v0.5.2 (Unity 2020.3.7 & Up)

Table of Contents

Contents

Welcome

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

If you've ever played any horror survival games then you know puzzles are very common, this asset aim's to replicate puzzles from well known horror games such as Lust From Beyond, Resident Evil, etc.

Puzzler is quiet flexible and easy to use, so get to puzzling!

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

Need some puzzles for your game but don't know where to start?
Puzzler has got you covered!

Create puzzles with ease utilizing built in puzzle types, let Puzzler do all the heavy lifting so you can focus on your game!

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

Puzzle Types:

- Solo Item
- Multi Items
- Sequential
- Rotate
- Rotate Advanced
- Lights
- Switches
- Wave
- Weight

What you get:

- Action Bar
- Character Action
- Simple Fade
- 8 Main Scripts
- 11 Utility Scripts
- 3 Helper Extensions (i.e Scenes Updater, Script Editor & Version Detect)
- 3 Demo Scenes
- 5 Custom Puzzle Models
- 9 Example Usages
- Documentation

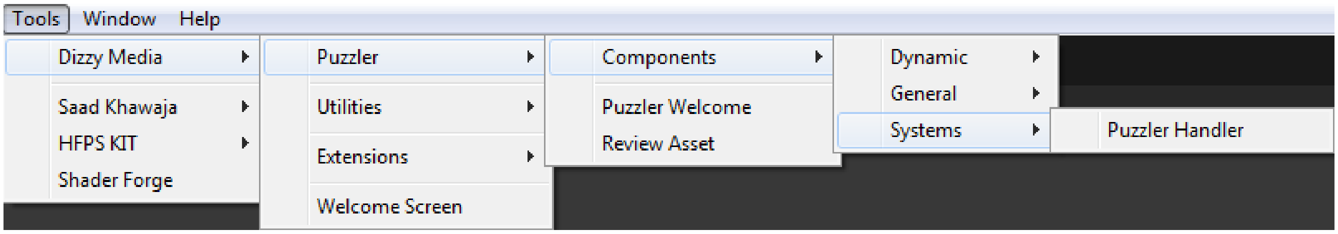
Usage: Basic

Puzzler is a straight forward and easy to use system giving you the ability to create all kinds of puzzles that can be interacted with 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.



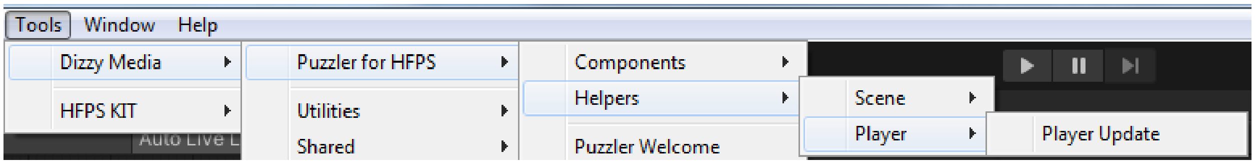
Usage: Custom/Integrations

- Audio Diaries for HFPS
- Diaries Pack for HFPS
- Text Diaries for HFPS
- Components for HFPS
- Durability for HFPS
- Easy Hide for HFPS
- Extensions for HFPS
- Mobile Controls for HFPS
- Vendor for HFPS

Usage: Player

Updating

If at any point you want to update a player setup to support Puzzler right away but don't want to manually do it yourself, you can use the top menus helpers to update a player setup.

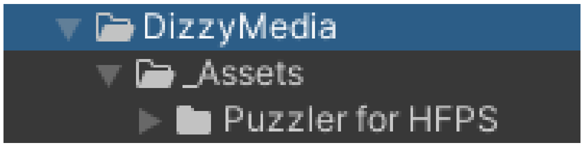


Keep in mind this will update the players root objects layout in order to support the **Puzzler Camera Controller** usage.

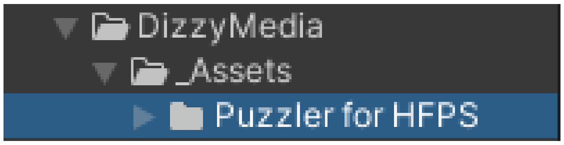
Usage: Updating

General

If you are updating Puzzler versions in your project it is generally recommended to do a clean import of the asset. This means delete the parent Dizzy Media folder.



If you have any of my other assets already imported, you can delete the Puzzler 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

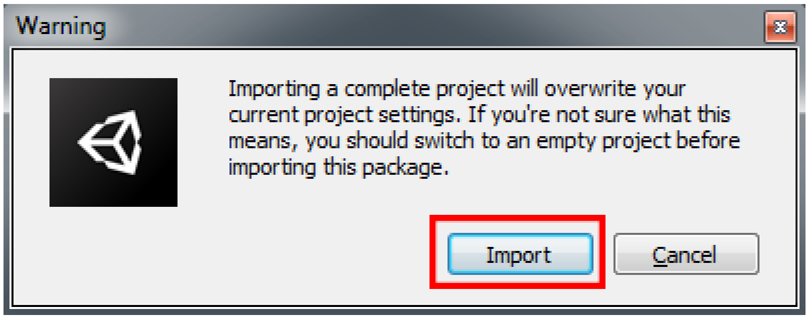
In this case the only action needed from the user (you) is to run the Script Editor extension again to add the newly added edits.

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

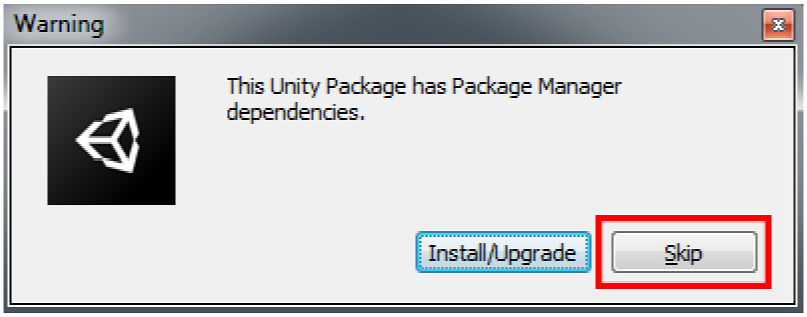
In this case the user may need to re-import scripts that were edited, from the HFPS asset pack in order to restore them back to their original state prior to the edits.

1.



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.

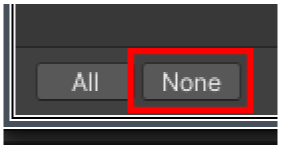
2.



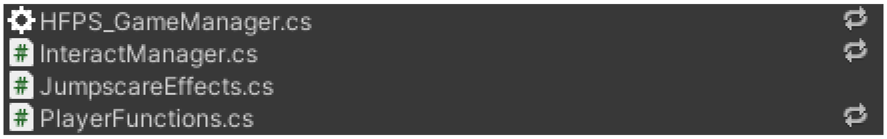
Warning

This Unity Package has Package Manager dependencies.

3.

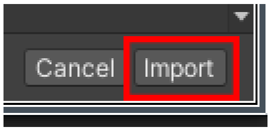


4.



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:

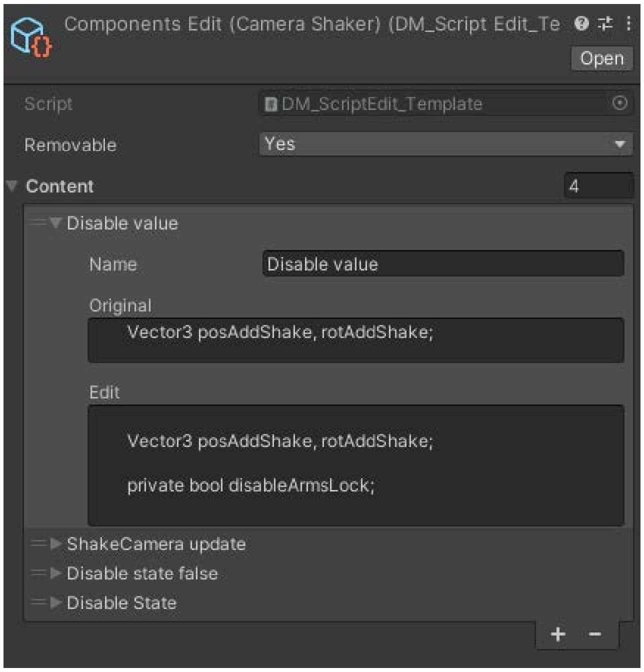
1. Copy / Paste Edits

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

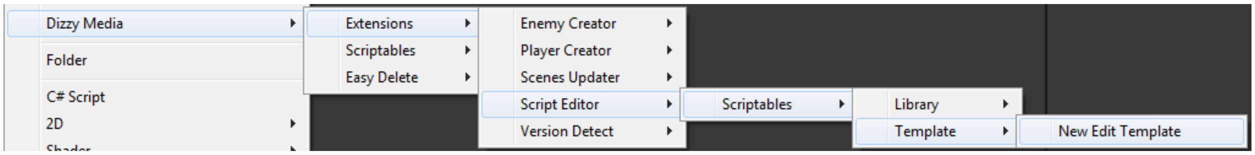
2. Create Edit Scriptables

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 Puzzler script edits and then make your custom edits, using the library of choice.



Usage: Version

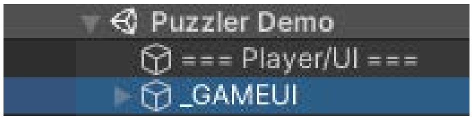
Scenes

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.

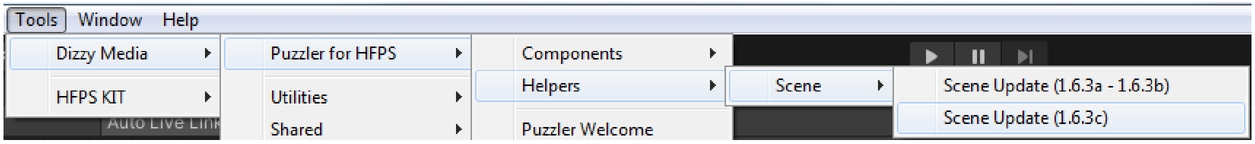
If you are using HFPS 1.6.3c here's how to fix the scene(s) setup issue:

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 Puzzlers 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 puzzler uses.

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

Scripts: Components

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

Components: Camera

- **Puzzler Camera Controller** - Handles camera movement action for puzzles (i.e Rotate Advanced)

Components: Dynamic

- **Puzzler Dial** - Handles dynamic dials which can have slots or use a continuous rotation.

- **Puzzler Wave -** Handles wave pattern creation and updating.

Components: General

- **Puzzler Holder** - Holds references for puzzles (i.e utilized by Puzzler Handler)

Components: Player

- **Puzzler References** - Holds scripts references used by Character Action

Components: System

- **Puzzler Handler** - Handles all puzzle interactions and actions.

- **Puzzler Item Viewer** - Handles displaying non action items for use with puzzles (i.e Rotate Advanced > Answer Sheet, etc.)

- **Puzzler Item Viewer Connect** - Connects actions to Item Viewer

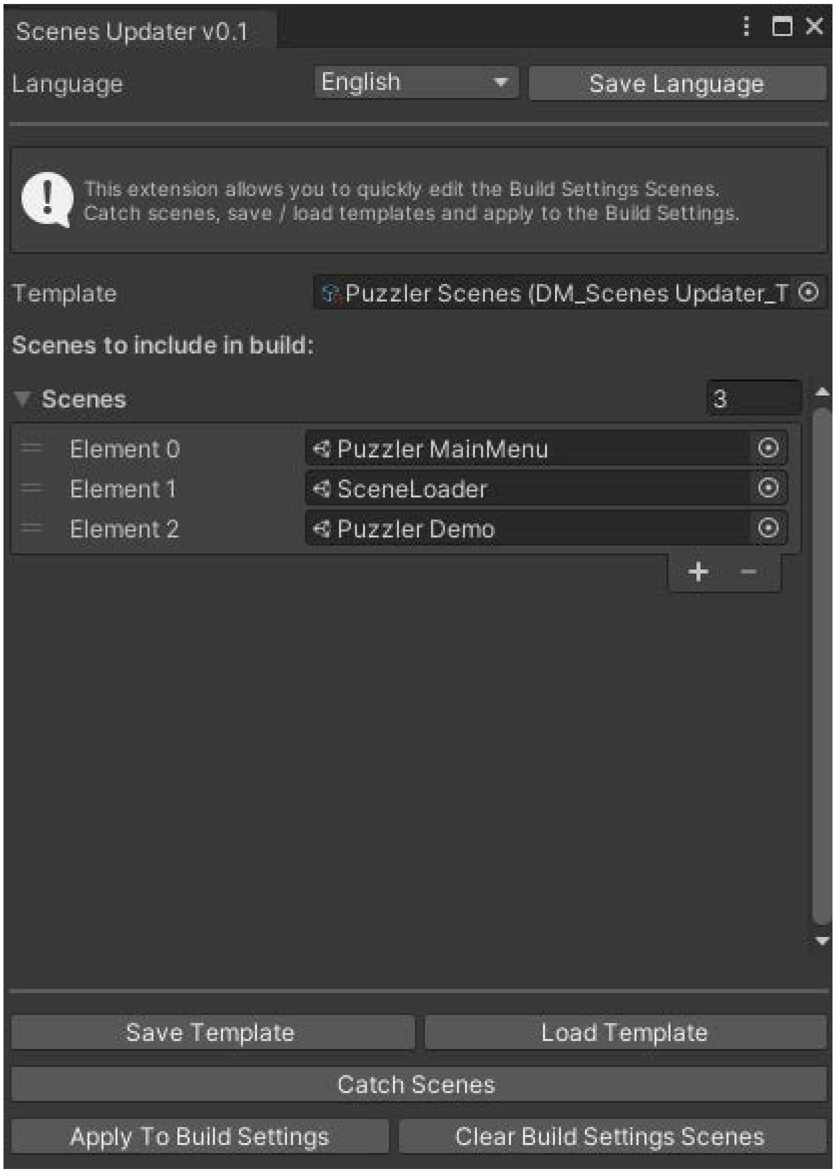
Scripts: Extensions

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

Extension: 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: Puzzler Scenes (DM_Scenes Updater_T

Scenes to include in build:

Scenes: 3

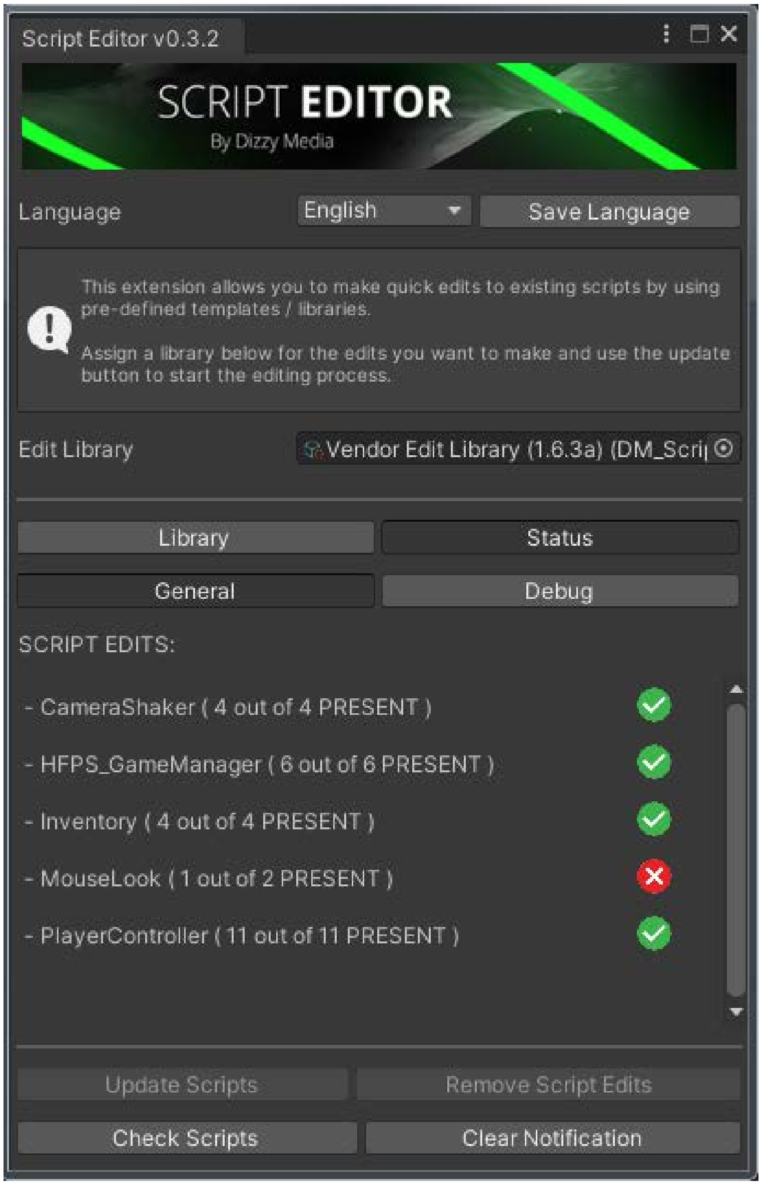
```
Element 0    Puzzler MainMenu
Element 1    SceneLoader
Element 2    Puzzler Demo
```

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

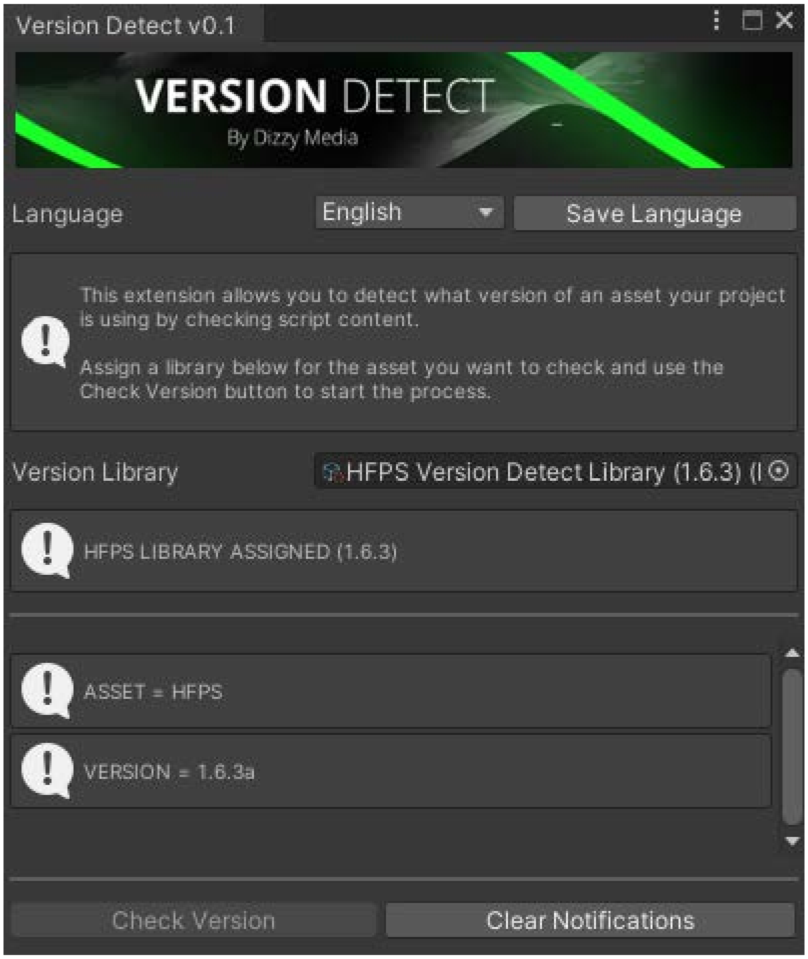
SCRIPT EDITS:

- 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

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

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.

Simple Icon - 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.

Transform Indicator - 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.

Puzzle Types

Puzzler comes with various types of puzzles that can be created easily, the provided puzzle examples are one way those puzzle types can be used, however each can be expanded on or used in various ways.

Below i will cover each puzzle type, it's general usage and specific options available only to each puzzle type.

Puzzle: Solo Item

The **Solo Item** puzzle allows you to use a single item which can placed in multiple slots, such as placing candles, statues, cups, etc.

All puzzles that use a single item type also allow for 2 different detection methods

- **Open Inventory** - Opens the players inventory when the puzzle is interacted with, to select the appropriate item.
- **Auto Detect** - Automatically detects if the item is in the players inventory and selects/uses that item for the puzzle.

Puzzle: Multi Items

The **Multi Items** puzzle allows you to use multiple items which can placed in multiple slots, each slot has it's own item identifier which is used for detecting the appropriate item placed in the appropriate slot.

Each slot has it's own **Slot Check** options allowing for 2 different detection methods

- **Empty** - This allows the slot set to be empty and not require an item for complete detection
- **Item** - This allows the slot set to detect an item and ignore empty checks

Each slot has it's own **Slot Use** options allowing for 2 different detection methods

- **All Prefab Items** - This allows all items listed in Multi Prefabs to be used on this slot.
- **Only This Item** - This limits item usage to the specific item indicated on this slot (for use with items such as specific body parts or specific items)

Puzzle: Sequential

The **Sequential** puzzle allows you create a puzzle that requires the user to interact with the puzzle in a specific order or sequence. This is commonly used with interactions such as buttons, panels, etc. but can be expanded in various ways.

Sequential comes with a few different options such as Reset Type

- **On Fail** - Resets the puzzle / sequence on incorrect input or puzzle fail
- **On Finish** - Resets the puzzle / sequence after all interactions are interacted with (i.e 4 out of 4 buttons pressed) if incorrect or puzzle fail

Puzzle: Rotate

The **Rotate** puzzle allows you create puzzles that rotate on a specific axis in order to form a complete picture, symbol, etc. when completed.

Rotate puzzles allow you to set the start rotation and the activate rotation for each rotate slice / slot, as well as keeps track of the current rotation for debug purposes.

Puzzle: Rotate Advanced

The **Rotate Advanced** puzzle allows you create puzzles that have multiple rotate able modules with their own slices / slots, which can be moved around or across each rotation module.

Each slice / slot has it's own values for the correct module and slot that specific slice / slot should be placed in, which allows the puzzle to be completed when all slices / slots are in the correct module and slot.

Each Rotate Slot has it's own **Slot Type** options allowing for 2 different detection methods

- **Empty** - This allows the slot set to be empty and not check correct module / position
- **Filled** - This allows the slot set to check the correct module / position

Puzzle: Lights

The **Lights** puzzle allows you create puzzles that require specific lights to be active or not active, this puzzle uses the Interactable Light component from HFPS to check whether they are active or not.

This puzzle does not require lights to be activated in a certain order like the Sequence puzzle, however it does require specific lights to be on and specific lights to be off, as indicated in the Lights Active list.

Puzzle: Switches

The **Switches** puzzle allows you create puzzles that require specific switches to be active or not active, this puzzle uses the Dynamic Object component from HFPS to check whether they are active or not.

Similar to the Lights puzzle, switches do not need to be activated in a specific order, however certain switches need to off and certain switches need to be on, as indicated in the Switches Active list.

Puzzle: Wave

The **Wave** puzzle allows you create puzzles that display a wave shape, which can be altered or modulated, to change the shape of the wave to whatever you want.

You can have multiple modules, each holding their own wave, which once completed activates and waits for all waves to be active before completing the puzzle.

Each wave has it's own set of wave checks such as amplitude, wave length, etc. which is used for checking if the wave is correct.

Puzzle: Weight

The **Weight** puzzle allows you create puzzles that require a certain amount of weight to be placed in specific modules, this can be anything from actual weighted items to liquid poured in a glass.

This puzzle allows single or multi item use for weight addition as well as multiple modules for weight placement. Each module has the correct and max weight of each module and can have multiple slots where weight items can be placed.

Add Ons

Puzzler 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 Puzzler currently uses a coin and jukebox model from the asset store, they are **FREE**:

Coin:

https://assetstore.unity.com/packages/3d/props/gold-coin-indian-goddess-low-poly-231931

Jukebox:

https://assetstore.unity.com/packages/3d/props/electronics/classic-jukebox-3-69085

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

https://bit.ly/3OWkoSY

Credits

A few of puzzlers custom models come with paintings that were actually hand painted by a friend of mine, you can find his info below if you are interested in seeing more of his work.

Artist:
Benjamin Azaxor

Links:

https://azaxorart.bigcartel.com/

https://www.instagram.com/azaxor.art/

Models art is included on:

- Painting - Crow
- Painting - Skull

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