

Status Effects

[Asset Store Link](#)

© 2022 Justin Garza

PLEASE LEAVE A REVIEW OR RATE THE PACKAGE IF YOU FIND IT USEFUL!

Enjoy! 😊

Table of Contents

- [Status Effects](#)
 - [Table of Contents](#)
 - [Contact](#)
 - [Description Features](#)
 - [Terms of Use](#)
 - [Overview/Setup](#)
 - [Scripts](#)
 - [Status_Effect_Manager.cs](#)
 - [Other Scripts](#)
 - [Shaders](#)
 - [CharacterGlow.shadergraph and ZomBunny_Eyes.shadergraphs](#)
 - [RimGlow](#)

Contact

Questions, suggestions, help needed?

<https://github.com/jgarza9788>

Description Features

Displays a Status Effect using a shader and VFX particles.

- 5 different statuses
- Particles + Shader
- Easily Customize
- Easily Add new statuses

- Fully Commented C# code

Terms of Use

Required:

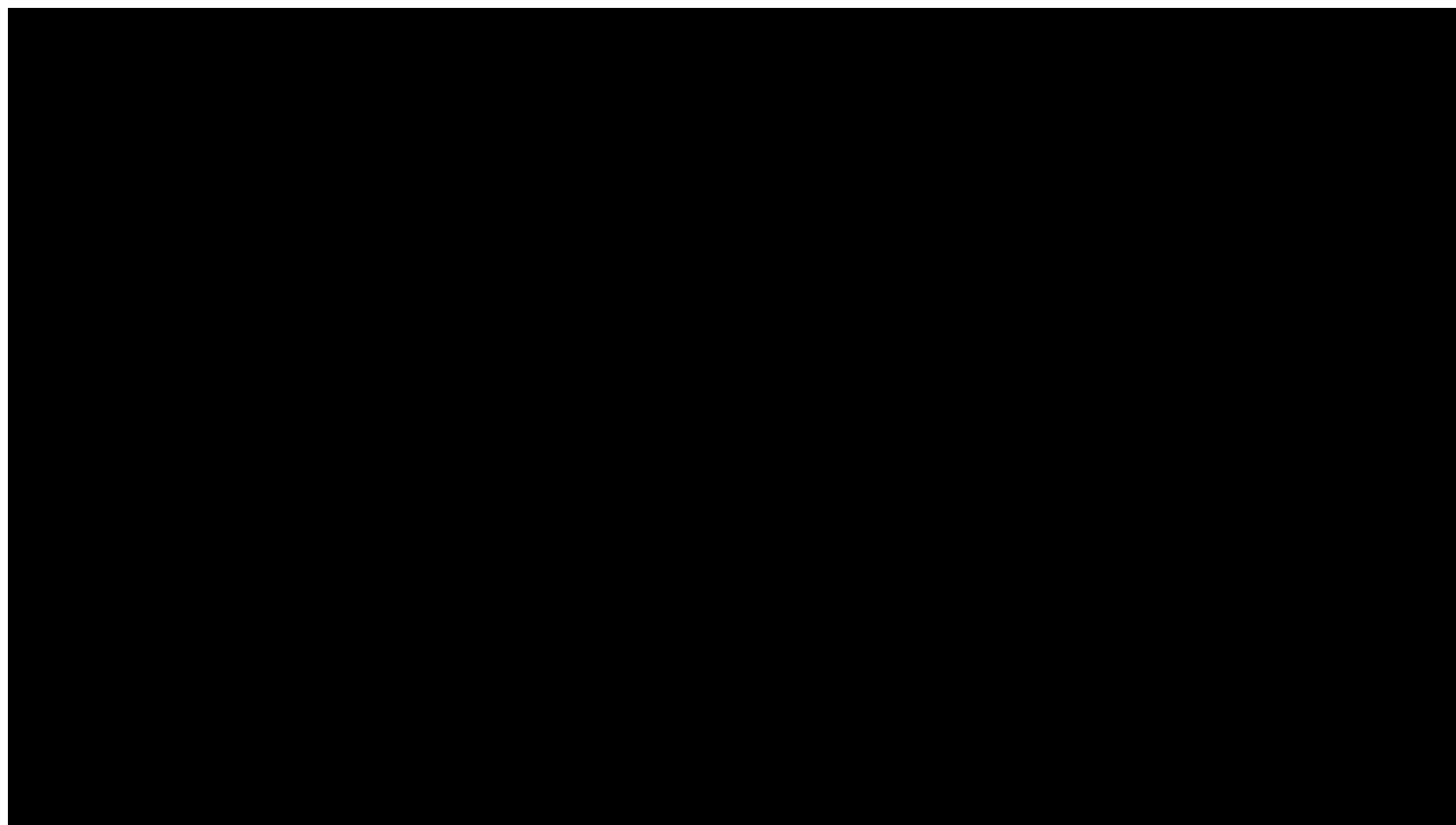
please follow [Unity's EULA](#)

Suggestion/Optional:

please put my name in the credits, or in the special thanks section. 😊

Overview/Setup

Status_Effect_Manager.cs manages the statuses of the character and updates the color and sends events to the VFX (for particles).



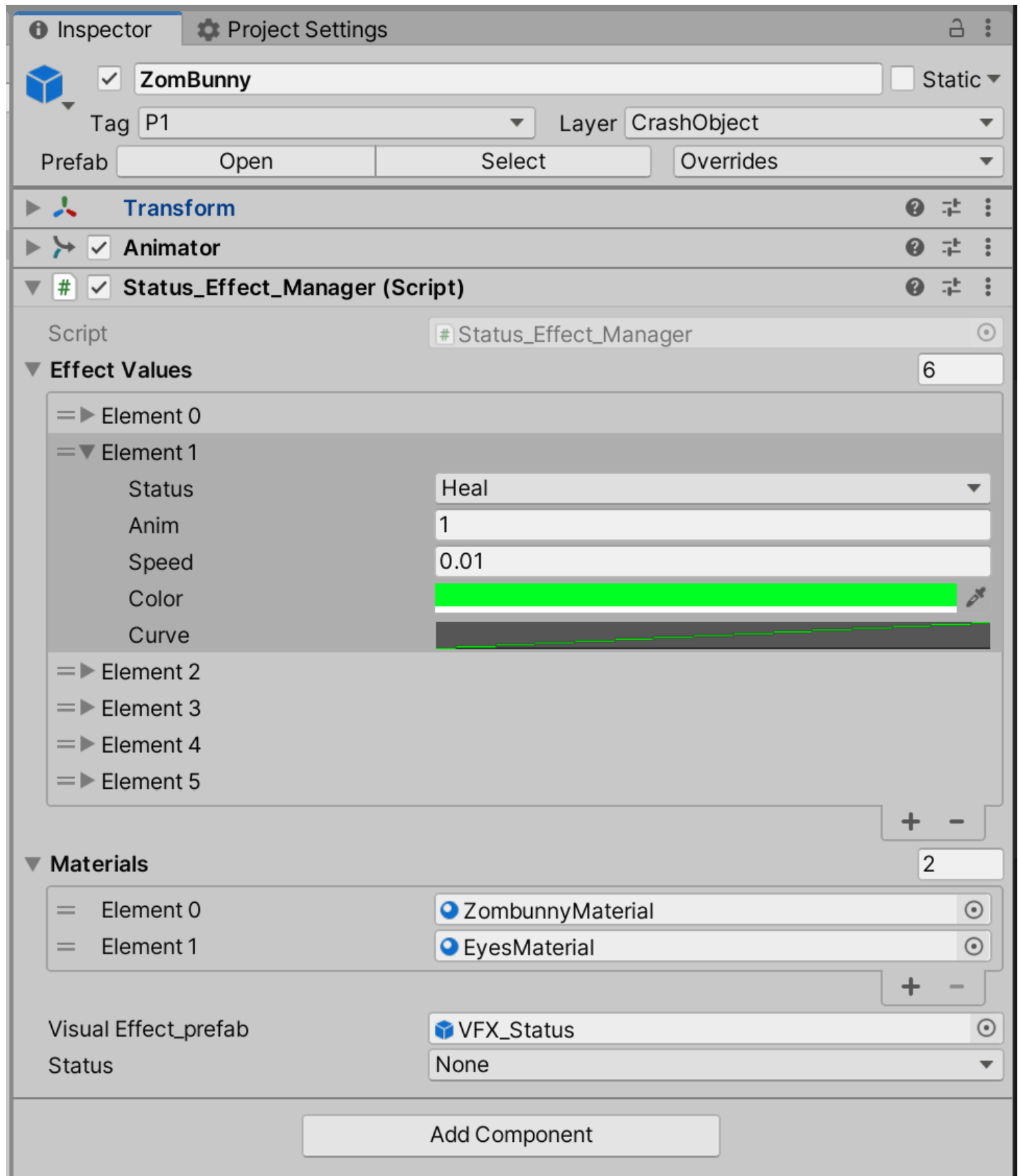
[Link to Video](#)

Scripts

Status_Effect_Manager.cs

Description:

This Script will trigger the status of the character(s)



Effect Values:

This is a list of all the different effects.

- Status:
the name of the effect, this will be sent to the VFX object to trigger the correct particles for the status
- Anim:
the current play value of the animation (0.0 -- 1.0)
- Speed:
the speed at which the animation plays
- Color:
the color used for the effect
- Curve:
the Curve for the animation

Materials:

This is a list of the materials with shaders that need to be updated.

the shaders must have 2 properties (GlowAnim (float), and GlowColor (a Color))
Please see the CharacterGlow.shadergraph as an example.

VisualEffect_prefab:

this is a prefab that will be created on Awake.

and it will spawn the particles when the event is triggered.

Status:

This is the effect that will be played.

Other Scripts

The Other scripts are basically just used for the Demos.

AlwaysFace.cs:

Turns the gameObject to face the Target.

Rotate.cs:

Used to rotate the camera.

Shaders

CharacterGlow.shadergraph and ZomBunny_Eyes.shadergraphs

these are the main shaders for rendering the ZomBunny.

both of these are updated by the Status_Effect_Manager.cs script,
and both use the RimGlow subshader.

RimGlow

this takes in the GlowAnim, GlowColor, and the Normal Vector (world space).