

Unity 3D, system cząsteczkowy,

1. Wstęp:

Wykonanie projektu symulującego ogień z użyciem systemu cząsteczkowego..

2. Oprogramowanie:

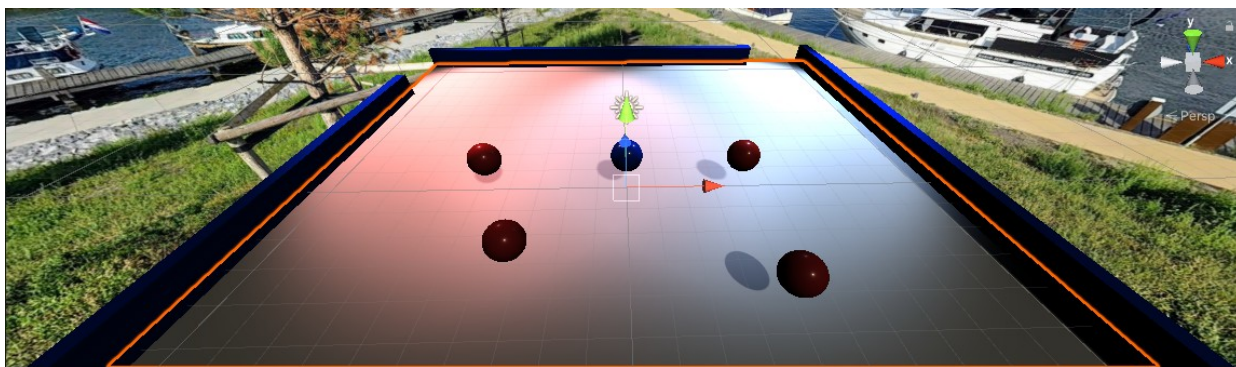
Do wykonania projektu konieczne jest zainstalowanie Unity3D, co wiąże się z utworzeniem konta, oraz dowolnego edytora kodu (IDE).

3. Ćwiczenie:

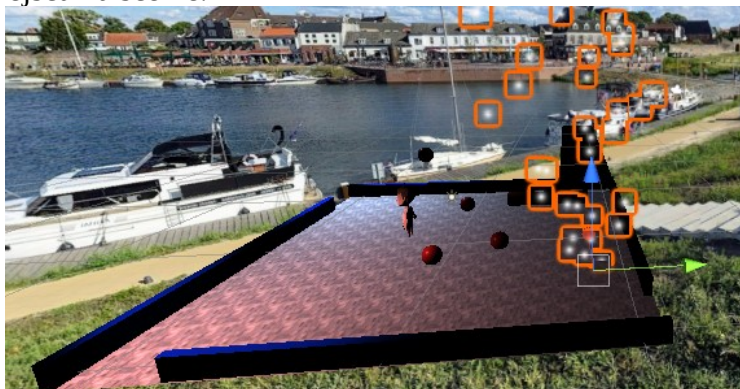
Zmodyfikować poprzedni projekt poprzez dodanie do sceny efektu ognia w oparciu o system cząsteczkowy.

4. Wykonanie zadania:

1. Uruchomić Unity3D.
2. Otworzyć projekt z poprzednich zajęć.



3. Dodać do sceny system cząsteczkowy (GameObject → Effects → Particle System) i ustawić w wybranym miejscu na scenie.

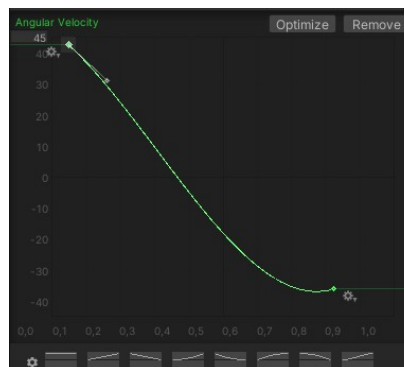


4. Poustawiać podstawowe parametry takie jak:
Emission → RateOverTime=20;
Shape → Shape=Cone;
Shape → Angle=0;
Shape → Radius=0.2;

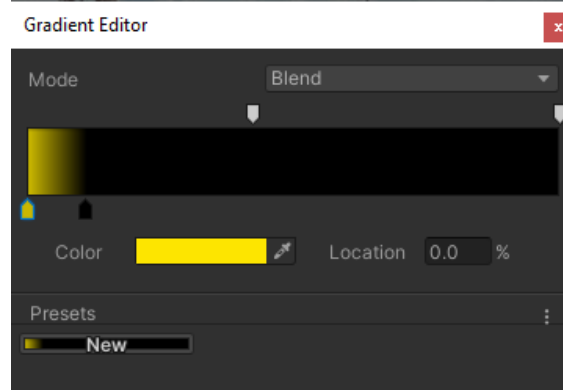
SizeOverLifeTime → Size → Krzywa;



RotationOverLifeTime → AngularVelocity → Krzywa



5. Utworzyć nowy materiał i przypisać dowolną teksturę ognia. (Create → Material).
6. Przełączyć shader na Mobile → Particles → Additive i przeciągnąć teksturę.
7. Ustawić ColorOverLifeTime → Color → Gradient



8. Dodanie światła do ognia:

1. Dodać światło i ustawić w miejscu systemu cząsteczkowego, ustawić jako child. GameObject → Light → PointLight
2. Dobrać wstępnie zasięg i intensywność. Wyłączyć DirectionalLight.



3. Utworzyć skrypt odpowiedzialny za zmianę intensywności światła w czasie. AddComponent → NewScript

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class blinking : MonoBehaviour
6 {
7     public Light myLight;
8     public float minWaitTime;
9     public float maxWaitTime;
10    public float minLightIntensity;
11    public float maxLightIntensity;
12
13    |
14    // Start is called before the first frame update
15    void Start()
16    {
17        StartCoroutine(Blinking());
18    }
19
20    IEnumerator Blinking()
21    {
22        while (true)
23        {
24            yield return new WaitForSeconds(Random.Range(minWaitTime,maxWaitTime));
25            myLight.intensity = Random.Range(minLightIntensity,maxLightIntensity);
26        }
27    }
28 }
29
30

```

9. Wykonać zamknięte pomieszczenie o dowolnych teksturach, umieścić wewnątrz 4 światła w postaci ognia (płonące beczki).

