

# SPRAWOZDANIE

Zajęcia: Grafika i Multimedia

Prowadzący: prof. dr hab. inż. Vasyl Martsenyuk

Laboratorium Nr 1 Data 4.02.2021 <b>Temat:</b> "Modelowanie gry 3D. Roll and Ball" Wariant 3	Kamil Pająk Informatyka II stopień, stacjonarne, 1 semestr
---	---

## 1. Polecenie: wariant 2 zadania

Opracować grę typu Roll a Ball z takimi parametrami

Wariant 3. Kolor materialu „playera” – zielony,

obiekty „pick up” dwóch typów

1 typ obiektu „pick up” - capsule,

2 typ obiektu „pick up” - cylinder,

ilość obiektów „pick up” typu 1 - 7,

ilość obiektów „pick up” typu 2 - 5,

Kolor materialu obiektów „pick up” typu 1 – żółty,

Kolor materialu obiektów „pick up” typu 2 – brązowy,

Kolor materialu „ścian” - niebieski

Reguły gry:

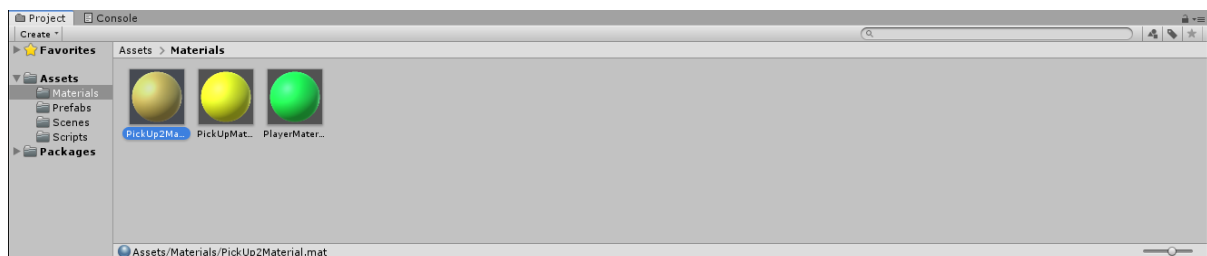
trafianie w obiekt typu 1 – 1 punkt

trafianie w obiekt typu 2 – 4 punkty

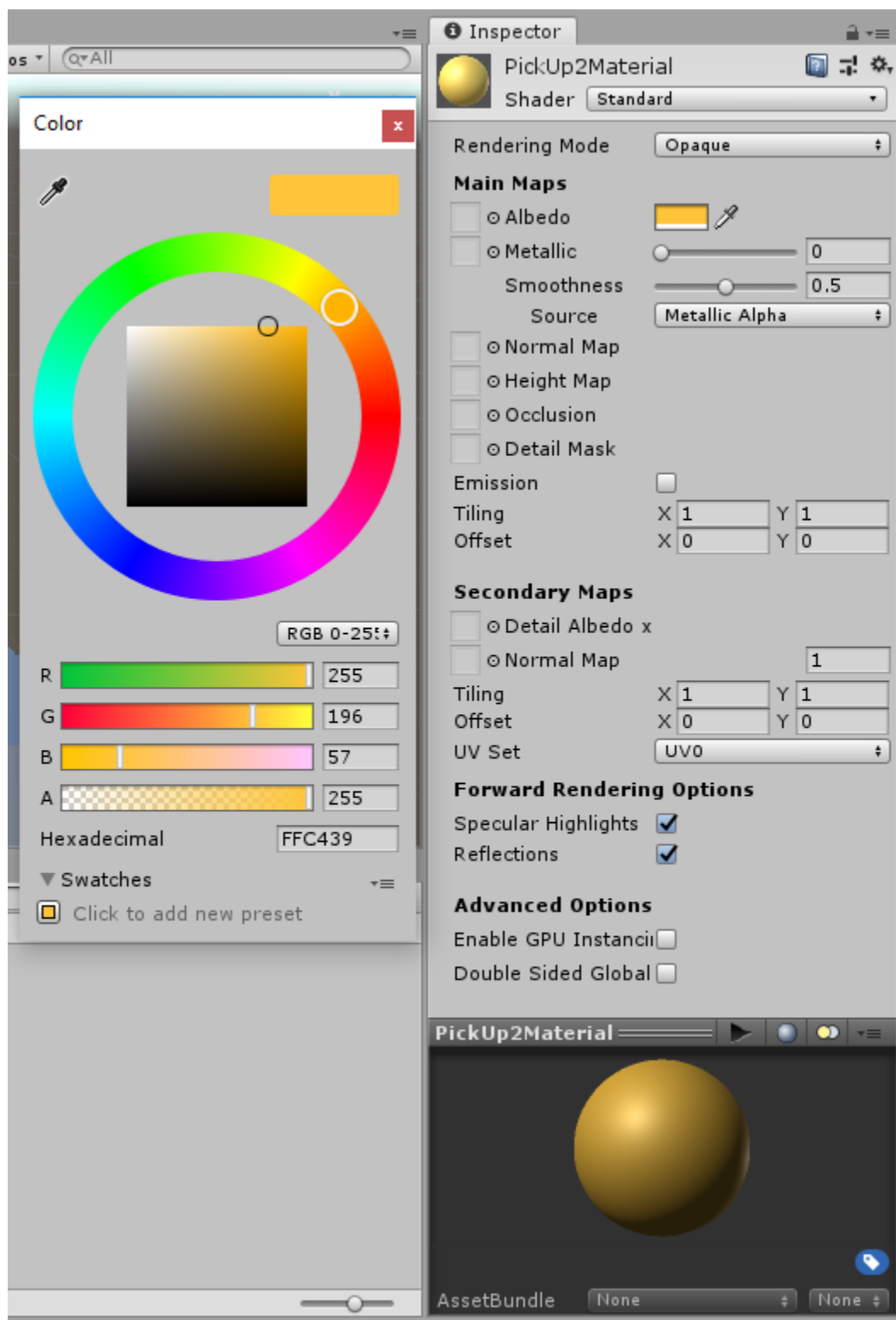
warunek zakończenia gry – 9 punktów

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

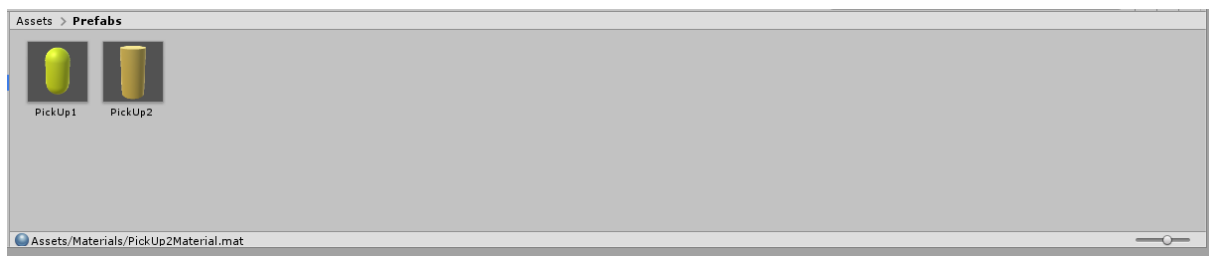
### Stworzenie materiałów obiektów



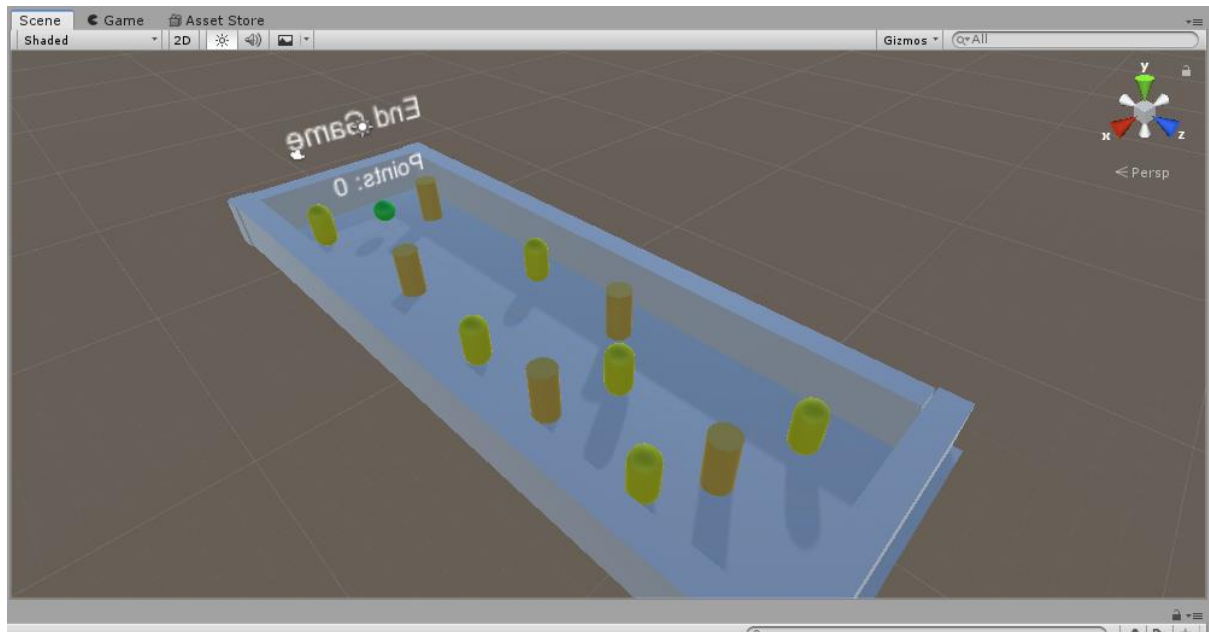
Nadanie koloru materiałom



## Stworzenie dwóch typów PickUp



## Stworzenie wymaganej liczby obiektów PickUp:



## Obiekty na drzewie obiektów



## Skrypt sterujący rozgrywką

Listing 1: Pola

```
public float Speed;
private int _points;

private const int PointPickUp1 = 1;
private const int PointPickUp2 = 4;

private bool _finishedGame;

private GameObject _endText;
private Rigidbody _rigidbody;
```

Listing 2: Funkcja Start

```
void Start()
{
    _points = 0;
    _endText = GameObject.Find("EndText");
    _endText.SetActive(false);
    _rigidbody = GetComponent<Rigidbody>();
}
```

Listing 3: Funkcja Update

```
void Update()
{
    if (_finishedGame)
    {
        _rigidbody.Sleep();
        return;
    }

    var moveHorizontal = Input.GetAxis("Horizontal");
    var moveVertical = Input.GetAxis("Vertical");

    var movement = new Vector3(moveHorizontal, 0.0f, moveVertical);
    _rigidbody.AddForce(movement * Speed * Time.deltaTime);
}
```

Listing 4: Funkcja OnTriggerEnter

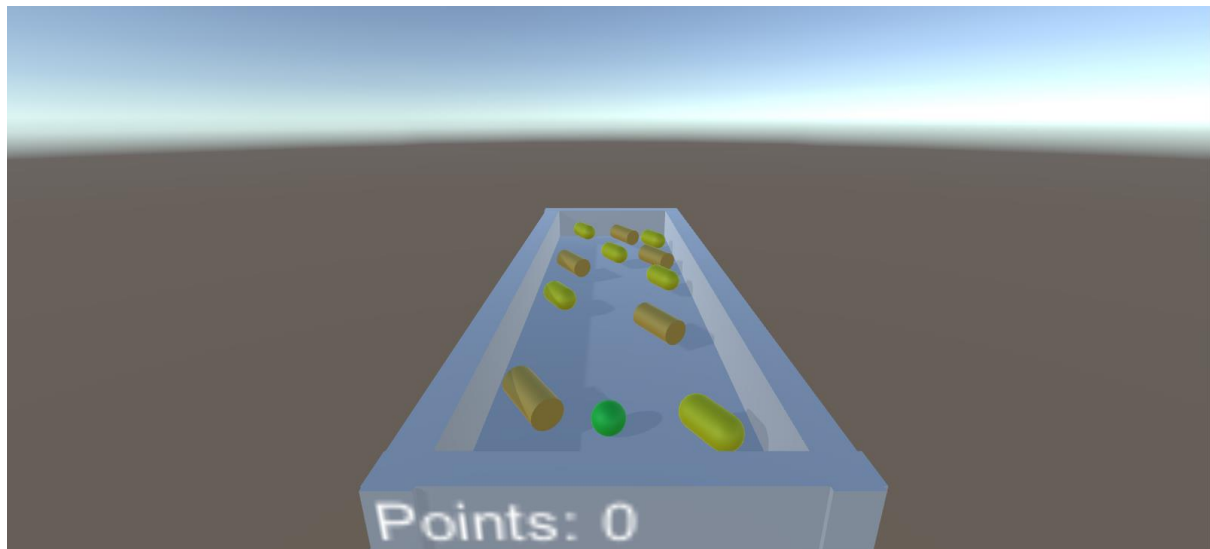
```
void OnTriggerEnter(Collider other)
{
    if (other.gameObject.tag.Equals("PickUp"))
    {
        other.gameObject.SetActive(false);
        _points += PointPickUp1;
    }

    if (other.gameObject.tag.Equals("PickUp2"))
    {
        other.gameObject.SetActive(false);
        _points += PointPickUp2;
    }

    if (_points > 9)
    {
        _finishedGame = true;
        _endText.SetActive(true);
    }

    var pointText = GameObject.Find("PointsText").GetComponent<TextMesh>();
    pointText.text = $"Points: {_points}";
}
```

Początek rozgrywki



Koniec rozgrywki



3. Wnioski:

Środowisko Unity pozwala w wygodny i szybki sposób tworzyć gry z wykorzystaniem języka C# w roli skryptów oraz gotowych elementów takich jak kształty geometryczne itp.