Contents

MATTHEW RUKAS – VOLUMETRIC GAMES
WWW.VOLUMETRIC-GAMES.COM
VOLUMETRICGAMES@GMAIL.COM

GAS MASK SYSTEM V1.0
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, as long as you give the correct tags and logic. 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.

The asset includes:
- Raycast system that allows for pickup of gas mask and filters independently
- Add as many filters to your game or scene, with easy modification
- Basic UI for Health / Filter percentage / No of filters
- Lots of controls for customising it for your own game
- PBR Gas mask and filters.
- Scripts which can be easily modified.
- Sound effects for the system are included, for breathing and choking.

MATTHEW RUKAS – VOLUMETRIC GAMES
WWW.VOLUMETRIC-GAMES.COM
VOLUMETRICGAMES@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). I was advised by Unity to delete the "**Standard Assets**" > "**Characters**" from the asset, so please make sure to right click in the "**Project Panel**" > "**Import Package**" > "**Characters**" so the FPSController will be updated for you to use!

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). I'm having trouble getting the interaction to work. What can I do?

A). Make sure your Interactive objects (Mask and filters) have the "**Layer**" at the top right of the inspector as "**Interact**", and the appropriate tags are added to each. See setup for more details.

Q). What image effects do I need to use this system?

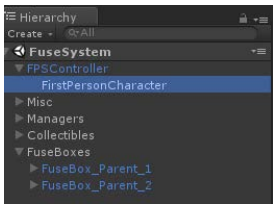
A). Image effects or standard assets aren't allowed to be uploaded by me as a publisher because Unity do not allow it, so please import the "Legacy Standard Assets" from the Unity Store. Unity 5.5 is required to import this account, if you're using a lesser version. You can import them into a higher version of Unity first and then copy the folder into a previous version.

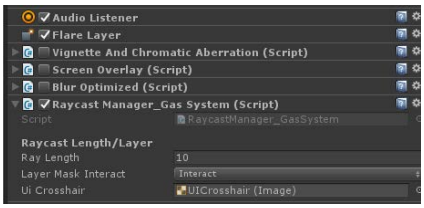
Manual Setup – Initial #1

1. When starting your new project please import the "**Characters**" Standard assets or any FPSController you wish. (Right click in the "**Project**" > "**Import Package**" > "**Characters**"). Add an "**FPSController**" to your scene. (Unless you're using the demo which will already have one added but the package will be required to be imported before use).
2. Please select your "**FPSController**" and open the "**FirstPersonController**" script in your code editor. Look for the two lines relating to "**Walk**" and "**Run**" speed variables and change the "**Private**" to "**Public**" like below. Make sure FPSController has the tag of "**Player**".

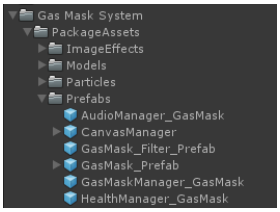
```
[SerializeField] private bool m_IsWalking;
[SerializeField] public float m_WalkSpeed;
[SerializeField] public float m_RunSpeed;
[SerializeField] [Range(0f, 1f)] private float m_RunstepLenghten;
[SerializeField] private float m_JumpSpeed;
[SerializeField] private float m_StickToGroundForce;
```

3. Please Import the **"Legacy Image effects"** from the Unity asset store into your project. It is required to use some of the image effects we want!
4. Import the "**ParticleSystems**" package from the standard assets. ("**Right Click**" > "**Import Package**" > "**ParticleSystems**")
5. Please navigate to the scripts folder in the gas mask package and add the "**RaycastManager_GasSystem**" script to your "**FirstPersonCharacter**" or "**MainCamera**".
6. Also add the "**Vignette and Chromatic Aberration**" & "**Screen Overlay**" & "**Blur Optimized**" to this camera, as seen below from the legacy image effects package. Make sure these image effects are turned off before gameplay.
7. Add the "**Worn_Mask_Overlay**" from the prefabs folder to the "**Screen overlay**" image effect on your main camera. Set the Intensity to -0.27.

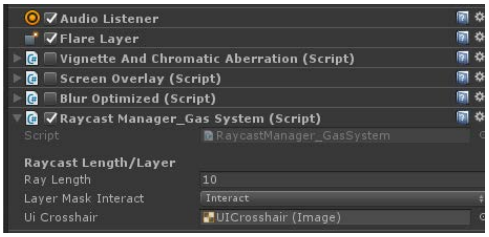




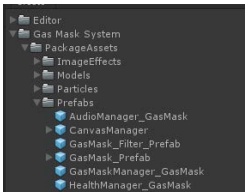
8. Now drag the "**CanvasManager**" from the "**Prefabs**" folder into the hierarchy.
NOTE: Only drag this prefab into the hierarchy, not the scene.



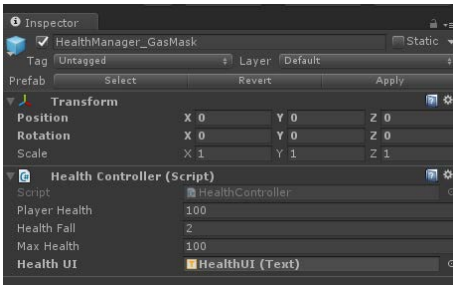
9. Add the crosshair image from the "**CanvasManager**" components to the "**FPSController**" > "**FirstPersonCharacter**" "**RaycastManager_GasSystem**" crosshair slot in the inspector.
10. Make sure the "**Layer Mask Interact**" is set to "**Interact**". **NOTE: If this isn't available in the dropdown you will need to create it at the top right of the Unity inspector, by choose the "Layer" dropdown and choosing "Add Layer". Once you create the layer it should auto fill the LayerMask box in the inspector!**



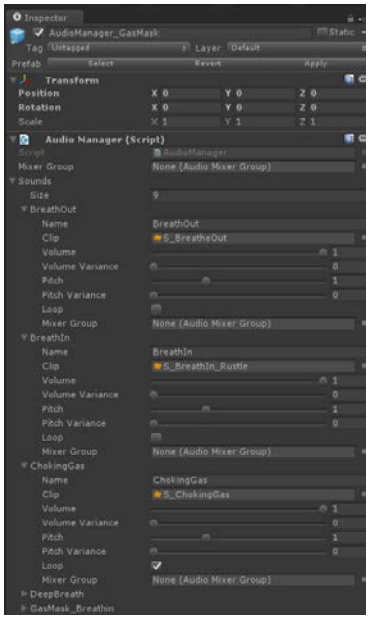
11. Add "**AudioManager**" and "**GasMaskManager**" and **"HealthManager"** to your hierarchy or scene from the assets "**Prefabs**" folder.



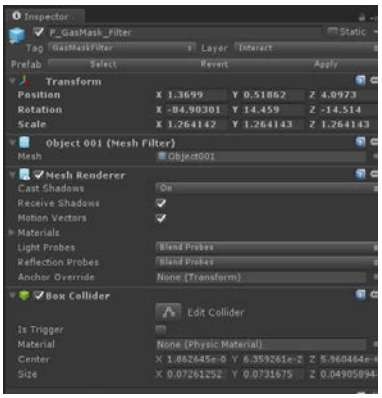
12. **NOTE**: Make sure your "**GasMaskManager**" has the tag of "**GasMaskController**". You can create this in the tag manager at the top of the inspector.
13. Add the "**HealthUI**" from the "**CanvasManager**" to the "**Health UI**" slot in the "**HealthManager**". Have **"Player Health"** and **"Max Health"** at 100. "**HealthFall**" at 2.



14. Make sure the "**AudioManager_GasMask**" has the sounds you require in the appropriate slots, see example. **NOTE: Keep volume and pitch at a minimum of 1 and give them the names as seen in the screenshot below.**



15. Place a "**Gas_Mask_Filter**" into your scene from the prefabs folder and make sure it has the tag of "**GasMaskFilter**" and Layer of "**Interact**". Also add "**GasMask_Prefab**" to your scene and give it a tag of "**GasMask**" and Layer of "**Interact**". **NOTE: You can duplicate the Gas mask filters later on.**



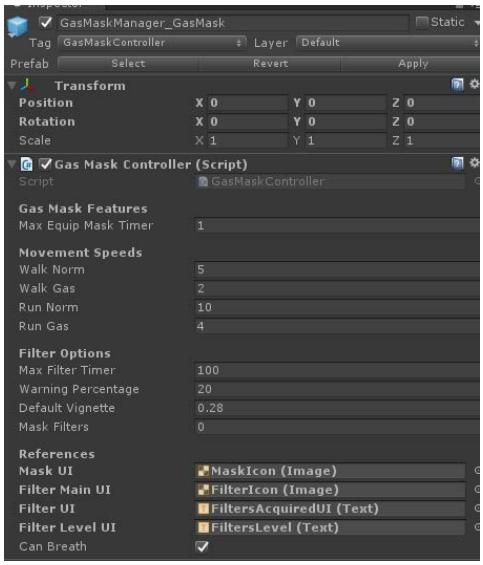
16. Look at the "**GasMaskManager**" in the hierarchy (so long as you have added this to your scene or it already exists. NOTE: Read through the information below and attach the correct GameObjects in the "**GasMaskManager**" script in the inspector. See the image on the next page for more details!

    Max Equip Mask Timer: The time in seconds, it takes to equip the gas mask using the held button press.

    Walk Norm: The normal walk speed of the FPSController
    Walk Gas: The speed of the FPSController inside Gas
    Run Norm: The normal running speed of the FPSController
    Run Gas: The speed of the FPSController running inside gas

    Max Filter Timer: The maximum timer value usually good to keep it at 100.
    Filter Fall Rate: The speed at which the filter percentage falls, increase for more speed.
    Warning Percentage: At what percentage should we show the UI warning?
    Default Vignette: The default value of the vignette
    Mask Filters: How many filers does the player currently have?

    Mask UI: Add the "MaskIcon" from the CanvasManager
    Filter Main UI: Add the "FilterIcon" from the CanvasManager
    FilterUI: Add "FiltersAcquiredUI" from the CanvasManager
    Filter Level UI: Add "FiltersLevel" from CanvasManager



17. You can now add one of the "**Mist_Green_Particle_Prefab**" to your scene, or you can create your own inaccessible area by creating a collider, and adding the "**GasDamage**" script to it. Remember to set this collider to "**isTrigger**"

Final Notes

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

1). If you're having errors on start-up, remember to import the standard assets, characters and also the "Legacy Standard Assets" for the desired image effects.

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

3). Controls:
- Press "e" to interact with a pickup
- Hold "g" to equip/unequip the gas mask
- Press "z" to replace the filter

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

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!

MATTHEW RUKAS – VOLUMETRIC GAMES
WWW.VOLUMETRIC-GAMES.COM
VOLUMETRICGAMES@GMAIL.COM

Extending the Gas Mask System

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

In the "**HealthController**" script you can find line 33 which has a public method called "**Death()**" In this method you can add your own outcome for player death. This could be the player respawning, reloading a scene and so on. If you feel any sort of confusion about this, please don't hesitate to send me an email!

```
    public void Death()
    {
        //DO SOMETHING HERE!
    }
}
```

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
}
```

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.

Email: volumetricgames@gmail.com

Website: http://www.volumetric-games.com