

Game Programming

Sci-Fi Demo: Lecture V Destructibles

Izzet Fatih Senturk

Destructible Setup

- Drag and drop the “Wooden Crate” object under Game Models folder to the Hierarchy window
 - Add a box collider to the new game object if it already doesn’t have it
- When you shoot the Wooden Crate object, disable the object and replace it with the “Wooden Crate Cracked” game object
 - Drag and drop the “Wooden Crate Cracked” object to the Hierarchy window
 - Copy and paste the position of “Wooden Crate Cracked” object to “Wooden Crate” object so that both of them have the same position

Destructible Behavior

- Update Player script, the shoot method
 - Check if we hit the crate
 - Start the crate destroy procedure
- Create a new C# script named Destructible_sc
 - Attach it to the Wooden crate game object
 - Define the DestroyCrate function

```
if (Physics.Raycast(rayOrigin, out hitInfo))
{
    Debug.Log("Raycast hit: " + hitInfo.transform.name);
    GameObject hitMarker = Instantiate(_hitMarkerPrefab, hitInfo.point, Quaternion.LookRotation(hitInfo.transform.forward));
    Destroy(hitMarker, 1f);

    //Check if we hit the crate
    Destructible_sc destructible_sc = hitInfo.transform.GetComponent<Destructible_sc>();
    if (destructible_sc != null)
    {
        destructible_sc.DestroyCrate();
    }
}
```

Destructible Behavior

- Update Destructible_sc script
 - Define a private handler variable for the destructible crate, serialize it and set its value from the Inspector window
 - Define a public method to be called from the Player script to destroy the crate

```
[SerializeField]  
0 references  
private GameObject _crateDestroyed;
```

```
public void DestroyCrate()  
{  
    Instantiate(_crateDestroyed, transform.position, transform.rotation);  
    Destroy(this.gameObject);  
}
```