Multi Device Preview & Gallery

v1.7.0

for Unity 5.0 to 2018.3

January 24, 2019

Contents

1 Overview **6**
1.1 Introduction 6
1.2 Features 6
1.3 Requirements 8
1.4 Limitations 8
1.5 Known Issues 8

2 Quick Start Guide **9**

3 Gallery Window **10**

4 Device Preview Window **12**

5 Settings Window **13**
5.1 Gallery 13
5.2 Resolutions 14
5.3 Managing Resolution Presets 15
5.4 Destination 15
5.5 File Name 16
5.6 Remove iOS or Android Permission Needs 17
5.7 Hotkeys 17

6 Screenshot Quick Start Guide **19**
6.1 Screenshot Capture Workflows 19
6.1.1 Take screenshots in editor: ScreenshotWindow 19
6.1.2 Easy to use in-game screenshots: ScreenshotManager 19
6.1.3 Capture and export screenshots from script: SimpleScreenshotCapture . . 19
6.1.4 Capture textures from script: ScreenshotTaker 20
6.2 Quick Screenshot Configuration 20
6.3 Platform Specific Configuration 20
6.3.1 Android 20
6.3.2 iOS 21
6.3.3 WebGL 22

7 Screenshot Window **23**

8 Screenshot Creator Configuration Guide 24
8.1 Capture Mode . . . 24
8.2 Resolutions . . . 24
8.3 Cameras . . . 25
8.4 Overlay Canvas . . . 27
8.5 Composition . . . 28
8.6 Batches and Process . . . 30
8.7 Preview . . . 31
8.8 Capture . . . 31
8.9 Hotkeys. . . . 32
8.10 Utils . . . 32
8.11 Screenshot Taker Config . . . 33

9 Extra Features 34
9.1 In-game Gallery and Screenshots Management . . . 34
9.1.1 Customizable Grid Gallery . . . 34
9.1.2 Customizable Greybox . . . 35
9.2 Display a Validation UI . . . 36
9.3 Thumbnail . . . 36
9.4 Screenshot Cutter . . . 36
9.5 Message Canvas . . . 37

10 Simple Localization 38

11 FAQ 39
11.1 How to 39
11.1.1 Hide Some Scene Objects . . . 39
11.1.2 Access the screenshots taken by the ScreenshotManager . . . 39
11.1.3 Share the screenshots on social medias . . . 39
11.1.4 Create a Screenshot with a Transparent Background . . . 39
11.1.5 Export using separated layers . . . 40
11.1.6 Capture an offscreen scene . . . 40
11.1.7 Capture using an UI button . . . 40
11.2 Common issues . . . 40
11.2.1 I have a compilation error on iOS . . . 40
11.2.2 The PPI mode preview is different from what I see on my device . . . 40
11.2.3 I have some artefacts with temporal anti-aliasing, or other special effects . 41
11.2.4 I can not create screenshots with the ScreenshotTaker . . . 41
11.2.5 I can not take screenshots on Android . . . 41
11.2.6 My GameView is blinking when I update the device or gallery previews. . 41
11.2.7 My GameView and layout are doing strange things when I capture a screenshot . . . 41
11.2.8 Nothing Happens when I try to Update the Preview(s) . . . 42

12 Programming 43
12.1 API . . . 43
12.1.1 ScreenshotTaker . . . 43
12.1.2 SimpleScreenshotCapture . . . 44
12.2 Delegates . . . 45
12.2.1 Screenshot Manager . . . 45

12.2.2 Screenshot Taker 45

13 Versions **46**

14 Support **50**

Thank you for purchasing *Multi Device Preview & Gallery*! I wish this package will meet your needs and expectations. Please do not hesitate to contact me if you have a feature request, or any question, issue or suggestion.

Arnaud,
support@wildmagegames.com

1 OVERVIEW

1.1 Introduction

Multi Device Preview & Gallery gives you an accurate preview of your game for multiple devices, resolutions, and aspects, at a glance.

With more than 200 device presets and more than 40 full device preview, check that your game content and UI are properly scaled, and use the popularity presets to ensure that your game is adapted to the most popular devices or resolutions of your target platforms.

No more bad surprises, no more tedious checks, no need to buy dozens of devices. This is the very essential tool for developers, which will save you a lot of time!

1.2 Features

- Use the Gallery to preview your game at several resolutions, aspects, and devices at a glance.
- Edit your game and UI and see immediatly what your changes look like on all selected devices.
- **(New feature)** Full device preview for more than 40 devices, to get a good idea of what your game looks like on the most popular devices, including phones with notches.
- Preview at the device physical sizes with the ppi simulation, to better scale your UI content. No more bad surprises, unclickable buttons and unreadable texts, what you see is what you get!
- More than 200 mobile phone and tablet presets with their PPI, including the most popular smartphones on the market: Apple, Samsung, Google, HTC, etc.
- Use the Mobile, Standalone and PC popularity presets, to be sure that your game works perfectly for the most popular devices or resolutions of your target platforms.
- **(New feature)** Play your game and check that everything looks fine on any device with the Live Preview.
- **(New feature)** Automatically add your target devices to the GameView presets to play or edit at the device resolution.
- More than 50 resolution presets sorted by ratios to test all the cases.
- Easily export screenshots for all Stores (Amazon, Google Play, App Store, Windows Store, etc.)
- Easily add your custom resolutions and devices.
- **(New feature)** Update using the hotkeys, or enable auto-refresh in play mode and/or in edit mode.

Also, this package comes with all the features of *Ultimate Screenshot Creator*, which is is the ideal tool to create professional marketing and PR assets, wallpapers, mobile store screenshots, and more.

Main in-game Features.

- **(New feature)** Capture sreenshots in-game on all platforms.

- **(New feature)** Export your screenshots to Android Gallery.
- **(New feature)** Export your screenshots to iOS Camera Roll.
- **(New feature)** Customizable in-game gallery to manage screnshots.
- **(New feature)** Display a customizable validation UI to the user to save the screenshot or discard it.
- **(New feature)** Display a screenshot thumbnail when taking screenshots.
- **(New feature)** Export the screenshots to platforms Picture Folder.
- **(New feature)** Capture off-screen scenes.

Main Marketing Features.

- **(New feature)** Generate hundreds of multilingual promotional pictures with the customizable composer and batch process.
- Capture multiple resolution screenshots in one click.
- **(New feature)** Create custom presets and collections that can be shared between projects.
- **(New feature)** A localization feature has been added to easily localize your screenshots, and can be used independently to localize your whole game.
- Customizable set of cameras with custom rendering properties, such as culling mask, clear mode, clear color and field of view.
- Customizable overlay system to automatically include your game logo, watermarks, and more.
- **(New feature)** Export to separated layers.
- **(New feature)** Create custom process to be used in automated screenshot generation.
- In-game preview with photography guides to better frame your screenshots.
- Burst mode not to miss the best moments, or to be used as input of a GIF creator software.
- Ultra HD screenshots.
- **(New feature)** Capture only a sub-part of the screen with the Capture Area Selection.
- Powerful naming system with symbols to customize the export folder and file names.
- Easily export screenshots for all Stores (Amazon, Google Play, App Store, Windows Store, etc.)

More Features.

- **(New feature)** Use the ScreenshotWindow to easily take screenshots in editor.
- **(New feature)** Use the Simple Screenshot Taker to easily take screenshots from one line of code.
- Export to PNG and JPG.
- Robust transparent backgrounds solution.
- Landscape and portrait mode.
- Screenshot preview, with photography guides.
- Resolution scaling.
- Customizable shot sound.
- Customizable hotkeys.
- Customizable export folder.
- Align cameras to view utils.
- Use the delegates to call your code during the capture process.

- Time management utils.
- Increment file name or override existing files.
- **(New feature)** Compatible with the new Unity post process stack.
- **(New feature)** Support multi-display settings.
- **(New feature)** Support of WebGL.
- Perfectly works with Unity 4.6 and later UI system.

1.3 Requirements

- The solution works with Unity 5.0 and later free or pro edition.
- Unity 5.4 and later is recommended for live preview.
- On iOS, adding the picture to the phone gallery requiers the Photo Gallery access rights. By disabling some of the asset features it is possible to remove that requirement, see Section 5.6 for more details.
- We advise to use Unity 5.4 and later for a better GameView management using *GAMEVIEW_RESIZING* in editor (see FAQ 11.2.7).

1.4 Limitations

- Unity WebPlayer platform is not supported.
- SpeedTree leaves can not be rendered with a transparent background.

1.5 Known Issues

- On Unity 2018.1 and later, UI element with a stretch anchor within canvas using a constant physical size canvas scaler may not be positioned correctly in the preview.
- Possible incompatibility with obfuscators. Whitelist the namespace AlmostEngine.Screeenshot.

2 QUICK START GUIDE

1. Open the gallery window in the menu *Window/AlmostEngine/Multi Device Preview and Gallery/Gallery*, or the preview window in the menu *Window/AlmostEngine/Multi Device Preview and Gallery/Device Preview*.

2. Open the settings window in the menu *Window/AlmostEngine/Multi Device Preview and Gallery/Settings*, or using the *Settings* button in the gallery or preview window.

3. Set your screen PPI in the Gallery section. A wrong PPI would result in a bad device scaling in ppi mode.

4. Select the resolutions to be used for the preview, create your owns, or use the list + button to add one of the preset resolutions.

5. Set the destination folder and the filename for exporting the screenshots.

6. Select the display mode: *RATIO*, *PPI*, or *PIXELS*.

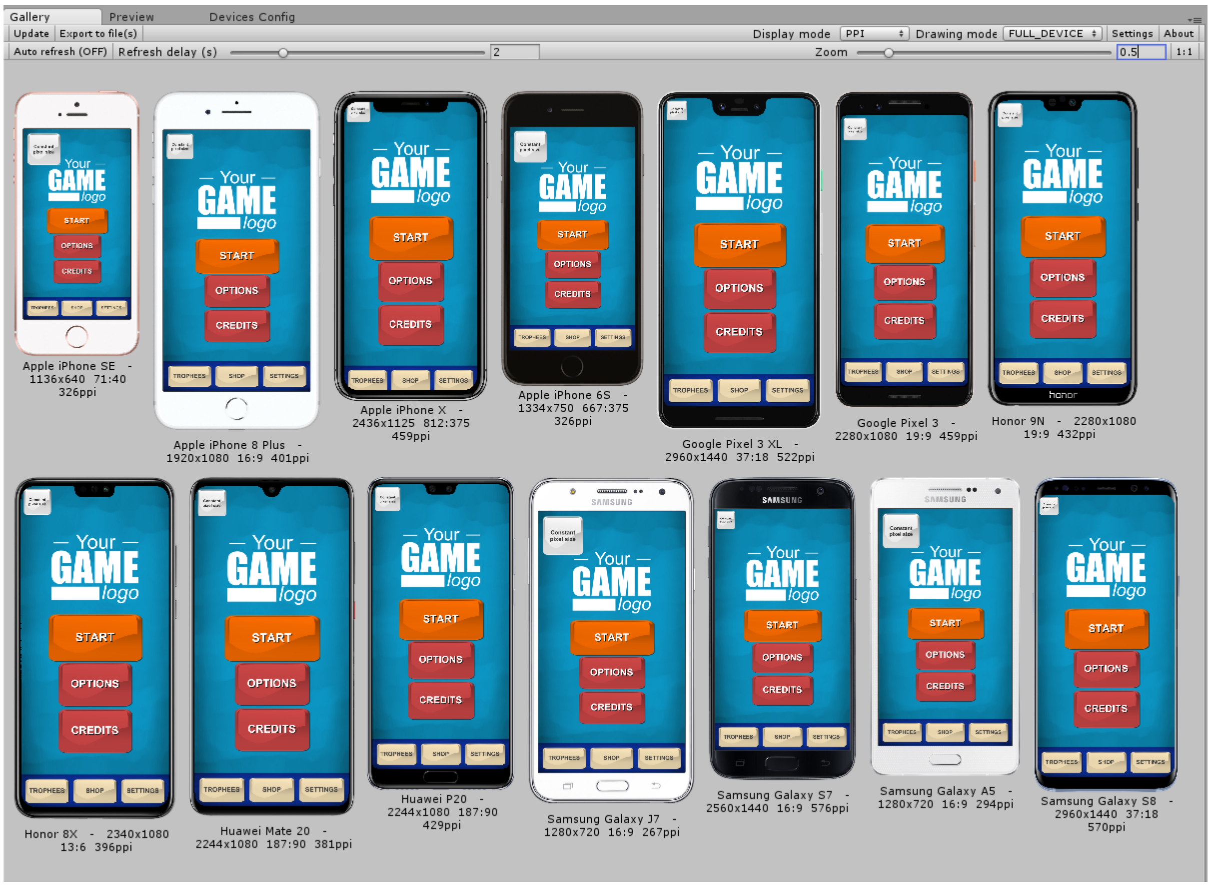
7. Select the drawing mode: *TEXTURE_ONLY*, *SCREEN_MASK*, or *FULL_DEVICE*.

8. Click on Update in the window to create a preview of all selected resolutions. Note that you can also use and customize the menu item hotkeys (Section 5.7).

Important Android and iOS configuration. *Multi Device Preview & Gallery* comes with all *Ultimate Screenshot Creator* features, including the ability to save to iOS and Android galleries. You need to configure it properly (see Section 6.3) or to exclude those features (see Section 5.6). Not doing so will result in compilation issues on iOS.

3 GALLERY WINDOW

The Gallery Window allows you to preview multiple devices or resolutions at a glance. The displayed devices are managed in the Settings Windows (see Section 5). Open the gallery window in *Window/AlmostEngine/Multi Device Preview and Gallery/Gallery.*


Gallery
Preview
Devices Config
Update
Export to file(s)
Auto refresh (OFF)
Settings
About
1:1
Apple iPhone SE - 1136x640 71:40 326ppi
Apple iPhone 8 Plus - 1920x1080 16:9 401ppi
Apple iPhone X - 2436x1125 812:375 459ppi
Apple iPhone 6S - 1334x750 667:375 326ppi
Google Pixel 3 XL - 2960x1440 37:18 522ppi
Google Pixel 3 - 2280x1080 19:9 459ppi
Honor 9N - 2280x1080 19:9 432ppi
Honor 8X - 2340x1080 13:6 396ppi
Huawei Mate 20 - 2244x1080 187:90 381ppi
Huawei P20 - 2244x1080 187:90 429ppi
Samsung Galaxy J7 - 1280x720 16:9 267ppi
Samsung Galaxy S7 - 2560x1440 16:9 576ppi
Samsung Galaxy A5 - 1280x720 16:9 294ppi
Samsung Galaxy S8 - 2960x1440 37:18 570ppi

Display mode: PPI
Drawing mode: FULL_DEVICE
Refresh delay (s): 2
Zoom: 0.5

Figure 1: The gallery window

Update. Press this button to update all active resolutions. Note that you can use the hotkey to update the resolution previews when focused on the gallery window or on the sceneview.

Export. Will export the preview images to files, using the export settings of the *PreviewManager*. Note that it will also export the device image is you are in *FULL_DEVICE* drawing mode. To export only the screen content, switch to the *TEXTURE_ONLY* mode.

Display Mode. There are three display modes:

- *RATIO*: the preview width is proportional to the window width, and its height is computed based on the resolution ratio. A zoom of 1 means the preview width is equals to the window width.
- *PIXEL*: the preview dimensions are proportional to the resolution dimensions. A zoom of 1 means one pixel on screen is equal to one pixel on the device.

- *PPI*: the preview dimensions are proportional to the resolution physical dimensions. A zoom of 1 means the preview physical size on screen is equal to the device screen physical size in real life.

Drawing Mode. Select the drawind mode. See Section 5 for more details.

Settings. Open the settings window.

Auto Refresh. Auto refresh can be set to play mode and/or edit mode in the settings window.

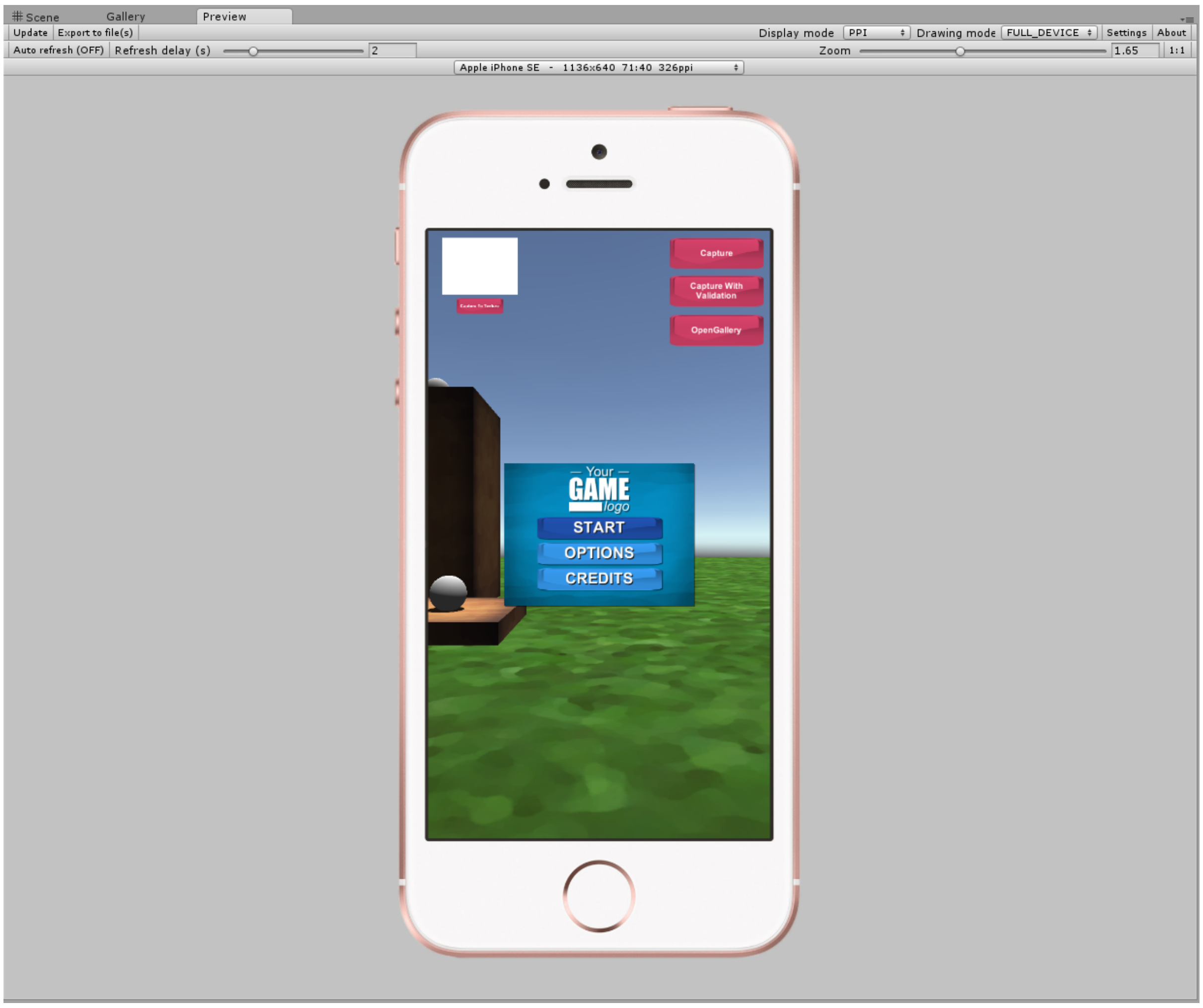
Refresh Delay. You can set the time waited between each refresh pass. Set to 0 for a real-time preview, set to a higher value to reduce performance costs.

Zoom. Use the zoom slide to change the preview dimensions.
Note that you can also use Ctrl + Mouse scrollWheel.

1:1. Use that button to quickly reset the zoom value to 1.

4 DEVICE PREVIEW WINDOW

The Device Preview Window allows to preview a specific device, and do a live preview in play mode. Open the preview window in *Window/AlmostEngine/Multi Device Preview and Gallery/Device Preview.*



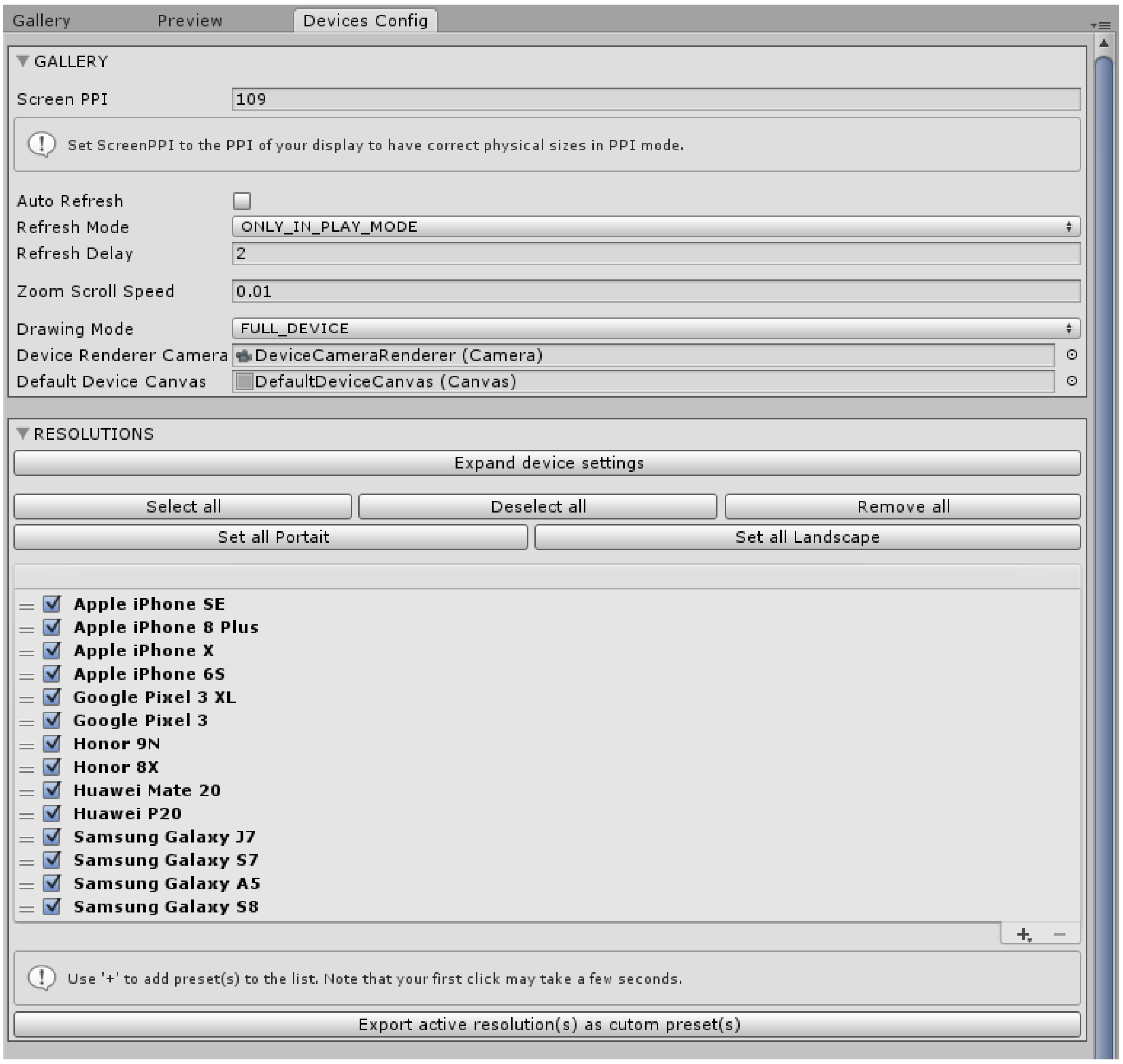
Display mode PPI
Drawing mode FULL_DEVICE
Refresh delay (s) 2
Zoom 1.65
Apple iPhone SE - 1136x640 71:40 326ppi

Figure 2: The preview window

Device Selection. With this window, you can select one of the active resolution to be previewed, update it and export it.

5 SETTINGS WINDOW

The settings window allows to configure the gallery and device preview window. Open the settings window in *Window/AlmostEngine/Multi Device Preview and Gallery/Settings*. The Settings Window settings are saved in the file *MultiDevicePreviewConfig.asset*.



GALLERY

Screen PPI	109

Set ScreenPPI to the PPI of your display to have correct physical sizes in PPI mode.

Auto Refresh	☐
Refresh Mode	ONLY_IN_PLAY_MODE
Refresh Delay	2
Zoom Scroll Speed	0.01
Drawing Mode	FULL_DEVICE
Device Renderer Camera	DeviceCameraRenderer (Camera)
Default Device Canvas	DefaultDeviceCanvas (Canvas)

RESOLUTIONS

- ✓ Apple iPhone SE
- ✓ Apple iPhone 8 Plus
- ✓ Apple iPhone X
- ✓ Apple iPhone 6S
- ✓ Google Pixel 3 XL
- ✓ Google Pixel 3
- ✓ Honor 9N
- ✓ Honor 8X
- ✓ Huawei Mate 20
- ✓ Huawei P20
- ✓ Samsung Galaxy J7
- ✓ Samsung Galaxy S7
- ✓ Samsung Galaxy A5
- ✓ Samsung Galaxy S8

Use '+' to add preset(s) to the list. Note that your first click may take a few seconds.

Figure 3: The settings window

5.1 Gallery

Screen PPI. To use the gallery in PPI mode, you must correctly set your display screen ppi. In ppi mode, with a zoom of 1, the physical preview dimensions on your screen will physically match the device dimensions.

Auto Refresh. Can be enabled in play mode, in editor mode, or in both mode. It is recommended to enable auto refresh only with one active resolution and in *TEXTURE_ONLY* mode to prevent the gameview to be constantly resized, but it works with any number of resolutions.

Drawind Mode. You can chose between three device drawing mode.

- *TEXTURE_ONLY* draws the screen at the correct scale and resolution.
- *SCREEN_MASK* draws the device screen shape, for instance phone with notches will have the notches part cut.
- *FULL_DEVICE* draws the device using a device canvas, perfect to really see your game on the target device.

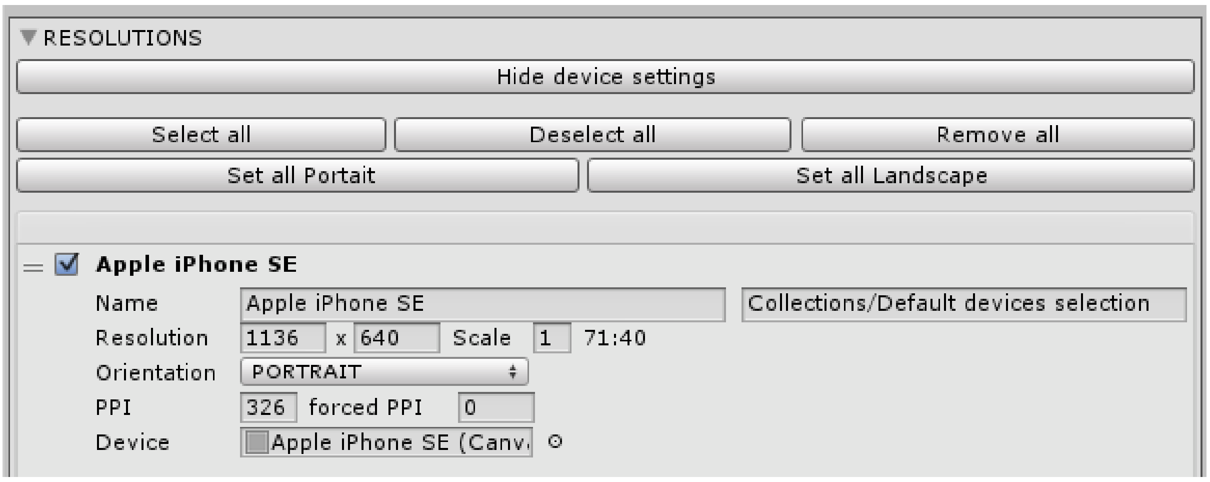
Note that devices with no custom device canvas will be rendered using a generic device contour.

Device Renderer Camera. The camera used to render the devices. Use the default Device-CameraRender prefab.

Default Device Canvas. When a device do not have a custom canvas, the specified default canvas is used. Note that it can be customized, for instance to change the device color.

Transparent Device Background. By default, the background color of the full device preview matches the editor color. If you want to export the full device preview images to files, enable the transparent device background.

5.2 Resolutions



☑ Apple iPhone SE

Name	Apple iPhone SE				Collections/Default devices selection
Resolution	1136	x 640	Scale	1	71:40
Orientation	PORTRAIT				
PPI	326	forced PPI	0		
Device	Apple iPhone SE (Canv				

Figure 4: Resolutions settings.

PPI. All the devices presets come with a *ppi* (pixel per inch) information, to be used to display the devices into the gallery window at their physical size.
Do not forget to set your screen *ppi* (in the Gallery section of the PreviewManager inspector) to the value corresponding to your display screen.

PPI ”Forced” value. Sometimes, Canvas with Constant Physical Size scale will not be scaled in the Unity editor as in the device. This can occur because on the device Unity uses the Screen.dpi value, which may be wrong. Set the forced ppi value to the Screen.dpi returned by your device to get the same scaling for Constant Physical Size canvas scaler in the preview than in your device.

Note that there is a know issue on Unity 2018.1 and later: the Canvas with Constant Physical Size scaler may not be scaled properly.

Device. The device renderer Canvas to be used to perform the integration of the picture into a real device picture.

Note that with an empty device, the default device canvas will be used.

5.3 Managing Resolution Presets

Resolution presets are stored as asset files in the project folder. When you click on the + button, the presets are loaded and sorted in categories. It is possible to easily create your own presets.

Creating Custom Presets. Click on ”Export active resolution(s) as custom preset(s)” to automatically create new preset files using the resolutions in your custom resolution list.

Also, you can right click on your project folder and select *Asset/Almost Engine/Ultimate Screenshot Creator/Custom preset.*

Creating Custom Collections. Collections are set of presets that can be grouped and loaded simultaneously.

To create a new collection, right click on your project folder and select *Asset/Almost Engine/Ultimate Screenshot Creator/Custom collection.* Then, add all desired resolution preset to the list.

Creating Custom Popularity Collections. Collections are set of presets that can be grouped and loaded simultaneously.

To create a new popularity collection, right click on your project folder and select *Asset/Almost Engine/Ultimate Screenshot Creator/Popularity preset.* Then, add all desired resolution preset to the list, and specify their associated popularity values.

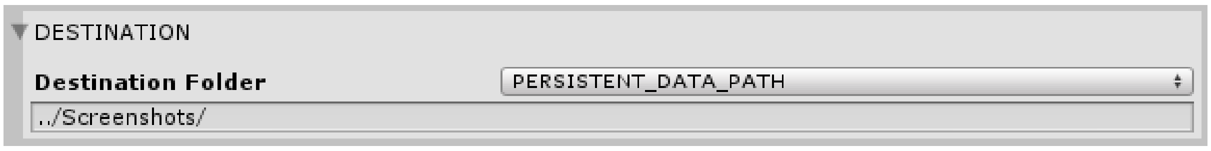
Copying Presets and Settings to an New Project. If you have some custom presets you want to copy to a new project, simply copy and paste your preset .asset and .meta files.

To copy your Screenshot Manager and Device settings, copy ScreenshotWindowConfig.asset and MultiDevicePreviewConfig.asset to your new project.

5.4 Destination

There are three export modes:

- *CUSTOM_FOLDER* allows you to select the export folder.
- *DATA_PATH* exports the screenshots relatively to the project data path.
- *PERSISTENT_DATA_PATH* exports the screenshots relatively to the persistent data path.



DESTINATION

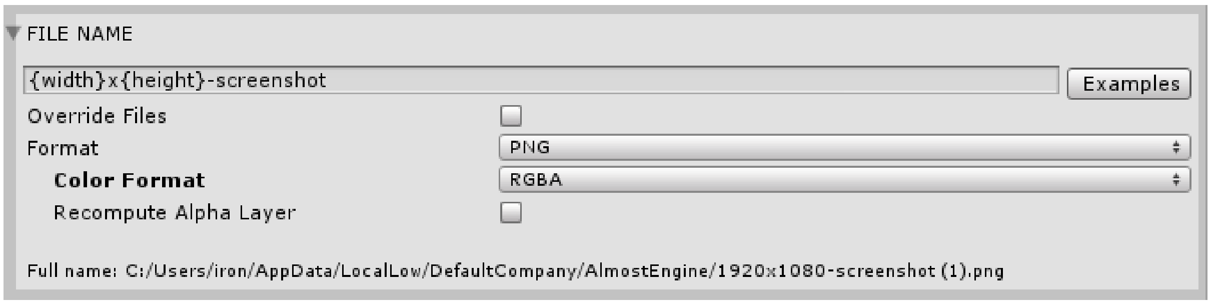
Destination Folder: PERSISTENT_DATA_PATH

../Screenshots/

Figure 5: Destination settings.

- *PICTURE_FOLDER* exports the screenshots relatively to the platform picture folder. It is the recommended mode for taking screenshot in-game on all platforms.

5.5 File Name



FILE NAME

{width}x{height}-screenshot

Override Files: ☐

Format: PNG

Color Format: RGBA

Recompute Alpha Layer: ☐

Full name: C:/Users/iron/AppData/LocalLow/DefaultCompany/AlmostEngine/1920x1080-screenshot (1).png

Figure 6: File name settings.

FileName. Defines the file name to be used for the screenshots. Use the *Examples* button to select one of the name presets.

You can use and combine the following symbols to better match your needs:

- {year}, {month}, {day}, {hour}, {minute}, {second} for the current time information,
- {width}, {height}, {scale}, {ratio}, {orientation}, {ppi}, {percent}, {name}, {category} for the resolution information.
- {layer} for the camera name in separated layer mode.
- {composer} for the current composer name in composition mode.
- {batch} for the current batch process name in batches mode.

Note that you can use the file name to group screenshots by resolution folders, etc. For instance *{width}-{height}/screenshot.png* will create one folder by resolution and create one *screenshot.png* in each of them.

Override Existing Files. By default, if you try to create a screenshot file that already exists, its name will be incremented. For instance: screenshot.png, screenshot(1).png, ... Set *override* to *true* if you want to override the existing files.

Format. You can export to *PNG* or to *JPG* with a custom quality.

Color Format. In *PNG*, you can export to *RGB* or to *RGBA*. Use *RGBA* to create screenshots with an alpha layer, enabling transparent backgrounds.

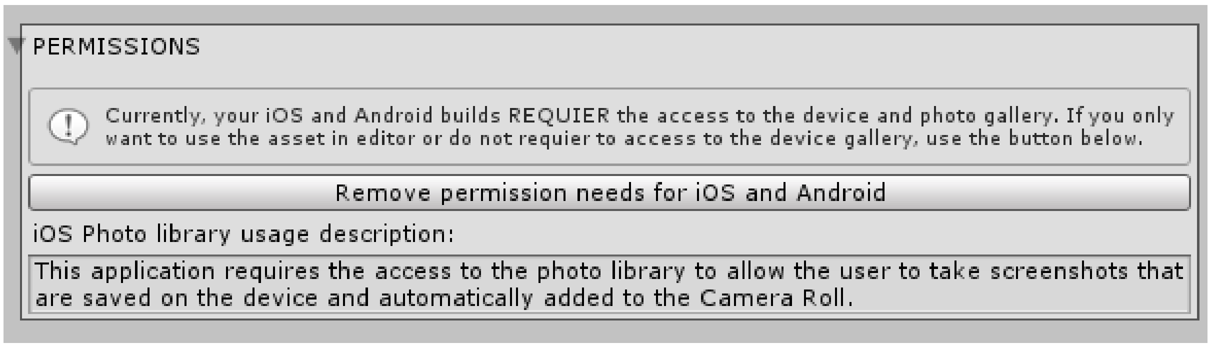
Recompute Alpha Layer. Force alpha layer to be recomputed. This is a costly process. Use only if you have alpha problems in *RGBA* mode.

Full Fame Preview. You can look at the full name preview to check if everything is correct. Note that the resolution information used for the full name preview is the one of the first resolution in the list.

5.6 Remove iOS or Android Permission Needs

If you want to use Multi Device Preview and Gallery in editor only, or if you want to remove the iOS and Android permission needs, you can partially exclude some features of Ultimate Screenshot Creator from your Android and iOS builds.

When the permission is removed, the ability to export screenshots to files and to add the screenshot to the gallery will be disabled on iOS and Android, but the other features will still be enabled. For instance, it will still be possible to capture screenshots to textures on iOS and Android. All editor features are still enabled.


PERMISSIONS

Currently, your iOS and Android builds REQUIER the access to the device and photo gallery. If you only want to use the asset in editor or do not requier to access to the device gallery, use the button below.

iOS Photo library usage description:

This application requires the access to the photo library to allow the user to take screenshots that are saved on the device and automatically added to the Camera Roll.

Figure 7: Permission settings.

Automatically removing the permission. The Permission section in the ScreenshotWindow settings contains a button to automatically remove or restore the permission needs.

Manually removing the permission.

1. Remove the Plugins/iOS folder.
2. In the Player Settings, in the Android tab, add *IGNORE_ANDROID_SCREENSHOT* to Scripting Define Symbols.
3. In the Player Settings, in the iOS tab, add *IGNORE_IOS_SCREENSHOT* to Scripting Define Symbols.
4. In the Player Settings, set *Write Access* to *Internal*.

5.7 Hotkeys

To update or export screenshots, you can use the menu items hotkeys *Tools/Device Preview/Update Preview(s)* and *Tools/Device Preview/Export Preview(s)*.

The hotkeys can be edited by editing the scripts *UpdateDevicePreviewMenuItem.cs* and *ExportDevicePreviewMenuItem.cs.* More details can be found on the script files. If you change the hotkeys, remember not to overwrite the menu item scripts in the next asset updates.

6 SCREENSHOT QUICK START GUIDE

6.1 Screenshot Capture Workflows

Ultimate Screenshot Creator is a very flexible and easy to use asset, and there are several ways to use it. In the following section, we describe some of the most common way to use it.

6.1.1 Take screenshots in editor: ScreenshotWindow

The ScreenshotWindow is an editor window providing a quick way to take screenshots in editor. It can be fully configured and its settings are saved automatically.

1. Open the screenshot window in the menu *Window/AlmostEngine/Screenshot Window*.
2. Edit the window settings (see below).
3. The settings are automatically saved, except scene object references. To keep persistent references to scene objects, use a ScreenshotManager.

You will find more details about the ScreenshotWindow on Section 7.

6.1.2 Easy to use in-game screenshots: ScreenshotManager

The ScreenshotManager is a gameobject to be added on the scene to take screenshots in editor or in play mode. It is an all-in-one and easy to configure screenshot solution. It handles everything, like filenames, export directory, camera, canvas, etc.

1. Put the *ScreenshotManager* prefab into your scene.
2. Select the *ScreenshotManager* to configure it (see below).
3. Do not forget to Apply your modifications to the prefab to save them, or to create a new prefab.

You will find more details about the ScreenshotManager on Section 8.

Available Example: In the DefaultExample scene, the "ScreenshotManager" game object has a ScreenshotManager component.

6.1.3 Capture and export screenshots from script: SimpleScreenshotCapture

There are several way to take screenshot from scripts. For instance you can configure a ScreenshotManager game object and call the Capture() method from a custom script.

You can also take screenshots without any ScreenshotManager, by using the SimpleScreenshotCapture static methods. You can chose to capture the current screen, a set of cameras, or to specify all capture settings. It will take the screenshot and export it using the given fullname. See Section 12.1 for more details.

Available Example: In the DefaultExample scene, the "Simple Capture Script Example" game object has a CaptureScreenshotExample component.

6.1.4 Capture textures from script: ScreenshotTaker

The ScreenshotTaker component is the core of all screenshot taking methods of the asset. It provides several capture coroutines, for instance to capture the current screen, a set of cameras, etc. At the end of the coroutine process, the provided texture is updated and can be used directly. See Section 12.1 for more details.

Available Example: In the DefaultExample scene, the "CaptureTextureCanvas Example" game object has a CaptureCameraToTextureExample component.

6.2 Quick Screenshot Configuration

1. Select the capture mode. *GAMEVIEW_RESIZING* is the most robust capture mode, and captures the UI perfectly. It works in Editor and Windows Standalone only. *RENDER_TO_TEXTURE* works in editor and on all platforms, but Screenspace Overlay UI are not rendered. *FIXED_GAMEVIEW* works in editor and on all platforms, can only capture at the current screen resolution. It is the recommended settings for taking screenshots in-game. See Section 8.1 for more details.
2. Set the destination folder and the filename. See Section 5.4 for more details.
3. Select the camera mode: *GAME_VIEW* mode copies what you see in game view, *CUSTOM_CAMERAS* mode allows you to use a set of cameras. In custom mode, select the cameras to be used for the capture.

   Note that you can also customize their rendering settings. See Section 8.3 for more details.
4. Select the resolutions mode: *GAME_VIEW* mode uses the game view resolution, *CUSTOM_RESOLUTIONS* mode allows you to customize the resolutions to be used. In custom mode, select the resolutions to be used for the capture, or use the list + to create a custom resolution. See Section 8.2 for more details.
5. The overlay system enables the easy integration of your company or game logo into each screenshot. Edit the example canvas or create your own to display your game logo. You can reference scene objects or prefabs. Select the overlays to be used for the capture. You can customize the default overlay or create a new one using a *Canvas*.

   Note that you can also disable the overlay system if you do not want any canvas, by simply disabling or removing all canvas in the overlays list. See Section 8.4 for more details.
6. The photography guide helps framing the screenshots, and is particularly useful using *PreviewWhilePlaying*.

6.3 Platform Specific Configuration

This section details the platform specific configuration required to be able to take screenshots in-game.

6.3.1 Android

1. The capture mode the capture mode *FIXED_GAMEVIEW* is recommended to capture exactly what is on screen. *RENDER_TO_TEXTURE* is possible too.

2. The export mode must be *PICTURE_FOLDER* or *PERSISTENT_DATA_PATH*.

Permission. To export to the *PICTURE_FOLDER*, your application must have the *external storage* permission. In the Player Settings, set *Write Access* to *External*. In that case, the permission to access the image gallery should be asked at the first app launch. If your app does not have access to the external storage (for instance if the user rejects the permission), the export will fall back to *PERSISTENT_DATA_PATH* on all devices with an API > 23. On devices with API <= 23 an error message saying the directory or file could not be created will be displayed.

The *PERSISTENT_DATA_PATH* mode does not require any permission, and the screenshots will be saved in the specific app data directory.

Note that is it possible to remove the permission needs by excluding some features of the asset. See Section 5.6 for more details.

Primary Storage. Note that *PICTURE_FOLDER* tries to export to the primary storage, if available. That means that screenshots saved to that directory are not deleted if the app is uninstalled. If the primary storage is not available, or on *PERSISTENT_DATA_PATH*, the plugin save to the first media storage available. On that case, the pictures are visible on the gallery, but can be deleted from the device when the app is uninstalled.

6.3.2 iOS

1. Move the *Plugins* folder at the root of the *Assets* folder. The plugin should be located exactly here: *Assets/Plugins/iOS/iOSUtils.m*.
2. **IMPORTANT.** On Unity 5.6 and later, you need to add the dependency to the **Photos** framework to the iOSUtils.m plugin. This should be done automatically, but if you have compilation issues on iOS, you can do it manually. In Unity, select the iOSUtils.m file, look in the Plaform settings, and add the Photos dependency in the *rarely used frameworks* list.
3. The capture mode the capture mode *FIXED_GAMEVIEW* is recommended to capture exactly what is on screen. *RENDER_TO_TEXTURE* is possible too.
4. The export mode must be *PICTURE_FOLDER* or *PERSISTENT_DATA_PATH*.

Permission. Adding the picture to the phone gallery requiers the Photos access rights. This is done automatically by the script *iOsPostProcessBuild.cs*. Note that is it possible to remove the permission needs by excluding some features of the asset. See Section 5.6 for more details.

The permission to access the camera roll should be asked at the first screenshot capture. To ask the permission at startup, add the script *RequestAuthAtStartup* to your project, or call *iOsUtils.RequestGalleryAuthorization()* when you want to.

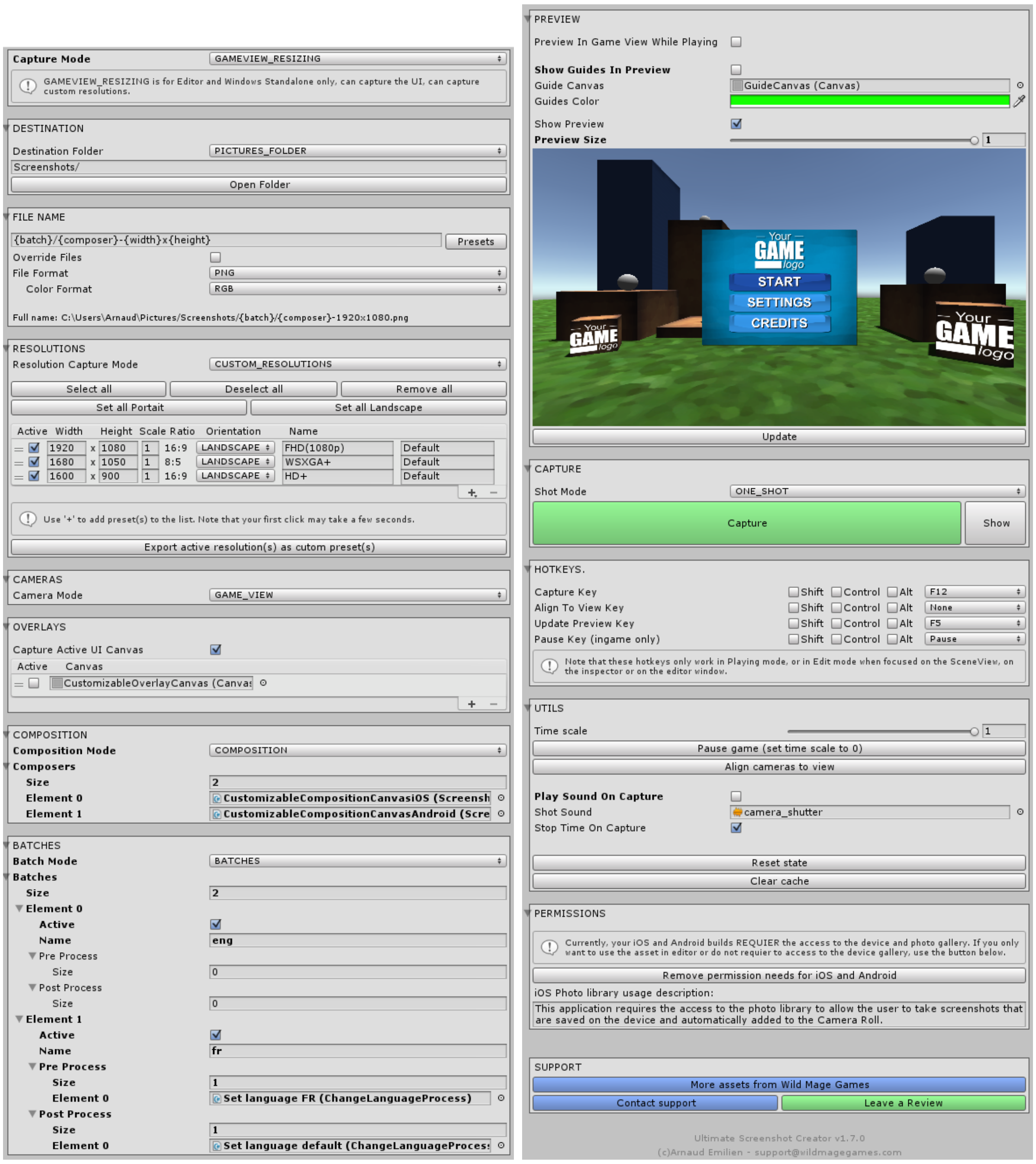
Usage Description. You can personalize the Photo usage description by editing the associated field in the ScreenshotManager inspector, or by editing the *UltimateScreenshotCreator/Assets/PhotoUsageDescription.asset* object.

6.3.3 WebGL

1. Move the *Plugins* folder at the root of the *Assets* folder. The plugin should be located exactly here: *Assets/Plugins/WebGL/WebGLUtils.jslib.*
2. The capture mode the capture mode *FIXED_GAMEVIEW* is recommended to capture exactly what is on screen. *RENDER_TO_TEXTURE* is possible too.
3. Note that with WebGL, the export mode is ignored since the image will be downloaded using the web browser.

7 SCREENSHOT WINDOW

The screenshot window allows to easily take screenshots in the editor, without the need to add anything on your scenes. Open the settings window in *Window/AlmostEngine/Screenshot Window*.



Capture Mode: GAMEVIEW_RESIZING
GAMEVIEW_RESIZING is for Editor and Windows Standalone only, can capture the UI, can capture custom resolutions.

DESTINATION
Destination Folder: PICTURES_FOLDER
Screenshots/
Open Folder

FILE NAME
{batch}/{composer}-{width}x{height} — Presets
Override Files: ☐
File Format: PNG
Color Format: RGB
Full name: C:\Users\Arnaud\Pictures/Screenshots/{batch}/{composer}-1920x1080.png

RESOLUTIONS
Resolution Capture Mode: CUSTOM_RESOLUTIONS
Select all — Deselect all — Remove all
Set all Portait — Set all Landscape

Active	Width		Height	Scale	Ratio	Orientation	Name	
✓	1920	x	1080	1	16:9	LANDSCAPE	FHD(1080p)	Default
✓	1680	x	1050	1	8:5	LANDSCAPE	WSXGA+	Default
✓	1600	x	900	1	16:9	LANDSCAPE	HD+	Default

Use '+' to add preset(s) to the list. Note that your first click may take a few seconds.
Export active resolution(s) as cutom preset(s)

CAMERAS
Camera Mode: GAME_VIEW

OVERLAYS
Capture Active UI Canvas: ✓

Active	Canvas
☐	CustomizableOverlayCanvas (Canvas

COMPOSITION
Composition Mode: COMPOSITION
Composers
Size: 2
Element 0: CustomizableCompositionCanvasiOS (Screensh
Element 1: CustomizableCompositionCanvasAndroid (Scre

BATCHES
Batch Mode: BATCHES
Batches
Size: 2
Element 0
Active: ✓
Name: eng
Pre Process
Size: 0
Post Process
Size: 0
Element 1
Active: ✓
Name: fr
Pre Process
Size: 1
Element 0: Set language FR (ChangeLanguageProcess)
Post Process
Size: 1
Element 0: Set language default (ChangeLanguageProces:

PREVIEW
Preview In Game View While Playing: ☐
Show Guides In Preview: ☐
Guide Canvas: GuideCanvas (Canvas)
Guides Color: 🟩
Show Preview: ✓
Preview Size: 1

Update

CAPTURE
Shot Mode: ONE_SHOT
Capture — Show

HOTKEYS.

Capture Key	☐ Shift	☐ Control	☐ Alt	F12
Align To View Key	☐ Shift	☐ Control	☐ Alt	None
Update Preview Key	☐ Shift	☐ Control	☐ Alt	F5
Pause Key (ingame only)	☐ Shift	☐ Control	☐ Alt	Pause

Note that these hotkeys only work in Playing mode, or in Edit mode when focused on the SceneView, on the inspector or on the editor window.

UTILS
Time scale: 1
Pause game (set time scale to 0)
Align cameras to view
Play Sound On Capture: ☐
Shot Sound: camera_shutter
Stop Time On Capture: ✓
Reset state
Clear cache

PERMISSIONS
Currently, your iOS and Android builds REQUIER the access to the device and photo gallery. If you only want to use the asset in editor or do not requier to access to the device gallery, use the button below.
Remove permission needs for iOS and Android
iOS Photo library usage description:
This application requires the access to the photo library to allow the user to take screenshots that are saved on the device and automatically added to the Camera Roll.

SUPPORT
More assets from Wild Mage Games
Contact support — Leave a Review

Ultimate Screenshot Creator v1.7.0
(c)Arnaud Emilien - support@wildmagegames.com

Figure 8: The screenshot window.

The Screenshot Window settings are saved in the file *ScreenshotWindowConfig.asset*. Note that the Screenshot Window can not save references to scene objects. Reference prefabs objects or use a Screenshot Manager game object.

See Section 8 for more details.

8 SCREENSHOT CREATOR CONFIGURATION GUIDE

8.1 Capture Mode

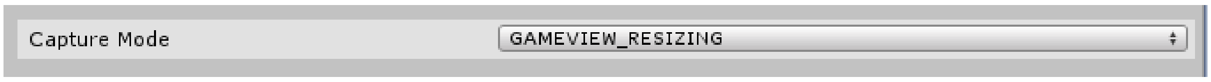

Capture Mode GAMEVIEW_RESIZING

Figure 9: Capture mode settings.

There are three capture modes:

- *GAMEVIEW_RESIZING* is the most robust capture mode, and captures the UI perfectly. It works in Editor and Windows Standalone only. Note that with this mode the editor gameview or game windows needs to be resized for a few milliseconds.
- *RENDER_TO_TEXTURE* works in editor and on all platforms, but Screenspace Overlay UI are not rendered.
- *FIXED_GAMEVIEW* works in editor and on all platforms, can only capture at the current screen resolution. It is the recommended mode for taking screenshot in-game on all platforms.

	GAMEVIEW_RESIZING	*RENDER_TO_TEXTURE*	*FIXED_GAMEVIEW*
Custom Cameras	$\surd$	$\surd$	$\surd$
Custom Resolutions	$\surd$	$\surd$	Screen resolution
UI	$\surd$	No UI	$\surd$
Overlays	$\surd$	No overlays	$\surd$
In Editor	$\surd$	$\surd$	$\surd$
In Game	Windows Standalone only	All platforms	All platforms

Table 1: Capture modes comparison.

8.2 Resolutions

Resolution Capture Mode. There are two resolution capture modes :

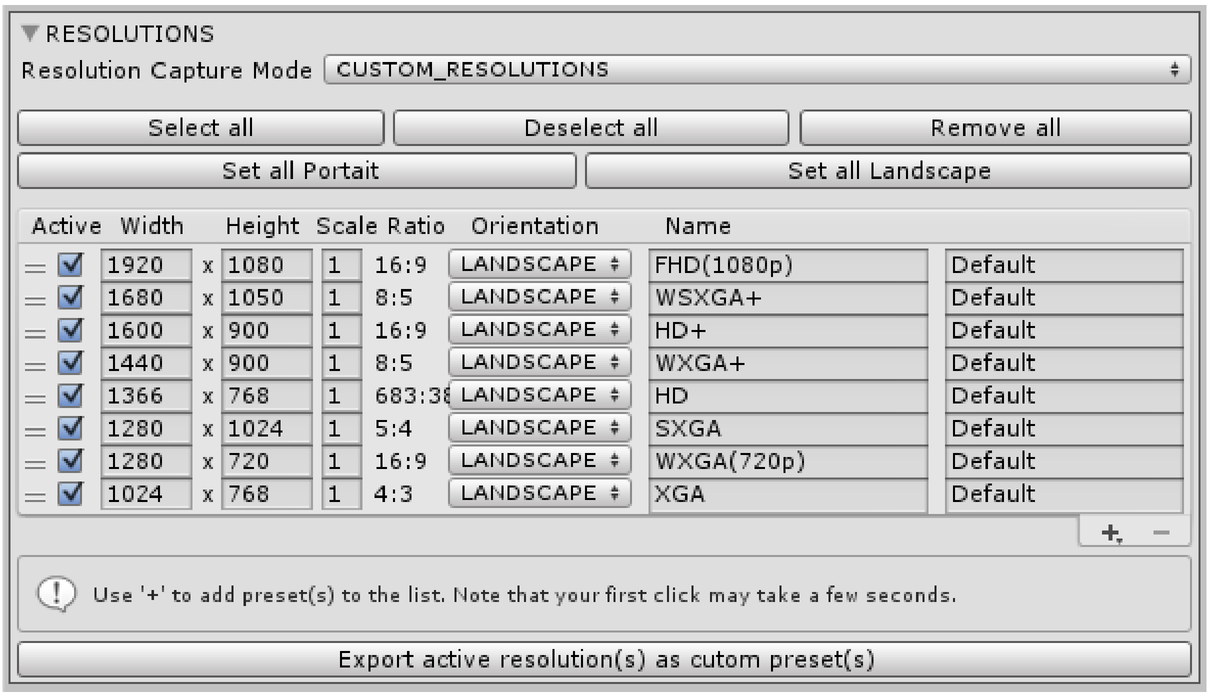
- *GAME_VIEW* just capture the game view.
- *CUSTOM_RESOLUTIONS* allows you to capture multiple resolutions in one click.

Managing resolutions. Use the resolution list to specify which resolutions are going to be captured. Use the list + button to add one of the resolution presets. Select *ALL* to include the whole group in one click.

Select the resolution to be removed and press the list − button, or right click on the resolution and select *remove item element*.

Scale. This setting allows you to easily take ultra HD screenshots. For instance, a resolution of 1600 × 1200 with a scale of 2 will take a screenshot with a resolution of 3200 × 2400.

Note that Unity limits the maximum resolution size to ∼ 8000 × 4000



Resolution Capture Mode: CUSTOM_RESOLUTIONS

Active	Width		Height	Scale	Ratio	Orientation	Name	
✓	1920	x	1080	1	16:9	LANDSCAPE	FHD(1080p)	Default
✓	1680	x	1050	1	8:5	LANDSCAPE	WSXGA+	Default
✓	1600	x	900	1	16:9	LANDSCAPE	HD+	Default
✓	1440	x	900	1	8:5	LANDSCAPE	WXGA+	Default
✓	1366	x	768	1	683:38	LANDSCAPE	HD	Default
✓	1280	x	1024	1	5:4	LANDSCAPE	SXGA	Default
✓	1280	x	720	1	16:9	LANDSCAPE	WXGA(720p)	Default
✓	1024	x	768	1	4:3	LANDSCAPE	XGA	Default

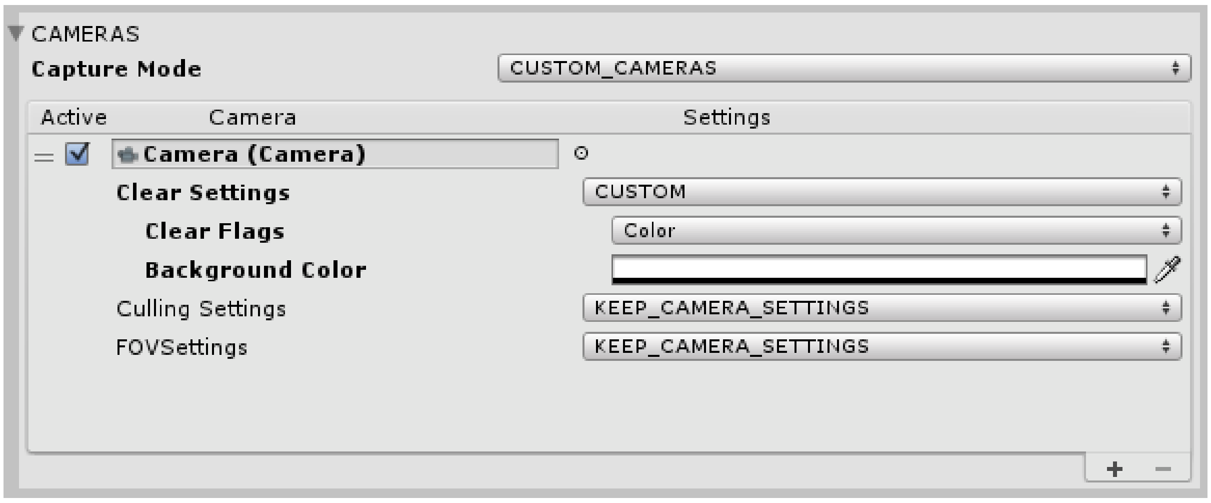
Use '+' to add preset(s) to the list. Note that your first click may take a few seconds.

Figure 10: Resolutions settings.

Orientation. Switch between *LANDSCAPE* and *PORTRAIT* mode.

Resolution Name and Category. The resolution name can be customized, and used in the file name field with the symbol {*name*}. You can also specify a category to be used with the file name symbol {*category*}.

8.3 Cameras



Capture Mode: CUSTOM_CAMERAS

Active	Camera	Settings
✓	Camera (Camera)	
	Clear Settings	CUSTOM
	Clear Flags	Color
	Background Color	
	Culling Settings	KEEP_CAMERA_SETTINGS
	FOVSettings	KEEP_CAMERA_SETTINGS

Figure 11: Camera settings.

Camera Capture Mode. There are two camera capture modes :

- *GAME_VIEW* just capture the game view.
- *CUSTOM_CAMERAS* allows you the customize the cameras to be used for the capture.

Export to separate layers. By enabling this option, a screenshot will be created for each camera. See Section 11.1.5 for more details.

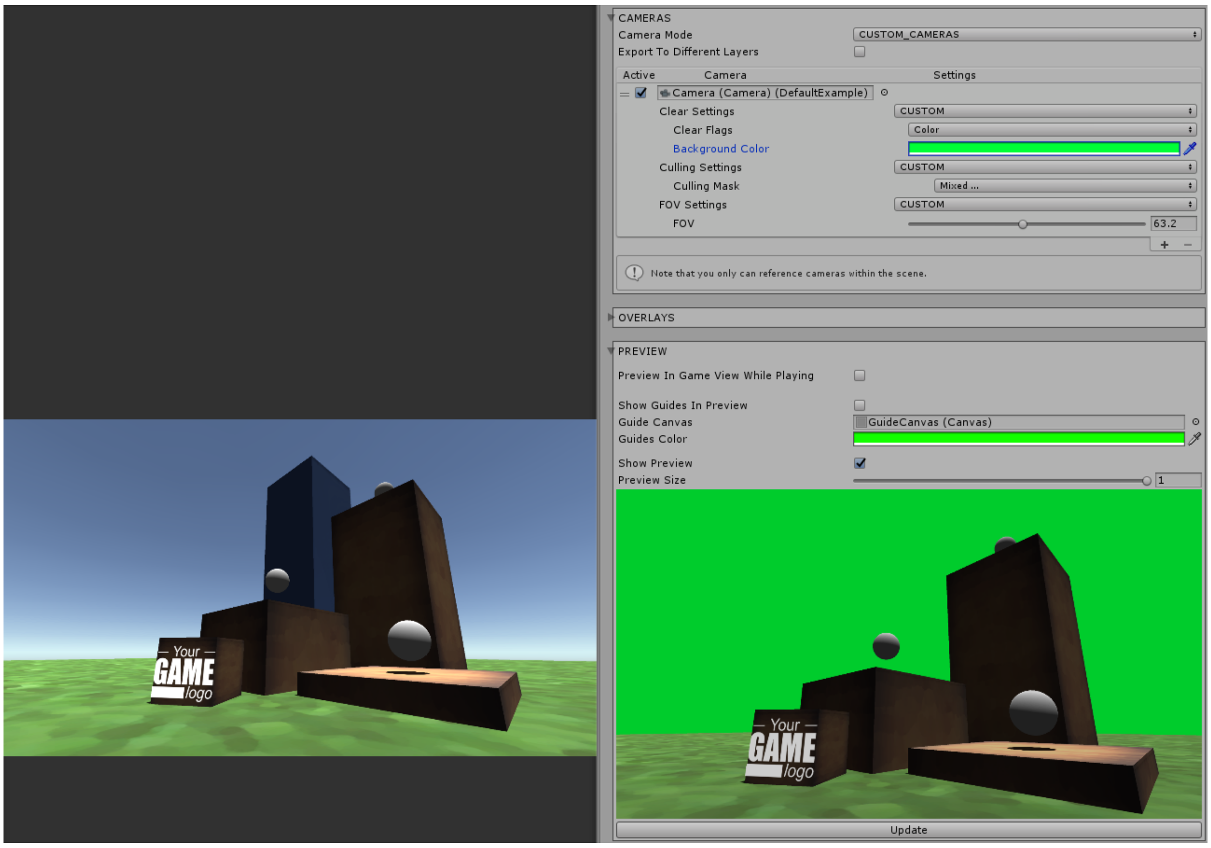
Managing Cameras. Use the camera list to specify which cameras are going to be used for the capture. You can select multiple cameras at a time if your game requires a layered camera rendering (for instance for UI elements or overlays, or if you have a camera for the scene rendering and an other camera for rendering the player gun).

Note that the other cameras of the scene will be disabled during the capture.

Custom Camera Settings. Camera settings can be overwritten during the capture:

- *KEEP_CAMERA_SETTINGS* keeps the camera settings unchanged,
- *CUSTOM* allows you to change the camera clear mode and background color, the camera culling mask, and the camera Field Of View.

Note that the custom settings override the current camera settings during the capture process, and should correctly restore them at the end of it.



CAMERAS
Camera Mode: CUSTOM_CAMERAS
Export To Different Layers: ☐

Active	Camera	Settings
✓	Camera (Camera) (DefaultExample)	
	Clear Settings	CUSTOM
	Clear Flags	Color
	Background Color	🟩
	Culling Settings	CUSTOM
	Culling Mask	Mixed ...
	FOV Settings	CUSTOM
	FOV	63.2

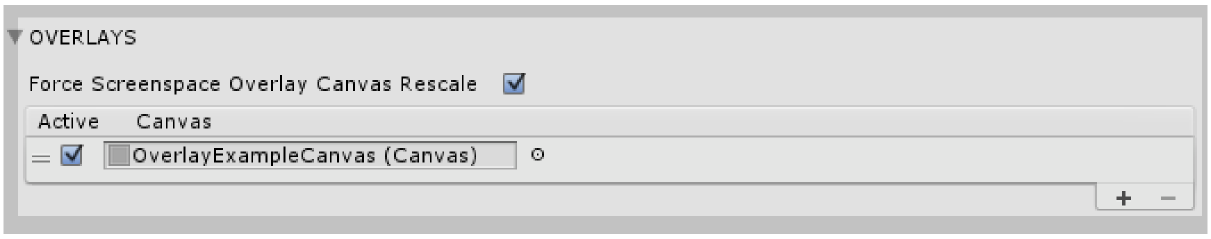
Note that you only can reference cameras within the scene.

OVERLAYS

PREVIEW
Preview In Game View While Playing: ☐
Show Guides In Preview: ☐
Guide Canvas: GuideCanvas (Canvas)
Guides Color: 🟩
Show Preview: ✓
Preview Size: 1

Figure 12: Example of custom camera settings.

8.4 Overlay Canvas


OVERLAYS

Force Screenspace Overlay Canvas Rescale ✓

Active	Canvas
✓	OverlayExampleCanvas (Canvas)

Figure 13: Overlay settings.

The overlay system allows you to easily integrate your game or studio logo into each screenshot, and much more. Use the overlay list to specify which overlays are going to be used for the capture.

Force Screenspace Overlay Canvas Rescale. Force Screenspace Overlay Canvas to be rescaled in *RENDER_TO_TEXTURE* mode.

Render Active UI Canvas. You can choose to hide or not all your active game UI during the capture.

Default Overlay Customization. The *OverlayExampleCanvas* is a *Canvas* prefab. You can customize it or create your own overlay canvas.

Creating Your Own Overlays. The overlay system is based on the Unity 4.6 Canvas system, and allows unlimited customization of the screenshot overlays. To create an overlay:

1. Create a new Unity *Canvas*.
2. Edit your new *Canvas* as you want.
3. Save the Canvas as a prefab and remove it from the scene.
4. Create a new overlay item in the screenshot manager list, and select your new *Canvas* prefab.
5. Disable the *Canvas* game object. This keeps the *Canvas* hidden during the game and only displays it during the capture.

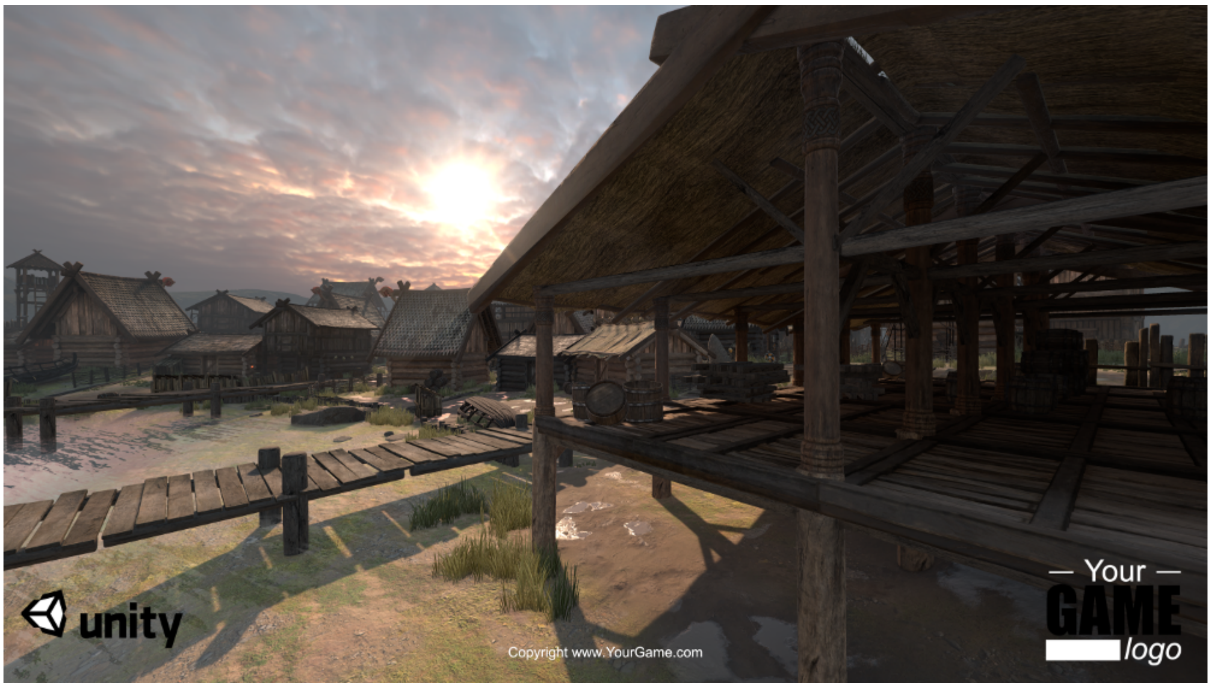
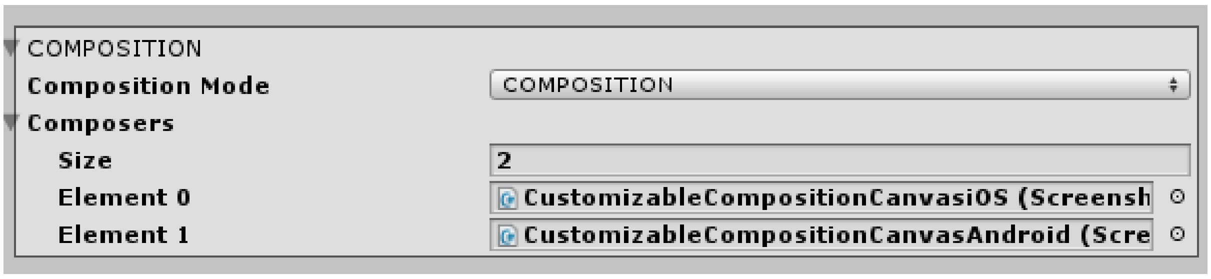


Figure 14: Example of screenshot captured using the default overlay (Viking scene, ©Unity).

8.5 Composition



COMPOSITION
Composition Mode: COMPOSITION
Composers
Size: 2
Element 0: CustomizableCompositionCanvasiOS (Screensh
Element 1: CustomizableCompositionCanvasAndroid (Scre

Figure 15: Composition settings.

The composition system allows to automatically capture your game and embed it in a promotional picture. Set the composition mode to *SIMPLE_CAPTURE* to disable it, or set it to *COMPOSITION*.

For each composer added to the composition list, the whole screenshot process is run to capture and integrate the screenshots within the composer canvas.

Creating Your Own Composition. Create a new Canvas game object and add a ScreenshotComposer component. Set the default UI Camera prefab to the camera field. The canvas field should reference the current composition canvas. In the textures list, add all RawImage game objects where your game will be embedded. For instance, it can be the UI image positioned in front of the device screen.

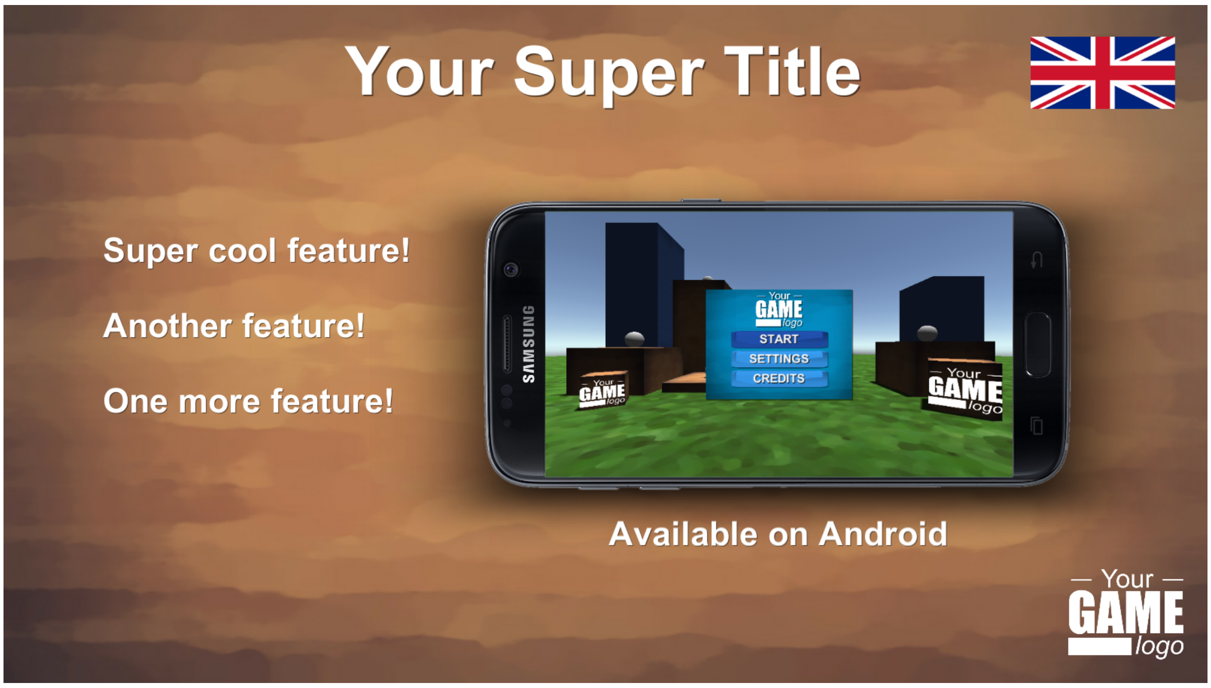


Figure 16: Example of composition using the Android template and the simple localization.

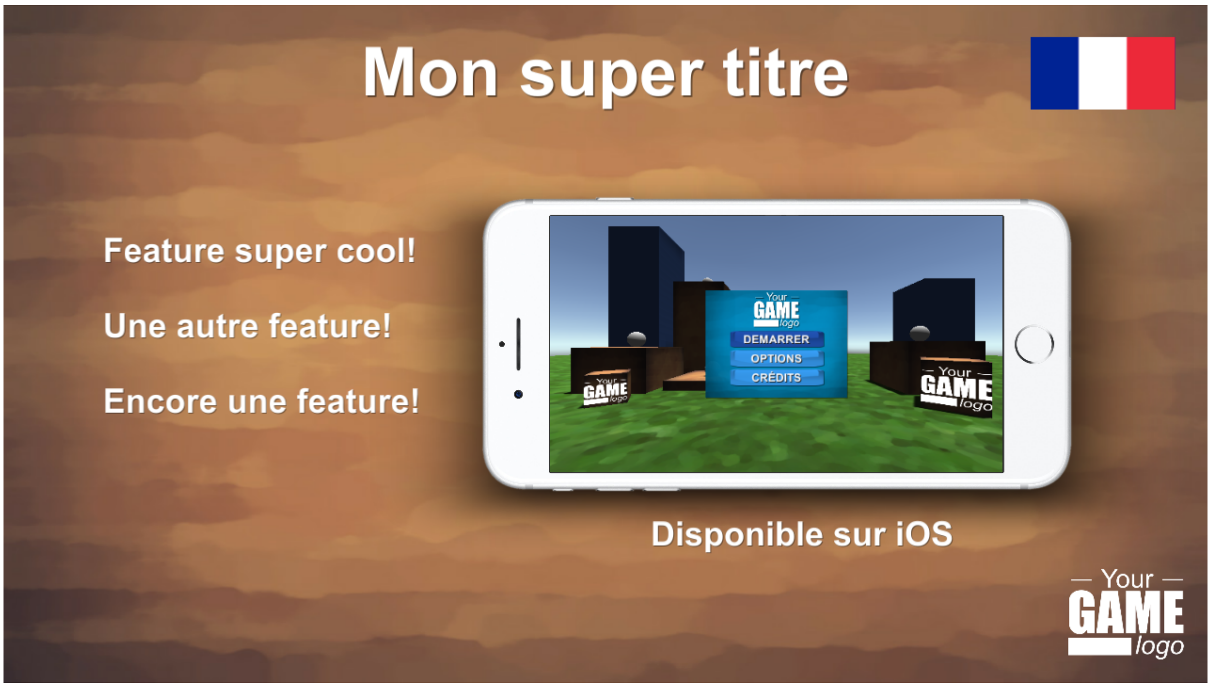


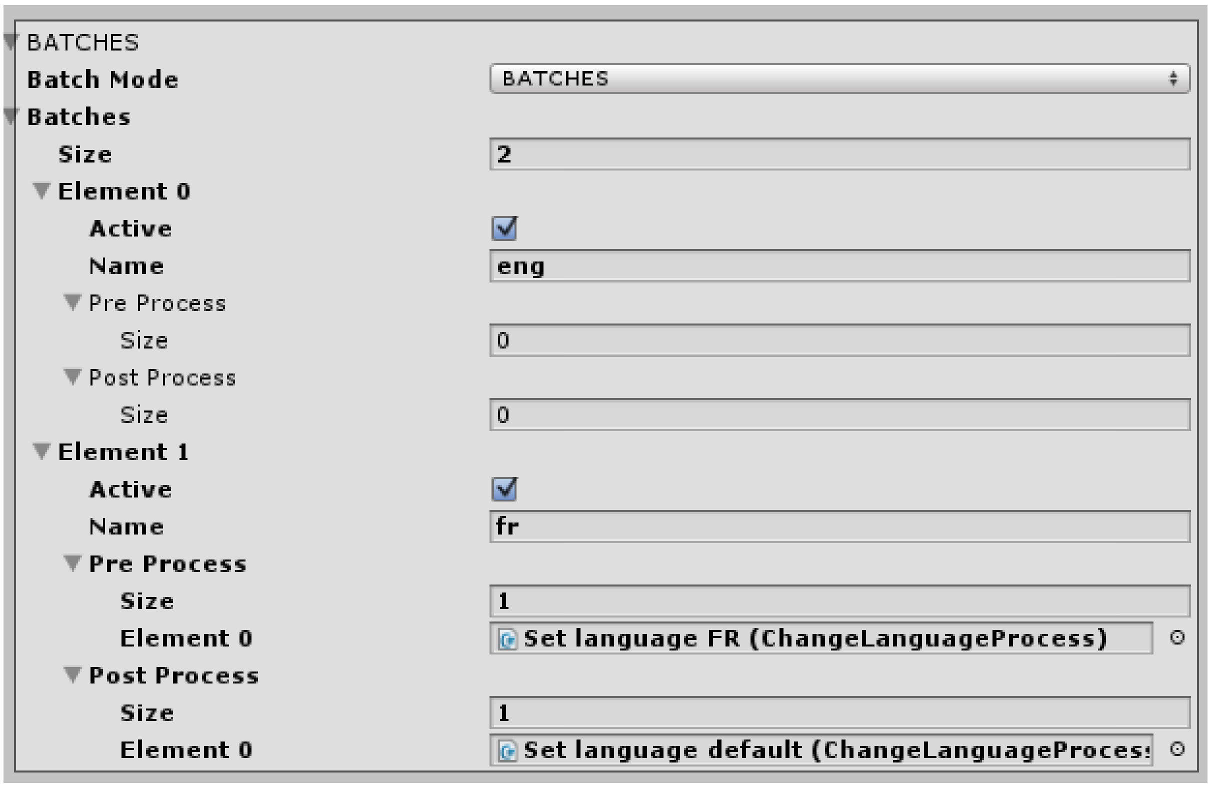
Figure 17: Example of composition using the iOS template and the simple localization.

8.6 Batches and Process

The batches system allows to automatically capture series of screenshots while applying some scripts before and after the capture. Set the batches mode to *SIMPLE_CAPTURE* to disable it, or set it to *BATCHES*.

For each batch added to the batches list, the whole screenshot process is run to capture with all pre process and post process.

For instance, to capture your game in multiple language, you can create two batches, one that does nothing, and one that changes the language in pre process and restore it in post process.



```
BATCHES
Batch Mode                 BATCHES
Batches
  Size                     2
  Element 0
    Active                 ✓
    Name                   eng
    Pre Process
      Size                 0
    Post Process
      Size                 0
  Element 1
    Active                 ✓
    Name                   fr
    Pre Process
      Size                 1
      Element 0            Set language FR (ChangeLanguageProcess)
    Post Process
      Size                 1
      Element 0            Set language default (ChangeLanguageProces
```

Figure 18: Composition settings.

Creating Your Own Process. Create a new script that inherits from ScreenshotProcess. Override the Process method or coroutine depending on your needs. Add your script to a new game object of prefab, and add the game object reference to a pre process or post process list.

8.7 Preview

Guides. You can display a photography guide to better frame your screenshots. The default guide is a *Canvas* prefab. This guide is visible in the preview picture only, and will not be rendered in the final image.

Preview in GameView while Playing. If you want to visualize how the screenshot will look while playing, you can set this option to true. When the game will start playing, the preview settings will be applied to the GameView, and restored when the game is stopped.

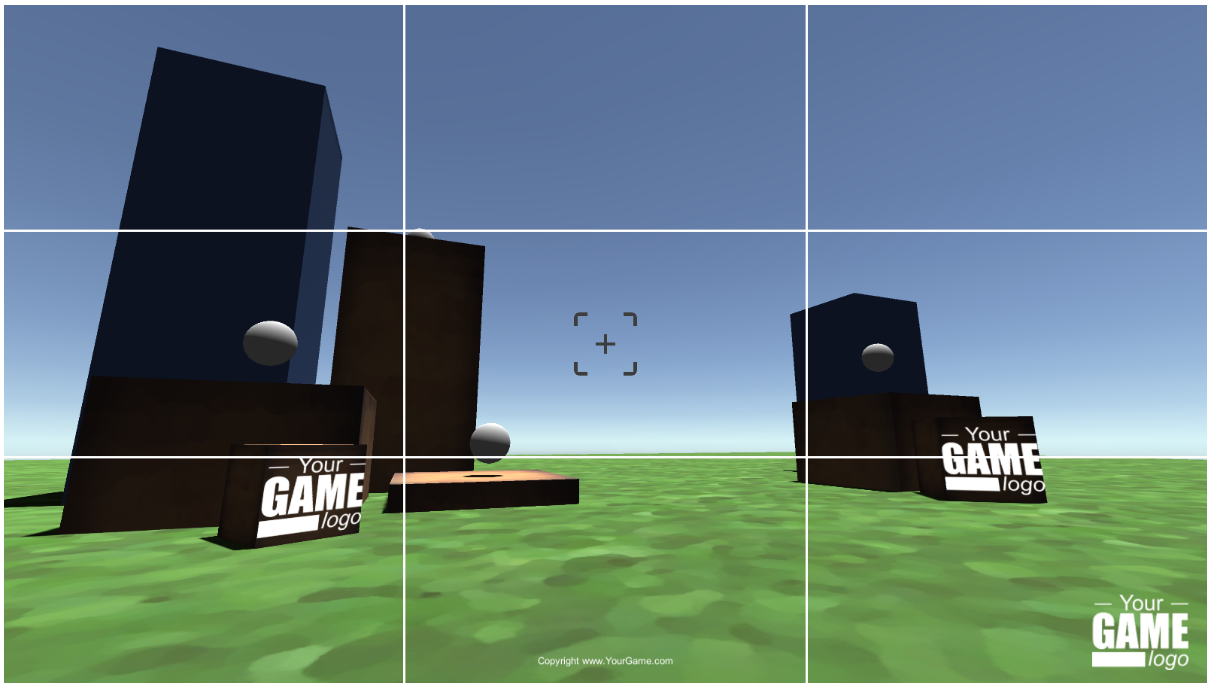
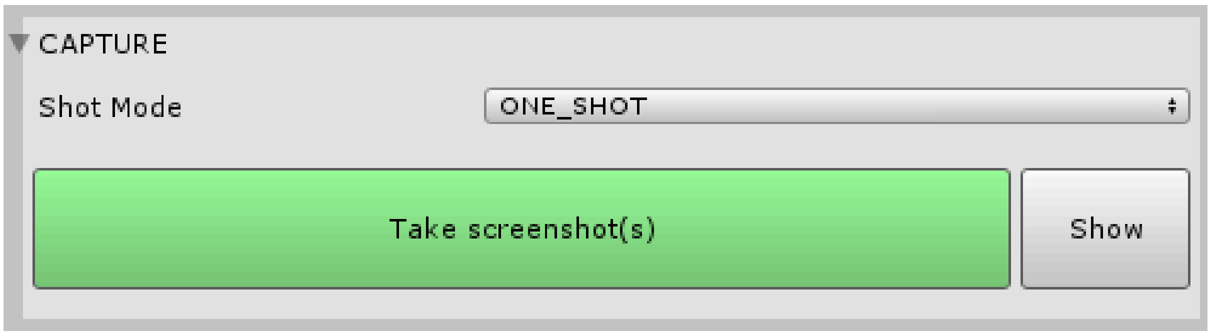


Figure 19: Use the framing guide in GameView while playing to better frame your screenshot.

8.8 Capture


CAPTURE

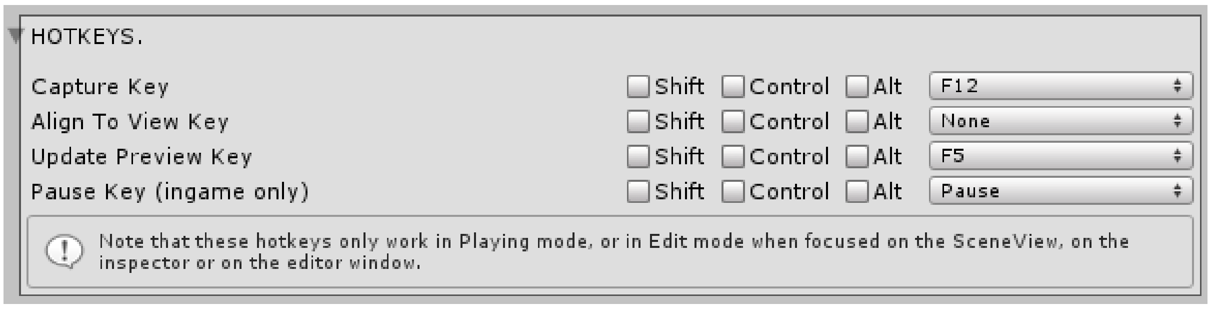
Shot Mode ONE_SHOT

Figure 20: Capture settings.

Capture Mode. There are two capture modes:

- *ONE_SHOT* captures one frame,
- *BURST* allows to take a series of screenshots with a fixed and customizable timestep. Note than you can use the screenshots as inputs of a GIF creator software.

8.9 Hotkeys.



HOTKEYS.

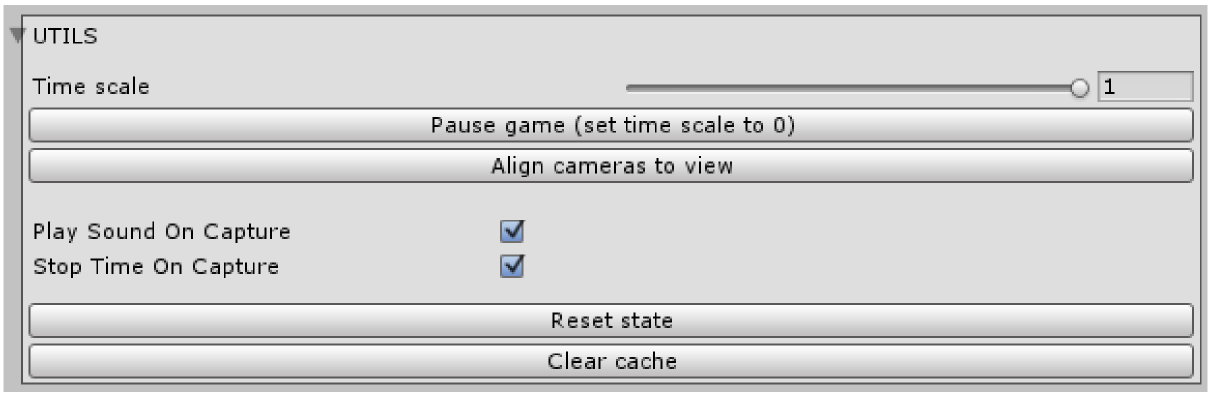
Capture Key	☐ Shift	☐ Control	☐ Alt	F12
Align To View Key	☐ Shift	☐ Control	☐ Alt	None
Update Preview Key	☐ Shift	☐ Control	☐ Alt	F5
Pause Key (ingame only)	☐ Shift	☐ Control	☐ Alt	Pause

Note that these hotkeys only work in Playing mode, or in Edit mode when focused on the SceneView, on the inspector or on the editor window.

Figure 21: Hotkeys.

In this panel you can set the various hotkey values, and specify the sound to be used for the capture. Note that for the hotkeys only works when focused on the SceneView, Screenshot-Manager Inspector or ScreenshotWindow, or on the GameView in play mode.

8.10 Utils



UTILS

Time scale: 1

Play Sound On Capture ✓

Stop Time On Capture ✓

Figure 22: Utils.

Align to View. You can align all the cameras in the camera list with the current scene view. Note that this is a reversible operation using Undo/Redo.

Time Scale. You can use these tools to pause or slow down the time while playing to edit the scene and/or better frame your screenshot.

Reset State. See FAQ 11.2.8.

8.11 Screenshot Taker Config

The ScreenshotTaker component is automatically added to each ScreenshotManager.

GameView Resizing Waiting Mode. In *GAMEVIEW_RESIZING* mode, you can specify the number of frame or time to wait between the screen resizing and the screenshot capture. It is particularly useful if you use some script or special effects that need some time to adapt to the screen size.

Force Layout Preservation. See Faq 11.2.7.

9 EXTRA FEATURES

9.1 In-game Gallery and Screenshots Management

The asset contains an in-game gallery system to display and manage the taken screenshots. It is based on the Grid Gallery to display the images, and the Greybox preview to perform actions on the screenshots.

9.1.1 Customizable Grid Gallery

The GridGalleryCanvas component is an example of component displaying a gallery based on the grid layout component. It inherits from the ScreenshotGallery, a more generic class which contains the basic in-game gallery methods.

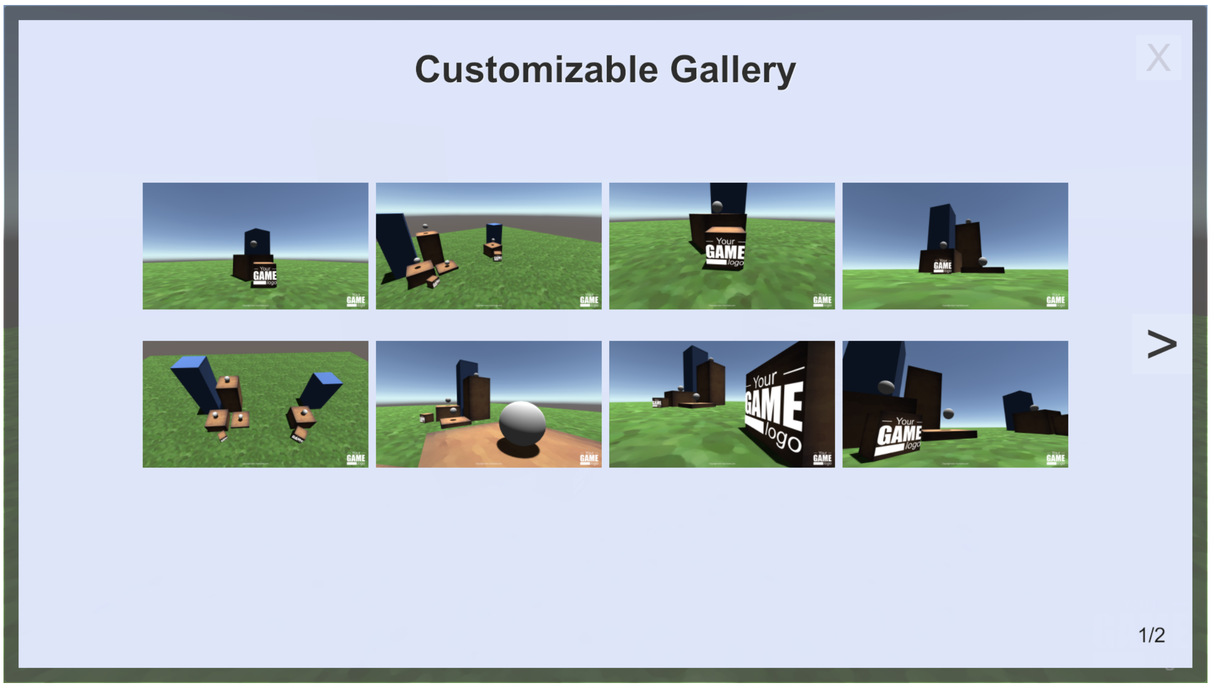


Figure 23: Example canvas for the GridGalleryCanvas component.

You can customize the canvas by creating a new canvas prefab and editing it as you want. You can also create a new component inheriting from GridGalleryCanvas or from Screenshot-Gallery, for instance to create a gallery script using an different layout than the grid layout.

Reference to the Greybox Canvas. By default, when an image is selected, a Greybox canvas is opened with that image to be displayed.

Automatic folder path. By default, you have to manually specify the folder path to be used for loading the screenshots from the device. If you are already using a ScreenshotManager, you can add a *SetScreenshotGalleryFolderPath* component to the *GridGalleryCanvas* gameobject to automatically set the path as the ScreenshotManager export directory.

Available Example: In the DefaultExample scene, you will find a functional example of the gallery grid, the greybox and the confirmation window.

9.1.2 Customizable Greybox

The GreyboxCanvas component is an example of component to handle the gallery image selection, for instance to zoom perform several actions.

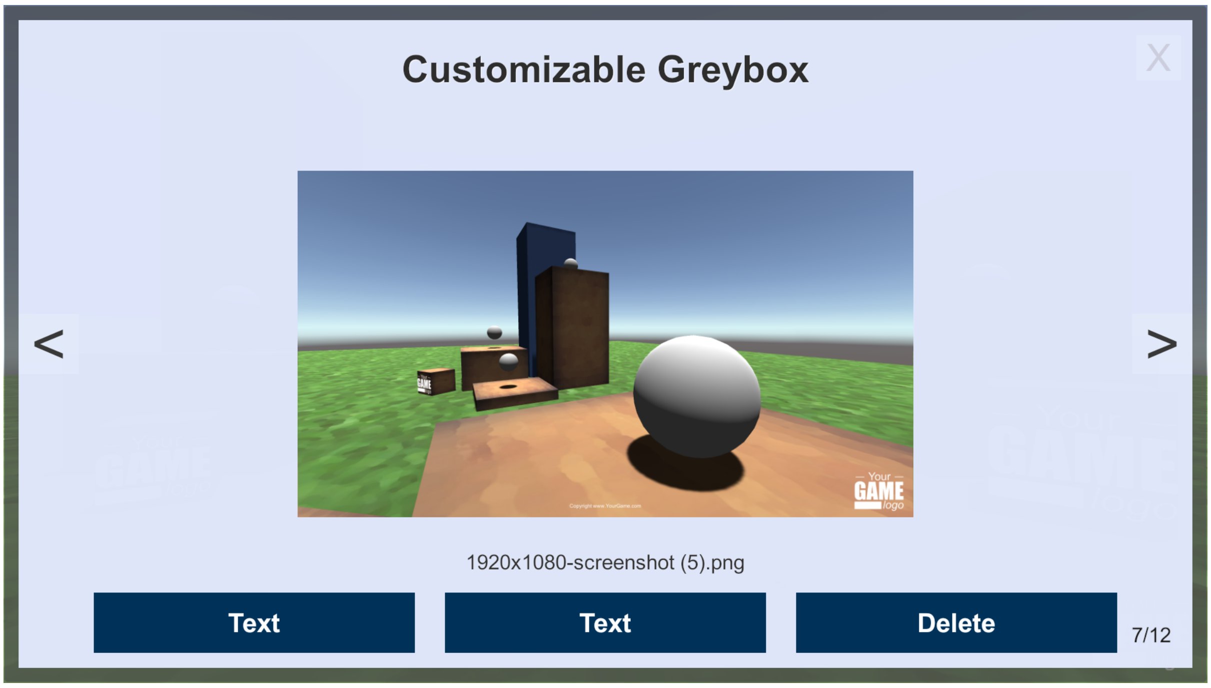


Figure 24: Example canvas for the GreyboxCanvas component.

One of the example action is to delete the selected screenshot. When clicked, the ui button shows the confirmation canvas, and the yes button calls the GreyboxCanvas delete method.

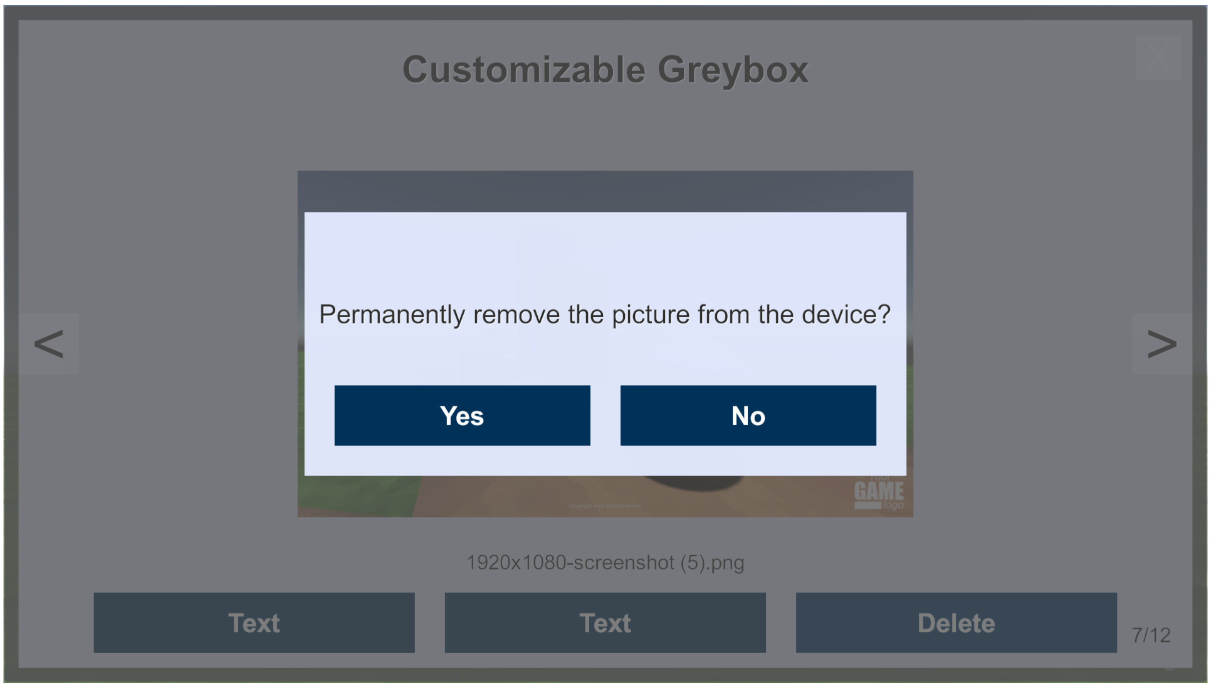

Customizable Greybox
Permanently remove the picture from the device?
Yes
No
1920x1080-screenshot (5).png
Text
Text
Delete
7/12

Figure 25: Example of confirmation window when removing a screenshot.

9.2 Display a Validation UI

The *ValidationCanvas* component shows how to call a capture event, and then display a validation ui to save or discard the screenshot. The idea is as follow:

1. Update the texture without exporting them using *UpdateAll()*.
2. Wait for the capture end event.
3. Display a UI and update its texture by accessing the screenshot texture.
4. Call *ExportAllToFiles()* if the user validates the screenshot.

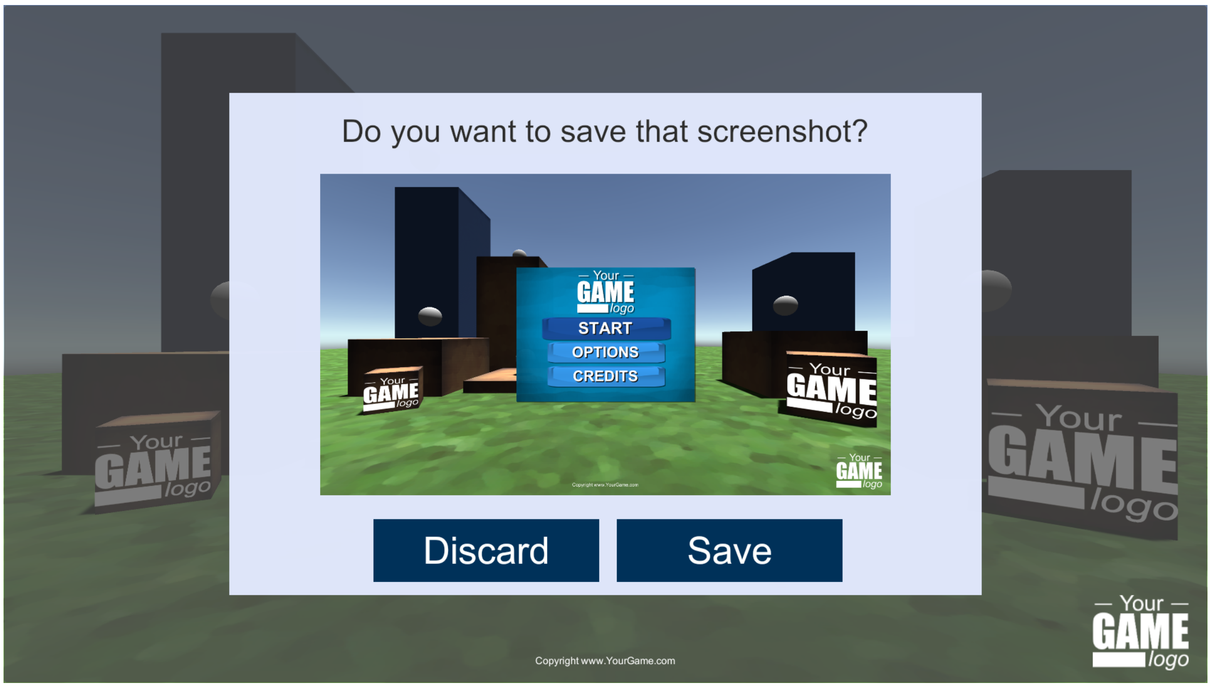

Do you want to save that screenshot?
Discard
Save
START
OPTIONS
CREDITS
Your GAME logo
Copyright www.YourGame.com

Figure 26: Example of validation canvas to preview the screenshot before saving it.

Available Example: In the DefaultExample scene, you will find a functional example of the ValidationGUIExample.

9.3 Thumbnail

The ShowScreenshotThumbnail script allows to temporarily show the screenshot thumbnail after each capture process. To show a thumbnail, just add the CustomizableThumbnailCanvas prefab to your scene. Note that you can customize it as you want, for instance by adding animations, etc.

9.4 Screenshot Cutter

If you want to capture only a sub-part of the screen, add the *ScreenshotCutter* component to one of your scene object. Set the SelectionArea property to the part of the screen you want to capture. It can be any RectTransform, such as an image or a UI panel. The *CutterExample* scene and *ScreenshotCutterExample* prefab illustrate a possible use of the screenshot cutter.

Note that the screenshot cutter only works on *GAMEVIEW_RESIZING* or *FIXED_GAMEVIEW* modes.

WARNING: The *ScreenshotCutter* will also cut the preview image of the gallery window and of the preview window. If a ScreenshotCutter is in your scene, disable its component to preview your game without the picture being cut.

9.5 Message Canvas

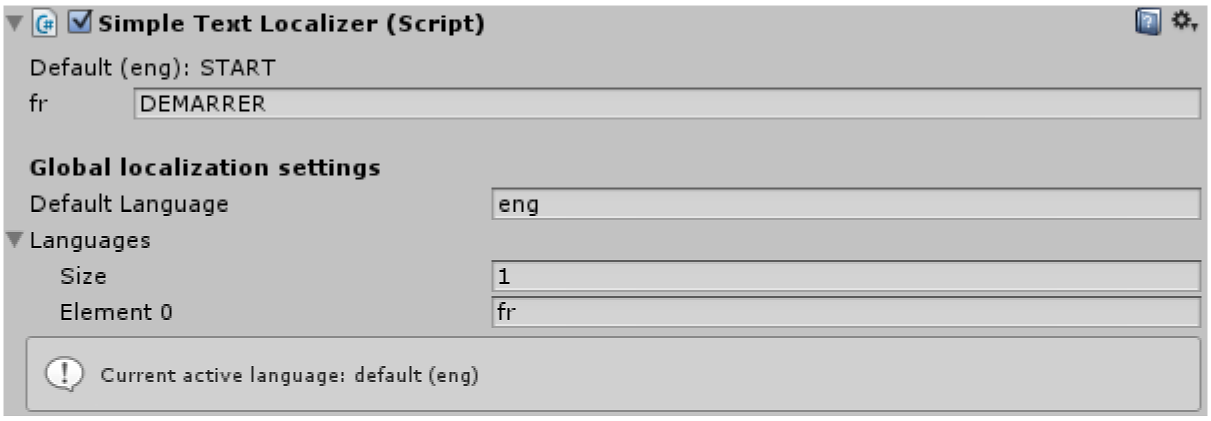
The message canvas component is a simple script to display some text when the screenshot manager export succeeds or fails. By adding the MessageCanvas prefab to your scene, the message will be displayed automatically. If you do not want any messages, simply remove the MessageCanvas game object.

10 SIMPLE LOCALIZATION

The Simple Localization asset allows to easily localize UI texts and images. It has been added freely to the Ultimate Screenshot Creator asset to enable the easy localization of your screenshots, but can be used independently to localize your whole game.

Language Settings. The LocalizationLanguages.asset file contains the list of the language IDs to be used by the localization components. It can be edited directly, or edited using the inspector of any SimpleTextLocalizer or SimpleImageLocalizer.

Text Localization. Add a SimpleTextLocalizer component to the UI text you want to localize. For each language id, write the localized text.



Simple Text Localizer (Script)

Default (eng): START

fr DEMARRER

Global localization settings

Default Language eng

Languages

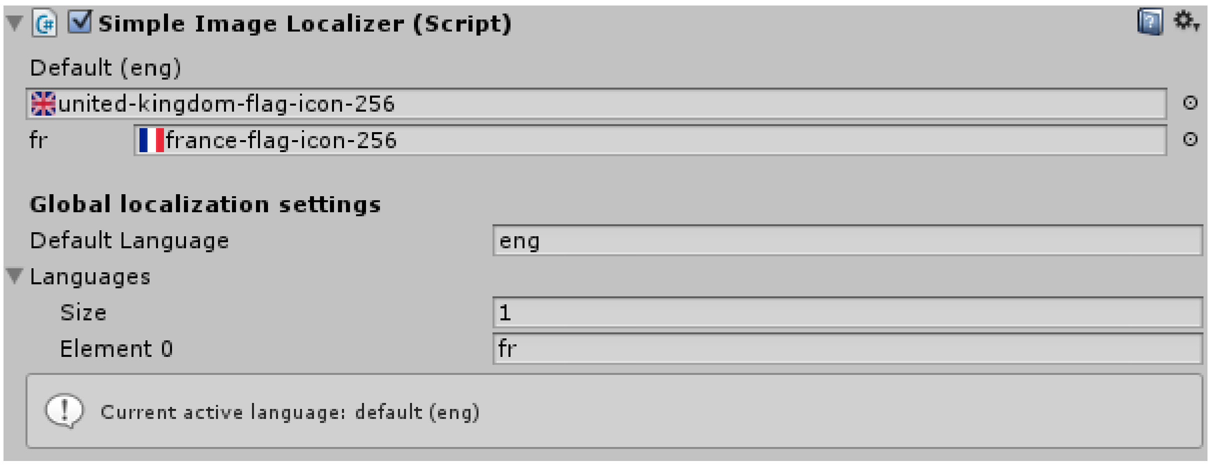
Size 1

Element 0 fr

Current active language: default (eng)

Figure 27: Text localization example.

Image Localization. Add a SimpleImageLocalizer component to the UI image you want to localize. For each language id, set the texture to be used.



Simple Image Localizer (Script)

Default (eng)

united-kingdom-flag-icon-256

fr france-flag-icon-256

Global localization settings

Default Language eng

Languages

Size 1

Element 0 fr

Current active language: default (eng)

Figure 28: Image localization example.

Changing the Language. In play mode, call the SimpleLocalizationLanguagesAsset.SetLanguage method to change the language of all text and images using the simple localization scripts.

11 FAQ

11.1 How to ...

11.1.1 Hide Some Scene Objects

Using a component You can hide a specific objet by adding the component *HideOnCapture* to the game object.

Using a culling layer You can hide all objects of a specific layer in *CUSTOM_CAMERAS* mode:

1. For each camera, select the camera and *CUSTOM_SETTINGS* culling mode.
2. For each camera, deselect the layers to be hidden in the culling list.

11.1.2 Access the screenshots taken by the ScreenshotManager

After the capture process, the textures can be accessed using the *ScreenshotManager GetActiveResolutions()* method, which returns a list of all active resolutions that were captured. It contains only one element in *FIXED_GAMEVIEW* mode. The texture used is stored in the *m_Texture* field, and its full file name in the *m_Filename* field.

11.1.3 Share the screenshots on social medias

This plugin does not have a feature to directly share on social medias. However, if you use an asset with social media sharing features (there are some assets on the asset stores providing native API calls to share on twitter, facebook, etc), you can call their methods with the created textures or screenshots.

For instance, using the the validation canvas feature, capture the screenshot and show the screenshot preview to the user (see Section 9.2). Customize the button text and script to add the functionality to share on social medias, by calling the appropriate methods of your social media asset. To get the screenshot texture or file name, please refer to Section 11.1.2.

11.1.4 Create a Screenshot with a Transparent Background

In *CUSTOM_CAMERAS* mode:

1. Select the first camera, and set *CUSTOM_SETTINGS* clear mode.
2. Select the *color* clear mode.
3. **IMPORTANT** Set the custom color to black and set its alpha to zero.
4. Set the export format to *PNG* and choose the *RGBA* color format.
5. If you have an issue with the alpha layer, try the *RecomputeAlphaLayer* settings.

11.1.5 Export using separated layers

Exporting to separated layers means that each camera will be rendered separately.

1. Set your export name. The {layer} symbol in the filename will be replaced by the name of the camera.
2. Set the capture settings to *PNG* and *RGBA*.
3. Set the camera mode to *CUSTOM_CAMERAS*, and toggle *Export to different layers*.
4. For each camera, expect the first one, set the custom color to black and set its alpha to zero, in order to have a transparent background.
5. Set your cameras culling masks and game object layers like you want.

Available Example: The SeparatedLayersExample scene contains a complete example.

11.1.6 Capture an offscreen scene

You can use the ScreenshotTaker to capture custom cameras to a texture, with your custom cameras being disabled in the scene. If your camera game objects are disabled, the asset will perform an offscreen rendering without enabling them, allowing you to capture something that is not displayed on the gameview.

To prevent any blincking, be sure to set the offscreen camera depth to -1.

Available Example: The OffscreenRenderingExample scene contains a "CaptureCamerasToTextureCanvas Example" gameobject with a CaptureCameraToTextureExample component.

11.1.7 Capture using an UI button

If you are using the ScreenshotManager, you need to call its Capture() method. Select the button, add an onClick event, set the manager as the target and select the Capture() method.

Available Example: The DefaultExample and OffscreenRenderingExample scenes contain several examples to call a capture method, using the ScreenshotManager, or direct script calls to SimpleScreenshotCapture or ScreenshotTaker.

11.2 Common issues

11.2.1 I have a compilation error on iOS

Check that you added the **Photos** framework dependency to the iOS plugin. Please refer to Section 6.3.2 for more details.

11.2.2 The PPI mode preview is different from what I see on my device

1. Check that your *Screen_PPI* value is correct.
2. Check that the device *PPI* value is correct. If you find a device preset with an incorrect value, please contact me.

3. Sometimes, Unity does not scale the device UI with the good PPI value. When your app is running on the target device, check that the *Screen.dpi* value returned by Unity matches the real screen PPI value. You can use the *ExampleCanvas* demo on your device to display the *Screen.dpi* value used by Unity to scale the UI. If it is different from the real device PPI, set the *Forced* PPI value to match the value returned by Unity.

4. Still not working? It's time to contact me.

11.2.3 I have some artefacts with temporal anti-aliasing, or other special effects

In editor, using the *GAMEVIEW_RESIZING* mode, if you have any artefact when taking a screenshot at a custom resolution, or a UI element at the wrong size, you can increase the *ScreenshotTaker m_GameViewResizingWaitingFrames* value. The *ScreenshotTaker* component is located on the same game object that the *ScreenshotManager*.

The *m_GameViewResizingWaitingFrames* value specifies how many frame the screenshot taker waits before to take the screenshot after the gameview has been resized. The default value of 2 should be enough for most settings. Increase this number when some elements are not well updated, like GUI, or when you see some post effects artefacts. Post effects like temporal anti aliasing requier a value of at least 10.

11.2.4 I can not create screenshots with the ScreenshotTaker

Check that you provide a valid full path as the filename. Please refer to Section 12.1 for more details.

11.2.5 I can not take screenshots on Android

If you get a message saying the folder or file can not be created, check the storage permissions. Please refer to Section 6.3.1 for a detailed configuration guide.

11.2.6 My GameView is blinking when I update the device or gallery previews.

This is expected. To render the previews, the plugin needs to render the game by deforming or resizing the GameView for a few milliseconds. More details on Section 11.2.7.

11.2.7 My GameView and layout are doing strange things when I capture a screenshot

Having the GameView deformed or blinking is expected when you use the *GAMEVIEW_RESIZING* capture mode.

For Unity 4.6 to 5.3, the algorithm rescales the GameView window to match the screenshot dimensions, and this undocks the GameView window. To prevent it, the editor layout is saved before the capture and restored after it, creating a sort of "flashing" effect. If this annoys you, you can set *Force Layout Preservation* to false.

With Unity 5.4 and later, the GameView has an inner "scale" which allows the modification of its dimensions without modifying the editor layout. During the capture, its resolution is changed several times before being restored, creating a sort of "blinking".

11.2.8 Nothing Happens when I try to Update the Preview(s)

Sometimes, the renderer may be locked in Editor and refuse to update the preview. This happens rarely, but if you do not have any response from the *ScreenshotManager*, stop and play the game.

Do not hesitate to contact me if you can reproduce this bug, so I can correct it.

12 PROGRAMMING

12.1 API

12.1.1 ScreenshotTaker

The ScreenshotTaker is the component used to capture the screenshots to textures. You can use it directly to capture a texture by calling one of its capture coroutine.

```
/// <summary>
/// Captures the current screen at its current resolution.
/// The texture will be resized if needed to match the capture settings.
/// </summary>
public IEnumerator CaptureScreenToTextureCoroutine (Texture2D texture,
                              bool captureGameUI = true,
                              ScreenshotTaker.ColorFormat colorFormat =
                                  ScreenshotTaker.ColorFormat.RGB,
                              bool recomputeAlphaMask = false)




/// <summary>
/// Captures the scene with the specified width, height, using the mode
    RENDER_TO_TEXTURE.
/// Screenspace Overlay Canvas can not be captured with this mode.
/// The texture will be resized if needed to match the capture settings.
/// </summary>
public IEnumerator CaptureCamerasToTextureCoroutine (Texture2D texture, int width
    , int height,
                               List<Camera> cameras,
                               int antiAliasing = 8,
                               ScreenshotTaker.ColorFormat colorFormat =
                                   ScreenshotTaker.ColorFormat.RGB,
                               bool recomputeAlphaMask = false)




/// <summary>
/// Captures the game with the specified width, height.
/// The texture will be resized if needed to match the capture settings.
/// </summary>
public IEnumerator CaptureToTextureCoroutine (Texture2D texture, int width, int
    height,
                            List<Camera> cameras = null,
                            List<Canvas> canvas = null,
                            ScreenshotTaker.CaptureMode captureMode = ScreenshotTaker
                                .CaptureMode.RENDER_TO_TEXTURE,
                            int antiAliasing = 8,
                            bool captureGameUI = true,
                            ScreenshotTaker.ColorFormat colorFormat = ScreenshotTaker
                                .ColorFormat.RGB,
                            bool recomputeAlphaMask = false)


public IEnumerator CaptureAllCoroutine (List<ScreenshotResolution> resolutions,
                       List<ScreenshotCamera> cameras,
                       List<ScreenshotOverlay> overlays,
                       CaptureMode captureMode,
                       int antiAliasing = 8,
                       bool captureGameUI = true,
                       ColorFormat colorFormat = ColorFormat.RGB,
                       bool recomputeAlphaMask = false,
                       bool stopTime = false,
                       bool restore = true)
```

12.1.2 SimpleScreenshotCapture

You can capture a screenshot using only the SimpleScreenshotCapture. That static class provides several capture methods, that you can choose depending on your needs.

Note that you need to provide a **valid full path** *fileName* to the Screenshot Taker, for instance *C:/Screenshots/myScreenshot.png*. You can use the ScreenshotNameParser.ParsePath() or ScreenshotNameParser.ParseFileName() to get one.

```
/// <summary>
/// Captures the current screen at its current resolution.
/// The texture will be resized if needed to match the capture settings.
/// </summary>
public IEnumerator CaptureScreenToTextureCoroutine (Texture2D texture,
                              bool captureGameUI = true,
                              ScreenshotTaker.ColorFormat colorFormat =
                                  ScreenshotTaker.ColorFormat.RGB,
                              bool recomputeAlphaMask = false)


/// <summary>
/// Captures the scene with the specified width, height, using the mode
    RENDER_TO_TEXTURE.
/// Screenspace Overlay Canvas can not be captured with this mode.
/// The texture will be resized if needed to match the capture settings.
/// </summary>
public IEnumerator CaptureCamerasToTextureCoroutine (Texture2D texture, int width
    , int height,
                               List<Camera> cameras,
                               int antiAliasing = 8,
                               ScreenshotTaker.ColorFormat colorFormat =
                                   ScreenshotTaker.ColorFormat.RGB,
                               bool recomputeAlphaMask = false)

/// <summary>
/// Captures the game with the specified width, height.
/// The texture will be resized if needed to match the capture settings.
/// </summary>
public IEnumerator CaptureToTextureCoroutine (Texture2D texture, int width, int
    height,
                            List<Camera> cameras = null,
                            List<Canvas> canvas = null,
                            ScreenshotTaker.CaptureMode captureMode = ScreenshotTaker
                                .CaptureMode.RENDER_TO_TEXTURE,
                            int antiAliasing = 8,
                            bool captureGameUI = true,
                            ScreenshotTaker.ColorFormat colorFormat = ScreenshotTaker
                                .ColorFormat.RGB,
                            bool recomputeAlphaMask = false)
```

12.2 Delegates

If you want to run some custom code before and after the capture processes, you can use the following delegates:

12.2.1 Screenshot Manager

```
public static void onCaptureStartDelegate ()
```

Is called when the capture starts.

```
public static void onCaptureEndDelegate ()
```

Is called when the capture ends.

```
public static void onResolutionExportSucessDelegate (ScreenshotResolution res)
```

Is called just after the screenshot is exported.

```
public static void onResolutionExportFailedDelegate (ScreenshotResolution res)
```

Is called just after the screenshot export fails.

12.2.2 Screenshot Taker

```
public static void onResolutionUpdateStartDelegate (ScreenshotResolution res)
```

Is called just before capturing the given resolution.

```
public static void onResolutionScreenResizedDelegate (ScreenshotResolution res)
```

Is called just after resizing the gameview in *GAMEVIEW_RESIZING* mode.

```
public static void onResolutionUpdateEndDelegate (ScreenshotResolution res)
```

Is called just after capturing the given resolution.

13 VERSIONS

1.7.0 - 13/01/2019

READ BEFORE UPDATING . Several files have been moved or removed in update 1.7. It is recommended to do a clean re-install by removing the AlmostEngine folder before importing the update. Do not forget to backup your custom prefabs before doing so. Otherwise, follow the upgrade process described below.

Upgrade Instructions from 1.6 to 1.7. If you want to manually move and remove the files instead of doing a clean reinstall, do as follow:

1. Move MultiDevicePreviewConfig.asset to AlmostEngine/MultiDevicePreview/Assets/. **Not doing it will result in large build resource files.**
2. Move ScreenshotWindowConfig.asset to AlmostEngine/UltimateScreenshotCreator/Assets/. **Not doing it will result in large build resource files.**
3. Move ScreenshotResolutionPresets.cs to AlmostEngine/UltimateScreenshotCreator/Assets/Scripts/Editor.
4. Remove DeviceResolutionPresets.cs
5. Remove PopularityResolutionPresets.cs
6. Remove RatioResolutionPresets.cs
7. Remove ResolutionDebugPresets.cs

Do not hesitate to contact me if you have any issues or questions.

Version Changes.

- **(New feature)** Full device preview for more than 40 devices, including phones with notches.
- Contains now more than 200 mobile phone and tablet presets.
- New popularity presets for most used smartphones, steam hardware.
- New preset collections.
- Improved resolution list display.
- **(New feature)** Generate hundreds of multilingual promotional pictures with the composer and batch process.
- **(New feature)** Create custom presets and collections saved as assets that can be shared between projects.
- **(New feature)** Display a screenshot thumbnail when taking screenshots.
- **(New feature)** A localization feature has been added to easily localize your screenshots, and can be used independently to localize your whole game.
- Possibility to automatically remove the permission requirements.
- Automatically add the Photos framework dependency.
- Support of Unity 2018.3.
- (Fix) Fixed gameview size on retina displays.
- (API) Presets are now asset files.

- (API) Popularity preset are now asset files, referencing preset assets.
- (API) Moved ScreenshotManager specific symbols parsing to the ScreenshotManager.
- (API) Moved all extra features to the namespace AlmostEngine.Screenshot.Extra.
- (API) Gallery and preview windows do not use the ScreenshotManager anymore.
- (API) Gallery and preview windows new delegates: onUpdateBeginDelegate, onUpdateEndDelegate.

1.6.2 - 16/11/2018

- iOS Photo Framework dependency now automatically set for Unity 5.6 or newer.
- (Fix) Compilation issue on WebGL.

1.6.1 - 11/09/2018

- (Fix) Compilation issue on iOS with photo usage description.

1.6.0 - 28/08/2018

- **(New feature)** Customizable ingame gallery to manage screnshots.
- **(New feature)** Export to separated layers.
- Customize iOS usage description.
- Improved API to simplify taking screenshots without the ScreenshotManager. The ScreenshotTaker now only contains the methods to capture screenshots. It can be called directly to capture textures. The sound and export have been moved to the ScreenshotManager script. The SimpleScreenshotCapture is a new static to easily capture and export screenshots from script. See Section 12.1 for more details.
- Support of Unity 2018.2.
- New scene example which shows how to capture offscreen scenes.
- New example scrips to better illustrate the asset possibilities.
- Improved documentation to better explain the different way to use the Ultimate Screenshot Creator asset.

1.5.3 - 17/05/2018

- **(New feature)** Added support for retina displays for Unity 5.4 and newer.
- **(New feature)** It is now possible to partially exclude the asset from iOS and Android builds to remove the need for any permission. See Section 5.6 for mode details.
- The preview and gallery windows have now independent display modes.
- Improved documentation for iOS and Android configuration.
- Added several sections in FAQ.
- (Fix) The preview, gallery and settings windows are now repainted when one setting is changed in any window.
- (Fix) Image format setting is now visible in $FIXED\_GAMEVIEW$ mode.

1.5.2 - 05/04/2018

- (Fix) Android crash with Unity 2017.3.

1.5.1 - 30/03/2018

- Changed the default hotkeys for Unity 5.0 to 5.2.
- (Fix) Removed automatic cameras when custom list is empty.

1.5.0 - 21/02/2018

Version 1.5 contains several important changes. **It is recommended to do a clean re-install by removing the MultiDevicePreview folder before importing the update.**

- There is no more PreviewManager prefab, it has been replaced by the Settings Window (Section 5).
- PreviewManager hotkeys have been replaced by MenuItems hotkeys (Section 5.7). They now works anywhere within the editor.
- **(New feature)** Preview device resolutions are automatically added to the GameView presets.
- Auto-refresh now works in edit mode.

Other modifications:

- Support of Unity 2018.1.
- **(New feature)** Screenshot Window to easily take screenshots in the editor without a manager.
- It is possible to use prefabs as screenshot overlay canvas.
- You can now specify a waiting time in seconds between each screen resize in $GAMEVIEW\_RESIZING$ mode.
- API changes. Many methods previously in the ScreenshotManager have been moved to other scripts. Resolutions and cameras methods have been moved to the ScreenshotConfig. The member m_GameViewResolution has been moved to the ScreenshotConfig. Name parsing has been moved to a new script ScreenshotNameParser. Name presets have been moved to a new script ScreenshotNamePresets.

1.4.3 - 09/01/2018

- Support of Unity 2017.3.
- **(New feature)** Validation canvas to preview the screenshot before saving.
- You can now request the iOS gallery authorization when you want.
- Added the possibility to increase the number or waiting frames in $GAMEVIEW\_RESIZING$, to prevent post effect artefact like temporal anti aliasing.
- Updated documentation and FAQ.
- (Fix) WebGL plugin error.
- 12 new phone presets.
- Updated steam hardware popularity presets.
- (Fix) Device selection memory in preview window.

1.4.2 - 01/11/2017

- (Fix) GameViewResizing in NET 4.6.

1.4.1 - 26/10/2017

- Support of Unity 2017.2.
- Asset renamed from *MultiResolution Preview Gallery* to *Multi Device Preview & Gallery*. Some folders and scripts have been renamed accordingly.
- Cosmetic update: logo updated.
- (Fix) File name update in burst mode.

1.4.0 - 17/10/2017

- **(New feature)** Screenshot cutter, capture only a sub-part of the screen.
- Added support to export to secondary storages on Android.

1.3.0 - 27/09/2017

In order to simplify the use of the preview gallery, I separated the features of the Gallery Preview from the *ScreenshotManager*. There are now two distinct managers: the *PreviewManager* is used by the preview windows for device preview; the *ScreenshotManager* is used only for taking screenshots. The documentation has been reorganized accordingly.

- **(New feature)** Live preview on a specific device on play mode.
- **(New feature)** Preview Window for previewing a specific device.
- Added scroll area on the Gallery Window.
- 8 new phone presets.
- Better API for developers.
- (Fix) iOS picture export folder.

1.2.2 - 12/09/2017

- Added support for WebGL.

1.2.1 - 29/08/2017

- (Fix) Android picture export folder.

1.2.0 - 16/08/2017

- Support of Unity 2017.1.
- Updated statistics.
- **(New feature)** Capture mode: $FIXED\_GAMEVIEW$ to capture the game screen on all platforms, including UI.
- **(New feature)** Export mode: $PICTURE\_FOLDER$, to export in the plaform specific picture folder.
- **(New feature)** Automatically export the screenshots to the Android gallery.
- **(New feature)** Automatically export the screenshots to the iOS gallery.
- Multi-display support.
- Added a message canvas do display a success or failure text in-game.

1.1.2 - 26/04/2017

- Fixed inspector background color in Unity Pro.
- Improved example camera controller.

1.1.1 - 03/04/2017

- Support of Unity 5.6.
- MultiSampling AntiAliasing support in *RENDER_TO_TEXTURE* mode.

1.1.0 - 23/03/2017

- Added more phone presets.
- Updated statistics.
- **(New feature)** Capture mode: *RENDER_TO_TEXTURE*.
- **(New feature)** Possibility to capture screenshots in builds on all platforms using *RENDER_TO_TEXTU*
- Orientation is now a ScreenshotResolution property.
- Extracted code from ScreenshotManager.cs to ScreenshotTaker.cs for all capture related code, and to TextureExporter.cs for all export related code.
- Added the possibility to recompute the alpha layer in RGBA if destructed by camera effects.
- Programmer friendly API, can now call a capture from code.
- Tooltips added for the inspector.

1.0.0 - 08/03/2017

- First public release.

14 SUPPORT

Please do not hesitate to contact me if you have a feature request, or any question, issue or suggestion : support@wildmagegames.com.