uap

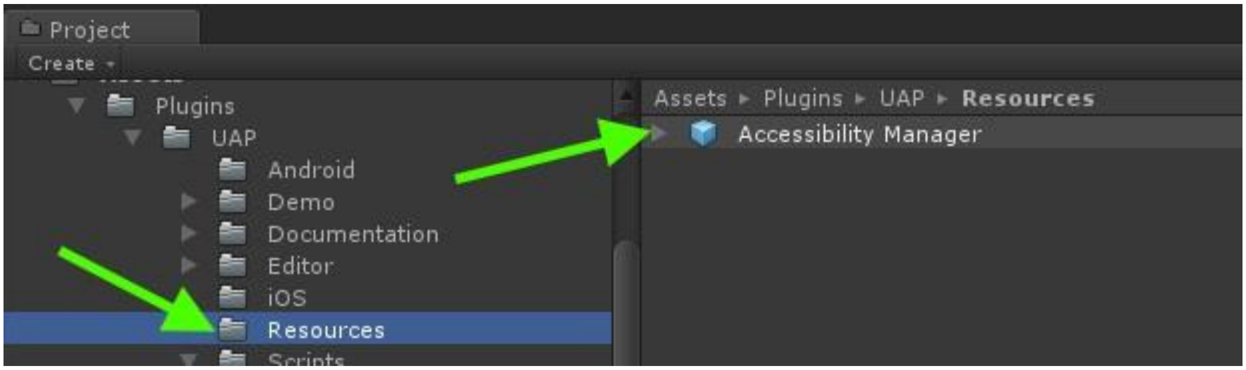
UI Accessibility Plugin

Quick Start Guide

Here's how to make your menus and UI accessible quickly - without unnecessary explanations or fine-tuning.

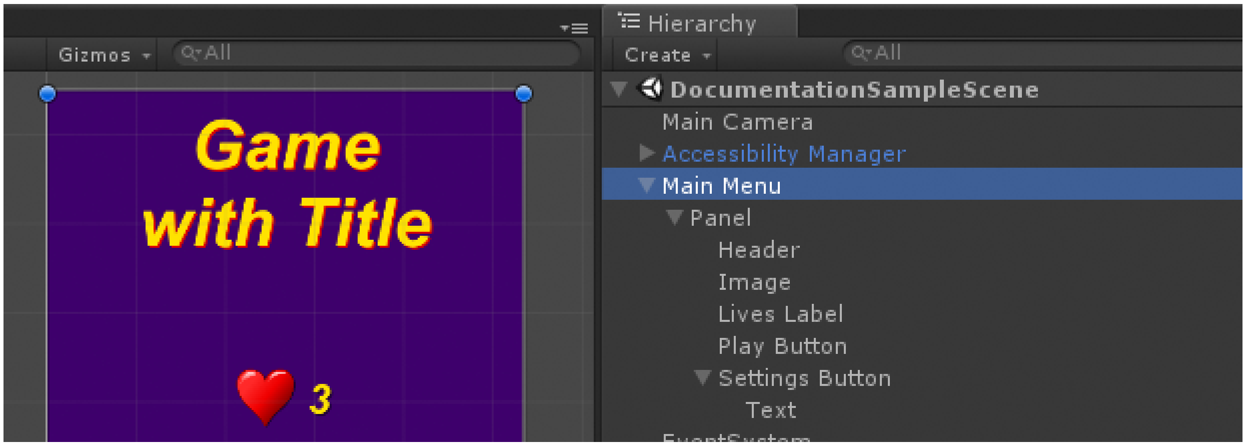
Step 1

Add the Accessibility Manager prefab to your scene.
You can play around with the settings later. For most applications, the default settings will be fine.



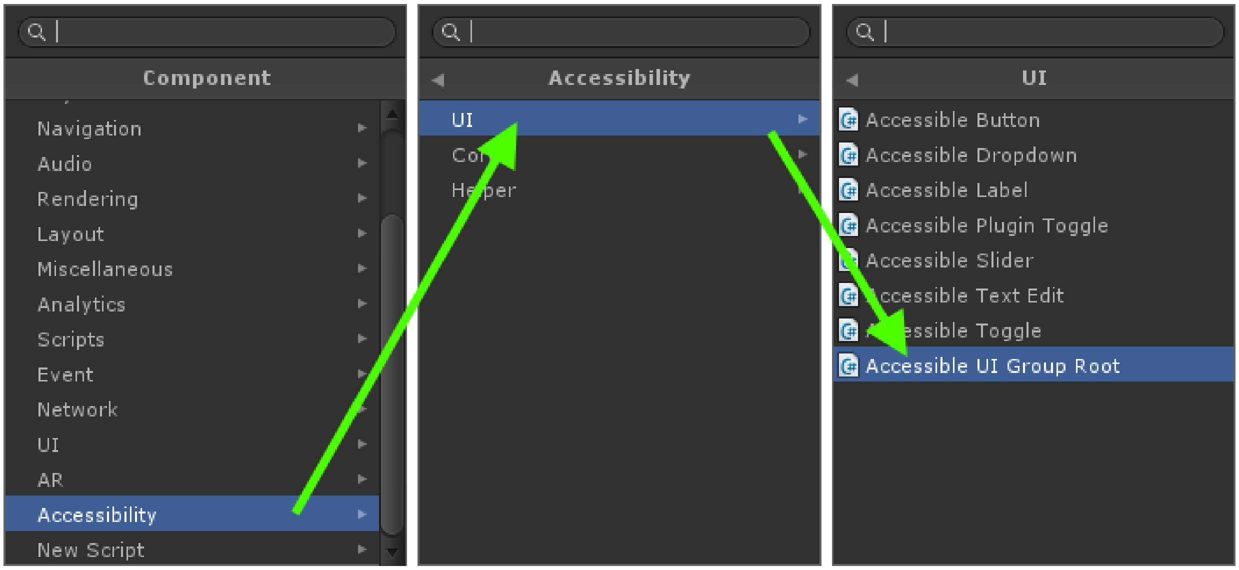
Step 2

Drag one of your menu screens into the scene.
Every UI element that needs to be accessible to the user will get an additional component attached to it in the next step. To read in more detail about these components and what they do, head over to Plugin Basics.

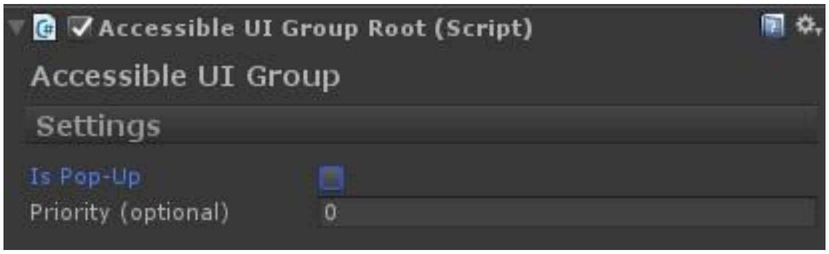


Step 3

Add an ***Accessible UI Group Root*** component to the root of your menu screen
Hint: The *Root* is usually a Canvas or a Panel, or simply the root of your prefab.



If your prefab/screen/dialog is a popup, you need to check the ***Popup* check box**.



☑ Accessible UI Group Root (Script)

Accessible UI Group

Settings

Is Pop-Up □

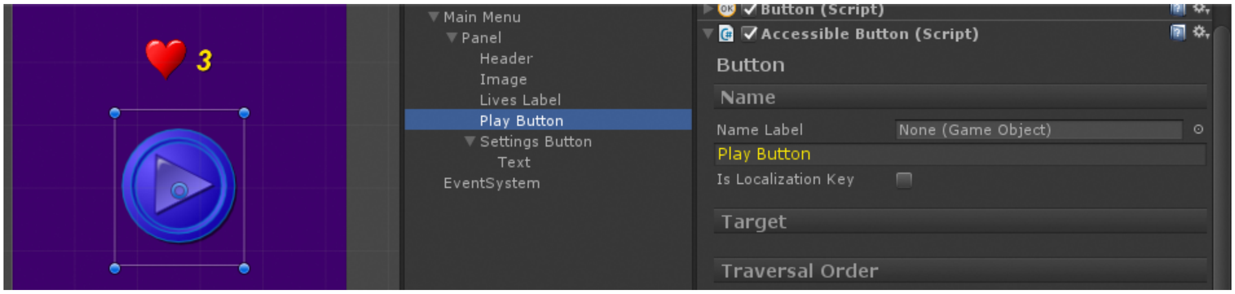
Priority (optional) 0

Step 4

Select a *GameObject* in your menu screen hierarchy that has a UI element on it that needs to be accessible.
These are *Labels, Buttons, Toggles, DropDown Lists, Sliders, Images and Input Edit Fields*

Step 5

Add the appropriate accessibility component to the *GameObject*
Example: Add the *Accessible Button* component to the *GameObject* containing the uGUI Button component (or UIButton in case of NGUI)



Name Label: None (Game Object)
Play Button
Is Localization Key: □

You do not need to add label components to the labels of buttons (or toggles). The *Accessible Button* component will find the label automatically if it is a child of the button and reference it internally.

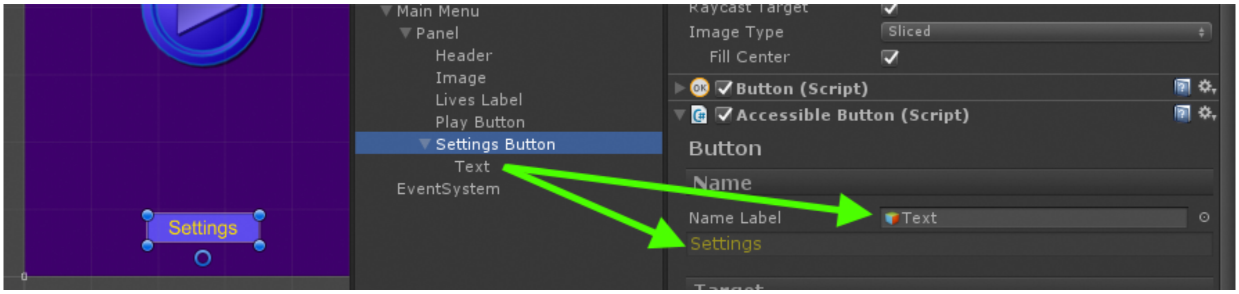


Image Type: Sliced
Fill Center: ☑
Name Label: Text
Settings

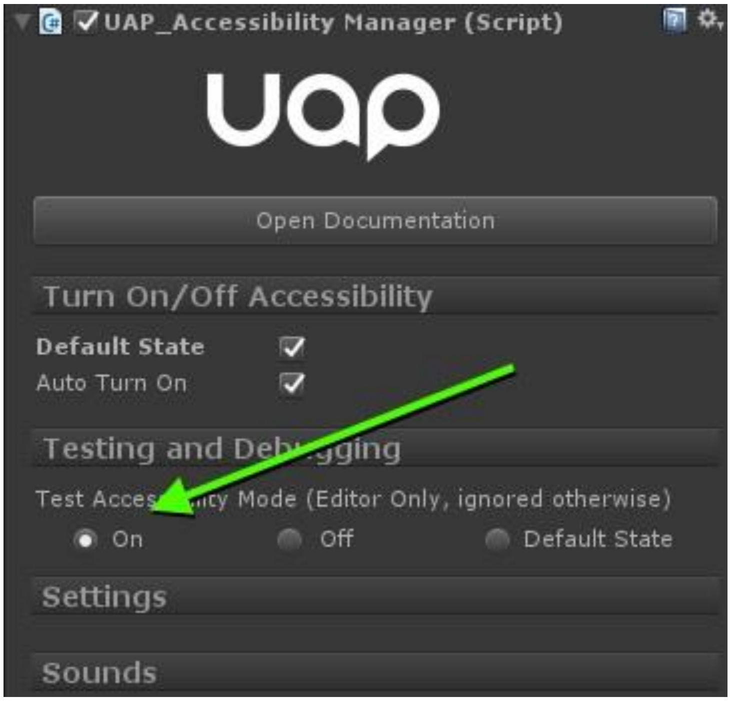
Hints: Use accessible labels to add text to images Write custom text into the 'Name' field of the accessibility component if the automatic text isn't appropriate.

Step 6

Repeat the steps 2 through 5 for all UI elements in all your menus until everything is marked up.

Step 7

Select the Accessibility Manager in your scene and enable Editor Test Mode in the *Testing and Debugging* rollout.
You can now jump into game mode and test your accessibility.


☑ UAP_Accessibility Manager (Script)
uap
Open Documentation
Turn On/Off Accessibility
Testing and Debugging
Settings
Sounds

Default State ☑
Auto Turn On ☑

Test Acces…ncy Mode (Editor Only, ignored otherwise)
● On ○ Off ○ Default State

Check out the How To Guides to find answers to the most common questions and use cases.
To make your app's accessibility shine, take a look at Best Practices.
To learn more about how the plugin works in detail, go to Plugin Basics.

Notes

Testing on Windows

If you are testing on Windows and do not have an actual screen reader installed, the plugin will use Window Speech Synthesis as a fallback for debugging purposes. This system works well, but speech is generated with a very noticeable **delay.**
This delay is caused by Microsoft SAPI and not the plugin. It will not appear on mobile devices or when testing with NVDA under Windows.
Since this can make testing cumbersome, it's recommended that you download NVDA, a free and very popular screen reader.

It can be downloaded here:
NVDA - NonVisual Desktop Access

UI Navigation

Keyboard navigation on a desktop computer is different from swipe navigation on mobile devices.
Use the up and down arrows to navigate.
See Navigation for more details.

Accessibility State

By default, the accessibility features will be disabled.
The plugin will turn itself on if an accessibility service (such as TalkBack or VoiceOver) is detected.
Alternatively, the user of the app can enable them either through a magic gesture, or through a menu (if you offer one).

Help and Support

The full documentation of the plugin can be found here. Head over to our support forum if you have any more specific questions.
You can also find a quick video tutorial here: UAP Basic Tutorial