

Poročilo SIGMA

PROGRAMSKI VZORCI - Vaja 2

Načrtovanje

Razmislek

Ugotovili smo, da za ta program, potrebovali za premikanje v prostoru 3 dimenzionalna polja, ki bodo beležila našo pozicijo, in naše trajektorje.

1. Korak

V orodju Unity smo izdelali preprost model drona, z imenom »DroneParent«. Ta je vseboval telo in 4 propelerje. Nato smo zanj še ustvarili `c#` skripto »DroneMovementScript«.

2. Korak

V omenjeni skripti smo definirali »ourDrone« (`Rigidbody` telo drona), »trenuten« (`int` zap. št. Trajektorja v seznamu) in »hitrost« (`float` hitrost premikanja drona) ter `List` »trajektorji« (`Vector3` cilji drona), ki so ključnega pomena za simulacijo.

V funkciji »Update« povemo da se naj pozicija drona spremeni v koordinate, ki so bližje naslednjemu trajektorju. In ko prispe, se naj začne približevati naslednjemu trajektorju, ki je v našem primeru za lažjo načrtovanje bil »Empty Object« z koordinatami.

3. Korak

Nato smo se odločili, da trajektorjev ne bomo dodajali, kot objektov v Unityu, ampak kot »position«, ki je spremenljivka objekta za vodenje drona.

Dronu smo telo odstranili in mu dodali visoko detajlno telo, ki smo ga najdlji na »Asset Store«

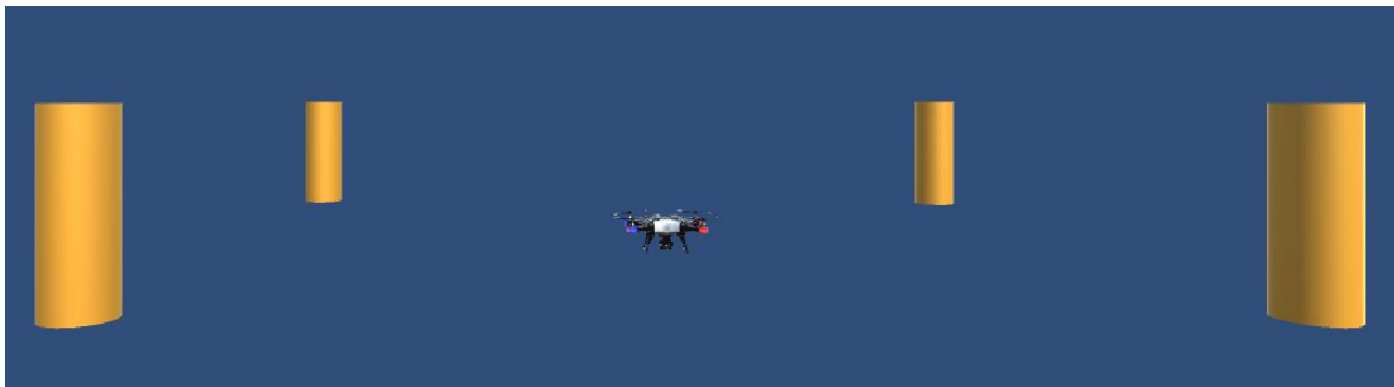
Dodali 4 objekte z teksturo, ki predstavljajo oddajnike.

Za zaključek smo še kameri dodali ozadje.

Končni rezultat

Drone ki kroži med lokalnimi cilji.

Opomba: cilji niso vidni in oddajniki so pobarvani rumeno.



Razred osnova (DroneMovementScript)

```
public class DroneMovementScript : MonoBehaviour {

    public Transform[] cilj;
    public float hitrost;

    private int trenuten;
    public List<Vector3> trajektorji = new List<Vector3>();

    Rigidbody ourDrone;

    public void Awake()
    {
        ourDrone = GetComponent<Rigidbody>();
    }

    // Use this for initialization
    void Start () {
        //trajektorji.Add(new Vector3((float)-1.48, 0, (float)-1.54));
    }

    // Update is called once per frame
    void Update () {
        if(transform.position != trajektorji[trenuten])
        {
            Vector3 pos = Vector3.MoveTowards(transform.position, trajektorji[trenuten], hitrost *
Time.deltaTime);
            ourDrone.MovePosition(pos);
        }
        else
        {
            trenuten = (trenuten + 1) % trajektorji.Count;
        }
    }
}
```