

SPRAWOZDANIE

Zajęcia: Grafika i Multimedia

Prowadzący: prof. dr hab. inż. Vasyl Martsenyuk

Laboratorium Nr 3 Data 4.02.2021 Temat: "Silnik fizyczny" Wariant 3	Bartosz Jarosz Informatyka II stopień, stacjonarne, 1 semestr
--	--

1. Polecenie: wariant 1 zadania

Stworzyć piramidę z użyciem elementów podstawowych wg wariantu

3. cylinder

Ilość poziomów piramidy jest 10.

Piramida jest położona na powierzchni ziemi (obiekt Terrain) pokrytą górami. Użyć materiały dla powierzchni ziemi (paczka **Terrain Textures - Snow - Free Samples**)

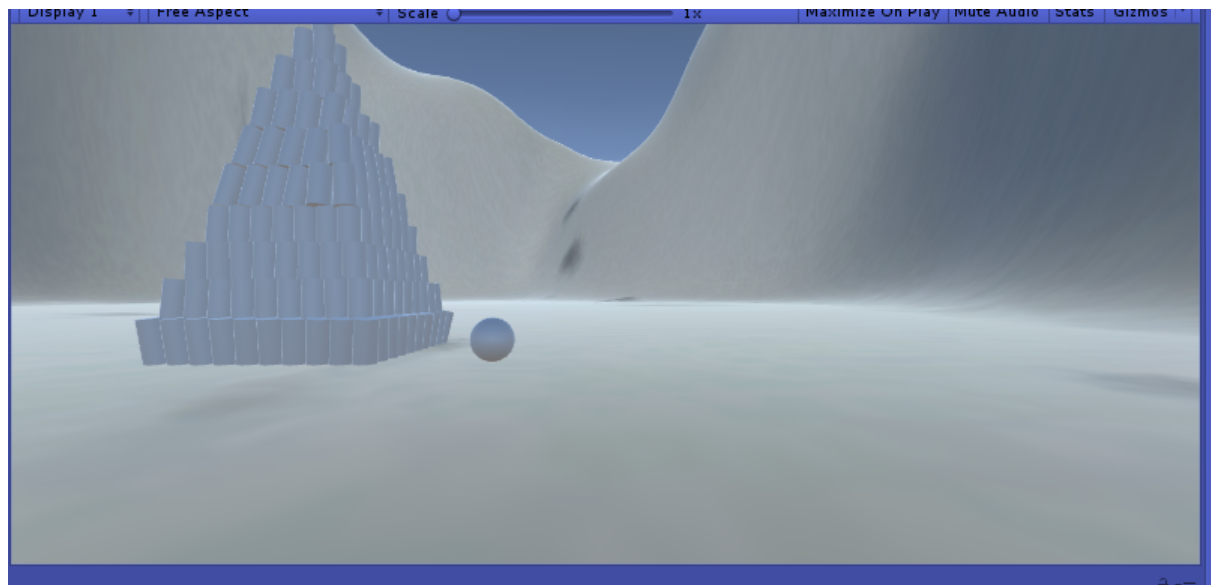
3. concrete

Połączyć elementy podstawowe piramidy sprężynami (Spring Joint)

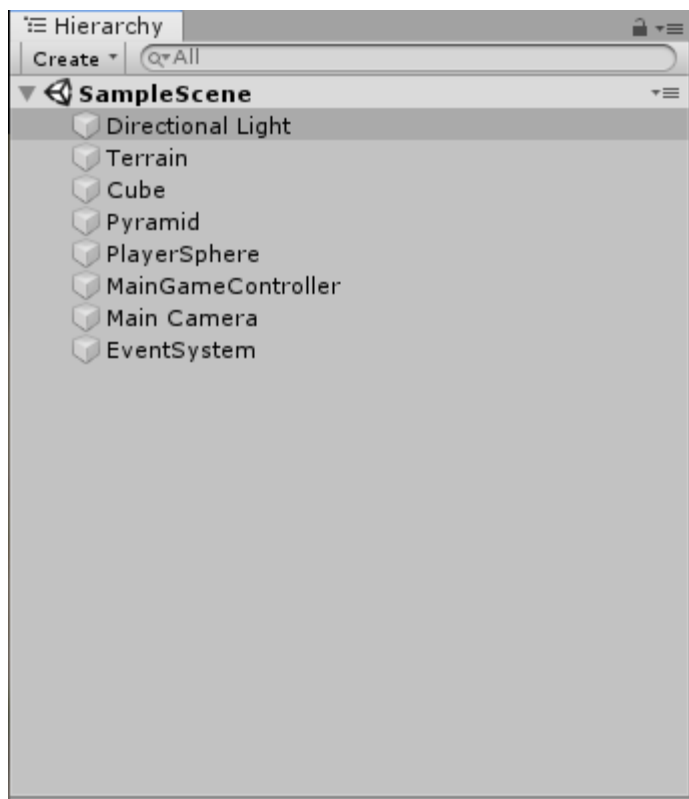
Opracować grę w której gracz za pomocą kuli chce rozbijać piramidę (używając Constant Force)

2. Opis programu opracowanego (kody źródłowe, rzuty ekranu)

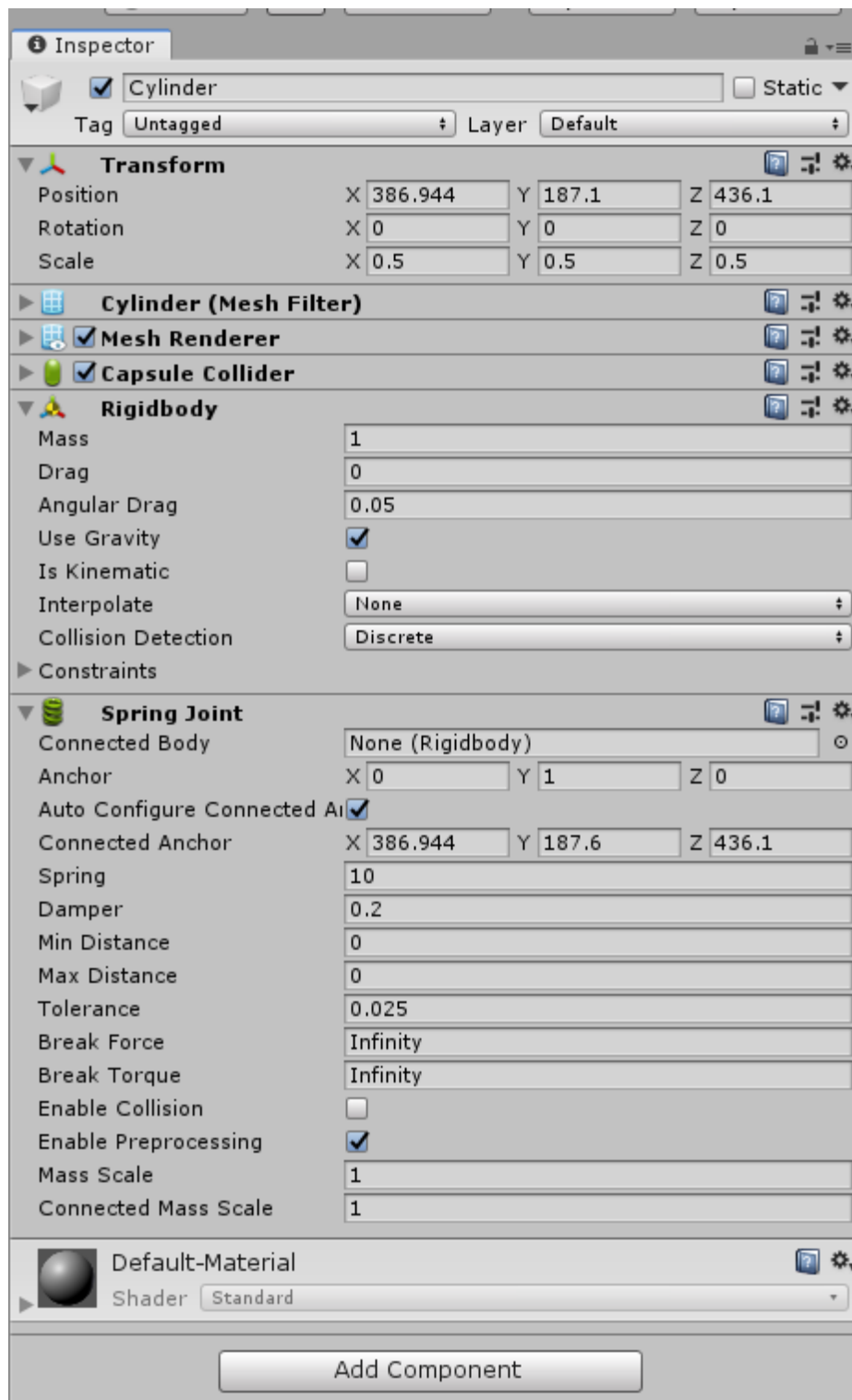
Główny widok sceny



Hierarchia sceny



Komponenty cylindra tworzącego piramidę



Fragmenty kodu:

4. Klasa MainGameController:

Listing 1: Pola

```
public GameObject PlayerSphere;  
public Camera MainCamera;  
  
private Vector3 _playerCameraForce;
```

Listing 2: Funkcja Start

```
void Start()  
{  
    _playerCameraForce  
        = PlayerSphere.GetComponent<ConstantForce>()  
            .force;  
    StartGame();  
}
```

Listing 3: Funkcja Update

```
void Update()  
{  
    MainCamera.transform.position  
        += _playerCameraForce;  
}
```

Listing 4: Funkcja StartGame

```
public void StartGame()  
{  
    var rigidBody = PlayerSphere  
        .GetComponent<Rigidbody>();  
  
    rigidBody.useGravity = true;  
  
    var constantForce = PlayerSphere  
        .GetComponent<ConstantForce>();  
  
    constantForce.force  
        = new Vector3(-10, 0, 10);  
}
```

5. Klasa PyramidGenerator:

Listing 5: Pola

```
public GameObject Objects;  
public GameObject Origin;
```

Listing 6: Funkeja StartGame

```

void Start()
{
    var objectScale = Objects.transform.localScale;
    var objectHeight = objectScale.y;
    var objectR = objectScale.x;
    var layerCount = 11;

    for (int k = 0; k < layerCount; k++)
    {
        var n = k;
        for (int i = 0; i < n; i++)
            for (int j = 0; j < n; j++)
                Instantiate(
                    Objects,
                    Origin.transform.position +
                        new Vector3(
                            objectR / 2
                            * (layerCount - k)
                            + i * objectR,

                            layerCount - k + objectR

                            objectR / 2 *
                            (layerCount - k)
                            + j * objectR),
                    Quaternion.identity
                );
    }
}

```