# Components

For HORROR FPS KIT

**v0.5.2 (Unity 2021.3.2 & Up)**

Copyright © 2024 Dizzy Media

All Rights Reserved.

# Table of Contents

## Contents

Welcome ........ 4

Asset Note (Quality Settings) ........ 4

About ........ 5

Usage: Basic ........ 6

Usage: Custom/Integrations ........ 7

Usage: Updating ........ 8

General ........ 8

Intermediate ........ 10

Advanced ........ 12

Usage: Version ........ 14

Scenes ........ 14

Scripts: Custom Edits ........ 15

Scripts: Components ........ 16

Components: AI ........ 16

Components: Camera ........ 16

Components: Destroyable ........ 16

Components: Enemy ........ 17

Components: Hit ........ 17

Components: Input ........ 17

Components: Player ........ 18

Components: UI | Display ........ 18

Components: UI | Menu ........ 18

Components: Weapons ........ 19

Components: World | Items ........ 19

Components: World | Player ........ 19

Components: World | Scene ..... 20

Scripts: Extensions ..... 21

Extension: Enemy Creator ..... 22

Extension: Floating Icon Helper ..... 23

Extension: Player Creator ..... 24

Extension: Ragdoll Creator ..... 25

Extension: Scenes Updater ..... 26

Extension: Script Editor ..... 27

Extension: Version Detect ..... 28

Extension: Weapon Creator ..... 29

Scripts: Utilities ..... 30

Scripts: Systems ..... 31

Systems: Action Bar ..... 31

Systems: Ignitable ..... 31

Systems: Save System ..... 32

Systems: Screen Events ..... 32

Systems: Widescreen ..... 32

Systems: Sub Actions ..... 33

Add Ons ..... 34

Add Ons: Models and Music ..... 35

Add Ons: Particles ..... 36

Thank You ..... 37

# Welcome

Hi there, thank you for purchasing **Components!**

After working with the Horror FPS Kit for a while now I noticed there were some pretty important features missing, after adding a few to my own personal project i decided to start taking a few requests which lead to some really interesting ideas for HFPS features.

Since then the asset has grown into something really useful, i hope it can help you in your game dev. journey!

### Asset Note (Quality Settings)

If you are experiencing low performance in the demo scene, download the file below and put the QualitySettings file in your ProjectSettings folder.

**Quality Settings Link:** https://bit.ly/3pEoWiZ

***WARNING***

Replacing ProjectSettings files will overwrite current settings, make a backup or test in a clean project.

## Contact info

For bugs, feature requests and a friendly chat, you can send an email to: dizzymediainc@gmail.com

I will always answer as soon as possible but please keep in mind that I am just one guy with a normal day job.

# About

Are You looking to take your HFPS project to the next level?
Components has got you covered!

Designed from the ground up to be a complete extension for the HFPS system, **Components for Horror FPS Kit** provides tons of useful systems that will take your project from zero to hero with ease!

In order to utilize Components you **MUST OWN** a copy of **Horror FPS KIT**!

**What you get**:

- Action Bar
- Character Action
- Complex Notifications
- Components Save System
- Ignitable
- Screen Events
- Simple Fade
- Simple Widescreen
- Sub Actions
- 30 Main Scripts
- 8 Helper Extensions
  (i.e Enemy Creator, Floating Icon Helper, Player Creator, Ragdoll Creator, Scenes Updater, Script Editor, Version Detect & Weapon Creator)
- 9 Utility Scripts
- 12 Demo Scenes
- 2 Breakable Models (Locker and Table)
- Demo Animations, Sounds, Graphics, etc.
- Documentation

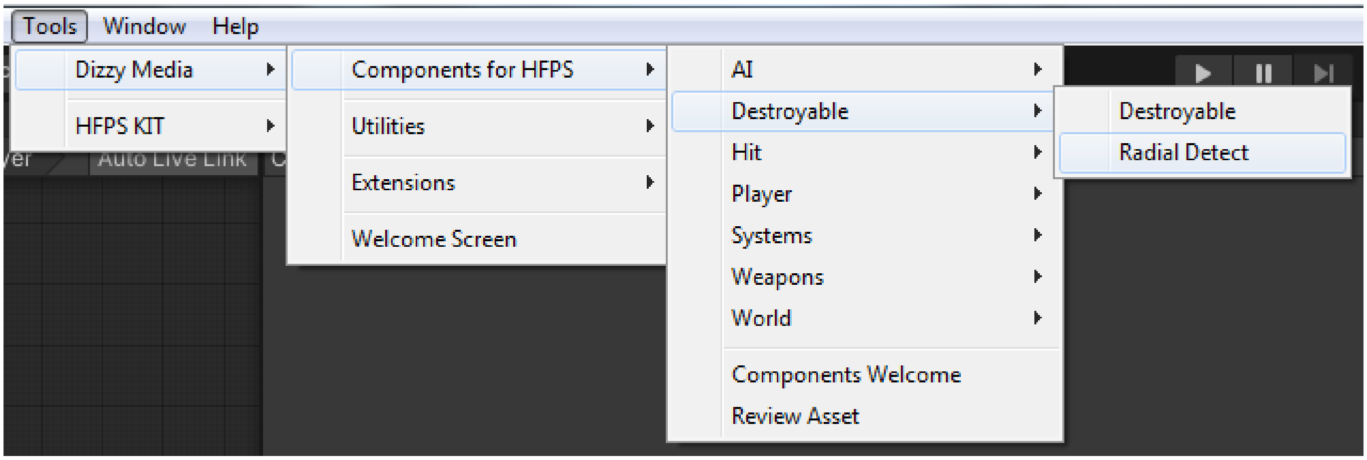
# Usage: Basic

**Components** is a straight forward and easy to use system giving you the ability to create all kinds of interactions in the game world.

The example scenes provide a few different ways the system can be used however the system is not limited by these examples.

Take a look at the example setups to get an idea of how they work, the value names are straight forward and should be easy enough to customize.

You can add components to objects by selecting an object and then selecting your choice from the top menu or use the "Add Component" button on an object.



# Usage: Custom/Integrations

## Dizzy Media:

- Audio Diaries for HFPS
- Diaries Pack for HFPS
- Text Diaries for HFPS
- Durability for HFPS
- Mobile Controls for HFPS
- Puzzler for HFPS
- Shooting Range for HFPS
- Vendor for HFPS
- Weapons for HFPS

*Components replaces Easy Hide, integration is **NOT** supported*

## 3rd Party:

- Dialogue System for Unity
- HFPS Localization System

### HFPS Localization System

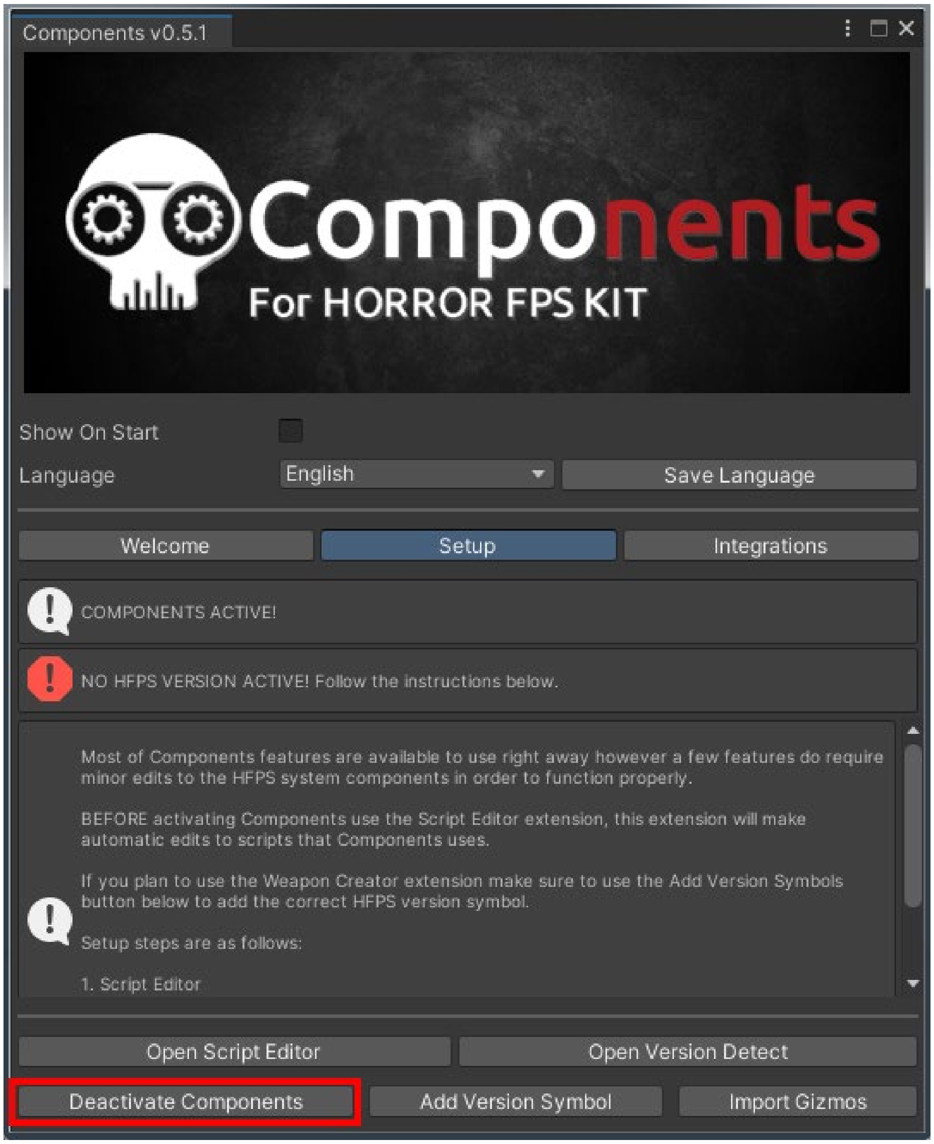
In order to use the HFPS Localization System with Durability you need to do the following:

1. Activate HFPS Localization System (via editor window)
2. Import Dizzy Media Localization Map (via editor window)
3. Import Durability Localization (via editor window)

# Usage: Updating

## General

If you are updating Components versions in your project it is generally recommended to do a clean import of the asset. This means deactivate the asset from the provided welcome window (i.e as indicated in the image below)



Show On Start ☐

Language: English

COMPONENTS ACTIVE!

NO HFPS VERSION ACTIVE! Follow the instructions below.

Most of Components features are available to use right away however a few features do require minor edits to the HFPS system components in order to function properly.

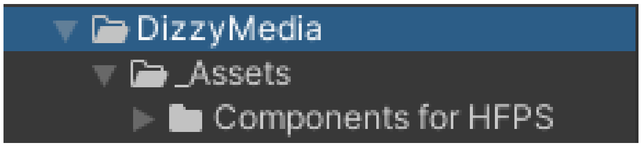
BEFORE activating Components use the Script Editor extension, this extension will make automatic edits to scripts that Components uses.

If you plan to use the Weapon Creator extension make sure to use the Add Version Symbols button below to add the correct HFPS version symbol.

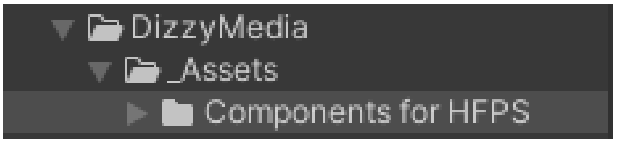
Setup steps are as follows:

1. Script Editor

Delete the parent Dizzy Media folder (i.e as indicated in the image below)



If you have any of my other assets already imported, you can delete the Components folder by itself.



# Intermediate

In certain cases Script Edits made with the Script Editor extension may be updated for new additions or fixes to issues that arise.

These changes are listed in the changelog and indicate which scripts edits are updated and what scripts they are for.

There are 2 cases of Script Edits updates:

1. **New Edits** - New edits have been added for new functions

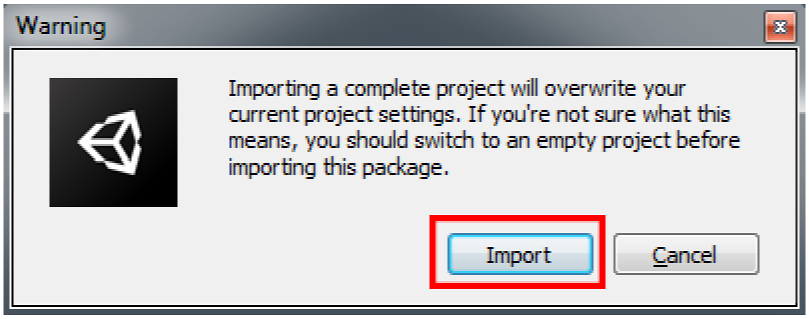
In this case the only action needed from the user (you) is to run the Script Editor extension again to add the newly added edits.

If any duplicate edits are added by the Script Editor extension, simply remove them and that should fix the issue.

2. **Existing Edits Changed** - Existing edits are changed to fix issues or version compatibility

In this case the user may need to re-import scripts that were edited, from the HFPS asset pack in order to restore them back to their original state prior to the edits.

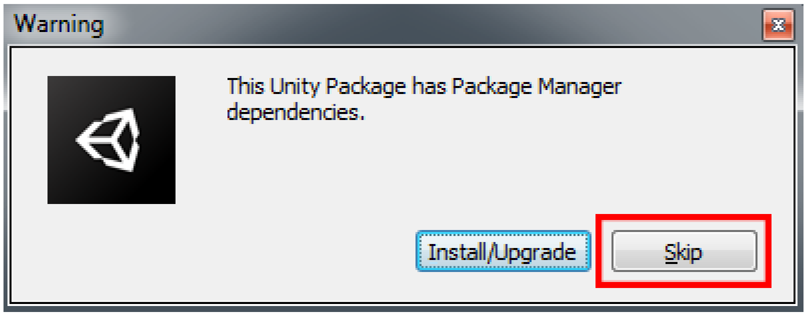
1.



Warning

Importing a complete project will overwrite your current project settings. If you're not sure what this means, you should switch to an empty project before importing this package.

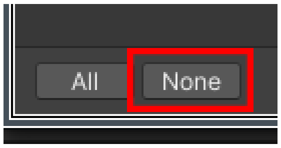
2.



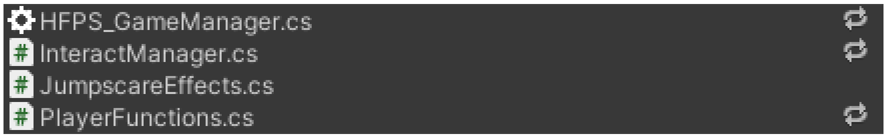
Warning

This Unity Package has Package Manager dependencies.

3.

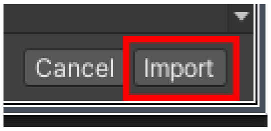


4.



HFPS_GameManager.cs
InteractManager.cs
JumpscareEffects.cs
PlayerFunctions.cs

5.



Once HFPS scripts have been restored to their original state, then you need to run the Script Editor extension in order to re-edit the scripts, which will add in the updated edits.

**DO NOT IMPORT ANYTHING ELSE FROM THE HFPS ASSET PACK EXCEPT THE EDITED SCRIPTS THAT HAVE THE CIRCLE ON THE RIGHT HAND SIDE!!!**

**IF YOU DO IT WILL OVERWRITE ALL CHANGES MADE TO THE HFPS ASSET AND PROJECT SETTINGS!**

# Advanced

In certain cases users may not want to re-import HFPS scripts due to custom edits made by the user to HFPS scripts.

There are 2 cases for this type of update:

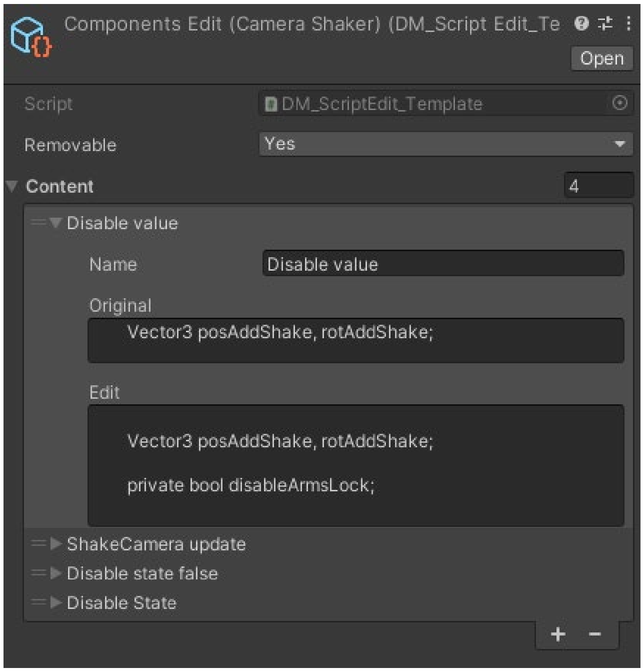
### 1. Copy / Paste Edits

In this case you can manually check Script Edits updates by finding the name of the Script Edit listed in the changelog and then look over the edits scriptable list.

You can then open the script linked to those edits and cross check the listed edits with the current scripts state.

Any changes with script edits can be copy / pasted into the script in question to update the script without using the Script Editor extension.

While this is a bit more cumbersome to do, if custom script edits were made for certain project needs, this is the advised method in order to not lose custom edits made by the user.



Script: DM_ScriptEdit_Template

Removable: Yes

Content: 4

Disable value

Name: Disable value

Original

```
Vector3 posAddShake, rotAddShake;
```

Edit

```
Vector3 posAddShake, rotAddShake;

private bool disableArmsLock;
```

ShakeCamera update

Disable state false

Disable State

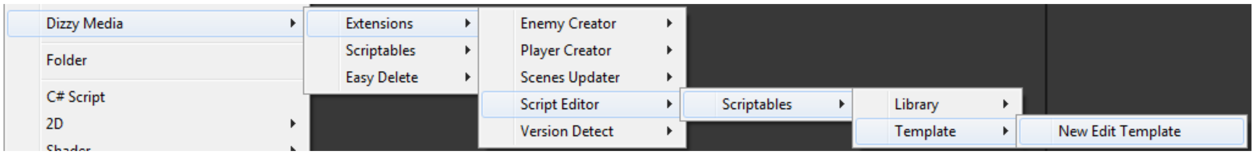
## 2. Create Edit Scriptables

In this case you can manually create new Script Edit templates / library scriptables for any custom edits you've made to HFPS scripts.

Once you've created the new templates / library with the custom edits made, the user can then use that to update HFPS scripts with the Script Editor extension.

In this way, having custom edits saved to these scriptables allows the user to re-import HFPS scripts and restore them to their original state, without losing any custom edits made prior.

The process would then be to make the Components script edits and then make your custom edits, using the library of choice.



# Usage: Version

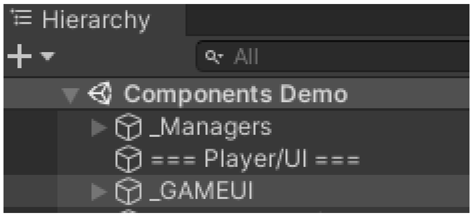
## Scenes

There are a few minor differences between HFPS 1.6.3a and 1.6.3c which introduce some minor breaking changes for player / scene setups.

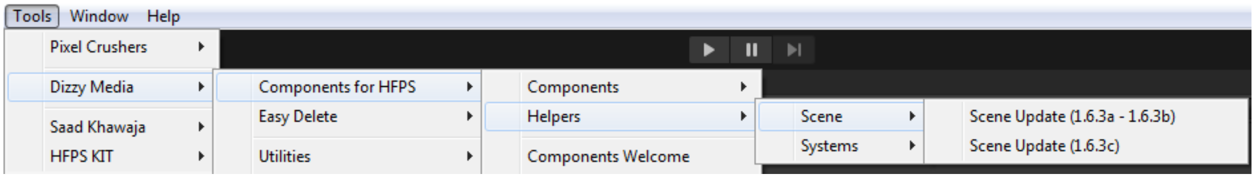
If you are using HFPS 1.6.3c here's how to fix the scene(s) setup issue:

1. First select the **_GAMEUI** object in the scene and then go to the top menu and use one of the scene helpers options to update the scene as needed.

1a.



1b.



Keep in mind this will replace the **Player Object** in the scene, the **Object References Scriptable** on Save Game Handler and the **Player Reference** on the Game Manager.

These helper options will also prompt the **Save Game Handler** to recollect **saveables** in the scene.

# Scripts: Custom Edits

Most of Components features are available to use right away however a few features do require minor edits to the HFPS system components in order to function properly.

Use the **Script Editor** extension to automatic edit HFPS scripts that components uses.

Once you've made the necessary edits use the activate and import buttons in the Components welcome screen to activate components and import the additional files.

# Scripts: Components

These are the main components/scripts you can utilize in your game/project

## Components: AI

- **Possessed** - Adds possession to any object and can attack player

## Components: Camera

- **FOV Manager** - Controls the cameras FOV (field of view) when called to, utilizes custom FOV states.

- **FOV Manager Connect** - Connects actions calls to FOV Manager

## Components: Destroyable

- **Destroyable** - Makes any object destroyable, with breakable stages, broken pieces with broken force, explosion settings and more!

- **Radial Detect** - Radial explosion detects / effects player and destroyables

## Components: Enemy

- **Enemies Add To** - Add enemy references to the Enemies Holder

- **Enemies Holder** - Holds references to all enemies in the game scene

## Components: Hit

- **Health Spot** - Holds health for specific enemy body spots, separate from total health for enemy, boss, etc.

- **Hit Receiver** - Receives damage from HFPS weapons, can send damage via event to any script

## Components: Input

- **Input Action** - Allows you to trigger events based on input(s) indicated in the input list (i.e button / vector, press, hold, direction, etc.)

## Components: Player

- **Audio Fader** - Allows fade between ambience and music audio sources with custom clips, keep ambience option and more!

- **Material Controller** - Controls material swap for body parts based on weapon or item shown i.e arms, etc. (i.e axe, glock, lighter, etc.)

- **Material Controller Connect** - Connects action calls to Material Controller

- **Player Manager** - Manages locking player features (i.e zoom, lean, etc.)

- **References** - Holds all references used by the Components system.

## Components: UI | Display

- **Complex Notifications** - Displays custom notifications which can be called to individually via action call.

- **Complex Notifications Connect** - Connects action calls to Complex Notifications.

- **Simple Fade** - Handles showing / hiding screen fade.

## Components: UI | Menu

- **UI Control** - Controls various aspects of the player UI
  (i.e UI state, save / load state, etc.)

## Components: Weapons

- **Dual Wield** - Allows dual wield of items / weapons with incompatible ID detection.

- **Lighter** - New item similar to candle but with unlimited use (i.e does not require refill, never expires, etc.)

## Components: World | Items

- **Start Items** - Sets items on scene start, can set per item to auto add item, auto add shortcut, display right away, etc.

## Components: World | Player

- **Character Action** - Handles more complex character actions, such as sitting in a chair to rest, side stepping through a crevice or crawling through a small opening.

- **Player Attention** - Allows you to indicate attention points (i.e player turns and looks at door, zombie, etc.)

# Components: World | Scene

- **HFPS Actions** - Contains a list of Character Actions in the scene
  (i.e used with Character Action > Teleport > New Level)

- **Active Handler** - Keeps track of activated objects (i.e broken planks, etc.)

- **Component Trigger** - Trigger area to trigger components functions (i.e destroyable, player attention, player heal, player damage, etc.)

- **Objective Push** - Pushes objective text updates to show for multiple complete checks (i.e 1 out of 5, 2 out of 5, etc.)

- **Scene Action** - Triggers unlimited amount of scene triggered events back to back with delays in between.

- **Sound Library** - Library of sounds which can be called to and played at any time.

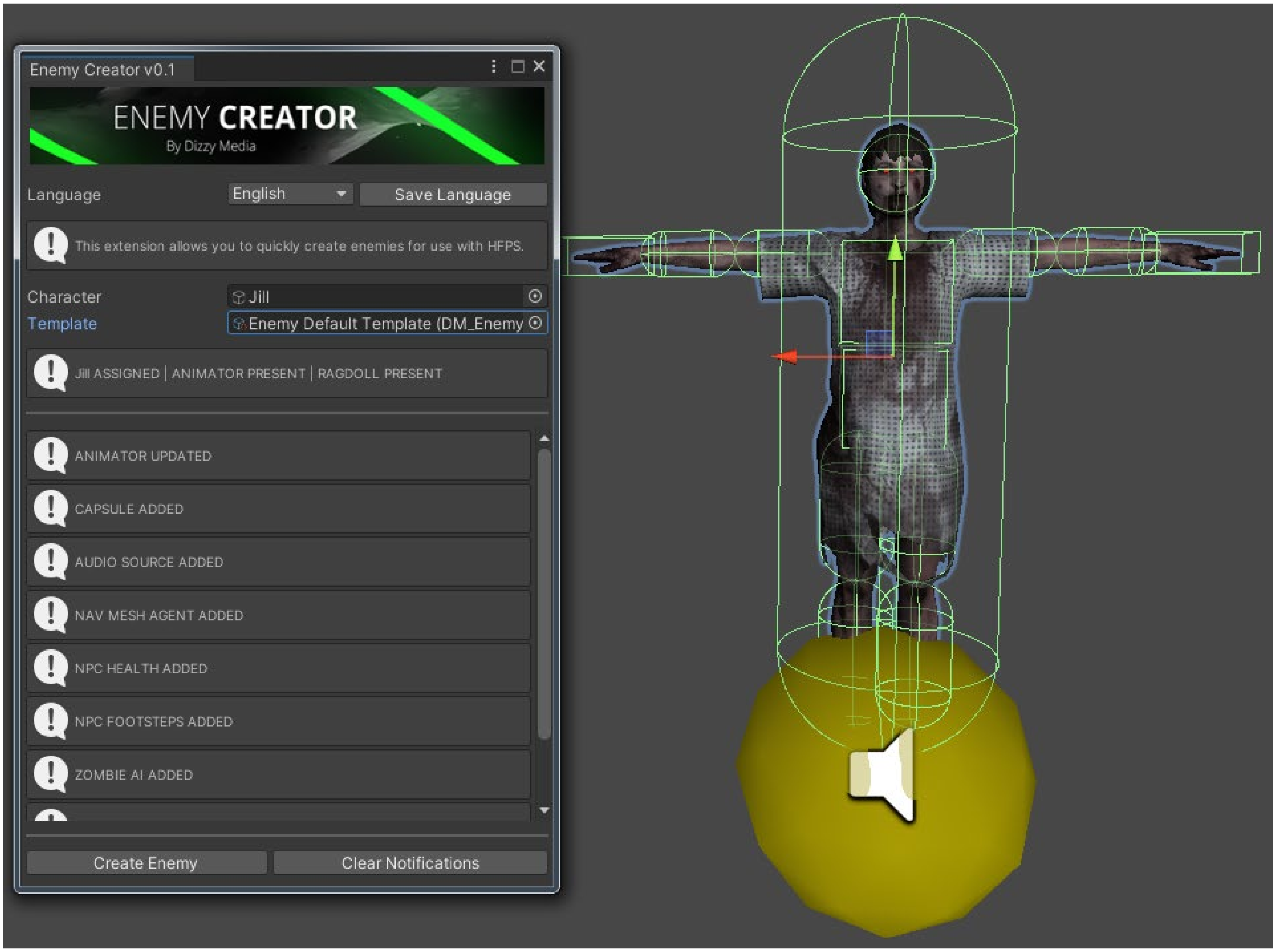
# Scripts: Extensions

Extension scripts are custom made editor extensions that allow extended functionality in various ways, either within the editor or for in game usage.

# Extension: Enemy Creator

This extension allows you to quickly make HFPS enemies by assigning a few values and utilizing templates which can be re-used across multiple enemies.

Assign a character and creator template in the editor window, then use the Create Enemy button at the bottom to start the creation process.



Language: English

This extension allows you to quickly create enemies for use with HFPS.

Character: Jill

Template: Enemy Default Template (DM_Enemy

Jill ASSIGNED | ANIMATOR PRESENT | RAGDOLL PRESENT

ANIMATOR UPDATED

CAPSULE ADDED

AUDIO SOURCE ADDED

NAV MESH AGENT ADDED

NPC HEALTH ADDED

NPC FOOTSTEPS ADDED

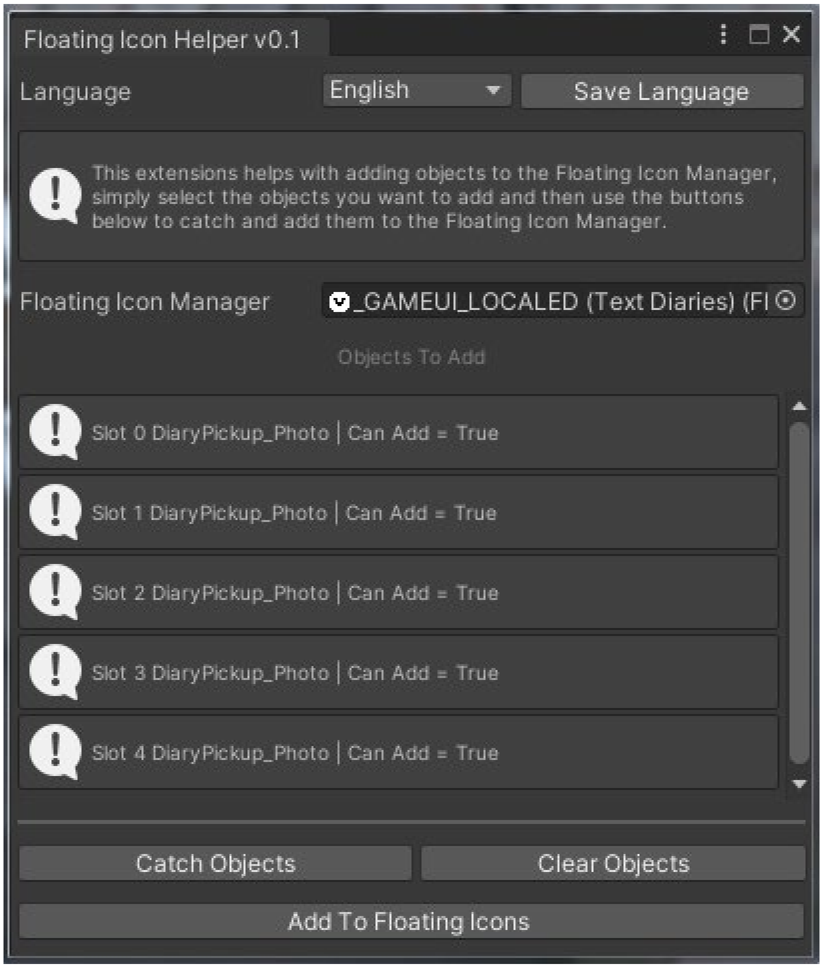
ZOMBIE AI ADDED

# Extension: Floating Icon Helper

This extension allows you to quickly add multiple items to the Floating Icon Manager, which handles displaying all floating icons in the scene.

The extension will auto detect if the selected object(s) can be added to the Floating Icon Manager based on the required features needed to properly display floating icons (i.e mesh renderer, etc.)

Catch the desired objects using the bottom button and then add to Floating Icons using the indicated button (make sure to clear objects before changing scenes)



Language: English

This extensions helps with adding objects to the Floating Icon Manager, simply select the objects you want to add and then use the buttons below to catch and add them to the Floating Icon Manager.

Floating Icon Manager: _GAMEUI_LOCALED (Text Diaries) (Fl

Slot 0 DiaryPickup_Photo | Can Add = True

Slot 1 DiaryPickup_Photo | Can Add = True

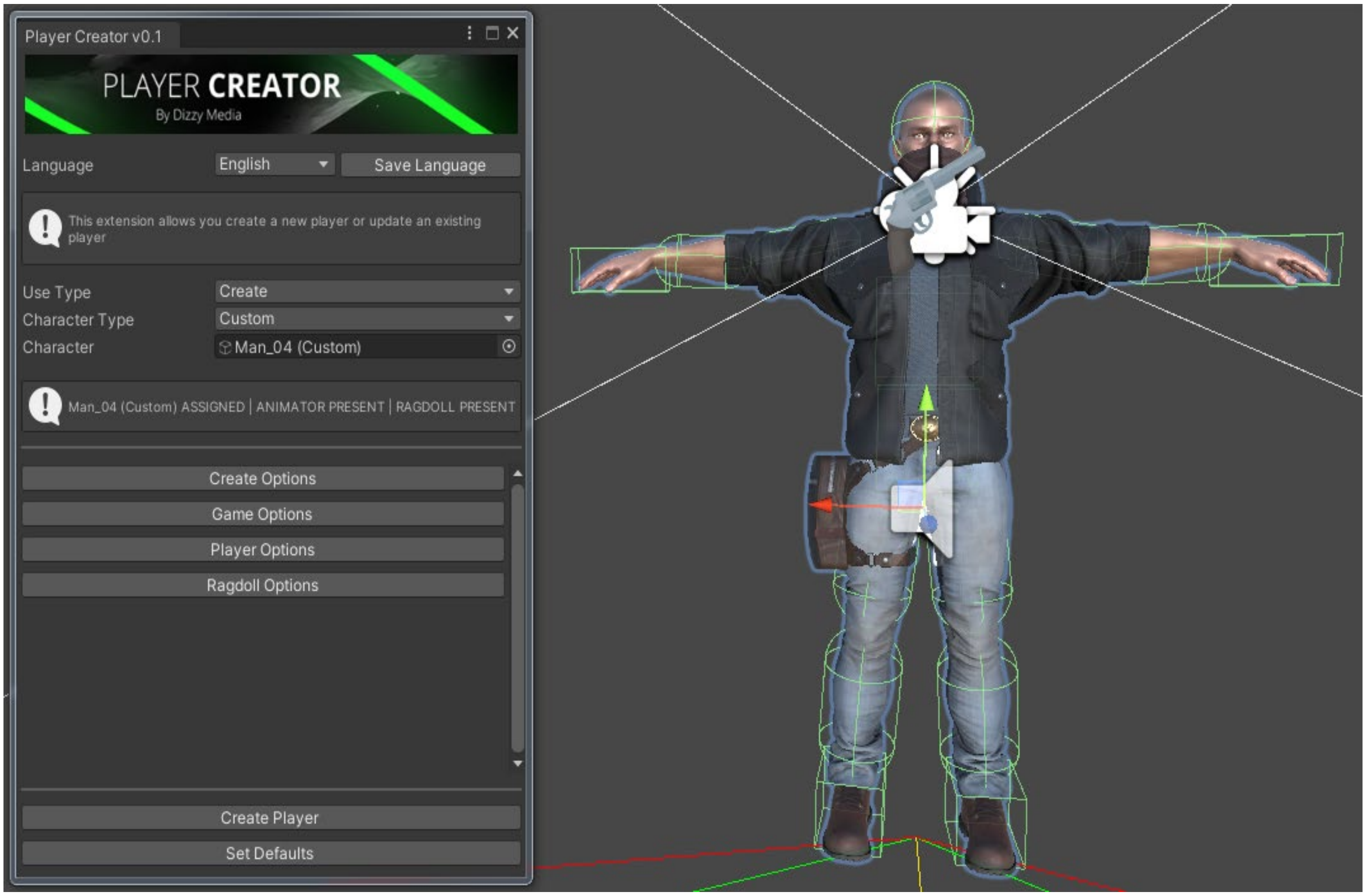
Slot 2 DiaryPickup_Photo | Can Add = True

Slot 3 DiaryPickup_Photo | Can Add = True

Slot 4 DiaryPickup_Photo | Can Add = True

# Extension: Player Creator

This extension allows you to quickly create a custom player or update an existing player for General or Components use.



Language: English

This extension allows you create a new player or update an existing player

Use Type: Create

Character Type: Custom

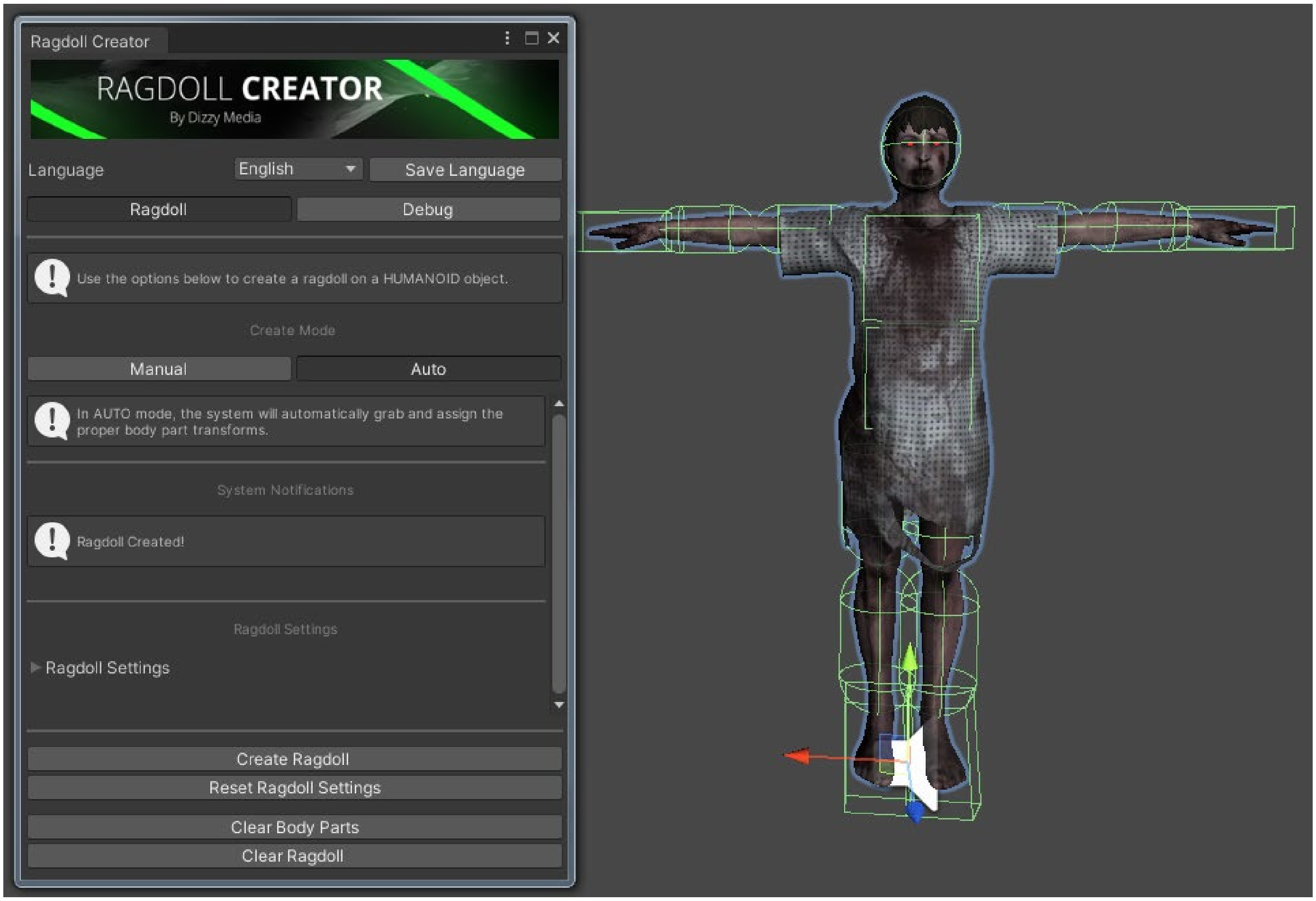
Character: Man_04 (Custom)

Man_04 (Custom) ASSIGNED | ANIMATOR PRESENT | RAGDOLL PRESENT

# Extension: Ragdoll Creator

This extension allows you to quickly add ragdoll functionality to your humanoid characters, either use the manual or auto option to set the body parts.

Once the body parts are assigned, set your ragdoll settings and then use the Create Ragdoll button.



Language: English

Use the options below to create a ragdoll on a HUMANOID object.

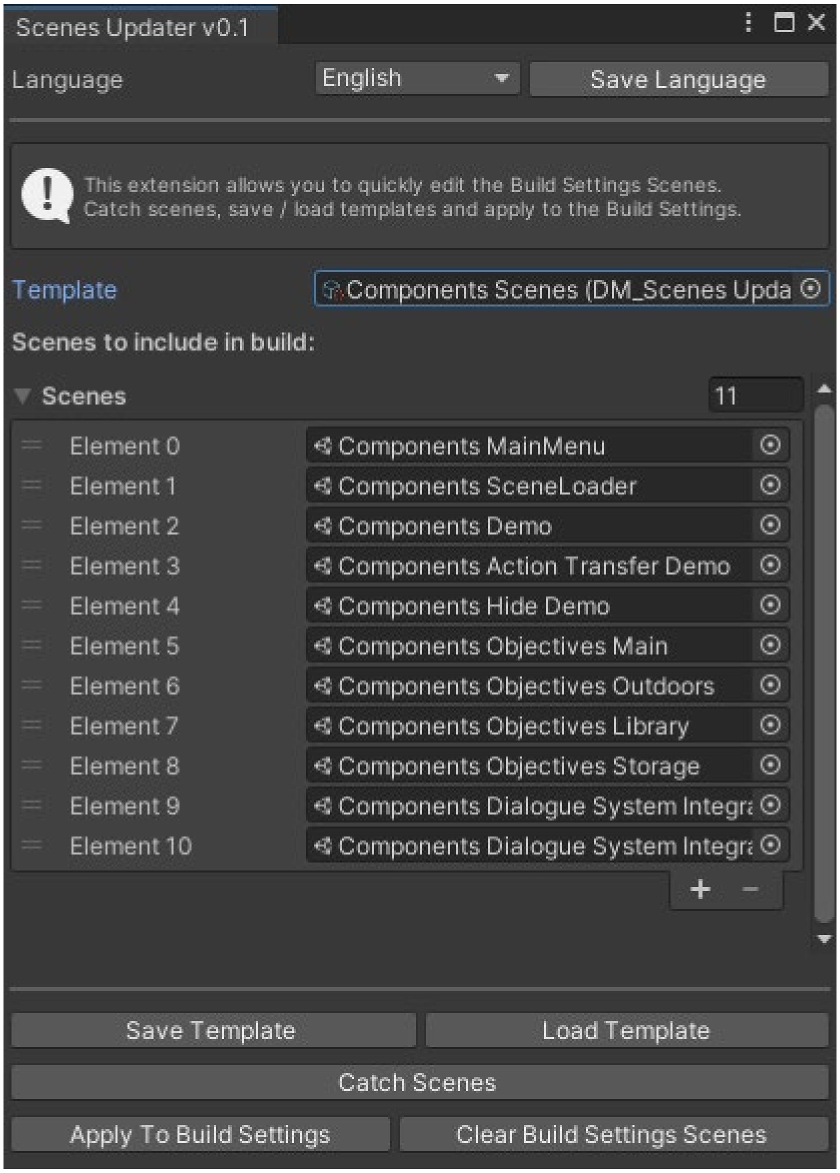
In AUTO mode, the system will automatically grab and assign the proper body part transforms.

Ragdoll Created!

# Extension: Scenes Updater

This extension allows you to make quick edits to the projects Build Settings by using pre-defined templates.

Assign a template in the editor window for the scenes you want to add and use the action buttons to update the Build Settings scenes.



Language: English

This extension allows you to quickly edit the Build Settings Scenes.
Catch scenes, save / load templates and apply to the Build Settings.

Template: Components Scenes (DM_Scenes Upda

Scenes to include in build:

Scenes: 11

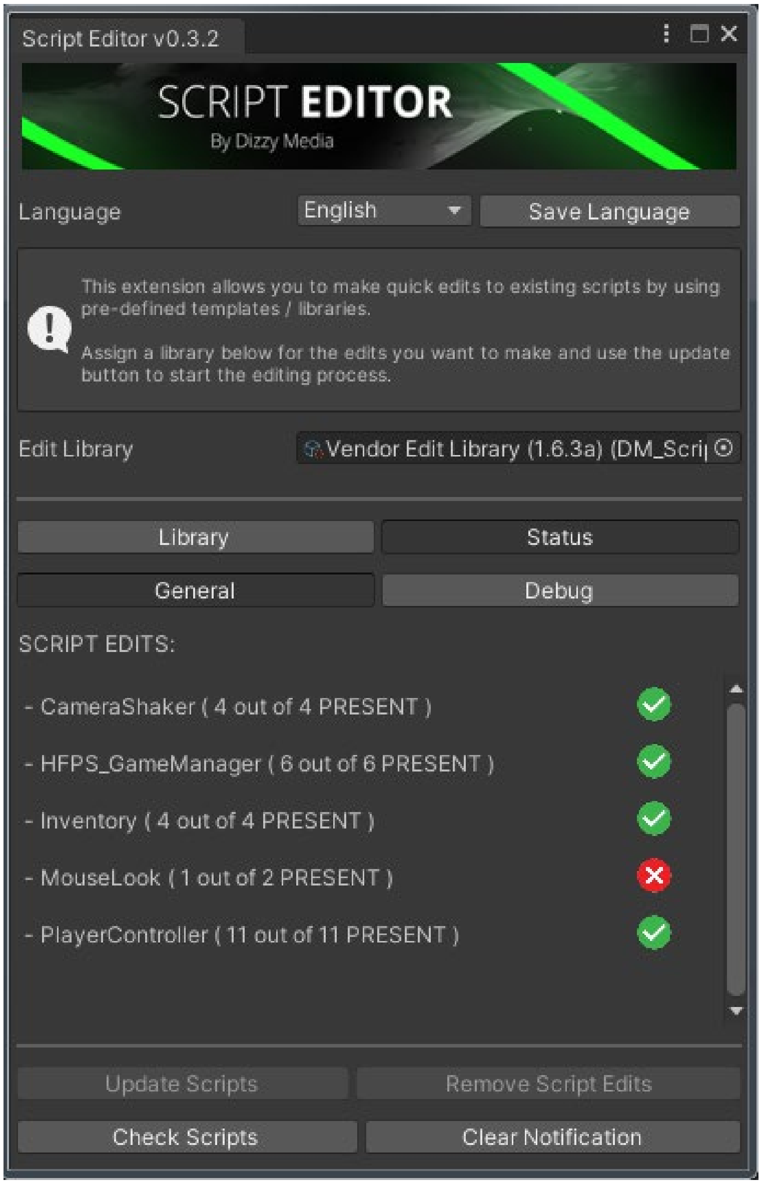
| | |
|---|---|
| Element 0 | Components MainMenu |
| Element 1 | Components SceneLoader |
| Element 2 | Components Demo |
| Element 3 | Components Action Transfer Demo |
| Element 4 | Components Hide Demo |
| Element 5 | Components Objectives Main |
| Element 6 | Components Objectives Outdoors |
| Element 7 | Components Objectives Library |
| Element 8 | Components Objectives Storage |
| Element 9 | Components Dialogue System Integr: |
| Element 10 | Components Dialogue System Integr: |

# Extension: Script Editor

This extension allows you to make quick edits to existing scripts by using pre-defined templates / libraries.

Assign a library in the editor window for the edits you want to make and use the update button to start the editing process.

You can also use the Check Scripts button to verify edits were properly made.



Language: English

This extension allows you to make quick edits to existing scripts by using pre-defined templates / libraries.

Assign a library below for the edits you want to make and use the update button to start the editing process.

Edit Library: Vendor Edit Library (1.6.3a) (DM_Scri

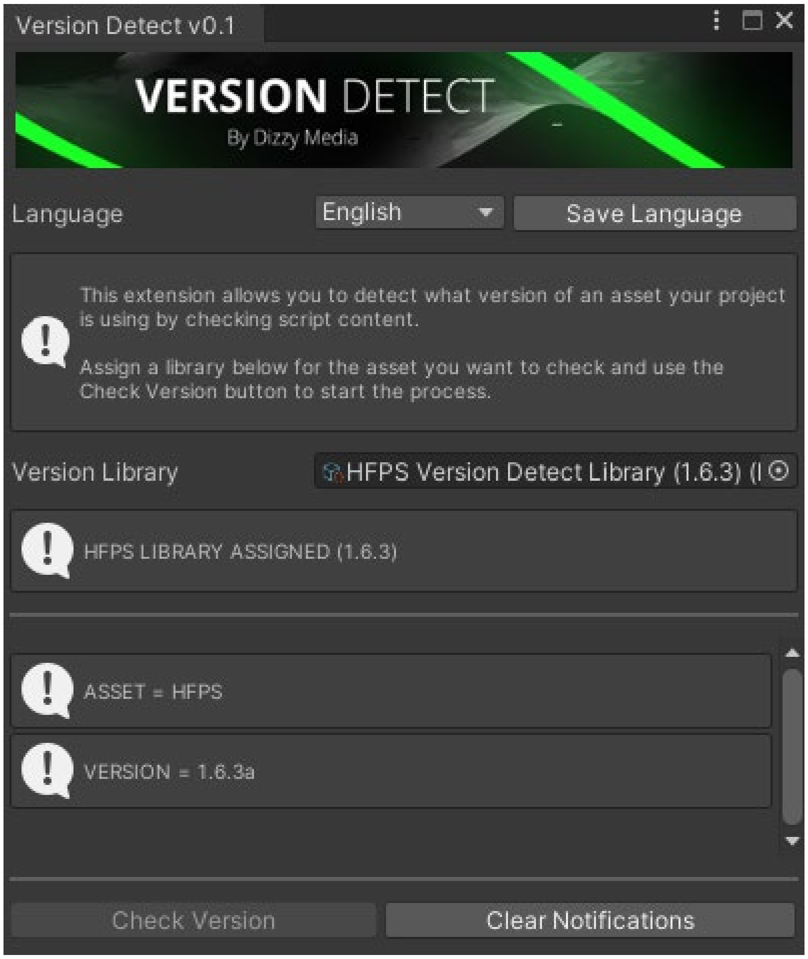
SCRIPT EDITS:

- CameraShaker ( 4 out of 4 PRESENT ) ✓
- HFPS_GameManager ( 6 out of 6 PRESENT ) ✓
- Inventory ( 4 out of 4 PRESENT ) ✓
- MouseLook ( 1 out of 2 PRESENT ) ✗
- PlayerController ( 11 out of 11 PRESENT ) ✓

# Extension: Version Detect

This extension allows you to detect what version of an asset your project is using by checking script content.

In this use case a provided library for HFPS 1.6.3 (A - C) is provided for version detection.



Language: English
Save Language

This extension allows you to detect what version of an asset your project is using by checking script content.

Assign a library below for the asset you want to check and use the Check Version button to start the process.

Version Library: HFPS Version Detect Library (1.6.3) (I

HFPS LIBRARY ASSIGNED (1.6.3)

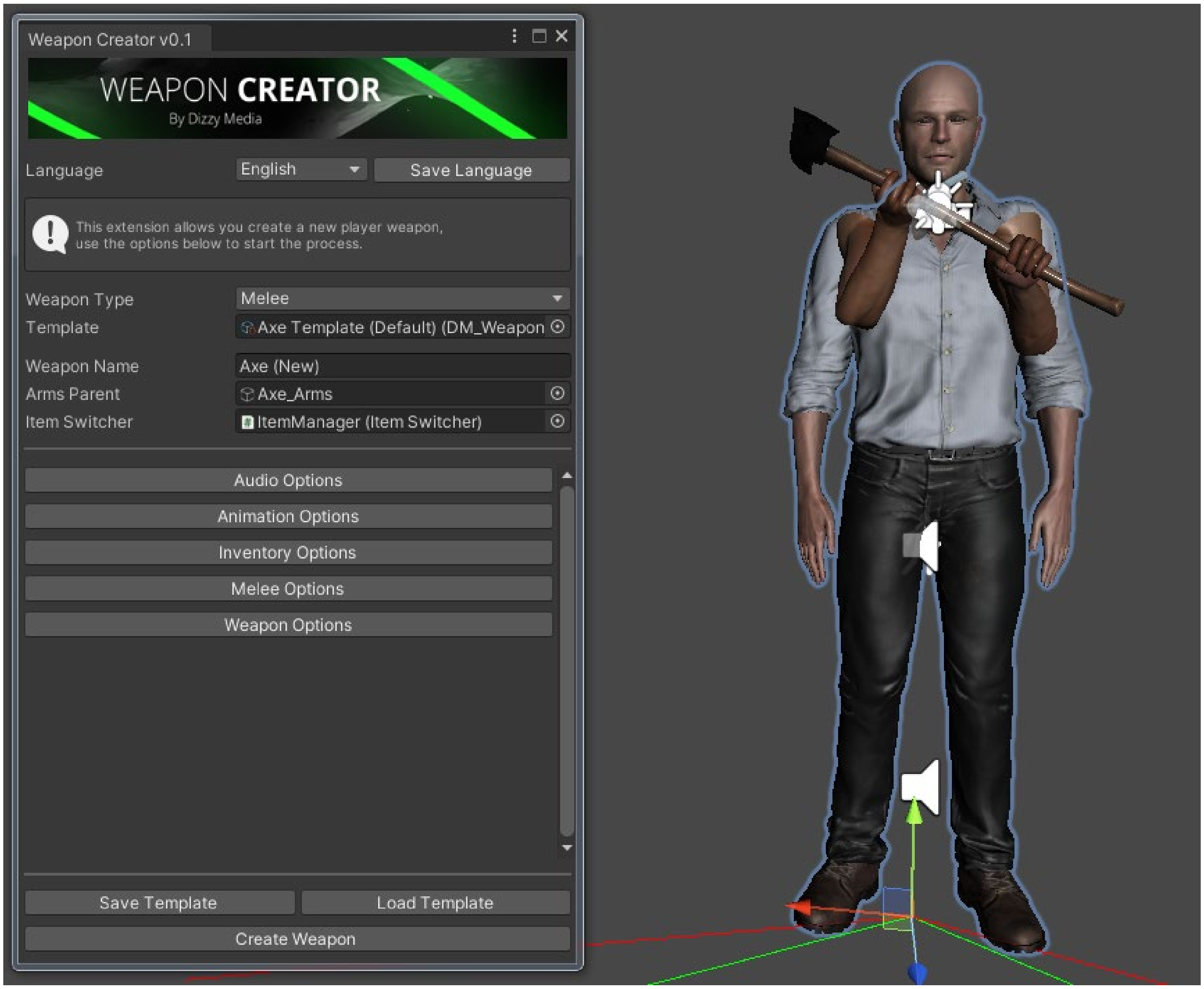
ASSET = HFPS

VERSION = 1.6.3a

# Extension: Weapon Creator

This extension allows you to quickly add player weapons to your existing player.

Assign a weapon template or fill in the necessary values to start the creation process.



Weapon Creator v0.1

Language: English

This extension allows you create a new player weapon, use the options below to start the process.

Weapon Type: Melee
Template: Axe Template (Default) (DM_Weapon
Weapon Name: Axe (New)
Arms Parent: Axe_Arms
Item Switcher: ItemManager (Item Switcher)

# Scripts: Utilities

These are the utility scripts that you can use for extended functionality.

- **DM Dissolve Controller**
- **Mini Audio**
- **Scare Handler**
- **Simple Icon**
- **Simple Pulse**
- **Transform Forward**
- **Trans Indicator**

**DM Dissolve Controller** - Easy to use script that helps control dissolving in/out when using the our custom dissolve shader.

The **Mini Audio** script handles playing 2D sound effects from anywhere via one call (Utilizes the HFPS sound effects audio source)

**Scare Handler -** Handles jumpscare effects similar to HFPS jumpscare trigger, except in this case you can call to the scare handler and trigger jumpscares whenever you want.

The **Simple Icon** script displays an icon on a object
(custom texture & offset options)

The **Simple Pulse** script handles material emission pulse offering  a few options.

The **Transform Forward** script offers a few gizmo options to display the forward direction of a transform.

The **Transform Indicator** script offers a few gizmo options for easier display of transforms.

# Scripts: Systems

These are the main systems you can utilize in your game/project

## Systems: Action Bar

**Action Bar** is handy for displaying a screen space action bar to signify interactable objects (i.e open chest, climb ladder, rest, etc.)

- **ActionBar** - Main AB script that will handle displaying the action bar when called to.
- **Action Bar Connect** - Connects action calls to linked Action Bars (i.e pause check, etc.)

## Systems: Ignitable

**Ignitable** is handy for lighting things on fire per the ignite time, per object (i.e candles, etc.)

- **Ignitable** - Object that can be lit on fire or ignited, holds ignite time.
- **Ignitable Handler** - Handles ignite actions (i.e lighting things on fire)

# Systems: Save System

- **Components Save System** - Saves scene & save data for custom save / load (i.e load previous scenes saved data, etc.)

- **Components Save Loader** - Handles loading saved items and objectives into new scenes (i.e with Components Save)

- **Components Save Connect** - Connects action calls to Components Save

# Systems: Screen Events

- **Screen Events** - Shows screen objects when triggered
(i.e jumpscares, moving bugs across screen, etc.)

# Systems: Widescreen

- **Simple Widescreen** - As the name states, simple system to show widescreen effect with a few options.

# Systems: Sub Actions

**Sub Actions** allows for multiple on screen sub actions which trigger assigned sub action per input (i.e left, right, up, down, etc.)

- **Sub Action** - Holds values for the sub action itself (referenced in Sub Actions Handler)

- **Sub Actions Handler** - Handles display for sub actions.

- **Sub Actions UI** - Handles UI display for sub actions.

# Add Ons

Components currently uses a few free assets from the store to better display demo scenes, you can find links to those assets below.

# Add Ons: Models and Music

The demo scene for Components currently uses a cup / mug model from this asset pack, it's **FREE** (i.e sub actions / coffee)

https://assetstore.unity.com/packages/3d/props/interior/plates-bowls-mugs-pack-146682

The demo weapons for Components currently use a knife model from this asset pack, it's FREE (i.e knife weapon)

https://assetstore.unity.com/packages/3d/props/weapons/hq-kitchen-rusty-knife-83994

The demo scene for Components currently uses royalty free music that must be stored externally, you can download it from the link below:

https://bit.ly/3QUY6iF

# Add Ons: Particles

Some of **Components** features use the basic HFPS particles however most of the demo scene and demo Components particles are created and owned by Unity, which need to be stored externally.

You can get them here for **FREE**:

https://assetstore.unity.com/packages/essentials/asset-packs/unity-particle-pack-5-x-73777

https://assetstore.unity.com/packages/essentials/tutorial-projects/unity-particle-pack-127325

# Thank You

Thanks a lot for purchasing this asset and I really hope that it helps you out with your project, I hope this documentation has helped explain the systems functions and intended use.

If you have any questions feel free to send me an email at:
dizzymediainc@gmail.com

If you like this asset and want to see more like it, please help me out by rating the product on the product page. Ratings let me know people like the products, encourage others to use them as well and increase our business overall, leading to more updates, expansions and new products.

Thanks again!

Tsuyoi Raion
**Dizzy Media**