

GUN SYSTEEM

Jordy Teunis

Inhoud

Inleiding.....	1
Methodes	2
Uitvoering	2
1. Onderzoeken wat een goed gun systeem is	2
2. Gun systeem voorbereiden	3
3. Gun systeem in het project	4
Stap 1: Setup van het Geweer in de Hierarchy	4
Stap 2: GunData Scriptable Object	4
Stap 3: Gun script.....	6
Validatie	7
Reflectie	7
Bronnen.....	7

Inleiding

Het doel van het gun systeem is dat het makkelijk is om een nieuw geweer toe te voegen aan de game met de benodigde criteria en dat het makkelijk is om te begrijpen en aan te passen. Het gun systeem gaat een scriptable object gebruiken om alle informatie van de gun op te slaan, daarnaast een script waar de geweer functies in zitten zodat je kan schieten in het project. Om dit systeem goed onder de knie te krijgen heb ik het onderzocht wat de beste manier is om dit in elkaar te zetten en mee geoefend met verschillende geweren, zodat het een snel en handig systeem wordt om te gebruiken.

Methodes

1. Analyseren: Onderzoeken wat een goed gun systeem is
 - Type: Literatuur- en audiovisuele onderzoek
 - Beschrijving: Onderzoeken wat de beste en meest aanpasbare gun systeem is om te integreren met het project.
 - Bronnen: Unity Documentation en YouTube
2. Ontwerpen: Gun systeem voorbereiden
 - Type: De benodigdheden voor het gun systeem
 - Beschrijving: De benodigdheden voor het gun systeem in kaart brengen zodat er een overzicht is voor het gun systeem.
3. Realiseren: Gun systeem in het project
 - Type: Gun systeem integreren in het project
 - Beschrijving: Het gun systeem bruikbaar maken in het project.

Uitvoering

1. Onderzoeken wat een goed gun systeem is

Uit de verschillende informatiebronnen blijkt dat er een aantal verschillende aanpak mogelijkheden zijn om een gun systeem te maken. In het algemeen heb je een gun script nodig die alle physics behandelt van het geweer, daarnaast heb je een script nodig waar je alle variables voor een specifiek geweer in opslaat zodat je gemakkelijk verschillende wapens kan toevoegen.

Ook wordt het geweer object altijd in de camera/cameraholder gezet zodat het geweer duidelijk zichtbaar is in firstperson en door het geïntegreerde Unity systeem automatisch meebeweegt met de camera, hierdoor kan je ook meteen het geweer op de locatie naar wens plaatsen zonder dat er veel bij komt kijken.

2. Gun systeem voorbereiden

Er zijn een aantal objecten nodig om het gun systeem compleet te maken. Het gun systeem dat het overzichtelijk, gemakkelijk aanpasbaar en simpel is een systeem waarbij er een cameraholder en een gunholder wordt aangemaakt in het speler object en daarin de camera en gun wordt toegevoegd. Als child van het gun object wordt het geweer model toegevoegd zodat het overzichtelijk is welk object het script gebruikt en welk het model.



Figuur 1: Hierarchy van het gun systeem.

De beste manier om geweer informatie op te slaan is doormiddel van scriptable objects, hierin wordt gedefinieerd welke variables de gun nodig heeft zoals reloadspeed. Door het een scriptable object te maken is het makkelijk om snel een nieuw wapen toe te voegen want je hoeft maar één script te gebruiken voor verschillende geweren.

Een scriptable object is snel en gemakkelijk aan te passen en verbruikt weinig tot geen performance om te gebruiken, daarom is het de beste oplossing om informatie op te slaan over het geweer.



Figuur 2: GunData

3. Gun systeem in het project

Stap 1: Setup van het Geweer in de Hierarchy

1. Voeg een CameraHolder toe aan de Player:

- Maak een nieuw GameObject als child van de Player en noem het "CameraHolder".
- Voeg de Camera toe aan de CameraHolder als child.

2. Maak een nieuw GameObject voor de Gun:

- Maak een nieuw GameObject als child van de CameraHolder en noem het "GunHolder".
- Voeg een nieuw GameObject toe als child van de GunHolder en noemt het "Gun".
- Voeg in het Gun object een script component toe met de naam Gun.

(Zie hierarchy in 2. Gun systeem voorbereiden)

Stap 2: GunData Scriptable Object

Maak een GunData script aan in je project, hierin slaan we alle informatie over een geweer op. In dit script maak een CreateAssetMenu aan, zodat je makkelijk een geweer met GunData kan toevoegen wanneer dat nodig is via het create menu. Vul de volgende variables toe:

- name
- damage
- maxDistance
- currentAmmo
- magSize
- fireRate
- reloadTime
- reloading

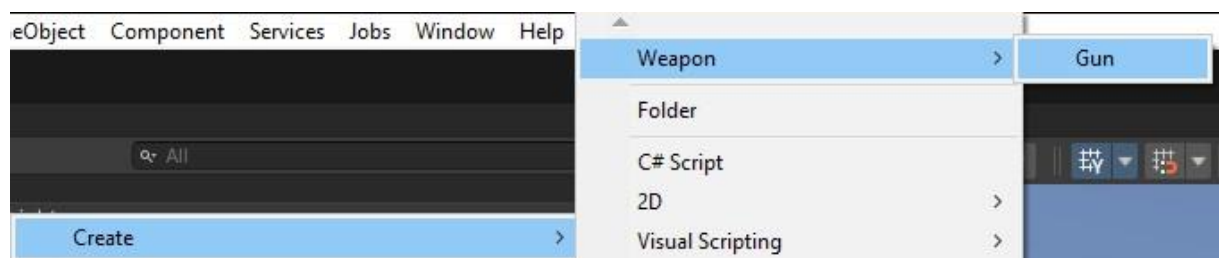
```

[CreateAssetMenu(fileName = "New Gun", menuName = "Weapon/Gun")] // Create a new gun in the weapon menu
0 references
public class GunData : ScriptableObject
{
    [Header("Info")]
    0 references
    public new string name; // The name of the gun
    [Header("Shooting")]
    0 references
    public float damage; // The damage that the gun does
    0 references
    public float maxDistance; // The maximum distance that the gun can shoot
    [Header("Reloading")]
    0 references
    public int currentAmmo; // The current ammo in the gun
    0 references
    public int magSize; // The magazine size of the gun
    0 references
    public float fireRate; // The fire rate of the gun
    0 references
    public float reloadTime; // The reload time of the gun
    [HideInInspector]
    0 references
    public bool reloading; // Is the gun currently reloading
}

```

Figuur 3: GunData script

Als dit script is opgeslagen kan je nu via het create menu in het project folder een gun aanmaken en deze een naam naar keuze geven. Vul daarna naar eigen keuze de waarden in die het beste past bij het geweer dat je maakt.



Figuur 4: Create Gun via GunData

Stap 3: Gun script

Pas het net aangemaakte Gun script aan waarin een referentie wordt gemaakt naar GunData zodat deze kan worden overgenomen door het Gun script. Maak daarna een simpel schiet systeem met de variables die zijn aangemaakt in GunData zodat het Gun script voor elk geweer gebruikt kan worden.

```
5 public class Gun : MonoBehaviour
6 {
7     [Header("References")]
8     [SerializeField] private GunData gunData;
9     [SerializeField] private Transform camera;
10    private float timeSinceLastShot;
11    private void Start()
12    {
13        PlayerController.shootInput += Shoot;
14        PlayerController.reloadInput += StartReload;
15    }
16    private void Update()
17    {
18        timeSinceLastShot += Time.deltaTime;
19    }
20    public void StartReload()
21    {
22        if (!gunData.reloadIng)
23        {
24            Debug.Log("Reloading");
25            StartCoroutine(Reload());
26        }
27    }
28    private IEnumerator Reload()
29    {
30        gunData.reloadIng = true;
31        yield return new WaitForSeconds(gunData.reloadTime);
32        gunData.currentAmmo = gunData.magSize;
33        gunData.reloadIng = false;
34        Debug.Log("Reloaded");
35    }
36    private bool CanShoot() => !gunData.reloadIng && timeSinceLastShot >= 1f / (gunData.fireRate / 60f);
37    public void Shoot()
38    {
39        if (gunData.currentAmmo > 0 && !gunData.reloadIng)
40        {
41            if (CanShoot())
42            {
43                if (Physics.Raycast(camera.position, camera.forward, out RaycastHit hitInfo, gunData.maxDistance))
44                {
45                    Debug.Log(hitInfo.transform.name);
46                }
47                gunData.currentAmmo--;
48                timeSinceLastShot = 0;
49                OnGunShot();
50            }
51        }
52    }
53 }
```

Figuur 5: simpel schiet script

Als de GunData wordt gebruikt zoals in het voorbeeld hierboven kan je elk verschillend soort gun die je aanmaakt gebruikt worden samen met dit script, je hoeft dus nu alleen de GunData te vervangen wanneer je een ander geweer wilt gebruiken.

Validatie

Ik heb verschillende geweren gebruikt op dit systeem en doordat de GunData verschillend is kan je makkelijk onderscheid maken tussen de verschillende geweren wanneer er mee wordt gespeeld.

Reflectie

Ik heb geleerd hoe ik een goed Gun systeem opzet en integreer in het project. Door de opzet die ik heb gekozen is het makkelijk te begrijpen hoe het werkt en is het makkelijk te hergebruiken. Door het leren van een scriptable object te gebruiken weet ik nu een snelle en efficiënte manier van variables opslaan en toe te passen op een project.

In het vervolg ga ik wanneer ik een scriptable object aanmaak meteen duidelijke namen gebruiken, zodat ik die later niet meer hoeft aan te passen. Voorderest ga ik het niet anders aanpakken, want ik heb eerst een onderzoek gedaan wat er nodig is voor een gun systeem en daarna het toegepast in het project.

Bronnen

Literatuur

SamMurphy. (z.d.). *GitHub - SamMurphy/Unity-Gun-System: Basic fps mechanics in Unity 5*. GitHub. <https://github.com/SamMurphy/Unity-Gun-System>

Tutorial - Make a ScriptableObject-Based Gun System from Scratch. (2022, 11 oktober). Unity Discussions. <https://discussions.unity.com/t/tutorial-make-a-scriptableObject-based-gun-system-from-scratch/896847>

Technologies, U. (z.d.). *Unity - Manual: ScriptableObject*. <https://docs.unity3d.com/Manual/class-ScriptableObject.html>

Audiovisueel

Rytech. (2022, 10 augustus). *MODULAR WEAPON SYSTEM in unity in under 4 minutes* [Video]. YouTube. <https://www.youtube.com/watch?v=47ZkulgnadI>

Plai. (2022, 23 januari). *Unity Basic Weapon System Tutorial* [Video]. YouTube. <https://www.youtube.com/watch?v=kXbQMhwj5Uc>