Contents

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

GAS MASK SYSTEM V1.4
DOCUMENTATION

Introduction

Thank you for purchasing the “Gas Mask System”. This is a Raycast based system, where the aim is to collect a gas mask and filters. These can be placed anywhere in your game world. You are required to collect the gas mask and using a held button press, you can equip the mask. You can then enter areas which were previously inaccessible due to poisonous substances, or other means in your game. (The inaccessible areas are portrayed by the green gas particles that are provided in the package) You can use any collider inside Unity as an inaccessible area. You will be able to consume the gas mask filter to increase the filter time. The system includes a basic UI and health system that can be easily customised or removed if you so wish.

ORE FEATURES

►*Raycast System*
►*Custom Unified UI design with equip circular UI*
►ScriptableObjects for all Audio
►Post Processing Stack Integration
►Toxic Gas Particles Included
►*UnityEvents for multiple interaction on the health manager*
►2 PBR *Gas Masks and filter models*
►*Sound effects: Breathing, choking, equipping, collecting and more!*
►*Works well for Horror games*
►*Fully Prefabbed*
►*Full Documentation*
►*Full C# Source Code*

Using Unity Version: 2019.4+
Build Using: Standard Render Pipeline – Suitable for all other versions.

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

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 "**Gas Mask System**".

Q). Before you start / Why isn't my character moving in the demo scene?

A). This package comes with the FPSController included in the package, so please don't overwrite with additional standard assets. IF YOU DO: Please, take a copy of the current FPSController script as it has some modifications!

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

A). Yes, you can open the "**GasMaskDemo**" to see the gas mask asset 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 "Page 2".

Q). Can I use my own custom effects for Vignette and blur?

A). Reach the "Extending the system" section for more information on that!

Manual Setup – Initial #1

PLEASE REFER TO ONLINE DOCUMENTATION FOR VIDEOS & DETAILED SETUP GUIDES:
https://speedtutoruk.gitbook.io/gas-mask-system-doc/

INPUT NOTE: Hold the "G" key to EQUIP or UNEQUIP the gas mask. (OR edit the "**InputManager**" script to change the key press)

QUICK START: Just add the "**ENTIRE DEMO PREFAB**" to your scene and you will have a quick start with everything you need!

STEP1: TAG AND LAYER SETUP:

- FPSController:
  - Tag: "Player"
  - Layer: "Ignore Raycast" - If you have raycasting detection issues
- MainCamera:
  - Tag: "MainCamera"
  - Layer: "PostProcessing"

STEP2: POST PROCESSING SETUP:

My system uses a post processing profile EXCLUSIVELY for the Gas Mask effects, as it changes profiles at runtime. This can help you have your own effects for normal gameplay and then a Gas Mask profile that ONLY activates when you entire the gas OR put on a gas mask. It's up to you whether you'd like to have both profiles very similar so there isn't a massive break in gameplay visuals.

Make sure to IMPORT the post processing package from the Window > Package Manager > Unity Registry > Search 'post' into your project.

- Main Camera:
  - Add a "Post-process layer" component
    - Set the layer to "PostProcessing"
    - The Trigger should be your MainCamera
- Create an Empty GameObject or use my "PostProcessingVolume" object:
  - Add the "Post-process Volume" script
  - Set the "IsGlobal" to true
  - Weight = 1
  - Add the "OriginalPostProcessing" profile or one you would use normally for your game

NOTE: I have a spare post processing profile called "GasMaskPostProcessing" - You can edit this to be similar to your current post processing but also needs to have Vignette and DoF as per my settings:

- Vignette:
    - Disabled By default
    - Intensity: 0.45
- Depth Of Field:
    - Disabled By Default
    - Focus Distance: 0.1

Step 3: Add an FPSController

- You can add this from the "Prefabs" folder or you can use your own FPSController for these purposes
- Make sure this has a tag of "Player" and a layer of "Ignore Raycast"
- On the MainCamera: Tag "Main Camera".
- Make sure the main camera has a "GMRayast" script and a "Post-process Layer"
- Set the "Post-Process layer" layer dropdown to "PostProcessing"

Step 4: Add collectable Objects to your scene

- Add a "Gas Mask" from the prefabs folder, this should have a "Box Collider" and "GMItem" script attached. Set the "GMItem" type to "Gas Mask"
- Add the number of "Gas Mask Filters" to your scene from the prefabs folder. Make sure this has a "Box Collider" and set the "GMItem" type to "Filter"

Step 5: Add a Gas Area (Particle Effect)

- Add the "GreenGasDamage" prefab or any effect you'd like
- Make sure this has a "Box Collider" set this to "IsTrigger"
- Make sure this has a "Gas Damage" script attached

Step 6: Add each of the following managers:

- GMCanvasContrainer (This has all the UI elements for the system
- GMHealthManager (You can edit all the variables here for controller health)
- GMController (You can edit all the variables for the gas mask system)
- GMAudioManager (This is where ScriptableObjects are added for audio
- GMInputManager (You can change inputs for the system here)
- GMUIManager (You can add all of the UI elements here and control post processing)

Final Notes

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

1). Make sure to have linked up all the important inspector items and play around with the settings to your desired effect!

3). Controls:
- Left Click to interact with a pickup
- Hold “G” to equip/unequip the gas mask
- Press “T” to replace the filter
- Press “TAB” to open the inventory

Of course all of these can be changed within the script as you wish!

REFERENCING THE AUDIO MANAGER:

1). If you click on the audio manager you can add addition elements to the array by incrementing the value by as many sound clips as you want to add, make sure to give them a name you will remember as you will reference this in your code!

2). Use the code, within your scripts to reference your sounds to play!
GMAudioManager.instance.Play(“NAME OF YOUR CLIP”);

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!

What is a namespace?

A namespace is almost like a folder that keeps scripts and classes from clashing with others that might have the same name. It's good practise to add them around classes which may have common names or specific use cases, so you can keep the code specific to your systems. This Gas Mask System using the namespace: "namespace GasMaskSystem" - you can reference this in other scripts to use the public methods in those scripts.

How to reference namespaces to use in other scripts?

If you need to access the Gas Mask scripts from another script that isn’t in relation to this asset you may need to use a namespace collection at the top of your script: “Using GasMaskSystem;”

ScriptableObjects for audio can be created by "Right Clicking anywhere in the project panel > Create > Sound" or duplicating an already created SO.

Creating more Audios with the ScriptableObject:

ScriptableObjects will all have specific settings that control the audio clip which is used, that you can change, the volume, pitch, variance and whether it can loop. All features of the default "AudioSource" component.

In the "AudioManager" is good to add the size of the "Sounds" array to the number of sounds which will play within this system. By default it has 8 SO audio clips, these should be added to access them all.

In the "GMController" - Look at the inspector and you'll see slots for each of the applicable SO which you create in the project panel, add any of these to the appropriate slots. This allows you to play that specific SO from the audio manager.

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!

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

Extending the Gas Mask System

What happens when I die and how do I edit this?

In the "**HealthController**" script has a UnityEvent added so you can use the "+" to add custom events! Email me if you have any troubles at all.

How can you use a different character controller with this asset?

In the "**GasMaskController**" script you will find a reference to the "**Player**" in the "**Public DamageGas**" and "**Public CanBreath**" that will control the walking and run speed. You will need to create a new reference to your player controller and change the values you wish when you're able to breath within your inaccessible area.

```
public void DamageGas()
{
    #region Damaging Gas Section
    canBreath = false;
    HealthController.instance.UpdateHealth();
    player.GetComponent<FirstPersonController>().m_WalkSpeed = walkGas;
    player.GetComponent<FirstPersonController>().m_RunSpeed = runGas;
    blurEffect.enabled = true;
    #endregion
}

public void CanBreath()
{
    #region Can Breath Region
    canBreath = true;
    player.GetComponent<FirstPersonController>().m_WalkSpeed = walkNorm;
    player.GetComponent<FirstPersonController>().m_RunSpeed = runNorm;
    blurEffect.enabled = false;
    #endregion
}
```

Patch Notes

Version 1.4 - October 2023

- Added debug reference for when the Gas Mask system is imported to tell the user to import post processing
- Fixed script GUID issues by duplicating all scripts to reduce clashing
- Added boolean for having gas mask at the start
- Added a new Gas Mask UI Background and renamed UI elements
- Added CanvasGroup to GasMask Canvas main
- Visor UI Canvas now uses a CanvasGroup to turn the object on and off meaning it will never have a disabled gameobject
- Updated GMUIManager to reflect this change
- Optimised this script for easier usage and better method naming
- Renamed maxHealthTimer to RegenerationDelay for better understanding, to control before the health will start regenerating
- Added a regeneration speed field so you can control health regeneration speed
- Optimised ObjectInteract script
- Optimised the crosshair method
- Optimised this to use the Sound ScriptableObjects rather than string name
- Changed the GMController script to use SO rather than name string
- GMRaycast renamed to GMInteractor
- Refactored this script to make it slightly more optimised and easier to read
- Renamed icons and textures
- Removed RotateScript from package
- Made Sound script generic so it doesn't clash with other assets

Version 1.3 - April 2022

- Added new Gas Mask / Barrel model to the system
- Added audio ScriptableObjects for easy creation for users
- Added information in docs and demo scene for holding "G" to equip mask
- Added new post processing for SRP version to replace legacy
- Added radial prompt for changing filters and equipping gas mask
- Create new UI Manager script for all UI management
- Created a method for editing player movement for easy editing
- Create Getter Setter for all public variables
- Refactored Health Controller and migrated UI code to UI Manager
- Refined Raycast script and remove exclusion layer
- Edited the "MaxEquip timer" variable to move from 0 - 10 on the range
- Removed baked lighting data in demo to save memory space
- Added and refined GMController to include new post processing
- Optimised Particle System for the Gas for better performance
- Renamed the Health Controller to Manager / GMItemController to GMItem

GAS MASK SYSTEM V1.4
DOCUMENTATION

Version 1.2 – September 2020

- Brand new unified UI design which includes a 360 filter level dial which morphs around the icon, re-positioned icons and indicators for filter amounts and indications. A new section for the health with moving numbers and health bar.
- Health will now regenerate when not inside gas (With or without the gas mask equipped
- Refined, refactored, renamed and tweaked the code to include new and remove redundant variables
- Added some range sliders for variables to keep them consistent
- Cleaned the Unity code to remove 40 default Unity exceptions

Version 1.1 – July 2020

- Edited the canvas layout and the way it is display in Unity
- Added custom colours to the Gas Mask controller for user customisation
- Updated the Health Controller to incorporate "**UnityEvents**" to make successful code interaction multi-use, meaning you can produce multiple outcomes with ease
- Added a new Audio Manager, so more sounds can be added and called within script
- Added a new input manager, which allows easy control of all inputs from one place
- Refactored the Raycast, GasMaskController, ItemControllers and more!
- Updated and edited sounds, textures and materials
- Added new manager examples to the prefabs and demo scene

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

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