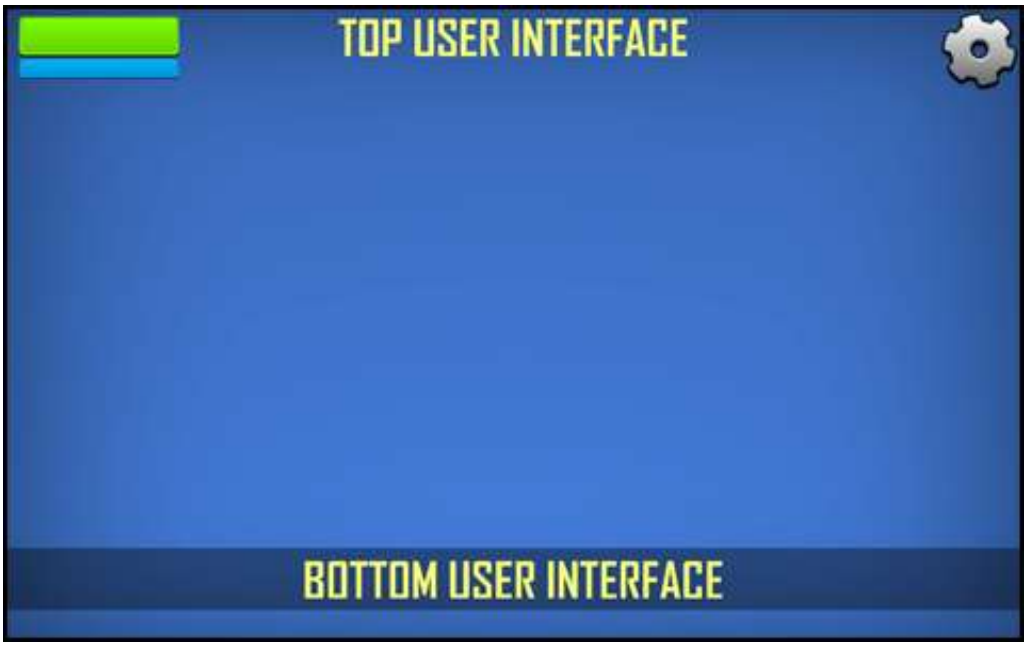
Auto Aspect Ratio

Game documentation and HowTo guide.



This document contains:

Package Description and features

This script allows you to fit your game for multiple aspect ratios and device screens. It is intended for 3D or 2D sprite based interfaces (Unity 4.3). Not intended for use with unity GUI.

WATCH PRESENTATION VIDEO

This project is a collaboration between Puppeteer & Jordan Swapp

Available in C# and JS

How to use?

Drag & drop to scene, assign your UI objects/background and that's all!

Features:

- Easy to use.
- Supports both Orthographic and Perspective camera.
- Works with 3D and 2D sprite based interfaces (Unity 4.3).
- Available in both C# and JS code.
- Includes presets for most common aspect ratios (4:3, 3:2, 5:4, 5:3, 16:9, 16:10, etc).

Current version 1.0

Please rate my file, I'd appreciate it 🙂

Overview of the game's library contents

Let's take a look inside the game files. Open the main AARAssets folder using Unity3D 4.3 or newer. Take a look at the project library, usually placed on the right or bottom side of the screen. Here are the various folders inside:

- **CS_Assets > CS_Prefabs:** Holds two prefabs with ready-made configurations for an Orthographic and a Perspective camera. You can drag and drop this in your scene, assign which UI elements/background you want to affect and start your game.

- **CS_Assets > CS_Scenes:** Contains the example scenes for both camera types (Orthographic and Perspective).

- **CS_Assets > CS_Scripts:** Contains the script written in C# which handles all the aspect ratio functions. You can practically drag this folder to another project and use it. But you will need to manually assign the aspect ratio presets.

- **Fonts:** Holds the font used in the example. The font is AGENCYB.

- **JS_Assets > JS_Prefabs:** Holds two prefabs with ready-made configurations for an Orthographic and a Perspective camera. You can drag and drop this in your scene, assign which UI elements/background you want to affect and start your game.

- **JS_Assets > JS_Scenes:** Contains the example scenes for both camera types (Orthographic and Perspective).

- **JS_Assets > JS_Scripts:** Contains the script written in JS which handles all the aspect ratio functions. You can practically drag this folder to another project and use it. But you will need to manually assign the aspect ratio presets.

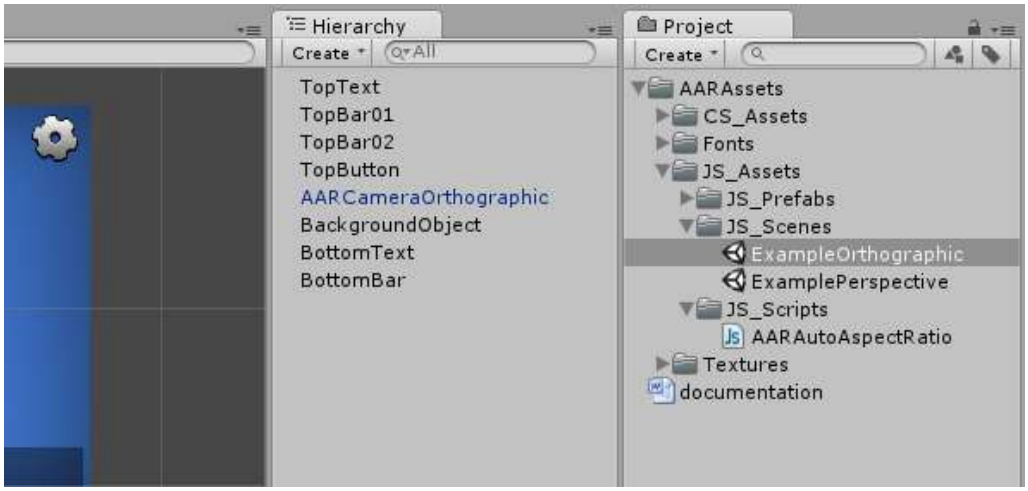
- **Textures:** Holds all the textures used in the examples.

How to use

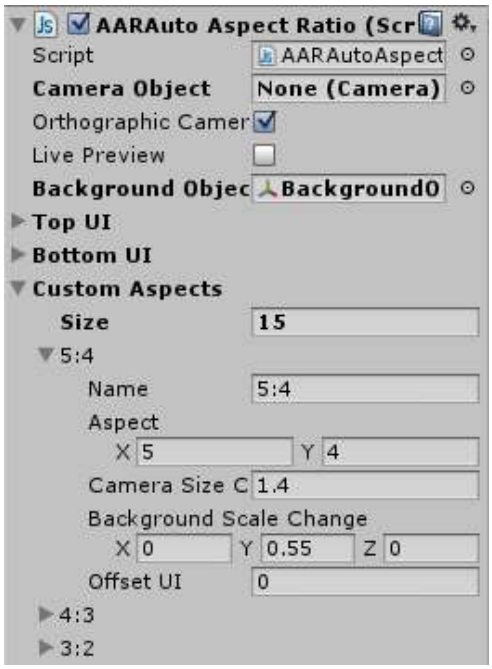
This script allows you to fit your game for multiple aspect ratios and device screens. It works by calculating the current aspect ratio of the camera, comparing it to the preset aspects, and then moving the UI elements, scaling the background, and resizing the camera based on those settings.

It is intended for 3D or 2D sprite based interfaces (Unity 4.3). It is not intended for use with the old default unity GUI.

Start by opening your project, and then importing AAR package to your project. A new folder will be created (AARAssets). Let's take a look at the ready example to see how it works. Open the orthographic example scene, and select **AARCameraOthrographic**.



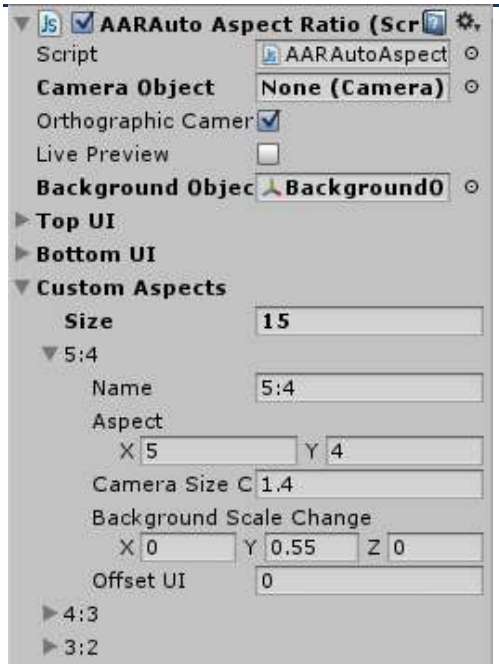
This object contains everything you need to make the aspect ratios work. Let's look at it.



Camera Object: Here you can assign the camera object that will be used for auto aspect. If you don't assign a camera, the main camera in the scene will be used instead.

Orthographic Camera: Set this to **true** if you want to use an Orthographic camera in your game (best for 2D games), or set this to **false** if you want to use a Perspective camera for 3D games.

Live Preview: Set this to **true** if you want to see changes made to the aspect ratio settings (camera/background size, UI positions) during gameplay. This is used to preview any changes you may want to make, but remember to set this to **false** before you publish your game.



Background Object: Here you can assign the background object which will be scaled based on the aspect ratio.

Top UI: Here you can set all the UI objects which are aligned to the top of the screen.

Bottom UI: Here you can set all the UI objects which are aligned to the bottom of the screen.

Custom Aspects: Here you can set all the settings for different aspect ratios or resolutions. We'll take the first element as an example:

Name: This is the name of the aspect ratio setting we are working with. It's just for convenience. You can name it anything you like (ex: iPad, iPhone, etc)

Aspect: This is the aspect ratio we are targeting. **5:4** means that any screen size in which the ratio of the width to the height is 5/4 (1.25) will get the relevant settings applied to it (camera size, UI position, background scale).
You can set a resolution instead of an aspect ratio and it will work the same. For example you can set the aspect value to **1024** by **768**, and it will detect that resolution and apply the changes accordingly.

Camera Size Change: This is the change in camera size (or zoom) that will be applied to the camera in this aspect ratio. If you are using an Orthographic camera the change is applied to the camera's orthographic size, and if you are using a Perspective camera the change will be applied to the camera's Z position (zoom in/out). In the example above we increased the camera size by 1.4, meaning that we zoomed out.

Background Scale Change: This is the change in scale that will be applied to the background in this aspect ratio. In the example we increased the Y scale of the background to make it fit the zoomed out camera.

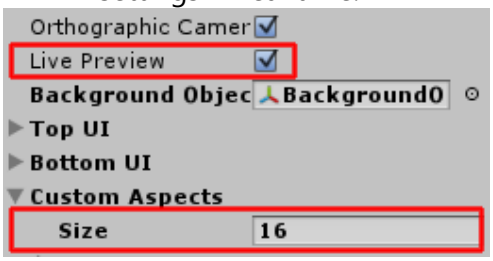
Offset UI: This is the change in the Y positions of the top and bottom UI elements. In the example we kept it at 0, meaning that the UI elements will not move at all. If we set it to 1, the top UI elements will move up 1 unit, while the bottom UI elements move down 1 unit.

Adding your own custom aspect ratios

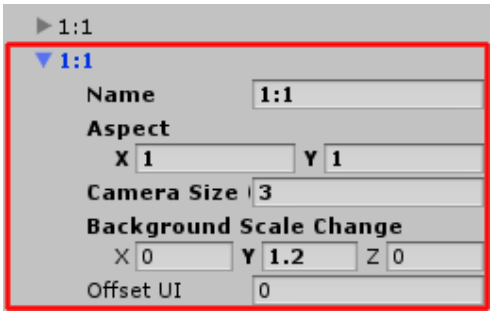
One of the easiest ways to set your own custom aspect ratios is to start the game in the editor while **Live Preview** is on, fine tune the settings for the custom aspect ratio, copy the component, stop the game, and then paste the component.

Let's do it step by step:

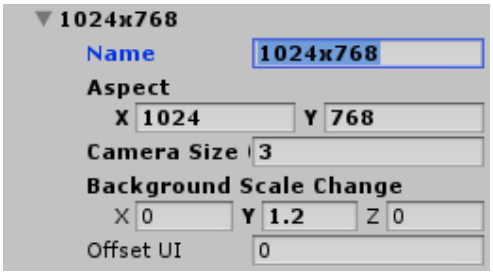
- Add a new aspect ratio setting to the list of **Custom Aspects** by increasing it to 16. Also don't forget to set **Live Preview** to **true**. This is so that we can see the changes in the settings in real time.



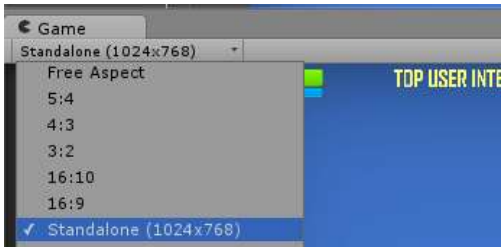
- The new custom aspect ratio is duplicated from the last element in the list (1:1).



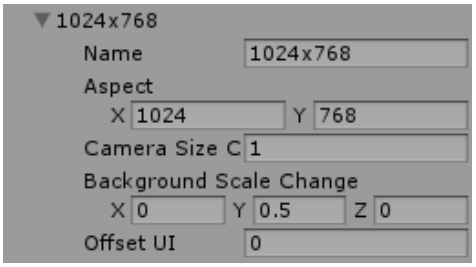
- Change the name to 1024x768 (just for display purposes).
- Change the aspect X to 1024 and the aspect Y to 768.



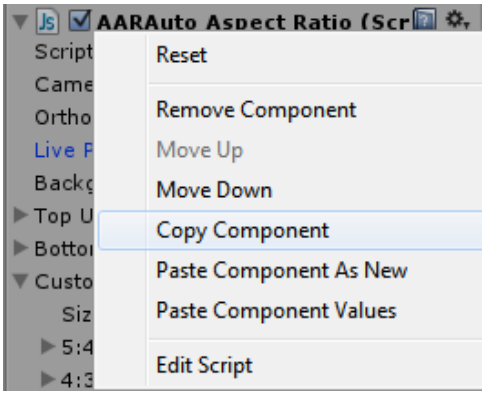
- Click the Play button to start the game, and then change the aspect ratio in the **Game** tab to **Standalone (1024x768)**.



- You'll notice that the game area changed a little. This is because the AAR script detected the new resolution and applied the settings to camera/background size and UI positions accordingly.
- Now you can go on and start tinkering with the settings to see the changes in the game area in real time. Try to change **Camera Size Change** for example and you'll see the results.
- After changing the settings we got to the right results that we want.



- And now we want to apply the changes in the game. If we stop the game now, all settings will be "forgotten" by Unity. To solve this we right click on the component and copy it.



- Now we can stop the game.
- After we stopped the game, right click on the component and Paste Component Values.
- This will apply the changes we made during play to the Unity editor.
- We're done!

It is highly advised, whether you are a designer or a developer to look further into the code and customize it to your pleasing. See what can be improved upon or changed to make this file work better and faster. Don't hesitate to send me suggestions and feedback to puppeteerint@gmail.com

Follow me on twitter for updates and freebies!

Good luck with your modifications!