
7.1 Free Version 19 7.2 Store Version 24 8 Listen To Callbacks 31 8.1 Callback Types And Registration 31 8.2 Trigger Callbacks In Your Own Implementation 34 9 Behavior 35 10 Export 37 10.1 One-Click 37	7	Import & Setup	19
8 Listen To Callbacks 31 8.1 Callback Types And Registration 31 8.2 Trigger Callbacks In Your Own Implementation 34 9 Behavior 35 10 Export 37 10.1 One-Click 37			19
8.1 Callback Types And Registration 31 8.2 Trigger Callbacks In Your Own Implementation 34 9 Behavior 35 10 Export 37 10.1 One-Click 37		7.2 Store Version	24
8.1 Callback Types And Registration 31 8.2 Trigger Callbacks In Your Own Implementation 34 9 Behavior 35 10 Export 37 10.1 One-Click 37	8	Listen To Callbacks	31
8.2 Trigger Callbacks In Your Own Implementation 34 9 Behavior 35 10 Export 37 10.1 One-Click 37			31
10 Export 37 10.1 One-Click 37			34
10.1 One-Click	9	Behavior	35
10.1 One-Click	10	Export	37
			37
		10.2 External Modification	37

11 Ads	39
12 Preference	43
13 Simulator	45
14 Migration 14.1 0.0.2	47 47
15 Changelog 15.1 0.0.2 (preview.2) 15.2 0.0.2 (preview.1) 15.3 0.0.1	

Version 0.0.2 preview.2 Date Nov 30, 2020 Contact justzht+unilwp@gmail.com

Introduction

UniLWP.Droid is a live wallpaper (LWP) solution for Unity to run on Android. It is used in most live wallpaper apps from FinGameWorks, including Metropolis, Vortex, and Diorama.

UniLWP.Droid is built with customization in mind. It works with Unity's default apk build pipeline, but what makes it different from other solutions is an alternative workflow provided to deeply customize the look and bahavior of your apps through external modifications.

1.1 Requirements

- Unity 2019.3 and up (certain features require Unity 2020 or more recent releases)
- Android 7.0 and up (API 24+)
- Android programming experience needed (only if advanced build mode is in use)

1.2 Features

- C# callbacks to build data-driven live wallpapers
 - Unlock state (Locked / Ambient / Screen-on / Unlocked)
 - Dark mode
 - Wallpaper scroll offset, with page count and progress on each page
 - Window insets (to avoid overlapped UI rendering with device notch)
 - Is in wallpaper / preview / activity mode
- Unity Ads support
- Screen saver (DayDream) support
- Non-intrusive integration

- Unity Cloud Build support
- Customization friendly design
 - Re-building project would still maintain your external modifications made using Android Studio, including java files, xml resources, and gradle dependencies

1.3 Version Comparison

UniLWP.Droid has two variants.

- One is UniLWP.Droid.Free, a free plugin in UPM format.
- Another is UniLWP.Droid.Store, an Asset-Store-listed plugin with more features.

UniLWP.Droid	Free	Store
Unity as Live Wallpaper		
Default Build Pipeline (One-Click Apk)		
Callbacks (Lock State, Scroll Offset, etc)		
Touch Events		
Modular Customization	(You need to do it yourself)	(Editor tools provided)
Advanced Build Workflow		

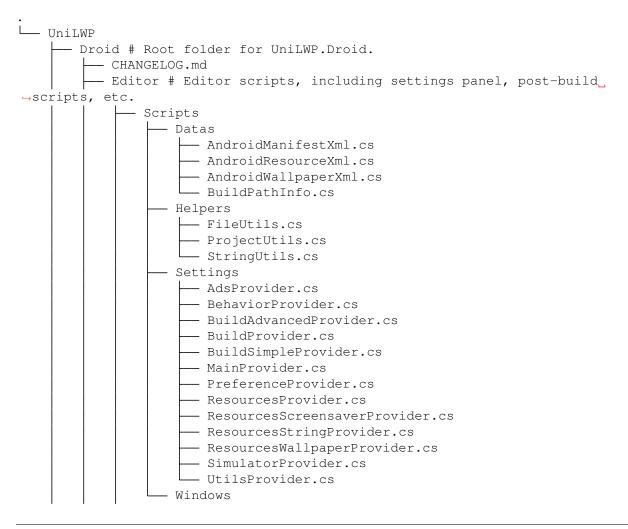
1.4 Demo

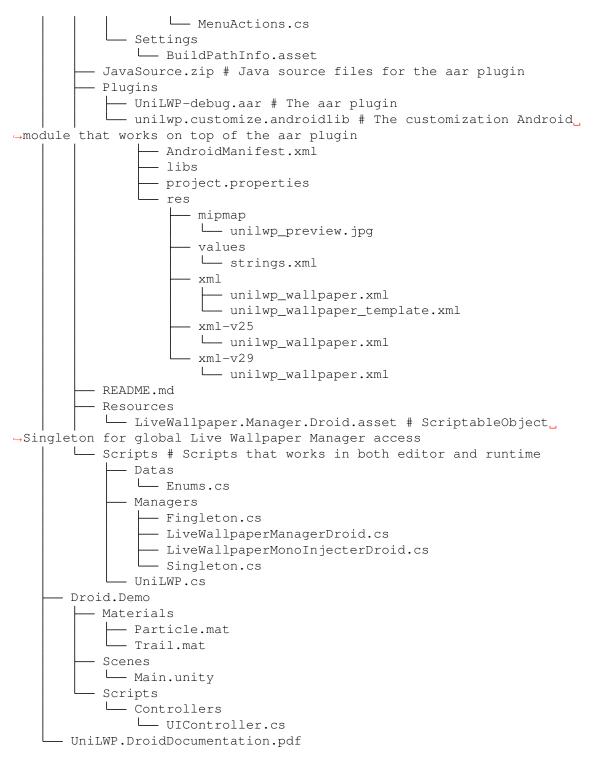
1.5 Legacy Documentation

For an earlier version of UniLWP.Droid documentation, please refer to Google Doc.

Structure

After the import of the store (paid) version, you should have those files in the Assets/FinGameWorks folder:





27 directories, 46 files

Android Plugin

The scope of this page is within Assets/FinGameWorks/UniLWP/Droid/Plugins and the source files that contributed to the plugins in that folder.

3.1 Android Library

The Android Library here refers to the UniLWP-(debug|release).aar file located in Assets/ FinGameWorks/UniLWP/Droid/Plugins folder and the source code to compile it.

The Android Library is responsible for major work of converting an ordinary Unity app into a live wallpaper, including lifecycle modification, manifest declaration, event hijacking, and C#-Java communications.

Note: The source code project as a Android Library gradle project, is included as a zip file (JavaSource.zip), which you can import using Android Studio to compile your own aar file with own logic.

3.1.1 Initialization

UniLWP is designed to initialize even before the Unity player itself for a total control of the player instance. To achieve this goal, it deploys an early-init technique called ContentProvider, which is mainly done in the LiveWallpaperInitProvider.java file.

Tip: For how Content Provider can be called even before Application.onCreate(), there is a detailed explaination on the Firebase Dev Blog as the Firebase SDK also deploys such techniques.

In the onCreate() method of LiveWallpaperInitProvider, UniLWP calls LiveWallpaperManager. getInstance().Init(Context context) with the acquired context, which would later be used to create the Unity player. This way, UniLWP is able to extend its lifespan beyond the Unity player and consequently maintain the control to it no matter the app is launched by system in activity mode or wallpaper mode. The drawback though, is that the Unity player, by default, would be initialized at the beginning, which would let the system took a performance and memory hit early. However, this behavior can be easily tweaked to fit your own need, with the possibility to launch Unity instance when you needed (and destroy it when you don't) through the flag unilwp.behavior.isolate and those Java APIs:

```
2
3
4
5
6
7
8
9
```

1

```
public enum LiveWallpaperManager implements IUnityPlayerLifecycleEvents {
    public static LiveWallpaperManager getInstance() {
        return INSTANCE;
    }
    public void LoadUnity(Context context); // Load Unity instance
    public void UnloadUnity(); // Unload Unity
    public void Init(Context applicationContext); // Load UniLWP, and optionally_
        JUniLWP if unilwp.behavior.isolate is false
        public void DeInit(Context applicationContext); // Unload UnitWP and Unity
    }
}
```

3.1.2 Display Target

UniLWP uses a single Unity instance for multiple render targets, including the surface view in activities, wallpaper engine in wallpaper services, and even surface view in screen saver (day dream) services. This shared instance ensures that the contents on those render targets are all the same, the only difference is the rendering resolution. However, UniLWP is required to switch render targets to only the currently active one, or the user would only see a static image.

To make this happen, UniLWP's java plugin calls this method of UnityPlayer when a switch is needed:

```
package com.unity3d.player;
1
    public class UnityPlayer extends FrameLayout implements IUnityPlayerLifecycleEvents,...
2

→com.unity3d.player.f {

       public boolean displayChanged(int var1, Surface var2) {
3
           if (var1 == 0) {
4
                this.mMainDisplayOverride = var2 != null;
5
                //...
6
7
            }
           return this.updateDisplayInternal(var1, var2);
8
       }
9
   }
10
```

The method, displayChanged(int var1, Surface var2), accept two parameters, the first being the display index, while the second is the java Surface object. Index 0 refers to the main display, as you can see in the highlighted code block. If we supply a value that is bigger than 0, Unity would be updated with the new surface being a Display that you can acquire through the C# API Display[] Display.displays.

However, we are not gonna use this multiple display setup. Instead, UniLWP only uses index 0 for display switches to make existing cameras work without additional efforts. Anytime a new surface is visible, UniLWP calls displayChanged(0, newSurface) so Unity renders to that surface.

If you are using a customized activity or service, chances are you want your customized surface to be registered into this switch logic. Please refer to *Customize Look and Feel* for more information.

3.1.3 Callbacks

One of UniLWP's unique features is the ability to monitor native events to develop data-driven wallpapers that adapts to dark mode, lock screen, and other device environment changes.

Tip: For how to use callbacks in your C# code, please refer to Listen To Callbacks

Callbacks are implemented via the LiveWallpaperListener Java interface and LiveWallpaperListenerManager Java class.

```
public enum LiveWallpaperListenerManager {
    // called in C# via AndroidJavaObject
    protected void setEventListener(LiveWallpaperListener eventListener) {
        this.eventListener = eventListener;
        // report initial status to Unity
        // ...
    }
    private LiveWallpaperListener eventListener;
}
```

When the Unity instance is initialized in Java by LiveWallpaperManager, the C# part of UniLWP, LiveWallpaperManagerDroid will be awaken along with the Unity instance, and before any scene loading, it will call setEventListener(LiveWallpaperListener eventListener) with a C# Android-JavaProxy. Any Android native events would be synced through this interface and dispatched to any listener attached.

It is also suggested that you attach listeners in Monobehavior's Start() or Awake() function to only register once after the internal setup of LiveWallpaperManagerDroid.

3.1.4 Behaviour & Flags

1

2

3

4

5

6

7

8

UniLWP comes with a handful options to tweak its native behavior. Certain times you want UniLWP to log in detail, other times you don't. The same applies to Unity initialization, initial variable value, setting button event, etc. All those fields are organized into a single Java class LiveWallpaperConfig, and the values would be retrieved from meta-data tags in the merged final AndroidManifest.xml.

Tip: For how to set behaviour flags in your C# code, please refer to Listen To Callbacks

3.2 Android Module

The Android Module here refers to the unilwp.customize.androidlib folder located in Assets/ FinGameWorks/UniLWP/Droid/Plugins folder. It is essentially an Android Library project, with the extension androidlib to identify itself as a plugin to Unity.

Note: The androidlib extension is an Unity 2020.x feature addition that, if applied, can make your plugin folder work anywhere. In other words, plugins folders with or without this extension will work if placed in the Assets/Plugins/Android folder, however, on 2020+ and with this extension, the plugin folder can be moved to anywhere within Assets.

In this sense, the Android Module will still largely work on 2019.3 if you move it into the global folder Assets/ Plugins/Android, but that is pretty in contrast with the 'non-invasive' design philosophy of UniLWP. If you really need it to work on 2019.3, please also modify the plugin path field in setting scripts so the editor UI will point to the correct file, but it won't receive offical support.

3.2.1 Resources

As described before, this folder is in a standard Android Library project layout. Aside from AndroidManifest. xml, values are mostly stored in res/xml and res/values folder. In the final compile stage, these values will be merged into the app and replace any previous or default value.

3.2.2 Goal

Why adding a new folder to the plugin folder when UniLWP already has an AAR file in it? Obviously AAR file cannot be modified easily after it is compiled. While the behavior can be redirected through meta-data entries mentioned earlier, all those resources files (names, images) are static-referenced.

An Android Library project that exposes its content to the outside is therefore more suitable to store these values. UniLWP since 0.0.2 comes with additional editor scripts to manage values within the editor so you don't have to use a xml editor or Android Studio.

Customize Look and Feel

4.1 Icon, Name, Metadata

Where to find settings

Unity Application Icon		
Project Settings	م	:
 Project Audio Device Simulator Editor Graphics Input Manager Package Manager Physics Physics 2D Player Preset Manager Quality Script Execution Order Services 	Player Company Name Product Name Version Default Icon Default Cursor	FinGameWorks UniLWP.Droid 0.0.0.1 None (Texture 2D) Select None (Texture 2D) Select
Ads Analytics Cloud Build Cloud Diagnostics Collaborate In-App Purchasing Tags and Layers TextMesh Pro Time Version Control XR Plugin Management ♥ UniLWP ♥ Droid Ads Integration	Cursor Hotspot	X 0 Y 0

Available Build Type Simple | Advanced

4.1.1 Application Icon

To change the application icon, you can just use the Unity Project Settings/Player/Icon settings.

4.1.2 Wallpaper Preview

To change the wallpaper preview icon, replace the jpg file within the unilwp.customize.androidlib folder. The exact path should be res/mipmap/unilwp_preview.jpg. For more about the unilwp.customize. androidlib folder, please refer to *Android Plugin*.

Note: Optionally, since 0.0.2 preview.2, you can utilize the mipmap manage tool under in Project Settings (UniLWP/ Resources/Mipmap) to upload images into the mipmap folder.

4.2 Advanced Behavior

Available Build Type Simple | Advanced

4.3 Native User Interface

Available Build Type Advanced Only

Build Pipeline

5.1 Simple (Default)

5.2 Advanced (Optional)

Preparation

Before import UniLWP.Droid, make sure that:

- You are using Unity 2019.3 and up.
- You have switched to Android build target.
- You have an Android testing device running Android 7.0 or later.

Then, proceed to Import & Setup.

Import & Setup

7.1 Free Version

Where is the package manager											
É	Unity	File	Edit	Assets	GameObj	ect	Compo	onent	Window	Help	
• • •	● ↔ ∽	2		×	Pivot 🤅	🔁 Glo	bal 🖽		Minimize Zoom		ЖM
	archy ⊶ All ≪) Untitle	'n			3		ŧ Scene isplay 1	■ ▼ 192	Bring All to Panels	o Front	>
	🖓 Mair 💮 Direo	n Came				•			Layouts Collaborat	e	>
									Asset Stor Package M		
									Asset Man	agement	>

while at the top of the Unity window in Windows.

How to add UPM packages	
 Package Manager Packages: Unity Registry - 	Sort: Name 🗸 🔻
Add package from disk	4.2.6
Add package from tarball	3.0.2
Add package from git URL	3.1.6
▶ 2D Sprite	1.0.0
▶ 2D SpriteShape	4.1.4

Fig. 2: If you cannot see the dropdown list, drag the title of package manager panel to un-dock the window, and then try again.

View	more	info	about	the	package
	onsole		-	-	a i
+ *	 Packages > Uni 	WD Dealed Free	٩		* 🕈 🖈 🕫 10
🔹 🚘 Assets	 Packages > On CHANGELC 				
Scenes	C⊐ Editor ■ LICENSE				
Custom NUr JetBrains Rie	it 📄 package				
🕞 🖿 Test Framev	vork ERUntime				
TextMeshPr	0.				
UniLWP.Dro					
🕞 🖿 Unity Ul	1. S. 11. (S.1)				
Visual Studie					•
🖬 Package Manag					1
The second	Project - Sort: Name 4 -			\$ 9	
The second s	(a.k.a Justin Fincher)		P.Droid Free Preview		
Unity Technolo	THE MARKET ACCORDING TO A DESCRIPTION OF		eng (a.k.a Justin Fincher)		
JetBrains Rider E		The second s	0.1-preview.1 git mentation • View changelog • 1		
Test Framework	1.1.16 🕢	74 500-000 (000-000-000-000-000-000-000-000-			TO AND DEPENDENCE
▶ TextMeshPro	3.0.1 🕤		ion of UniLWP.Droid. Works as a allpaper through one-click build.		
► Timeline	1.3.6 🗸		d was shrinked down in terms of For Java source, please check	features, please check LIC	CENSE for
Unity Collaborat	e 1.3.9 🗸	https://gith	ub.com/JustinFincher/UniLWP.D	roid.Free.JavaSource	Ţ
Last update Nov 25	, 17:07	2			Remove
N	ewly added files are groupp	ed and showr	n within the Packages sect	ion of the project panel	l

Ideal			project				settings
🖨 Asset Store 🛛 🛱 Scene	Inspector Pac	kage Upload	😎 Game	💠 Project Settings	🜻 Lighting		:
▼ Project Audio	Player						0 ≠ ≎
Device Simulator Editor		.				÷	^
Graphics Input Manager	Settings for Andro	ĥd					
Package Manager Physics		d Presentation					
Physics Physics 2D Player	Start in fullscree						
Preset Manager Quality	Render outside s Optimized Frame						
Script Execution Order	Resolution Scali Resolution Scali			Disabled			
Ads Analytics	Blit Type Supported Aspe			Auto			*
Cloud Build Cloud Diagnostics	Aspect Ratio Mo			Native Aspect Ratio			-
Collaborate In-App Purchasing	Orientation Default Orientati	on*		Auto Rotation			
Tags and Layers TextMesh Pro	Allowed Orienta Portrait	tions for Auto R	lotation	~			
Time Version Control	Portrait Upside			~			
XR Plugin Management	Landscape Ri Landscape Le			~			
🔻 Droid	Use 32-bit Displa						
Ads Integration Behavior	Disable Depth ar Render Over Nat						
Build Advanced Export	Show Loading In			Don't Show			-
One-Click Resources	* Shared setting be	tween multiple plat					
Screen Saver Strings	Splash Image						
Wallpaper Simulator	Other Setting						
Utils	▶ Publishing Se	ettings					•
	Project/	Player/Re	soluti	on and Presen	tation		

🚔 Asset Store 🛛 🛱 Scene	🛈 Inspector 🛛 Package Upload 🖙 Game	Project Settings Plighting :	
√ Project	Player	0 ± 4	2
Audio	Minimum API Level	Android 7.0 'Nougat' (API level 24)	
Device Simulator Editor	Target API Level	Automatic (highest installed)	
Graphics			
Input Manager	Configuration	IL2CPP 👻	
Package Manager	Scripting Backend		
Physics	Api Compatibility Level*		
Physics 2D Plaver	C++ Compiler Configuration	Debbg	
Preset Manager	Use incremental GC	×	
Quality	Mute Other Audio Sources*		
Script Execution Order	Target Architectures		
✓ Services Ads	ARMv7	×	
Adds Analytics	ARM64	~	
Cloud Build	Split APKs by target architecture (Experimen		
Cloud Diagnostics	Install Location	Automatic 👻	
Collaborate	Internet Access	Auto	
In-App Purchasing Tags and Layers	Write Permission	Internal	
TextMesh Pro	Filter Touches When Obscured		
Time	Sustained Performance Mode	~	
Version Control	Low Accuracy Location		
XR Plugin Management	Android TV Compatibility		
Droid			
Ads Integration			
Behavior			
⊤ Build	Scripting Define Symbols		
Advanced Export One-Click	UNILWP_AUTHOR;UNILWP_ADS		
Resources	Allow 'unsafe' Code		
Screen Saver	Use deterministic compilation		
Strings	Active Input Handling*	Input Manager (Old)	
Wallpaper Simulator			
Utils	Optimization Prebake Collision Meshes*		
	Frebake Collision Mesnes*		*
	Project/Player/O	ther Settings	

7.1.1 Import

The free version, UniLWP.Droid.Free, is hosted on GitHub as an UPM package.

- To import, first open package manager via menu Window/Package Manager.
- Then, at the top-left of the newly opened package manager window, you will find a plus sign. Click on it and select Add package from git URL...
- Paste the following URL into the field and press enter:

https://github.com/JustinFincher/UniLWP.Droid.Package.Free.git

• Unity should download and load UniLWP.Droid as an UPM-formatted dependency.

Or, if you are familiar with package. json, you are free to do paste this line into your json file:

7.1.2 Setup

Toggle Project Settings panel via menu path Edit/Project Settings...

- Go to Project/Player and adjust certain items:
 - In Player/Resolution and Presentation, make sure that both Optimized Frame Pacing and Render Over Native UI are unchecked.
 - In Player/Other Settings, make sure that both Mute Other Audio Sources and Filter Touches When Obscured are unchecked, the Minimal API Level is Android 7. 0 Nougat (API Level 24). You might also want to change Graphics APIs to OpenGLES3 only, but that is optional.
- Go to Project/Audio and adjust certain items:
 - Check Disable Unity Audio if you don't want your wallpaper to play sounds.

7.1.3 Build

It is the same as the default Unity build pipeline, that you only need to trigger a build through the default File/ Build And Run menu path. The plugin will be packed into the final apk file and be initialized as soon as possible to handle Unity lifecycles.

7.2 Store Version

Note: As Unity 2020.1 deprecated the Unity Asset Store web browser panel in favour of the package manager, you need to use package manager to import UniLWP.Droid on 2020.1+

Paid Version Import Guide

🗯 Unity	File	Edit	Assets	GameObject	Component	Window	Help	
● ● ₩ -13 - (5	2		×	∏ Pivot ⊕Glo	bal 😛	Minimize Zoom		ЖM
₩ € → <u>6</u> 5 Hierarchy ▼				3:1	Bal ∷ ≢ Scene isplay 1 🔻 192	Bring All to	o Front	
v & Untitle ⊕ Main				:		Panels		>
	ctional Li					Layouts		>
						Collaborat	te	
						Asset Stor	18	_
					1000	Package N	lanager	_
g. 3: Package Ma ile at the top of	-			enu Window. The dows.	menu bar would	Asset Mar		>
ile at the top of	the Unit	y windo	ow in Win			l be at the top		
ile at the top of w to find ass Package Manager	the Unit	y windo 9 packa	ow in Win ages you	dows. I bought before		l be at the top		
ile at the top of	the Unit	y windo e packa ort: Name :	ages you 2.2.2 ⊻	dows. I bought before	in package m d Materials	l be at the top		
Package Manager Packages: My A: Dunity Registry In Project I v My Assets	the Unit	y windo e packa ort: Name :	ages you 2.2.2 ↓	dows. ■ bought before ■ Clear Filters 10+1 Standar Rispat Momit	in package m d Materials st 18, 2015 asset	I be at the top	of the screet	
Package Manager Package Manager Packages: My As C Unity Registry IC Unity Registry IC Unity Registry IC Unity Registry IC Unity Registry IN Project 31 ✓ My Assets 32 Built-in 33 Visualizer Spect	the Unit et store	y windo e packa ort: Name	ages you $2.2.2 \pm$ $1.2.1 \pm$ $1.5 \mp$	dows. ■ bought before ■ Clear Filters 10+1 Standar <u>Rispat Momit</u> Version 2.0.0 - Augu <u>View in the Asset Sto</u> This is a limited selec	in package m d Materials st 18, 2015 asset e • Publisher Webs ion of	I be at the top	of the screet	
Package Manager Packages: My Assets	the Unit et store	y windo e packa ort: Name :ter	ages you $\downarrow \bullet$ Filters 2.0 \downarrow 1.2.1 \downarrow 1.0 \downarrow 1.0 \downarrow 2.1.0 \downarrow	dows. ■ bought before ■ Clear Filters 10+1 Standar <u>Rispat Momit</u> Version 2.0.0 - Augu <u>View in the Asset Sto</u>	in package m d Materials st 18, 2015 asset e • Publisher Webs ion of	I be at the top	of the screet	

How to import asset store packa	ges in package manager	
1 Inspector		a :
Package 'UniLWP.Droid - Unity Live Wall	paper For Android' Manifest	Edit View in Package Manager
Information		
Display name Ur	iiLWP:Droid - Unity Live Wallpaper For Android	1
Version 0.1	0.0	
Category To		
Description		
Introduction UniLWP.Droid is an Android live wallpaper frame Documentation Lite Version on GitHub Sample Features	work for Unity. With it, you can create Android Studio p e App	projects that compile to live wallpaper apps.
 One-click build to apk files, there is no manual Modular design that allows experienced Andro 	editing before or after the export/building process (so i id developers to customize your own wallpaper behavi ate, activity visible state, dark mode state, screen lock	or (manual editing needed)
 Open wallpaper service preview/activity from (C# side	•
■ Package Manager + ▼ Packages: My Assets ▼ Sort: Name ↓ ▼ Filter	Clear Filters	i 🔍 🗘 🗘
		Sumah o
UniLWP.Droid - Unity Live Wallpape 0.0 4	Unit WD Deside United live Wall	nomes Fest Andreid
▶ UniLWP.Droid - Unity Live Wallpape 0.0 🛓	UniLWP.Droid - Unity Live Wall FinGameWorks Version 0.0.0 - June 02, 2020 asset store	paper For Android
▶ UniLWP.Droid - Unity Live Wallpape 0.0 🛓	FinGameWorks	and addression and the stress locks
▶ UniLWP.Droid - Unity Live Wallpape 0.0 坐	FinGameWorks Version 0.0.0 - June 02, 2020 asset store	lisher Support
▶ UniLWP.Droid - Unity Live Wallpape 0.0 坐	FinGameWorks Version 0.0.0 - June 02, 2020 asset store View in the Asset Store Publisher Website Introduction UniLWP.Droid is an Android live wallpaper framework	lisher Support
▶ UniLWP.Droid - Unity Live Wallpape 0.0 🛓	FinGarneWorks Version 0.0.0 - June 02, 2020 asset store View in the Asset Store • Publisher Website • Publ Introduction UniLWP.Droid is an Android live wallpaper framewor Studio projects that compile to live wallpaper apps. More Images & Videos	lisher Support
▶ UniLWP.Droid - Unity Live Wallpape 0.0 坐	FinGameWorks Version 0.0.0 - June 02, 2020 asset store View in the Asset Store - Publisher Website - Publ Introduction UniLWP.Droid is an Android live wallpaper framewor Studio projects that compile to live wallpaper apps. More Images & Videos UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction UniLWP.Droid Introduction Introduction Introduction UniLWP.Droid Introduction Introduction Introduction UniLWP.Droid Introduction Intro	lisher Support
▶ UniLWP.Droid - Unity Live Wallpape 0.0 坐	FinGarneWorks Version 0.0.0 - June 02, 2020 asset store View in the Asset Store • Publisher Website • Publ Introduction UniLWP.Droid is an Android live wallpaper framewor Studio projects that compile to live wallpaper apps. More Images & Videos	Itsher Support
▶ UniLWP.Droid - Unity Live Wallpape 0.0 🛓	FinGameWorks Version 0.0.0 - June 02, 2020 asset store View in the Asset Store Publisher Website Introduction UniLWP.Droid is an Android live wallpaper framework Studio projects that compile to live wallpaper apps. More Images & Videos UniLWP.Droid UniWP.Droid UniWP.Droid Univer UniLWP.Droid Univer	Itsher Support rk for Unity. With it, you can create Android
▶ UniLWP.Droid - Unity Live Wallpape 0.0 🛓	FinGameWorks Version 0.0.0 - June 02, 2020 asset store View in the Asset Store Publisher Website Introduction UniLWP.Droid is an Android live wallpaper framework Studio projects that compile to live wallpaper apps. More Images & Videos UniLWP.Droid UniWebroid UniWebroid Univer Univer United to the wallpaper framework United to the wallpaper framework Studio projects that compile to live wallpaper apps. More Images & Videos United to the wallpaper for detect United to the wall of the detect United to	Itsher Support
▶ UniLWP.Droid - Unity Live Wallpape 0.0 🛓	FinGameWorks Version 0.0.0 - June 02, 2020 asset store View in the Asset Store - Publisher Website - Publisher Unit/WP.Droid is an Android live wallpaper framework Studio projects that compile to live wallpaper apps. More Images & Videos Unit/WP.Droid Webset Station for Acted View images & Videos On Asset Store Package Size Size: 220.94 KB (Number of files: 16) Purchased Date	Itsher Support rk for Unity. With it, you can create Android
▶ UniLWP.Droid - Unity Live Wallpape 0.0 🛓	FinGameWorks Version 0.0.0 - June 02, 2020 asset store View in the Asset Store - Publisher Website - Publ Introduction UniLWP.Droid is an Android live wallpaper frameword Studio projects that compile to live wallpaper apps. More Images & Videos UniLWP.Droid	Itsher Support rk for Unity. With it, you can create Android
▶ UniLWP.Droid - Unity Live Wallpape 0.0 🛓	FinGameWorks Version 0.0.0 - June 02, 2020 asset store View in the Asset Store - Publisher Website - Publ Introduction UniLWP.Droid is an Android live wallpaper frameword Studio projects that compile to live wallpaper apps. More Images & Videos UniLWP.Droid UniLWP.Droid Webset Solden for Added View images & videos on Asset Store Package Size Size: 220.94 KB (Number of files: 16) Purchased Date June 03, 2020 Release Details	Itsher Support rk for Unity. With it, you can create Android Supported Unity Versions 2019.3.13 or higher
▶ UniLWP.Droid - Unity Live Wallpape 0.0 ↓	FinGameWorks Version 0.0.0 - June 02, 2020 asset store View in the Asset Store - Publisher Website - Publ Introduction UniLWP.Droid is an Android live wallpaper frameword Studio projects that compile to live wallpaper apps. More Images & Videos UniLWP.Droid	Itsher Support rk for Unity. With it, you can create Android Supported Unity Versions 2019.3.13 or higher

Fig. 5: Choose the asset at the left sidebar, then press download and then import buttons at the bottom-right.

Ideal		project				settings
▲ Asset Store # Scene	Inspector Package Upload	co Game	Project Settings	Lighting		:
- 0	Discos		٩			
	Player					0 7 4
Device Simulator Editor	Ţ.				÷	
Graphics	Settings for Android					p
Input Manager Package Manager	▶ Icon					
Physics	Resolution and Presentation	on				
Physics 2D Player	Start in fullscreen mode Render outside safe area		~			
Preset Manager	Optimized Frame Pacing		Ě			
Quality Script Execution Order	Resolution Scaling					
v Services	Resolution Scaling Mode		Disabled			· •
Ads Analytics	Blit Type		Auto			
Cloud Build	Supported Aspect Ratio Aspect Ratio Mode		Native Aspect Ratio			-
Cloud Diagnostics Collaborate	Orientation					
In-App Purchasing	Default Orientation*		Auto Rotation			
Tags and Layers TextMesh Pro	Allowed Orientations for Auto Portrait	Rotation				
Time	Portrait Upside Down		-			
Version Control	Landscape Right					
XR Plugin Management v UniLWP	Landscape Left		~			
V Droid	Use 32-bit Display Buffer*		~			
Ads Integration Behavior	Disable Depth and Stencil*					
v Build	Render Over Native UI*					
Advanced Export One-Click	Show Loading Indicator		Don't Show			*
T Resources	* Shared setting between multiple p	olatforms.				
Screen Saver Strings	Splash Image					
Wallpaper	Other Settings					
Simulator Utils	Publishing Settings					
	Project/Player/F	Resoluti	on and Preser	ntation		

🖨 Asset Store 🛛 🛱 Scene	🚯 Inspector 🛛 Package Upload 🖙 Game	🕸 Project Settings 🌻 Lighting	1
⊤ Project	Player		0 ≠ ¢
Audio	Minimum API Level	Android 7.0 'Nougat' (API level 24)	
Device Simulator Editor	Target API Level	Automatic (highest installed)	
Graphics			-
Input Manager	Configuration	# 00DD	
Package Manager	Scripting Backend	IL2CPP	•
Physics	Api Compatibility Level*	.NET 4.x	
Physics 2D Player	C++ Compiler Configuration	Debug	
Preset Manager	Use incremental GC	~	
Quality	Mute Other Audio Sources*		
Script Execution Order	Target Architectures		
✓ Services	ARMv7	~	
Ads	ARM64	~	
Analytics Cloud Build	Split APKs by target architecture (Experime	enti	
Cloud Diagnostics	Install Location	Automatic	•
Collaborate	Internet Access	Auto	*
In-App Purchasing	Write Permission	Internal	T
Tags and Layers	Filter Touches When Obscured		
TextMesh Pro Time	Sustained Performance Mode	~	
Version Control	Low Accuracy Location		
XR Plugin Management			
⊤ UniLWP	Android TV Compatibility		
Droid			
Ads Integration Behavior			
T Build			
Advanced Export	Scripting Define Symbols		
One-Click	UNILWP_AUTHOR;UNILWP_ADS		
▼ Resources	Allow 'unsafe' Code		
Screen Saver Strings	Use deterministic compilation	~	
Wallpaper	Active Input Handling*	Input Manager (Old)	~
Simulator	Optimization		
Utils	Prebake Collision Meshes*		
	Keen Loaded Shaders Alive*		*
Project/Player/Other Settings			

The paid version, UniLWP.Droid.Store, is distributed on Unity Asset Store as a .unitypackage file.

7.2.1 Import On 2020.1 +

- Open package manager via menu Window/Package Manager.
- Switch package scope to Packages: My Assets and search UniLWP
- Choose UniLWP.Droid Unity Live Wallpaper For Android, then download and import

7.2.2 Import On 2019.3 +

- Open asset store panel via menu Window/Asset Store.
- Search for UniLWP in the newly opened web version of asset store.
- Choose UniLWP.Droid Unity Live Wallpaper For Android, then download and import

7.2.3 Setup

Toggle Project Settings panel via menu path Edit/Project Settings...

- Go to Project/Player and adjust certain items:
 - In Player/Resolution and Presentation, make sure that both Optimized Frame Pacing and Render Over Native UI are unchecked.
 - In Player/Other Settings, make sure that both Mute Other Audio Sources and Filter Touches When Obscured are unchecked, the Minimal API Level is Android 7. 0 Nougat (API Level 24). You might also want to change Graphics APIs to OpenGLES3 only, but that is optional.
- Go to Project/Audio and adjust certain items:
 - Check Disable Unity Audio if you don't want your wallpaper to play sounds.
- Go to UniLWP and change settings as you like. Please refer to Customize Look and Feel for options.

7.2.4 Build

Please refer to *Export*.

Listen To Callbacks

UniLWP.Droid provides callbacks for developers to register in C# so they can respond to Android native events.

8.1 Callback Types And Registration

 Currently, all available callbacks are declared within LiveWallpaperManagerDroid, a ScriptableObject singleton that you can access via a global variable.

LiveWallpaperManagerDroid would be initialized prior to scene loading, so you can safely call LiveWallpaperManagerDroid.Instance in Awake() or Start() without worrying null reference exceptions.

If you want to acquire the initial state of callback values, LiveWallpaperManagerDroid also has a set of static variables with corrsponding names to the callback methods as initial values.

Below is an example of how to acquire initial value of the screen status and also listen for future changes.

```
using FinGameWorks.UniLWP.Droid.Scripts.Managers;
public class Demo : MonoBehaviour
{
    private void Awake()
    {
        // acquire the initial screen status
        Enums.ScreenStatus screenStatus = LiveWallpaperManagerDroid.screenStatus;
        Debug.Log("ScreenStatus " + screenStatus);
        LiveWallpaperManagerDroid.Instance.screenDisplayStatusUpdated += status =>
        {
            // triggered every time screen status has change
            Debug.Log("ScreenStatus " + status);
        };
    };
    };
}
```

8.1.1 Insets

Insets refer to android.view.WindowInsets, a set of value describing the padding of android window where you should avoid drawing to. This is especially useful when you are trying to run Unity UI on Android devices with notches.

Called when

When the app has entered an Actvitiy or a Wallpaper Service.

Value

Padding values in pixel format. Please use in conjuntion with Screen.width to calucate the percentage if you are in a scaled UI space.

Listing 1: Declaration

```
public delegate void OnInsetsUpdatedDelegate(int left, int top, int right,

→int bottom);

public OnInsetsUpdatedDelegate insetsUpdated;

public static Vector4 insets;
```

Listing 2: Example

8.1.2 Offsets

Offsets refer to the distance the users have scrolled on their Android home (launcher). It is derived from the WallpaperService.Engine#onOffsetsChanged method with slight modifications.

Called when

When the user is swiping across pages on Android launchers.

Value

- float xOffset: Wallpaper horizontal scoll progress in a 0-1 scale.
- float yOffset: Wallpaper vertical scoll progress in 0-1 scale. Since there isn't much support of vertical launcher pages in Android apps, this value is mostly reported as 0.
- float xOffsetStep: The progress a horizontal full-page scroll would take in a 0-1 scale. Consequently, total horizontal page count can be calucated by 1 divided by this value (int) (1.0/xOffsetStep).
- float yOffsetStep: The progress a vertical full-page scroll would take in a 0-1 scale.

 bool simulated: Since certain stock launchers and ROMs do not follow the described behavior in Android documentation (specifically, Samsung's OneUI), the value UniLWP acquired from those devices are always 0 or 0.5, resulting in no way to know the total launcher pages and progress. To work around this limitation, UniLWP is designed to deploy a gesture recognizer to manually calucate the estimated progress, and simulated field would be true in this case.

Listing 3: Declaration

```
public delegate void OnWallpaperOffsetsUpdatedDelegate(float xOffset, float_

→yOffset, float xOffsetStep, float yOffsetStep, bool simulated);

public OnWallpaperOffsetsUpdatedDelegate wallpaperOffsetsUpdated;

public static Vector4 offset;

public static bool offsetSimulated;
```

Listing 4: Example

```
LiveWallpaperManagerDroid.Instance.wallpaperOffsetsUpdated += (xOffset, __

→yOffset, xStep, yStep, simulated) => {

    int xPageCount = xStep == 0 ? 0 : (int) Math.Round(1.0 / xStep);

    float xPageProgress = xPageCount * xOffset;

};
```

8.1.3 Dark Mode

Dark mode

8.1.4 Screen Display Status

Screen display status refers to the lock state of a phone. You can utlize this value to perform certian animations when the user lights up or unlocks the phone.

Note: For Android 9.0, this callback also include an always on display (AOD) value if you put androidprv:supportAmbientMode="true" into wallpaper.xml. However, since Android 10, this attribute is protected by a permission android.permission.AMBIENT_WALLPAPER, which is a system only permission that you normally cannot request except you are compling the ROM youself (i.e. you are also one of Android OEM companies or custom ROM makers)

Called when

When the user has turn on the screen in lock state / unlock the phone / lock the phone / leave the phone into always on display mode

Value

Enums.ScreenStatus, where:

- LockedAndOff = 0
- LockedAndAOD = 1
- LockedAndOn = 2

• Unlocked = 3

Listing 5: Declaration

Listing 6: Example

```
LiveWallpaperManagerDroid.Instance.screenDisplayStatusUpdated += status =>
{
};
```

8.1.5 In Activity

This callback reflects if the Unity instance is currently displaying in an activity.

Note: Notice that, this callback only works in the scope of UniLWP's own provided activities. If you are writing a customized activity and also want the UniLWP to receive this event in the C# side, please register your activity (refer to the Trigger Callbacks In Your Own Implementation section)

8.1.6 In Service

This callback reflects if the Unity instance is currently displaying in wallpaper mode.

8.2 Trigger Callbacks In Your Own Implementation

Behavior

Export

10.1 One-Click

10.2 External Modification

Ads

Warning: Unity Ads support is experimental but it should work on 0.0.2 and up.

Ads support has two limitations for now:

- The Initial Activity Check Bypass flag should be enabled (The editor script will do this automatically)
- You can only show an ad in activity, not wallpaper.

Ads Settings

Before enabling Unity Ads	support
Project Settings	i
	٩
in-App Purchasing	Ads Integration
Tags and Layers TextMesh Pro	
Time	Work in progress and API might change, use with caution
Version Control	
XR Plugin Management	Settings
▼ UniLWP	A Activate Ads will
⊤ Droid	1. add UNILWP_ADS flag to your symbol defines 2. set unilwp.behavior.activity.bypass.initial to true in Behavior settings
Ads Integration	2. set unilwp.behavior.activity.bypass.initial to true in Behavior settings
Behavior	Activate
Build Advanced Export	
One-Click	
Screen Saver	
Strings	
Wallpaper	
Simulator Utils -	
ouis 🗸	

After enabling Unity Ads support		
Project Settings	۹.	
Preset Manager Quality Script Execution Order Services Ads Analytics Cloud Build Cloud Diagnostics Collaborate In-App Purchasing Tags and Layers TextMesh Pro	Ads Integration	
	Work in progress and API might change, use with caution	
	Settings	
	Deactivate Ads will remove UNILWP_ADS flag from your symbol defines	
	Deactivate	
	Actions	
Time	Unity Ads Settings	
Version Control XR Plugin Management	User Guide	
▼ UniLWP ▼ Droid	Online Guide	
Ads Integration	1. Init Ads in Start()	
Behavior Build	Advertisement.Initialize (gameld, testMode);	
Advanced Export	2. Check if Ads module is loaded	
One-Click Resources	Advertisement.IsReady()	
Screen Saver Strings	3. Show Ads	
Wallpaper Simulator	Advertisement.Show();	
Utils 👻		

To enable it, you need to:

- Go to the UniLWP Ad setting panel at Project Settings/UniLWP/Droid/Ads Integration
- Click 'Activate' button. This step will:
 - Add UNILWP_ADS to your Android platform define symbols to enable certain blocks of code bound by #ifdef conditional compile flags
 - Set unilwp.behavior.activity.bypass.initial to true so that Unity Ads native library could acquire a proper context via UnityPlayer.currentActivity at startup.
- Click 'Unity Ads Settings' button to setup your ads and optionally add ads dependency.

Preference

Simulator

Migration

14.1 0.0.2

Issues worth noting from 0.0.0.1 to 0.0.2:

- The FinGameWorks/UniLWP/Droid menu items are grouped into sections for better readability. You can also use Project Settings/UniLWP/Droid panel which would provide the same functionality.
- Meta-data editing are now available directly within Editor (Unity 2020+).
- · Changes of meta-data flags in UniLWP aar plugins:
 - unilwp.behavior.floating removed
 - unilwp.behavior.settings.notifyButton replaced by unilwp.style.activity. launch.display
 - unilwp.behavior.resistToUpdate removed
 - unilwp.behavior.activity.expanded replaced by unilwp.behavior.preview. button.settings
 - unilwp.behavior.lifecycle.resumeWithoutHolder replaced by unilwp.behavior. screen.off.early
 - unilwp.behavior.activity.bypassCheck renamed to unilwp.behavior.activity. bypass.initial
 - unilwp.ref.class.screensaver.service added
 - unilwp.ref.class.launcher.activity added
 - unilwp.ref.class.wallpaper.service added
- UniLWP.Droid now launches LiveWallpaperLauncherRedirectActivity instead of LiveWallpaperPresentationActivity. LiveWallpaperLauncherRedirectActivity, as you can guess from its name, is a launcher activity (meaning it adds an icon to your app drawer) to launch different activities depending on different configurations. Please refer to unilwp.style.activity. launch.display flag for different styles.

Changelog

15.1 0.0.2 (preview.2)

Date Nov 29, 2020

Comment This is a preview release of 0.0.2, with many break changes both in C# and Java modules. Please delete UniLWP folder before upgrading to 0.0.2.

Added

• Added mipmap settings panel in the Project Settings editor window.

Fixes

N/A

Removed

N/A

15.2 0.0.2 (preview.1)

Date Nov 24, 2020

Comment This is a preview release of 0.0.2, with many break changes both in C# and Java modules. Please delete UniLWP folder before upgrading to 0.0.2.

Added

- Added UniLWP.Droid settings panels in the Project Settings editor window.
- Unity Ads integration (experimental).
 - To use Unity Ads (currently only within activities), you need to follow guidelines in Project Settings/UniLWP/Droid/Ads Integration.

Fixes

- Menu items within FinGameWorks/UniLWP/Droid are re-arranged into groups.
- Fixed a crash when Unity UI input field is activated.

Removed

N/A

15.3 0.0.1

Date Jun 2, 2020

Comment Initial Release