

Zajęcia 1. Modelowanie gry komputerowej 3D

Łukasz Hamera
University of Bielsko-Biala
lukasz.hamera@evatronix.com

20 marca 2019

Streszczenie

1 Treść zadania

Cele:

- Stworzyć grę Roll a Ball która wykorzystuje fizykę
- Realizować wejście z klawiatury
- Kolizje między obiektami
- Brak multimedia

Wariant 3

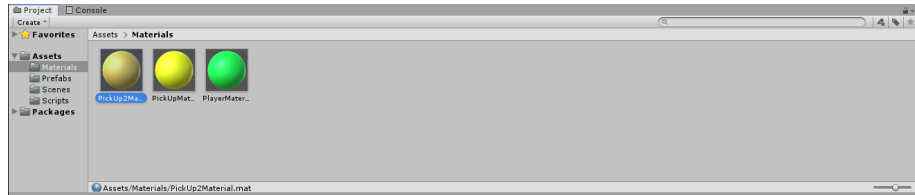
- Kolor materiału „playera” – zielony,
- obiekty „pick up” dwóch typów
- 1 typ obiektu „pick up” - capsule,
- 2 typ obiektu „pick up” - cylinder,
- ilość obiektów „pick up” typu 1 - 7,
- ilość obiektów „pick up” typu 2 - 5,
- Kolor materiału obiektów „pick up” typu 1 – żółty,
- Kolor materiału obiektów „pick up” typu 2 – brązowy,
- Kolor materiału „ścian” - niebieski

Reguły gry:

- trafiać w obiekt typu 1 – 1 punkt
- trafiać w obiekt typu 2 – 4 punkty
- warunek zakończenia gry – 9 punktów

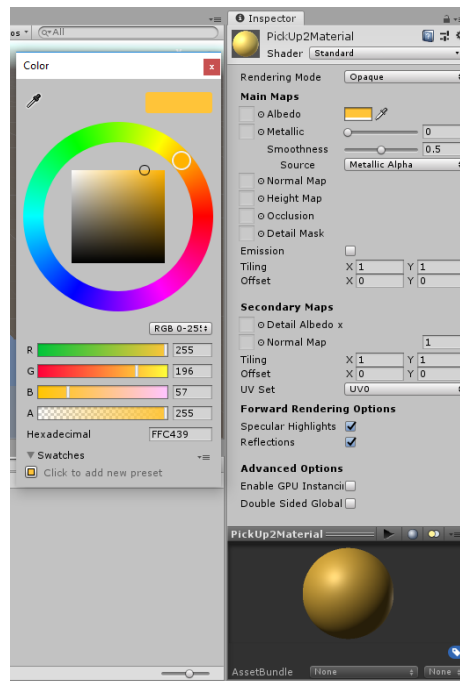
2 Rozwiązanie zadania

1. Stworzenie materiałów obiektów:



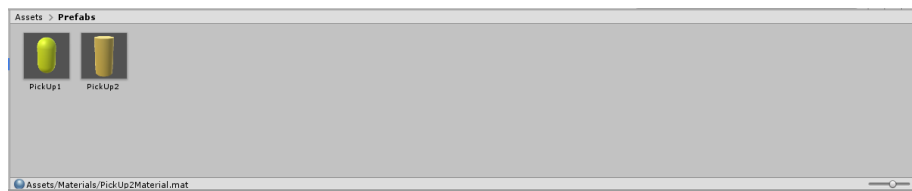
Rysunek 1: Materiały obiektów.

2. Nadanie koloru materiałom:



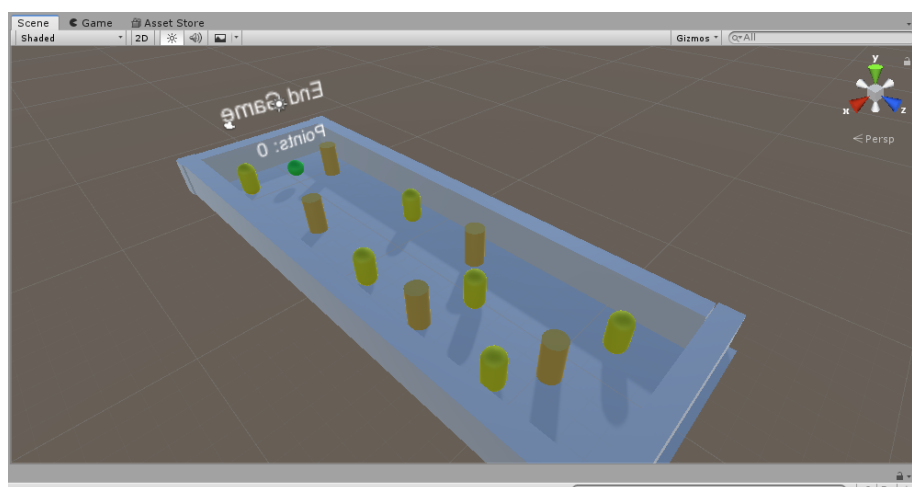
Rysunek 2: Zmiana koloru materiału.

3. Stworzenie dwóch typów PickUp:

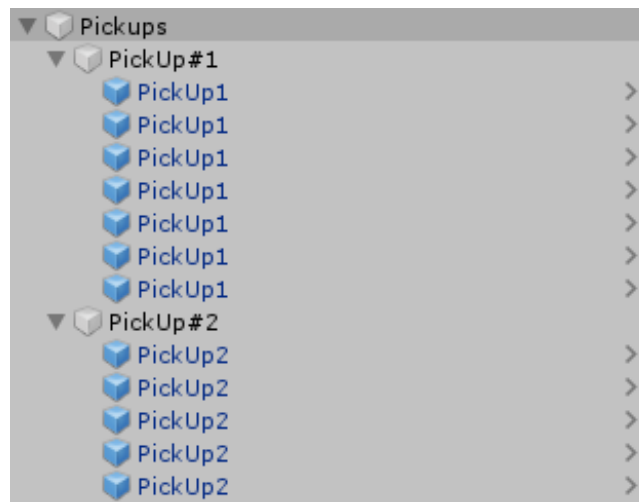


Rysunek 3: Dwa rodzaje obiektu PickUp.

4. Stworzenie wymaganej liczby obiektów PickUp:



Rysunek 4: Rozmieszczenie obiektów PickUp na mapie.



Rysunek 5: Obiekty PickUp na drzewie obiektów.

5. Skrypt sterujący rozgrywkę:

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: Funkcija 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: Funkcija OnTriggerEnter

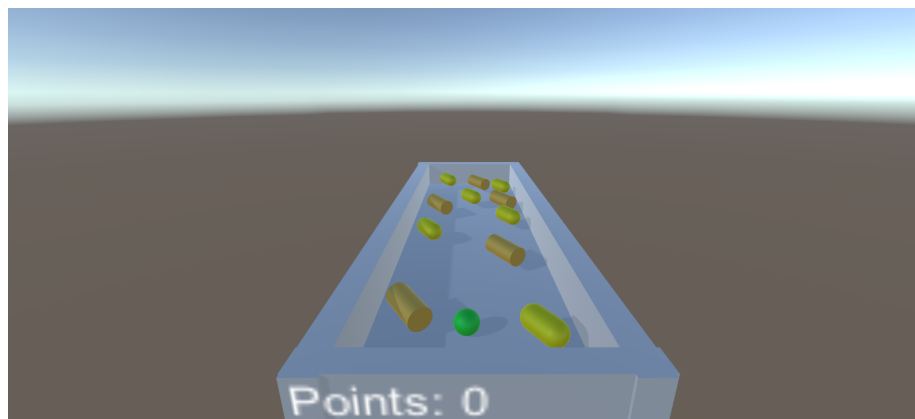
```
void OnTriggerEnter(Collider other)
{
    if (other.gameObject.tag.Equals("PickUp"))
    {
        other.gameObject.SetActive(false);
        _points += PointPickUp1;
    }

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

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

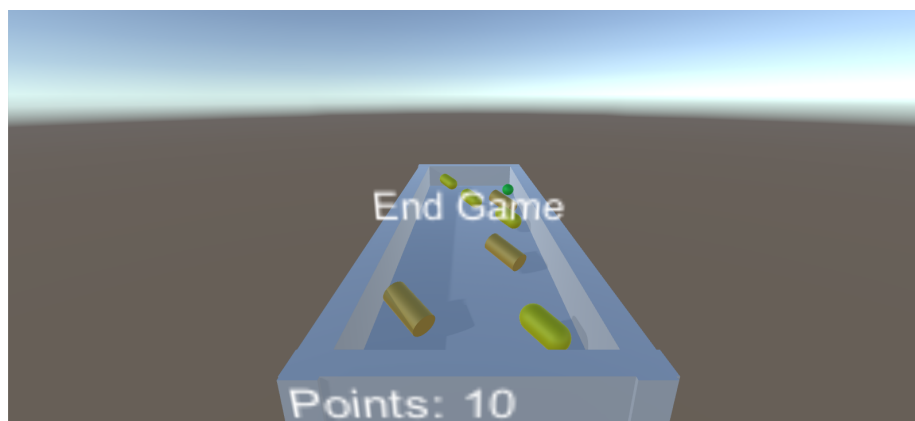
    var pointText = GameObject.Find("PointsText").GetComponent<TextMesh>();
    pointText.text = $"Points: {_points}";
}
```

6. Początek rozgrywki:



Rysunek 6: Widok z początku rozgrywki.

7. Koniec rozgrywki:



Rysunek 7: Widok z końca rozgrywki.