



v0.2.3 (Unity 2020.3.7 & Up)

Copyright © 2022 Dizzy Media

All Rights Reserved.

Table of Contents

Contents

Welcome	4
About	5
Usage: Basic	6
Usage: Custom/Integrations	7
Scripts: Custom Edits	8
Scripts: Components	9
Components: Dynamic.....	9
Components: General	9
Components: System	9
Scripts: Extensions	10
Extension: Script Editor	11
Scripts: Utilities	12
Puzzle Types	13
Puzzle: Solo Item	14
Puzzle: Multi Items.....	15
Puzzle: Sequential	16
Puzzle: Rotate	17
Puzzle: Rotate Advanced.....	18
Puzzle: Lights	19
Puzzle: Switches	20
Puzzle: Wave	21
Puzzle: Weight.....	22
Credits	23
Thank You.....	24

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:

- 4 Main Scripts
- 3 Utility Scripts
- 3 Demo Scenes
- 3 Custom Puzzle Models
- 9 Example Usages
- Documentation

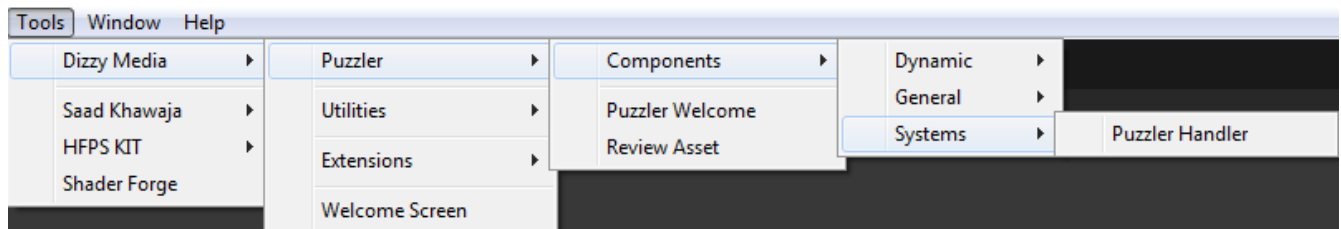
Usage: Basic

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

There are currently no custom methods or integrations for Puzzler however the systems are flexible enough where you can make your own edits/additions if you choose too.

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.

If you have any issues with the automatic edit process you can find example end result images for each edit here:

<https://bit.ly/3rd7VgS>

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: 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: System

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

Scripts: Extensions

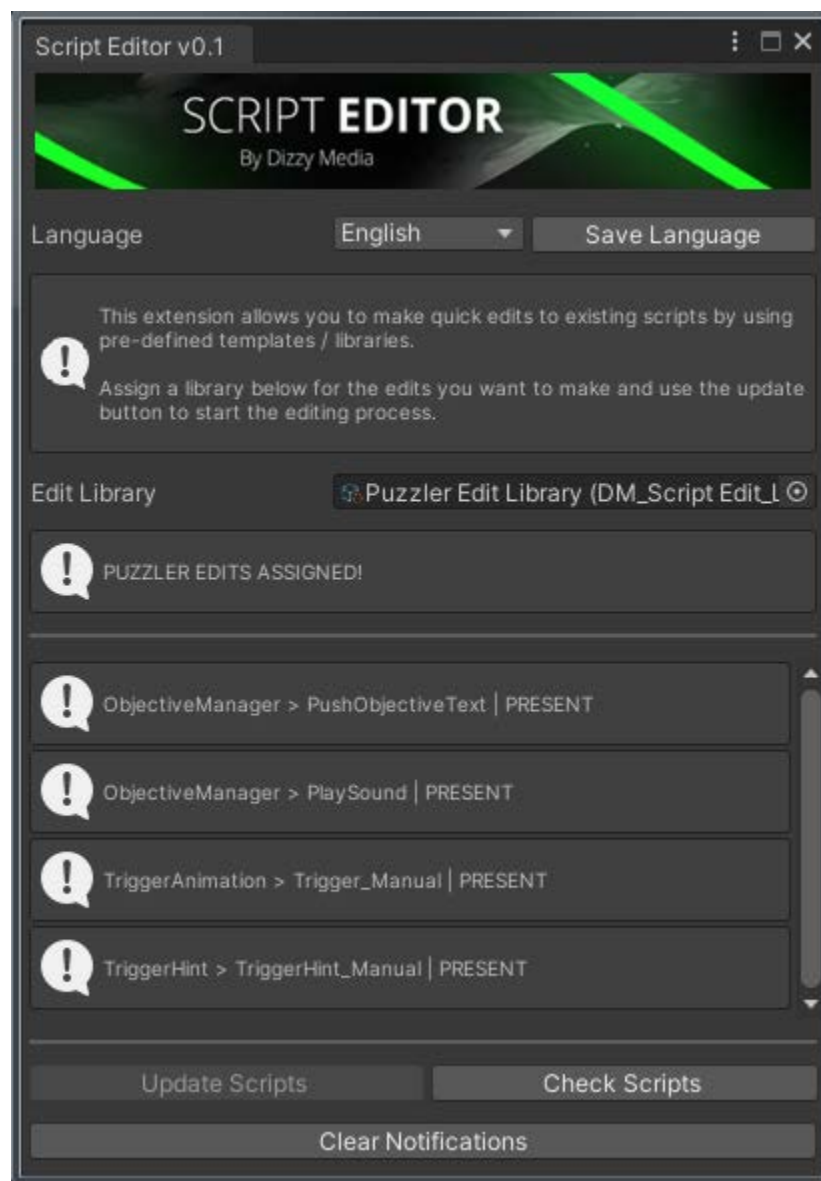
Extension scripts are custom made editor extensions that allow extended functionality in various ways, either within the editor or for in game usage.

Extension: 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.



Scripts: Utilities

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

- **DM Dissolve Controller**
- **Scare Handler**
- **Trans Indicator**
- **Simple Icon**

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

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.

Transform Indicator - Offers a few gizmo options for easier display of transforms.

Simple Icon - displays an icon on a object (custom texture & offset options)

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 be 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 player's inventory when the puzzle is interacted with, to select the appropriate item.
- **Auto Detect** - Automatically detects if the item is in the player's inventory and selects/uses that item for the puzzle.

Puzzle: Multi Items

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

Each slot has its own 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.

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.

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