

Lesson 11

Naša igra sada ima neku strukturu.

Prvo što trebamo promijeniti je da se životinje stvaraju bez pritiska na neki Button.

Isto tako, hrana nam prolazi kroz životinje, to je isto potrebno promijeniti.

Kada hrana dodirne životinju, i hrana i životinja moraju nestati sa scene.

Sada u SpawnManager scripti napravimo novu metodu - **SpawnRandomAnimal()**

U tu metodu kopirajmo kod koji se nalazi u Update metodi unutar IF-statementa. If statement ne kopiramo, nego ga brišemo, ne treba nam više.

```

// Update is called once per frame
void Update()
{
    .....
}

void SpawnRandomAnimal()
{
    int animalNumber = Random.Range(0, animals.Length);

    Vector3 spawnPos = new Vector3(Random.Range(-posX, posX), 0, posZ);

    Instantiate(animals[animalNumber], spawnPos, animals[animalNumber].transform.rotation);
}

```

Sada moramo isprogramirati vremenski interval stvaranja životinja.

To ćemo napraviti putem metode `InvokeRepeating`.

Tu metodu pozvati ćemo u `Start`, `Start` se poziva samo prvi frame, no metoda `InvokeRepeating` će se ponavljati. `Start` će ju pokrenuti prvi frame, i kad ju jednom pokrene metoda će raditi ono za što je zadužena.

```

// Update is called once per frame
void Start()
{
    InvokeRepeating("SpawnRandomAnimal", 2f, 1.5f);
}

void SpawnRandomAnimal()
{
    int animalNumber = Random.Range(0, animals.Length);

    Vector3 spawnPos = new Vector3(Random.Range(-posX, posX), 0, posZ);
}

```

Sada vremena koja smo stavili u tu metodu treba spremi u varijable.

```

public float startDelay = 2f;
public float spawnInterval = 1.5f;

// Update is called once per frame
void Start()
{
    InvokeRepeating("SpawnRandomAnimal", startDelay, spawnInterval);
}

```

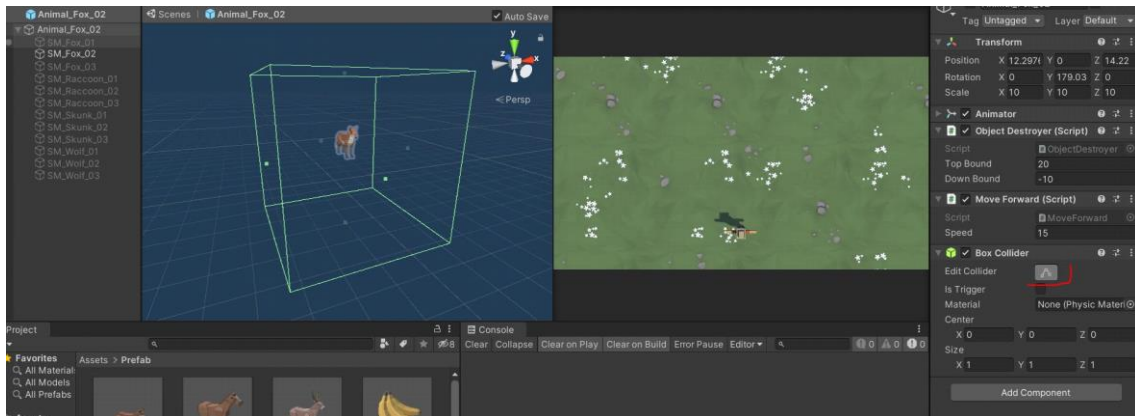
Sljedeće što trebamo napraviti je detekcija kolizije.

Prvo što trebamo je staviti okvir **Collider** oko životinje.

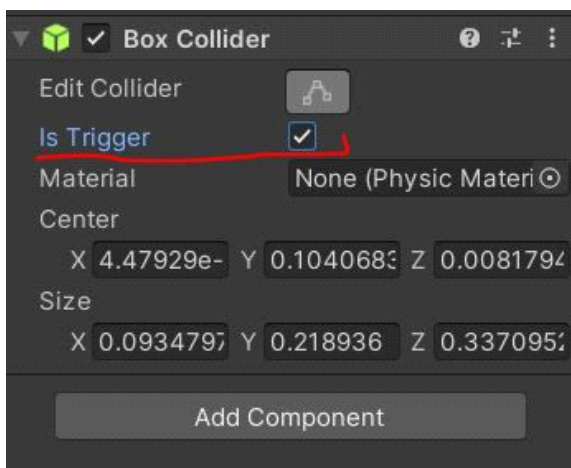
I uključiti trigger.

Kliknemo dva puta na Animal prefab, dodajmo **Box Collider**, namjestimo okvir i uključimo trigger - **Is Trigger**.

Okvir na životinji namještavamo klikom na button **Edit Collider**.



Nakon što složimo okvir uključimo trigger.

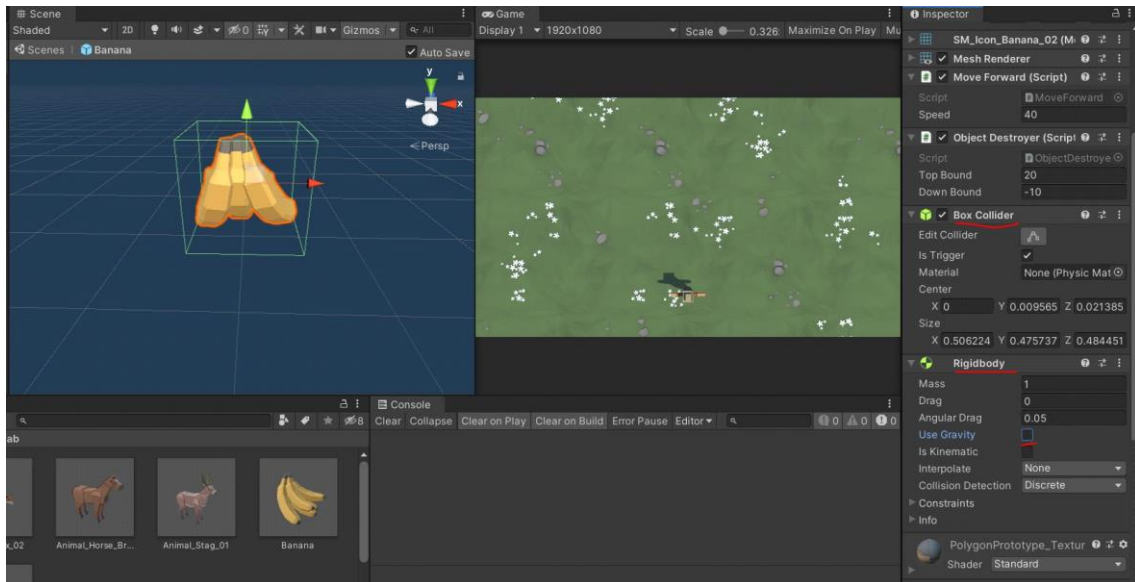


Razlog zašto smo dodali Collidere je kako bi **animal objekt** mogao detektirati kada ga je dotaknuo neki drugi objekt.

Isto to sve povimo i na **food objektu**.

No na food objekt ćemo dodati i **Rigidbody**. Jer da bi se detekcija objekata mogla dogoditi bar jedan objekt mora imati **Rigidbody**.

Na **Rigidbody** - isključite **UseGravity**. Kako na food nebi utjecala gravitacija.



Sada moramo napraviti skriptu koja će detektirati dodir objekata.

Napravimo skriptu **DetectCollisions.cs**.

Dodajmo ju na svaki Animal prefab i otvorimo skriptu.

U skripti pozovemo metodu **OnTriggerEnter**

```
public class DetectCollison : MonoBehaviour
{
    private void OnTriggerEnter(Collider other)
    {
    }
}
```

Ta metoda koristi se kako bi Unity znao dali je neki objekt dodirnuo drugi.

A ako je jedan objekt dodirnio drugi, nešto se mora dogoditi. A što se mora dogoditi određujemo mi.

A mi želimo kada neki objekt dodire Animal da animal nestane.

```
public class DetectCollison : MonoBehaviour
{
    private void OnTriggerEnter(Collider other)
    {
        Destroy(gameObject);
    }
}
```

Isto tako želimo i da se Food uništi pa ćemo dodati još jednu liniju koda. I reći da uništi i jedan i drugi game objekt koji se dodiruju.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class DetectCollison : MonoBehaviour
{
    private void OnTriggerEnter(Collider other)
    {
        Destroy(gameObject);
        Destroy(other.gameObject);
    }
}
```

Okej sada imamo skoro sve funkcionalnosti, fali nam još samo jedna.

Prvo izbrišimo sa scene sve animal objekte i food objekt.

Sada ćemo dodati Game Over poruku.

A to ćemo napraviti u skripti **ObjectDestroyer**.

Tamo smo već napravili logiku, kada životinja dosegne donju granicu.

```
void Update()
{
    if(transform.position.z > topBound)
    {
        Destroy(gameObject);
    }

    if(transform.position.z < downBound)
    {
        Destroy(gameObject);
        print("GameOver");
    }
}
```

Lesson Recap

Što smo naučili?

Kreiranje novih metoda

InvokeRepeating() - ponavljanje određenog koda.

Colliders i Triggers