Sci Fi Guns Documentation

Getting Started

To get started with this asset, you can either:

1. Create your own shooting system and VFX
2. Use our Plug and Play prefabs which are preconfigured with the shooting system and contain particle systems.

How to use the Plug and Play assets:

1. Add the prefab to the scene/spawn it through the Instantiate() function.
2. Call the Shoot() And Reload() functions on the gun based on user input.
3. Enjoy!

Scripting API

`Gun`: The Gun class handles:
- Initializing all particle systems.
- Spawning the gun projectile.
- Keeping track of bullets.

Provides Shoot() and Reload() functions.
All bullets move in the positive Z-axis, based on the rotation of the muzzle transform.

`GunStats`: This is a ScriptableObject class that is used to define the shooting behaviour of a gun, like its reload duration and fire mode (single shot or burst fire). Three preset GunStats objects are included in the asset per each gun variant but feel free to create your own and assign that to the gun script on any of the guns

`RecoilProfile`: This is a ScriptableObject class that is used to define the recoil behaviour of a gun. All Plug and Play guns come with a `RecoilController` and a preset recoil profile for each Plug and Play variant.

`RecoilController`: This script is optional and adds recoil to a gun given the recoil profile to use and whether the gun is right handed/left handed (changes the direction of the sideways recoil)
and whether the gun should “stack” its recoil up to the max defined in the profile or not.
`GunDynamicAnimationSpeedAdjustment`: It’s advised and HIGHLY recommended to use the dynamic animation speed adjustment script along with the gun script to ensure the VFX speeds stay synchronized with the stats given. If adding custom scripts I would recommend modifying this to suit your needs.
For most custom scripts that behave in a similar manner to `GunGlowManager`, i do advise to take a reference from this script and adapt to your needs, so you can adapt `GunDynamicAnimationSpeedAdjustment` in a way consistent with existing glow logic

VFX logic: `Railgun2Rails`, `GunChargeOverTime` and `GunGlowManager` are classes for the task of VFX.
`GunGlowManager`: It is used by the 2 railguns for their charged regions, lerping their glowing material’s emission color intensity from a base intensity and max intensity based on an animation curve. Used to add detail to the lightning particle system for the railguns.
The `invertGlow` boolean determines whether the material starts fully charged and discharges over time, or starts discharged and charges up.

`GunChargeOverTime`: This script behaves similar to the glow manager but modifies the glowing material’s color based on the amount of ammo in the gun. Used in the SMG and pistol presets.
The invertGlow boolean defines if it should be fully charged and get discharged over time or if it should stay fully discharged at rest and get charged over time.

The `Railgun2Rails` script adds a jolt effect to both rails of Railgun2 in the Plug and Play prefab.

Other Scripts:
`Bullet`: a script that moves a transform along the local positive z axis. This means the bullet's direction depends on its initial rotation which it derives from the muzzle transform.

`CameraController`, `NameTMPRotation`, `GunSpawner` and `DemoGunShooter` are all utility scripts just for the demo scenes (although feel free to use them if you like - i won’t mind :))
`CameraController` is used to move the camera within the demo scene with bounds.
The `NameTMPRotation` is used to rotate the pivot of all world space TMPs in the demo scene which showcase the name of the prefab used.
The `DemoGunShooter` is used to shoot all guns on left clicks in the all preset demo scene.
`GunSpawner` cycles between the guns.

Note: The demo scenes use TextMeshPro (TMP) so you may need to install that if you haven't already.

This asset was made by me and a friend — I handled the code and documentation while they worked on the models and visuals. Everything here is original, no third-party assets used.

FAQ

Q: My custom gun doesn't fire it's bullets the intended direction when using the gun class
A: Please ensure that the muzzle transform you provide the gun class is correctly oriented. One quick way to ensure that is to switch your handles mode to local from global (top left corner of the scene view) and click on your muzzle transform object, and rotate it till its z axis handle (the blue one) points away from the muzzle.

Q: My VFX is not working as intended (animations too slow or too fast) when using a custom GunStats scriptable object
A: Add the `GunDynamicAnimationSpeedAdjustment` script to the gameobject with your gun script, It should tune the speed and related parameters for the particle systems and the glowing (charging and discharging) of the elements in the guns which use them.

Customization

Q: How can I add my own gun models to use with this shooting system
A: Add the `Gun` Script (and optionally the `GunDynamicAnimationSpeedAdjustment` if you also use particle systems in the gun or the glow manager script) to the gun root prefab, provide

it with an existing `GunStats` object or a new one, give it a muzzle transform (ensure its correctly oriented), and give it the bullet prefab this asset comes with or a custom one.

Feel free to contact us for suggestions/improvements/bugs at thedevelopertrain@gmail.com or reach me out on my discord: thedevelopertrain_phoenix

Your feedback and reviews would be much appreciated to help us improve this asset over future updates

Thanks for checking out our asset!
Have a good day :)