SIMPLE FLASHLIGHT SYSTEM V1.7
DOCUMENTATION

Contents

MATTHEW RUKAS – SPEEDTUTOR
WWW.SPEED-TUTOR.COM
SPEEDTUTORUK@GMAIL.COM

Introduction

Thank you for purchasing the "Simple flashlight System", it is a complete template for implementing your very own flashlight system. The system allows you to collect an included PBR flashlight, batteries and allows you to add additional intractable objects, with a Raycast system. This package comes with a basic UI system showing battery level, count and whether your flashlight is on and off. It is the basis to start any style of adventure/horror game with this functionality.
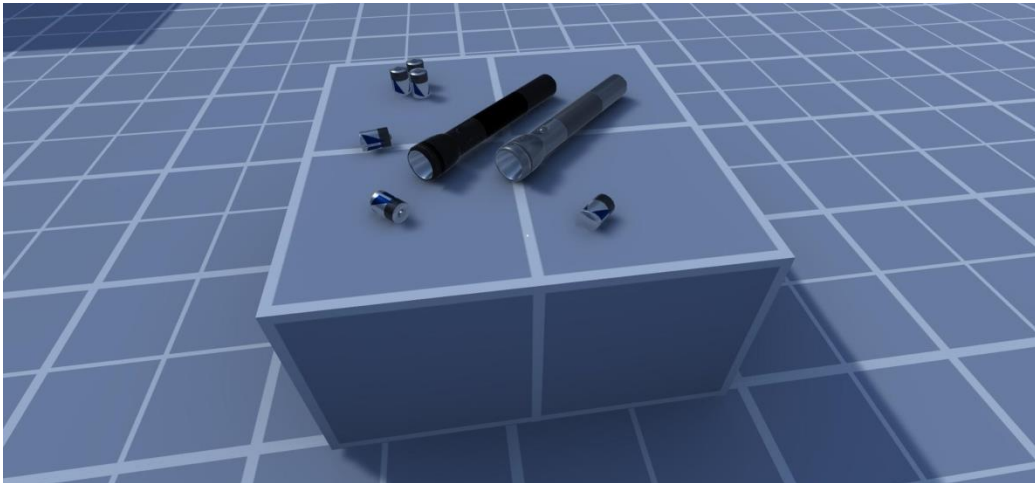
The asset includes:
- ►*Raycast System*
- ►*Trigger Example (3rd Person)*
- ►*Easily Customisable Flashlight Controller*
- ►*Reload the flashlight with simple button commands*
- ►*Battery gauge and intensity decrease over time*
- ►*PBR Flashlight included*
- ►*2x PBR Textures at 2k resolution*
- ►*Sound effects: Collecting, reloading and turning on / off*
- ►*Basic UI for battery level, count and flashlight on / off*
- ►*Full Documentation*
- ►*Full C# Source Code*

TAKE A LOOK AT THE ONLINE DOCUMENTATION HERE: https://speedtutoruk.gitbook.io/flashlight-system-doc/

FAQ

Q). How do I Import the asset?

A). Go to the Unity asset store and visit your "**Download manager**". Download the asset if not already downloaded and click "**Import**", import all required features of the asset for your use. It should have appeared in your project under "**Flashlight System**".

Q). Is there an example of this asset working?

A). Yes, you can open the "**FlashlightScene**" to see the flashlight system in action, or use this scene as your initial base of your project.

Q). How can I manually setup this asset?

A). See the manual setup instructions on "This page".

Q). My raycast isn't working properly, what can I do?

A). Make sure your FPSController has a layer of "Ignore Raycast" as the player collider can sometimes get in the way of the raycast.

Q). Do you have online Documentation?

A). Yes, you can see it all here: https://speedtutoruk.gitbook.io/flashlight-system-doc/

Manual Setup – Initial #1

TAKE A LOOK AT THE ONLINE DOCUMENTATION: For Videos, Updates, Tips and more
https://speedtutoruk.gitbook.io/flashlight-system-doc/

1. Add an "PlayerControllerParent" to your scene (From the prefabs folder). (Unless you're using the demo which will already have one added but the package will be required to be imported before use). As the system specifically requires the "FlashlightSwing" empty gameobject to be attached to a empty parent.
    a. (Make sure the "FlashlightSwing" is parented to the "PlayerControllerParent" and NOT the "FPSController")
    b. On the "**FPSController" -** Make sure you have the layer of **"Ignore Raycast"** - If you have troubles with the raycast not registering when you look towards the ground

2. Add "**FlashlightController**" / "**AudioManager**" / "**InputManager**" & "UIManager" to your hierarchy or scene from the assets "**Prefabs**" folder. Add this asset wherever you want in your scene. See the example below.
    a. **NOTE:** Only drag the "**FLUIManager**" prefab into the hierarchy, not the scene
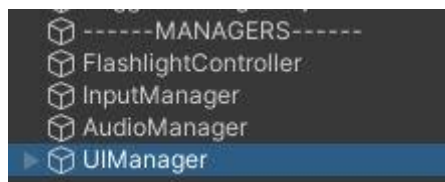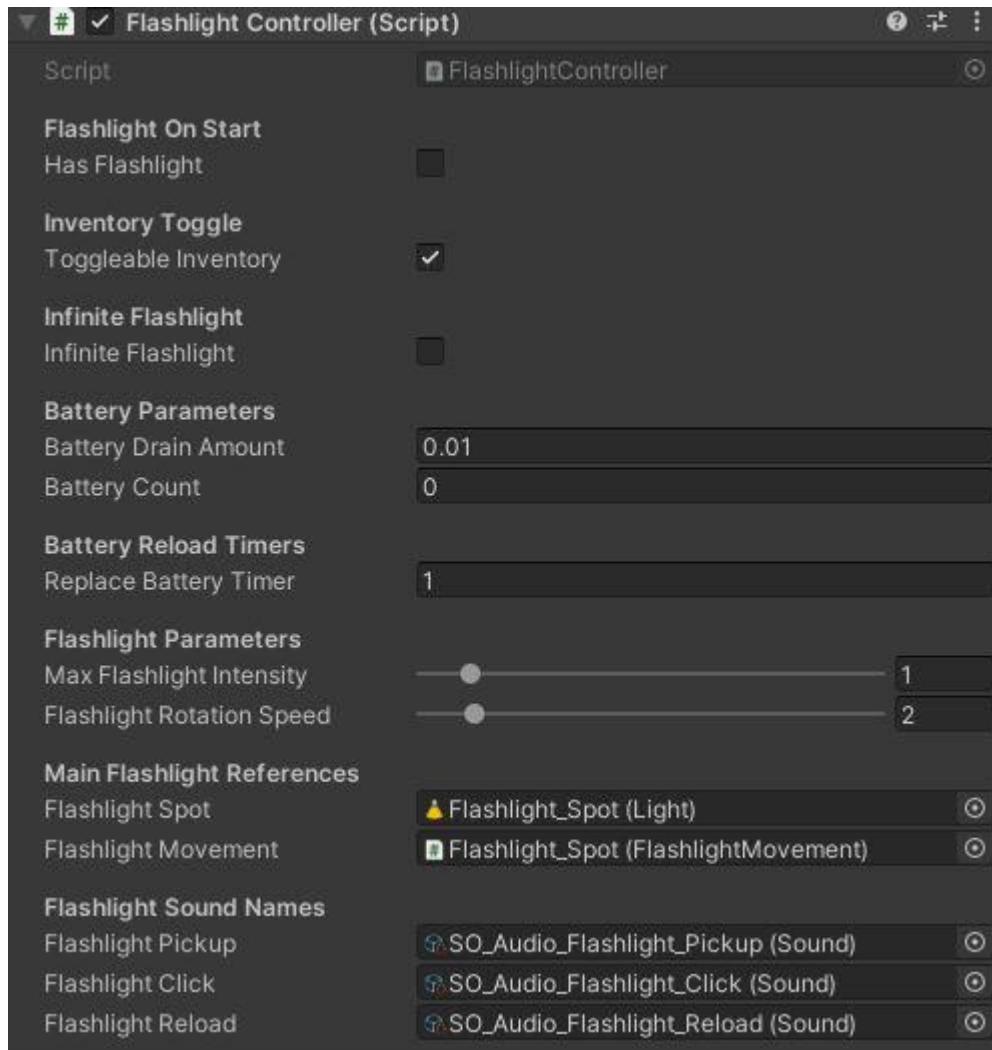

3. On the "**FLUIManager**" object make sure to attach all UI elements as per all the children underneath that asset, refer to the screenshot below:
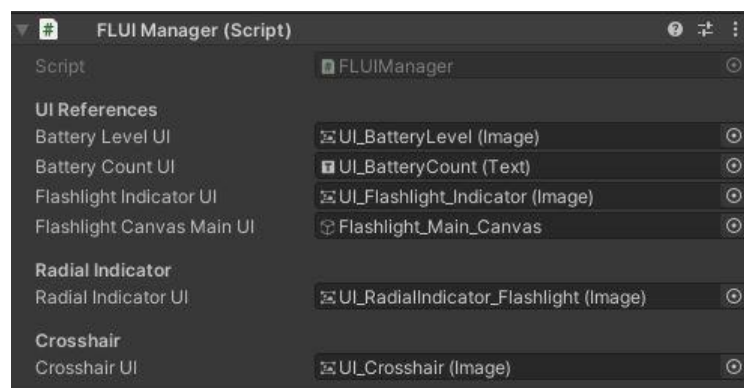
4. Look at the "**FlashlightController**" object and look at the script and set references accordingly, use values for your game.

5. Place a "Flashlight_Battery_1" into your scene from the prefabs folder.
   a. Add the "FlashlightItem" script to it, if not already attached and select the object type.
   b. Add the number of batteries you will gain from picking up this object.

6. Add the "Flashlight_DarkMetallic" or "Flashlight_LightMetallic" to your scene and place it where you want.
   a. Add the "FlashlightItem" script to it, if not already attached and select the type again from the dropdown box.

Final Notes

Your asset is setup and ready to use in your scene, please remember a few things.

1). If you need to access the Flashlight scripts from another script that isn't in relation to my asset you may need to use a namespace collection at the top of your script: "Using FlashlightSystem;"

2). You can add multiple batteries to the scene by dragging more battery prefabs.

3). Remember that the "**FlashlightSwing**" needs to be a child of an empty parent, and the flashlight light object a child of the "**FlashlightSwing**" object. Or the swaying will not work correctly! (See demo if you're confused in any way.

Remember to take a look at the demo scene if you have any troubles, it might give you an idea on how to fix an issue!

If you find the package helpful, please leave a positive review and star rating as it would really help me out! ☺ If you have any problems, feel free to send an email to me!

Contact

If you have any problems with the pack, or have some ideas for new features you'd be interested in, please feel free to contact me.

Thank you very much for downloading! Please be sure to leave a **5*** rating and **review** if you liked the package! Support me on
Patreon: https://www.patreon.com/SpeedTutor
Make a small donation: http://www.paypal.me/speedtutor

If you have any problems about the setup, usage, customisation or have any suggestions for future updates of this kit, please feel free to contact me.

Email: speedtutoruk@gmail.com
Discord: https://discord.gg/FguwaJHfSY
YouTube: https://www.youtube.com/user/speedtutor
Website: http://www.speed-tutor.com

Online Documentation

https://speedtutoruk.gitbook.io/flashlight-system-doc/

MATTHEW RUKAS – SPEEDTUTOR
WWW.SPEED-TUTOR.COM
SPEEDTUTORUK@GMAIL.COM

Patch Notes

Version 1.7.0 – July 2023

- Fixed script GUID issues by duplicating all scripts and replacing
- Made **Sound** script generic so it will work with any of my assets
- Refactored the **FlashlightMovement** script to make it more efficient
- Refactored **FlashlightController** script
    - Added new ScriptableObject references for sounds
    - Split all functionality into separate methods for easier usage
        - Added new methods for cool down timer and reloading
        - Added a refactor for collecting a flashlight
        - Added a new PlayerInput method to split logic from content
        - Refactored entire drain logic section into a few lines

- Update all fields with headers, titles and tooltips
- Refactored the **FlashlightItem** script to use a switch/case for better optimisation
    - Flashlight pickup method to now matches new **FlashlightController**
- Refactored **FlashlightPickupTrigger**
    - Flashlight pickup method to now matches new **FlashlightController**
- Updated the AudioManager to be slightly more refined
- Renamed FlashlightRaycast to FlashlightInteractor
    - Refactored this script to make it more efficient and shorter
- **FLUIManager** additions
    - Added new functionality of canvas to turn on / off the canvas group instead of the main canvas object so it is always suitable for visability in the scene to avoid confusion.
- Customised the flashlight UI to be more aligned with other assets
    - Removed the text from the UI
    - Renamed all UI elements to make them uniformed
- Added new large yellow flashlight
- Added new smaller flashlight

Version 1.6.0 - June 2022

- Renamed all textures for consistency
- Organised hierarchy of demo scene for ease of use
- Cleaned up spare UI sprites from project folder
- Created new UIManager
    - Moved Crosshair UI from raycast manager script

- o Moved all UI elements under the "UIManager" parent
- o Updated all UI references from FlashlightManager to new UIManager
- Optimised Flashlight Controller script and functionality
- Added ScriptableObjects for all audio effects for easy management
- Flashlight Controller now has an option to disable inventory input if you don't want to see or use it (Especially if you're using the infinite flashlight)
- Added an additional radial circle sprite so the package now has one thin and one thick option

Version 1.5 – June 2021

- Updated the "FlashlightRaycast" script to make the system more refined
    - o No need for layers or tags for general objects
    - o Set the FPSController layer to "Ignore Raycast"
- Added Radial UI for replacing batteries
    - o Hold or let go of the replace battery key to do the UI interaction
    - o Added new canvas "Flashlight_Radial_Canvas"
- Added a boolean to on the "FlashlightController" to select whether you have a flashlight at the start (Not needing the requirement of picking it up
- Re-imported Flashlight and Battery 3D assets with correct scaling of 1, 1, 1
- Updated all prefabs and refined scripts

Version 1.4.1 - March 2021

- Online documentation created at:
- Reduced asset size by removing light-baking
- Updated to lowest Unity V2018.4 (As per Unity rules)
- Removed Unity warnings from scripts by adding variable endings
- Refactored scripts with slight optimisations and shortening where required
- Added easy variable names to change sounds in the "FlashlightController" script
- Refactored and tweaked the "FlashlightTrigger" script to allow the user to pickup a battery or flashlight on trigger event
- Added custom battery number, so you can pickup multiple batteries at once

Version 1.4 – September 2020

- UI Changes:
    - o Redesigned the flashlight UI to incorporate a new design
        - ▪ New flashlight basic icon
        - ▪ Radial 360 dial to show battery indication
        - ▪ Bottom corner battery number indicator
        - ▪ Re-written to combine this new design
- Scripting Changes:
    - o Hold "Reload Battery" key to replace the battery instead of just pressing the button
    - o Optimised and re-factored the code to match with changes
    - o Re-based the scene to have a much darker demo

Version 1.3 - July 2020

- UI Changes:
    - o Edited the canvas layout and the way it is display in Unity
- Scripting Changes
    - o Added a new Audio Manager, so more sounds can be added and called within script
    - o Added a new input manager, which allows easy control of all inputs from one place
    - o Refactored the Raycast, ItemController and main flashlight manager scripts to make them more efficient, optimised, and easier to use
    - o The newly created "ItemController" allows an enum to be used to select the object type, the developer can easily add more objects to the enum to extend the system.
    - o Added a trigger script to pickup batteries by walking over them
    - o Added a new "Infinite Battery" tickbox in the controller, so you never need to pickup batteries, if the system is to be used just as a standalone flashlight mechanic.
- Miscellaneous
    - o Updated and edited sounds, textures and materials
    - o Added new manager examples to the prefabs and demo scene
    - o Update all prefabs and set them to 0,0,0 for future use
    - o Changed tag type to only require "Pickup" on each item