Energy Shield

Contact

Questions, suggestions, help needed?

Contact me at:

Email: jgarza9788@gmail.com

Cell: 1-818-251-0647

Contact Info: justingarza.net/contact

Description/Features

Displays an energy shield/force field effects when impacted by projectiles, or raycasts.

- Easily customizable effect. Size, Color, * Speed, etc.
- Renders on simple and complex meshes.
- Simultaneous impact points support.
- Support for culling and non-culling.
- Unity Free friendly.
- Fully commented C# code.
- Awesome demos!

Terms of Use

You are free to add this asset to any game you'd like However:

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

please do not re-distribute.

Table of Contents

Overview/Setup

The Energy Shield is composed out of a Mesh, Mesh Renderer, Collider, and the EnergyShieldManager.cs. The child GameObject ("RimShader") is only there to add an outline to the shield and allows it to be seen even if it's not rendering an effect. The "RimShader" GameObject can be moved or changed with out causing an error in the Asset.
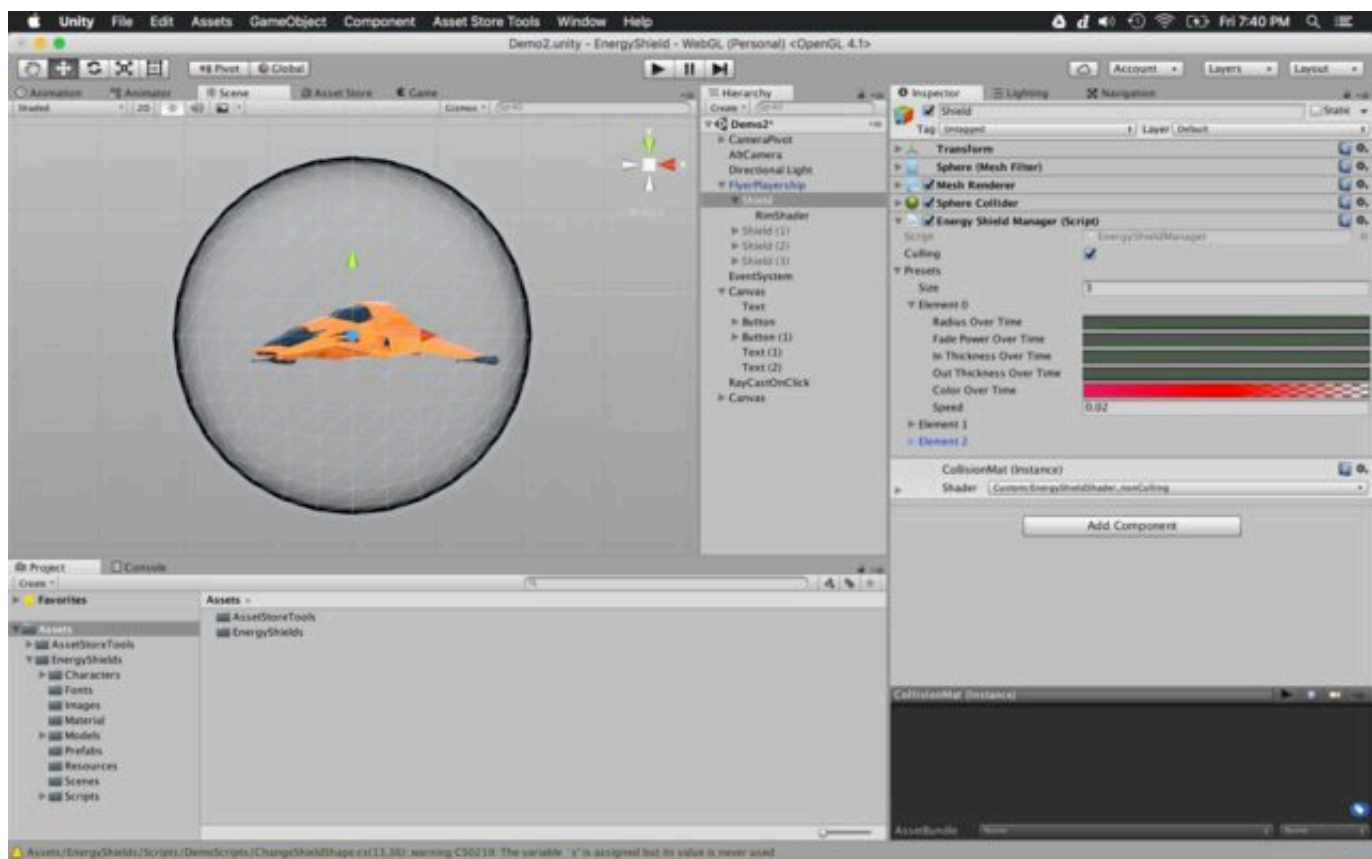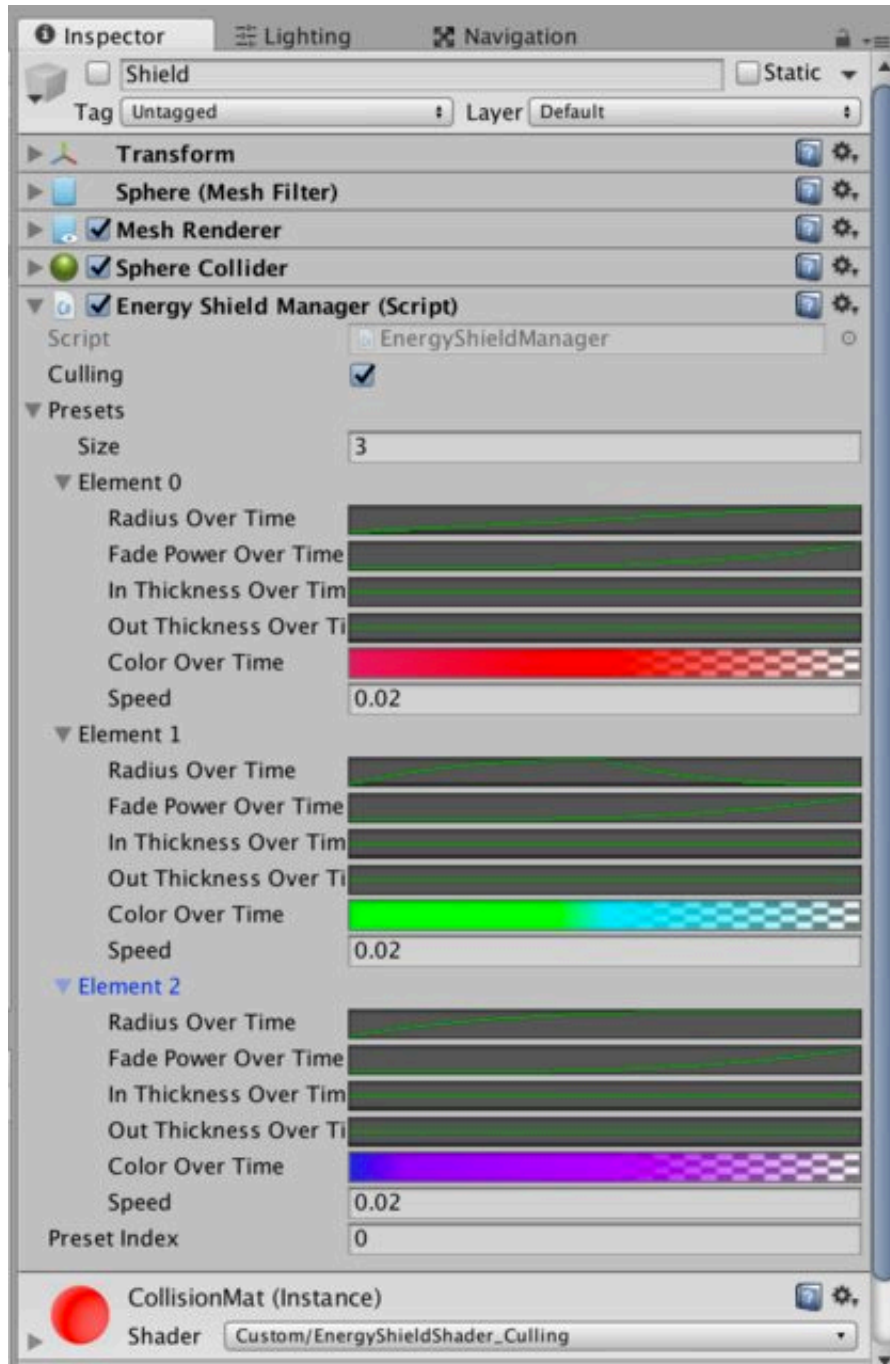

C# Scripts

EnergyShieldManager.cs

Description:
This Script Creates/Updates the shader to render the energy shield effects, and stores the presets for the energy shield effects.

Culling:
Culling will optimize performance by not rendering polygons facing away from the camera.
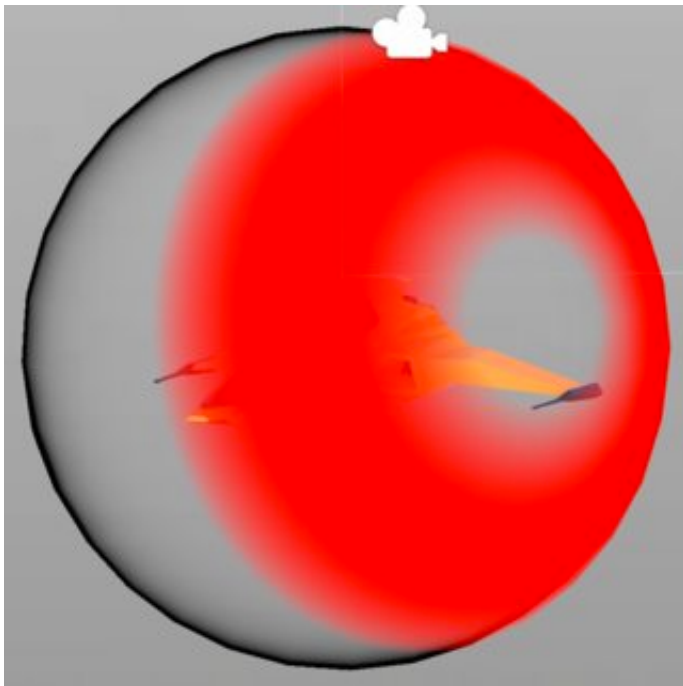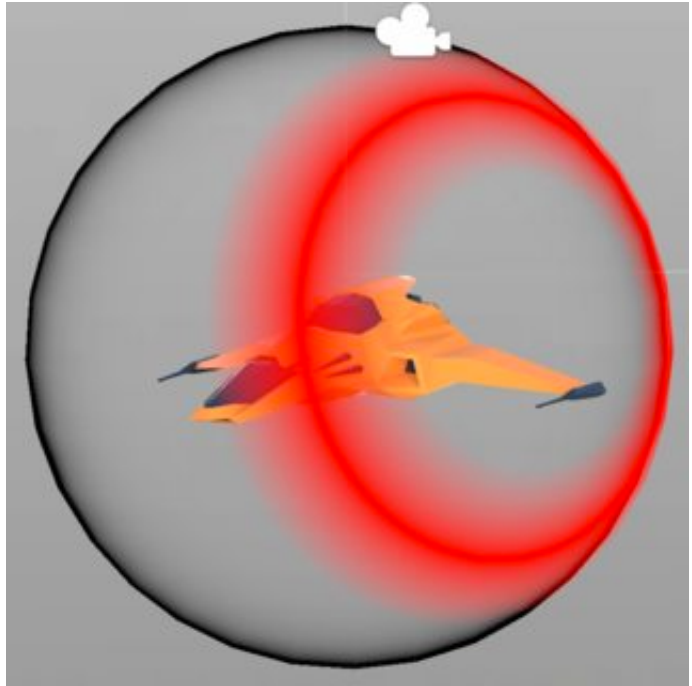
Presets:

Groups of values that will be used to determine how the energy shield effects will render.

RadiusOverTime:
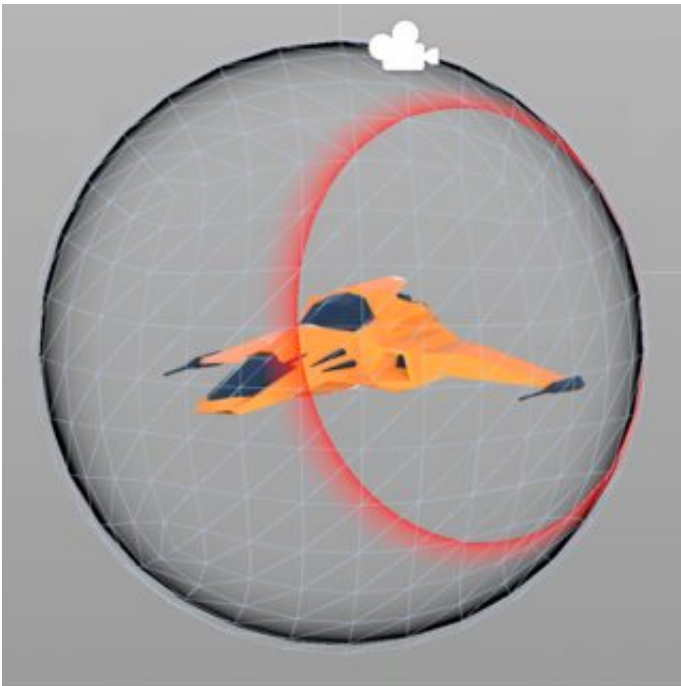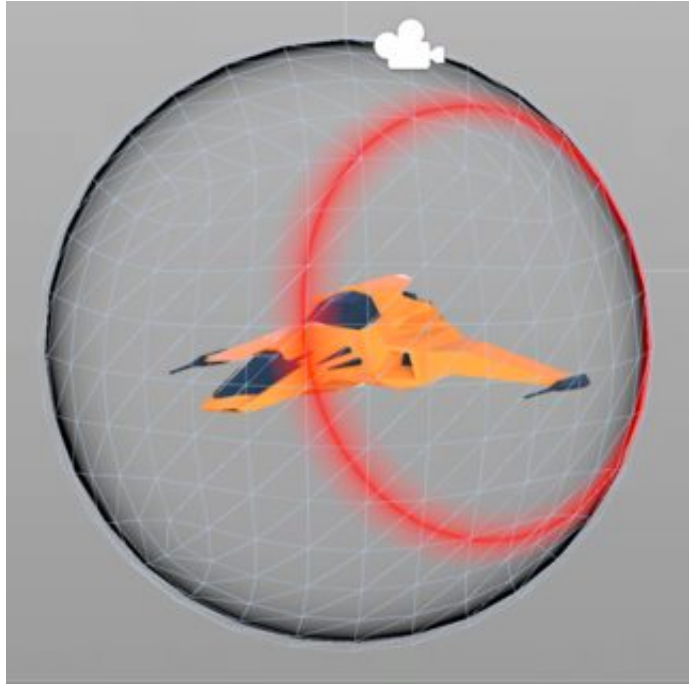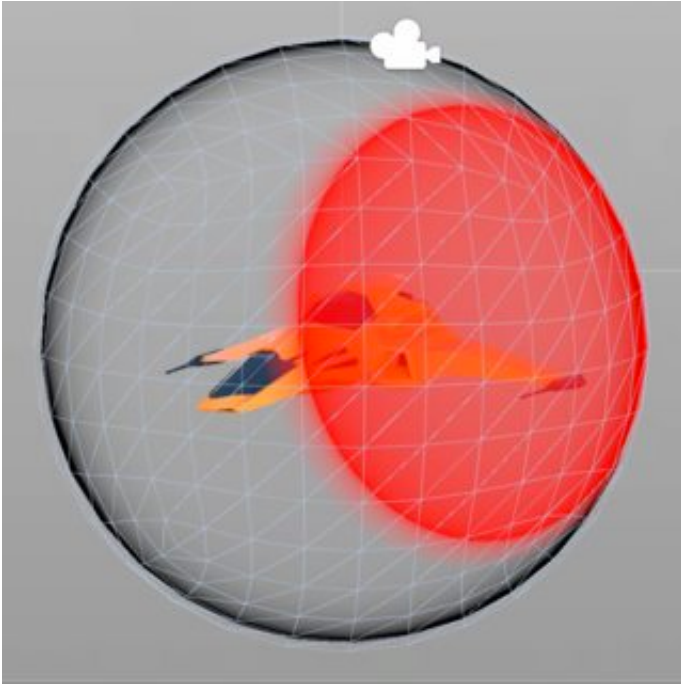The radius of the effect over time. The value should be between 0.5 and 10.0

FadePowerOverTime:
Controls how the circle fades on both sides of the radius. The value should be between 0.5 and 15.0

inThicknessOverTime:

Controls the thickness of the effect on the inside of the radius. The value should be between 0.05 and 15.0

outThicknessOverTime:
Controls the thickness of the effect on the outside of the radius. The value should be between 0.05 and 15.0

ColorOverTime:
This controls the color of the effect over time.
Should end with full transparency.

Other Scripts

Ther Other scripts are basically just used for the Demos.

AlwaysFace.cs:
Turns the gameObject to face the Target.

ChangeShieldShape.cs:
Changes the Shape of the shield for the demo.

DestroyAfter.cs:
Destroys an object after a set period of time.
Used in the projectiles.

Rotate.cs:
Used to rotate the camera.

ShootOnClick.cs: Controls the shooting of rays and projectiles.

SyncPresetIndex.cs:
makes sure all the shields use the same PresetIndex.

Shaders

Culling vs Non-Culling

The culling shader is more optimized since it won't render the parts of the mesh that are facing away from the camera.

However, this means if an effect is going on on the otherside of a transparent sphere it will not be viewable by the camera.

Using non-culling will fix this issue, however it doesn't alway render perfectly.

Properties

There is a set limitation to how many properties a shader can have. This means there is a limit of 10 effects that should occur on the same shield at the same time, since most effects do not last long i do not expect for this to be a problem with most users.

FAQs

Why Are Collisions Not Being Detected?

One reason why a collision is not being detected is because the shield is a child of an object with a rigidbody component. I'm calling this the "Rigidbody_ShieldChild" Problem. The solution to this problem is to have a script on the parent object that will detect the collision (Aka use the OnCollisionEnter method) and trigger the shield effect to occur.