Bullet Balistics - Documentation

Overview

Ballistic Settings (editor Window):

- Ballistic Quality:
  - Bulletdrop: are the bulltes affected by gravity?
  - Bulletdrag: are the bullets affected by air resistance and wind?
  - Ballistic Materials: will the bullets use the 'BallisticMaterial'-system on impacts?
- Drag Settings:
  - Air Density: density of the ingame air
  - Wind Direction: a Vector3 that defines the direction and strength of the wind
- Material Settings:
  - Name: preview name of the material
  - Energyloss Per Unit: the energy a bullet looses when traveling through 1 unit of this material
  - Spreadangle: spreadangle when the bullet leaves the material
  - Spreadanlge: spreadangle when the bullet is ricochet by the material (ricochet)
  - Ricochet Propability: this curve defines for each impactangle (90°-0° → 0-1) the propability (0%-100% → 0–1) of a bullet being ricochet
  - Impact Game Object: GameObject that is being instantiated on the impact with this material with a script attached that inherites from 'ImpactObject'

Weapon (component):

- General Settings:
  - Visual Spawnpoint: the transform where the visual representation of the bullet is instantiated at
  - Bullet Spawnpoint: the transform where the bullet calculations begin
  - Lifetime of Bullet: the time the bullet automatically gets destroyed after
  - Muzzle Damage: the damage one bullet deals with maximum velocity
  - Hit Mask: a LayerMask defining with wich layers the bullet can collide
- Bullet:
  - BulletPrefab: the visual representation of the bullet (attached scripts should inherit from 'PoolingObject' and use 'ReAwake()' instead of 'Awake()' when trying to perform actions on bulletspawning
  - Bullet Speed: Muzzlespeed of the bullet in m/s
  - Random Speed: speed differences between bullets
  - Mass of Bullet: mass of each bullet in kg
  - Dragcoefficient: drag coefficient of the bullet for drag calculations. (https://en.wikipedia.org/wiki/Drag_coefficient)
  - Diameter: diameter of the bullet

BulletHandler (component):

- There always has to be one BulletHandler in the scene
- Settings:
  - Max Bullet Updates Per Frame:

    the maximum amount of bullets that get processed each frame

    (to set a maximum cpu load if you have a **lot** of bullets in your game)
  - Visual To Real Bullet Movement Time:

    time it takes for the visual bullet to reach its 'physical' position; if VisualSpawnPoint and BulletSpawnPoint are not the same
  - Settings: reference to the BallisticSettings file (very important)

    create new BallisticSettings file under ‚Ballistics/Create BallisticSettings‘

Falloff Map Generator (component)

- Attach this component to the camera that draws the scope / to an object at the position from where the camera will look through the scope
- General:
  - My Weapon: reference to the weapon you want to create the falloff map for
  - Gizmo Size: Size of the editor Gizmos
- Zeroing:
  - Scope Distance: the distance in +z-direction from this object, where the falloff map will be created (>0)
  - Barrel Positon: the relative offset of the bulletspawnpoint, represented by a circle-gizmo (you can move the gizmo approximately to the end of the barrel)
  - Texturesize: the resolution of the generated falloff texture
  - Scope Zeroings: the distances for wich zeroing-markings will be drawn

Ballistic Object

- put this script on objects in your scene to assign them a ballistic material, wich you can define in the BallisticSettings Window
- use 'DefaulBallisticObject' or write your own scirpt, that inherites from 'Ballistic Object'
- 'BulletImpact(...)' is called by the BulletHandler when the BallisticObject is hit

Impact Object

- abstract class you can inherit from
- 'Hit(BallisticObjectData, RayCastHit)' gets called by an 'BallisticObject' when it gets hit
- used to play sounds, particleSystems etc. on bullet hits

Setup

Here is a link to a video tutorial on how to set it up:

https://www.youtube.com/watch?v=F9VINJOlnpM

(keep in mind, that this tutorial was created for an older version of this package and some steps might have changed)

- Create ‚BallisticSettings‘ File under ‚Ballistics‘ → ‘Create BallisticSettings‘
- create an empty GameObject and attach a 'BulletHandler' component
  - assign the BallisticSettings file there
- open the 'Ballistics' → 'Ballistics Settings' Window
  - adjust the world settings
  - create your own ballistic materials
- add a 'Weapon' component to your weapon
  - assign the fields as explained above
- add a 'SimpleWeaponInput' Component and assign your 'Weapon' to it;

  or write your own script, wich calls the Shoot() methode of the `Weapon`

hint: for example, if your 'maximal kinetic energy' equals 1000J and you have defined a ballisticmaterial with an 'Energyloss Per Unit' value of 990J you will be able to shoot through about 1 unit of this material when standing directly in front of it

Basic Weapon Controller

For an more advanced Weapon behavior you can use the 'Basic Weapon Controller'.

It gives you the possibility to use 'SingleShot', 'Full Auto', 'Brust' and 'Shotgun' weapons out of the box and also includes interfaces to use a 'SpreadController' and 'MagazineController' (as an example there is a 'DefaultSpreadController' and 'DefaultMagazineController' included)

On the 'Basic Weapon Controller' you have to call the 'Shoot', 'StopShoot' and 'Aim(bool)' methodes to controll shooting. Look inside the 'Basic Player Weapon Input' for an example how to set this up.

Access Weapon directly

Of course you also have the possibility to write a weaponcontroller completely by yourself, if the included one does not fit your purposes.

In order to fire your weapon you just have to call the 'ShootBullet(Vector3 direction)' on your weapon. The 'direction' parameter gives the direction the bullet will be fired in.

The rest will be taken care of by the 'BulletHandler'.

If you should have any problems, bugs, suggestions or questions please contact me:

Mr3d.CS@gmail.com

Best regards,

Christian