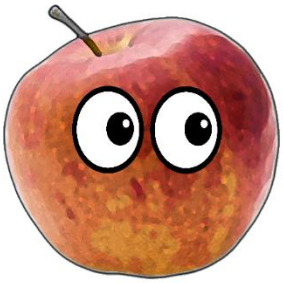
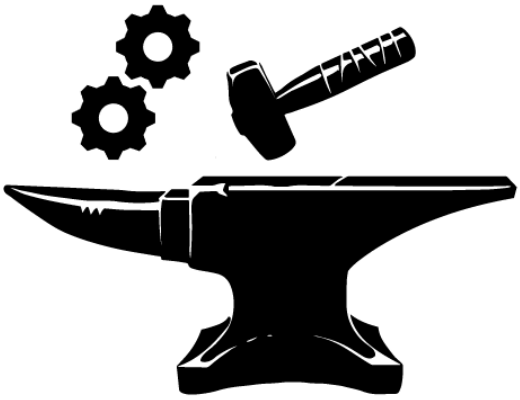


Daft Apple Games Presents…



3D Forge Components

User Guide

Version: 1.0

Contents

Introduction

Thank you for downloading 3D Forge Components by Daft Apple Games. This Unity package consists of components, scripts and prefabs that will help you get the most out of the beautiful buildings and blueprints from 3D Forge.

Getting Help

Please read this document carefully, and if you have any questions, comments, suggestions, or issues, please contact me through one of these channels:

- **Discord**: https://discord.com/channels/937367787444260905/949273238477893663

Please seek help for any issues via the channels above before you leave a review on the Unity Asset Store. Reviews should be based on the package, and I'd like an opportunity to resolve any issues you might have before you leave one.

Components Provided

The package currently contains the following components:

Building Door – scripts and components to provide opening and closing door functionality, complete with collider triggers, sound effects and animations.

Building Lights – scripts and components to provide light functionality, including setting global lighting parameters and functionality to turn off and on all lights within buildings or groups of buildings.

You'll find instructions for each in the following sections.

Package Contents

On installing this package from the Unity Asset Store, you should see the following folders in your Unity Project "Assets" folder:

```
3D Forge Components
|_ Animation – contains the animator controller and animations
|_ Documentation – contains this document.
|_ SFX – contains free to distribute sound effects in WAV format.
|_ Prefabs – contains several drag and drop prefabs, for use in your scenes
|_ Scripts – contains the scripts used to provide functionality to the components
|_ Scenes – contains two sample scenes to demonstrate the use of the components
```

In a nutshell

Before we go into the detail of the components, the following is a brief summary of what to expect.

Building Door

- Automatic configuration – just drag and drop the prefabs.
- Configurable “action prompt” displayed when player approaches the door.
- Vanilla and Invector prefabs with standard colliders and Invector’s vGenericAction component.
- Pre-configured animation for opening and closing, with a delay to allow the player to move.
- Open and closing sounds, driven by Animation Events.
- Basic “action” function (default ‘E’ key) to allow player to approach and open doors.

Building Lights

- Automatic configuration – just drag and drop the prefabs.
- Building level or global lighting configuration.
- Option to control candles and torches only, leaving fires and ovens unaffected.
- Global lighting controller, with configurable “on” and “off” times.
- Integration into Enviro to automatically control global lighting at specified time of day.
- Works with HD Render Pipeline as well as Standard.
- Debug toggle lighting key (default ‘L’ key) to allow testing of lighting toggle.

Exploring the Demo Scenes

There are two demo scenes included in the package. In order to use the sample scenes, you’ll need to download the following free assets from the Unity Asset Store:

- Blacksmith
- Blacksmiths Forge
- PB Thunderhammer Forge
- Free Fly Camera

To use the Invector sample scene, you will need to have any of the Invector controller packages installed in your project.

The “generic” test scene is a bit clunky, as it uses a very basic controller. Use ASWD to move around and E to trigger the door opening action.

Building Door Component

Using the “Building Door” prefab

The Building Door prefab can be added to any door Game Object in a 3D Forge model or scene. For example:

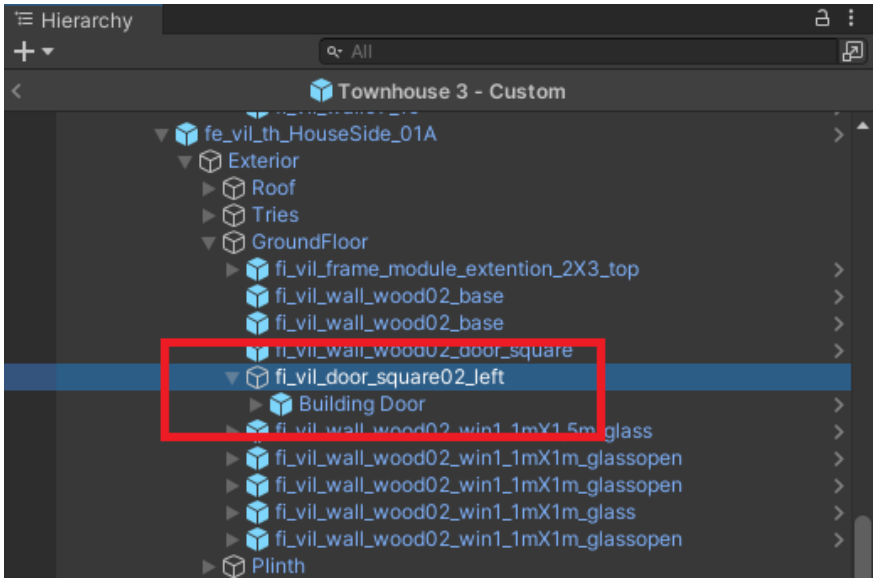


Figure 1 - Door prefab in hierarchy

How it works

The prefab brings together several parts that combine to give the door it’s behaviour. You shouldn’t have to amend any specific configuration, but a brief description is provided if you’d like to do so.

Animator

On the root of the prefab Game Object, you’ll find the reference animator. This will be cloned down onto the actual door mesh, and will control the two animations that open, pause and close the door.

Audio Source

The Audio Source is used to play the door open and closing sounds. You can configure this how you want, using a mixer or other configuration. The sounds are triggered via Animation Events, configured within the Animator controller.

Triggers / Colliders

The references triggers / colliders are automatically reconfigured at run time. These are re-sized and re-positioned relative to the parent door mesh, and per the configuration that we’ll look at in the next section.

Action Prompt

This UI configuration defines the layout and UI elements that make up the “Open” prompt. These elements are rendered in World Space so as to appear on top of the door. The Invector prefab contains additional configuration to display the configured key and controller action button configuration.

Configuring the Building Door

Once the prefab has been added, you can configure the behaviour of the door to your like via the Inspector.

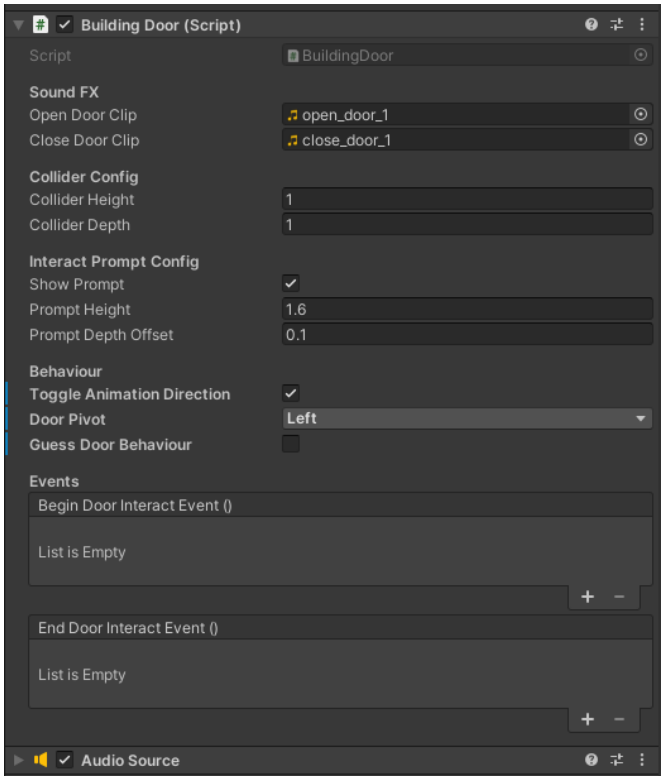


Figure 2 - Door prefab configuration

The parameters available are described below:

Parameter	Description	Suggested value
Open Door Clip	The sound that will be played when the door starts to open. This clip is controlled via an Animation Event on the Animation controller found within the prefab.	open_door
Close Door Clip	The sound that will be played when the door finishes closing. This clip is controlled via an Animation Event on the Animation controller found within the prefab.	close_door
Collider Height	This is the height of the two colliders that will be configured on either side of the door.	1.0f
Collider Depth	This is the depth of the two colliders that will be configured on either side of the door.	1.0f
Show Prompt	Controls whether an “Open” prompt is displayed on the door when the player enters either collider.	True
Prompt Height	This is the height at which the “Open” prompt will appear, when the player enters either collider.	1.6f
Prompt Depth Offset	This is an offset that is added to the door prompt to “pop out” from the door object. Use this if the prompt is hidden when the player enters either collider.	0.1f
Toggle Animation Direction	Toggling this value will mirror the open and close animation. Tick this if the door opens into the player, instead of opening out away from the player	False
Door Pivot	This specifies which side of the door pivots, when viewing	Right

Parameter	Description	Suggested value
	the door from the outside in. This influences the positioning of the colliders. If the colliders are positioned to the left or right of the door, instead of flush, then check this box to correct.	
Guess Door Behaviour	Check this box for the component to "imply" the animation and pivot selections, based on the door name. This will look for "left" or "right" in the name and adjust accordingly.	True

Table 1 – Building door configuration

Events

The component supports several Unity Events that you can use to hook code into the building door component:

Event	When invoked
BeginDoorInteractEvent	Just before the door animation is started, after the player has interacted with the door.
EndDoorInteractEvent	Just after the door animation has finished.

Table 2 - Building door events

Building Lights Component

Using the “Building Lights” prefab

Drag the Building Lights prefab directly onto a building Game Object in your scene. By default, the component will configure and manage all Light components within that Game Objects children and all subsequent sibling instances of Light. You can override this behaviour by specifying a Parent Building Game Object, which will then act as the corresponding parent.

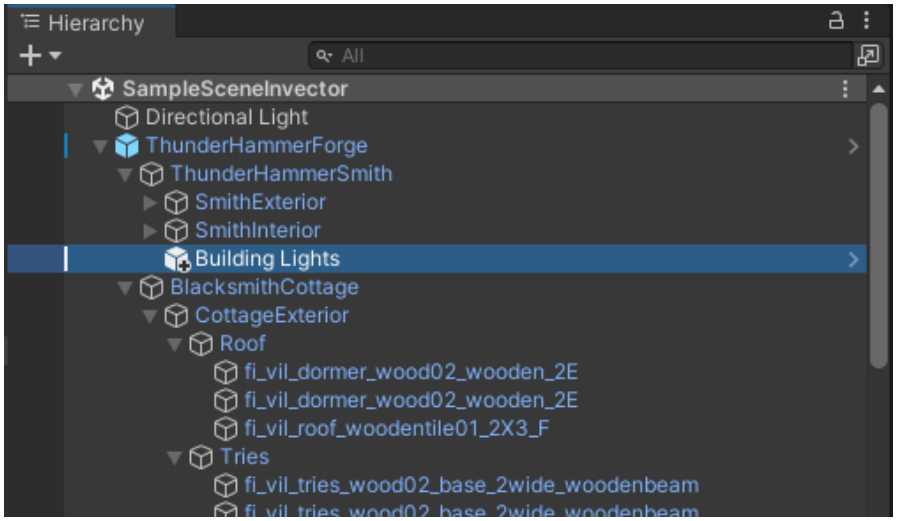
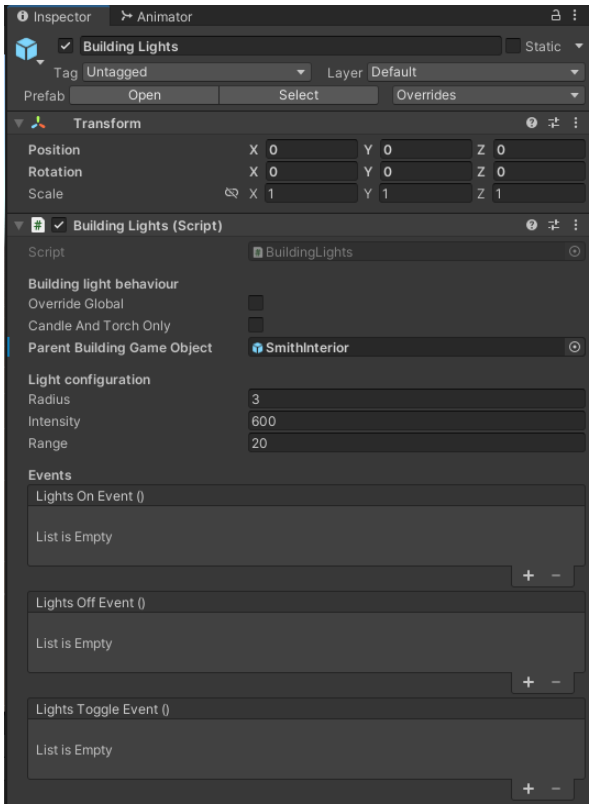


Figure 3 - Building light prefab in the hierarchy

Configuring the “Building Lights”

Once you’ve dropped an instance of the prefab into your scene hierarchy, you can configure it via the inspector:



The parameters available are described below:

Parameter	Description	Suggested value
Override Global	Buildings lights will be configured as per the local settings, and not overridden by the Building Lights Controller settings.	False
Candle and Torch Only	Look for “candle” and “torch” in the names of parent Game Objects and manage only these corresponding lights. Useful to avoid changing fires or stoves.	True
Parent Building Game Object	Signifies the parent Game Object for which all child Light objects will be managed. If left empty, the component will assume the prefab parent Game Object as the parent.	Empty
Radius	The radius to configure all applicable lights.	3
Intensity	The intensity, in lumens, to configure for all applicable lights.	600
Range	The range to configure for all applicable lights.	10
Debug Toggle Lights	Keyboard or keypad KeyCode that can be used to toggle lights on and off for testing	L
Lights on hours	Configurable list of integer “hours” that will result in lights being turned on.	18, 6
Lights off hours	Configurable list of integer “hours” that will result in lights being turned off.	22, 9

Table 3 - Building lights configuration

Building Lights Events

The component supports several Unity Events that you can use to hook code into building lights:

Event	When invoked
LightsOnEvent	Whenever the building lights are toggled or switched on.
LightsOffEvent	Whenever the building lights are toggled or switched off.
LightsToggleEvent	Whenever the building lights are toggled, either on or off.

Table 4 - Building lights events

Using the “Building Lights Controller” prefab

To use the Building Lights Controller prefab, simply drag the prefab anywhere in your scene:

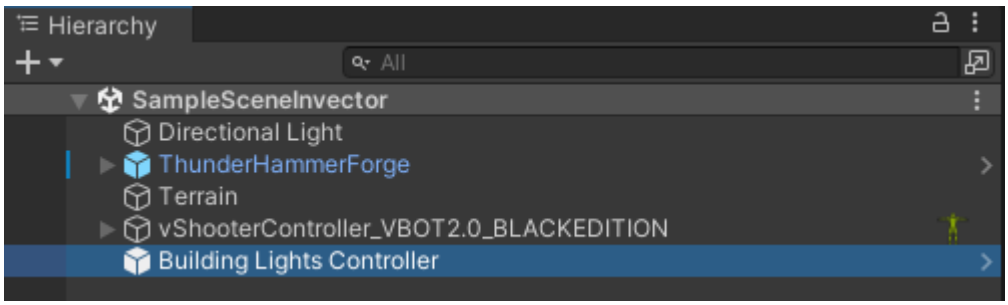
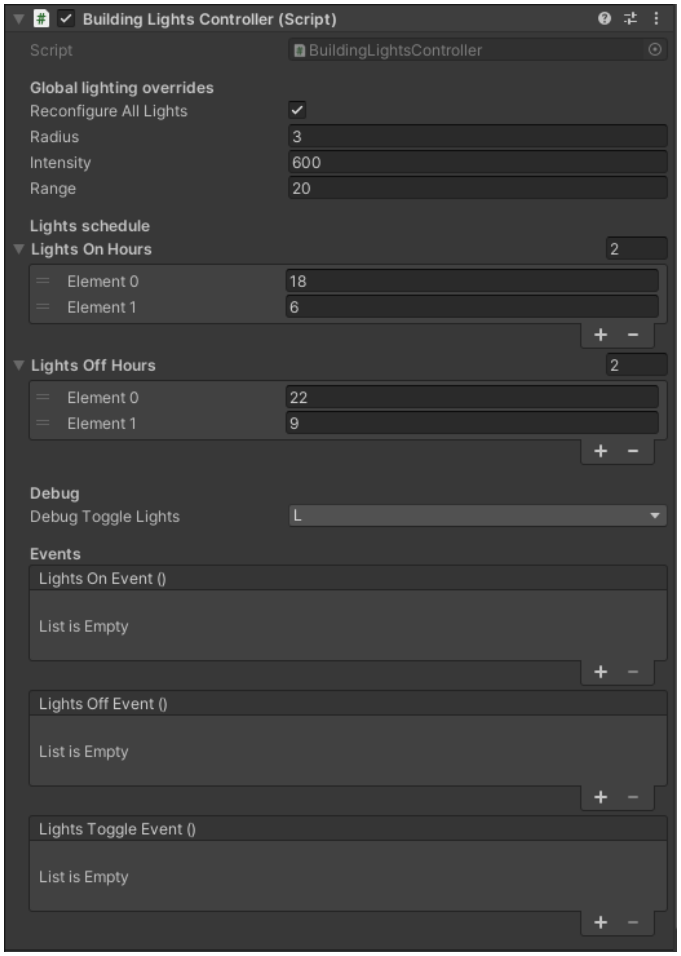


Figure 4 - Building lights prefab in hierarchy

Configuring the “Building Lights Controller”

Once you’ve dropped an instance of the prefab into your scene hierarchy, you can configure it via the inspector:



The parameters available are described below:

Parameter	Description	Suggested value
Reconfigure all lights	Will configure all Building Lights components within the scene with the values shown. If false, all lights will use their local configuration. Any local Building Lights that have “Override Global = True” will not be configured	True

Parameter	Description	Suggested value
	regardless of global settings.	
Radius	The radius to configure all applicable lights.	3
Intensity	The intensity, in lumens, to configure for all applicable lights.	600
Range	The range to configure for all applicable lights.	10

Table 5 - Building lights controller configuration

Building Lights Controller Events

The component supports several Unity Events that you can use to hook code into the building lights controller:

Event	When invoked
LightsOnEvent	Whenever all lights are toggled or switched on.
LightsOffEvent	Whenever all lights are toggled or switched off.
LightsToggleEvent	Whenever all building lights are toggled, either on or off.

Table 6 - Building lights controller events