

## 1. Cel i wariant zadania:

Opracować grę używając ThirdPersonController jako gracza oraz Terrain jako podstawę.

Celem jest zdobycie punktów przy trafianiu gracza w obiekty-"triggery" (zawierają w sobie triggery)

Przy trafianiu gracza w obiekt-"trigger" odbywa się (wg wariantu)

### 1. Zmiana koloru światła

2. Nagrywanie dźwięku

3. Animacja

Obiekty mają takie postaci (wg wariantu)

### 1. Sześćcian

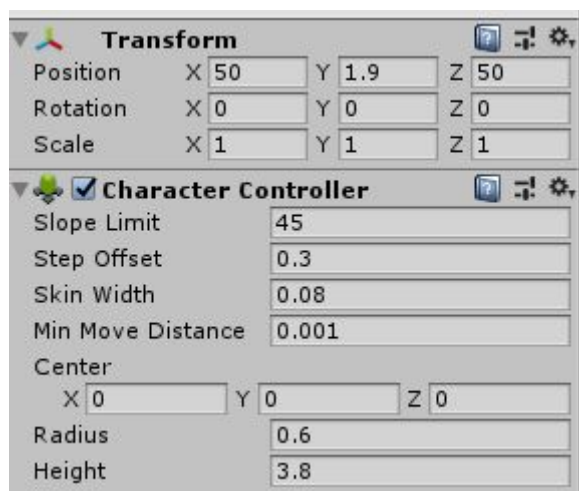
2. Kula

3. Kapsuła

## 2. Przebieg ćwiczenia:

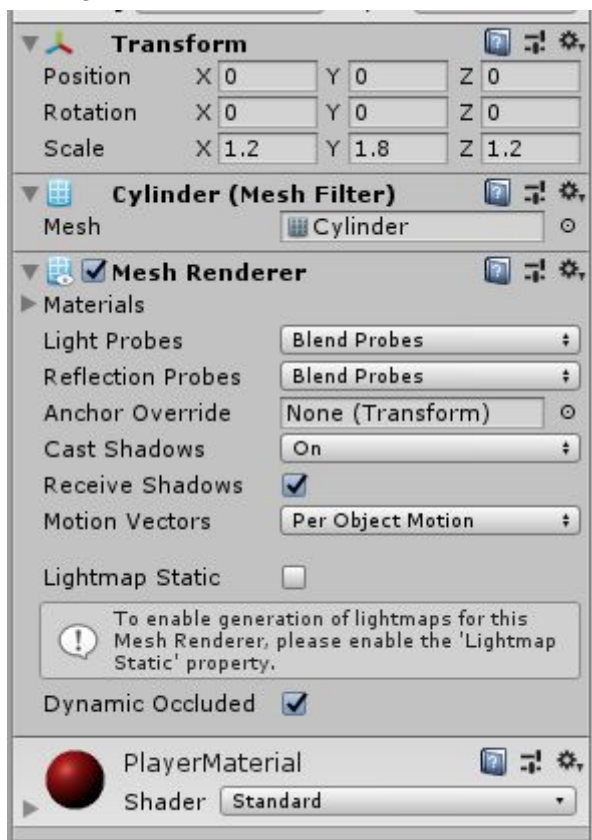
Po dodaniu obszaru pracy (terrain), należy utworzyć gracza.

W tym celu tworzymy pusty obiekt (Empty Object), do którego dodajemy "Character Controller".



W tym momencie posiadamy gracza, który nie ma ciała, w tym celu tworzymy ciało Playera (w przypadku niniejszego zadania jest to po prostu Cylinder) oraz dodajemy do gracza kamerę (przesuwamy "Main Camera" na nasz "Character Controller" lub dodajemy nową kamerę).

Ciało gracza:



Zgodnie z wariantem zadania, będziemy operować na świetle, więc światło również zostało dodane do gracza.

Kolejnym krokiem jest umożliwienie obrotu postaci w tym celu do Kamery dodajemy skrypt:

```
public class MouseLook : MonoBehaviour
{
    public float mouseSensitivity = 100f;
    public Transform playerBody;

    float XRotation = 0;
    // Start is called before the first frame update
    void Start()
    {
        Cursor.lockState = CursorLockMode.Locked;
    }

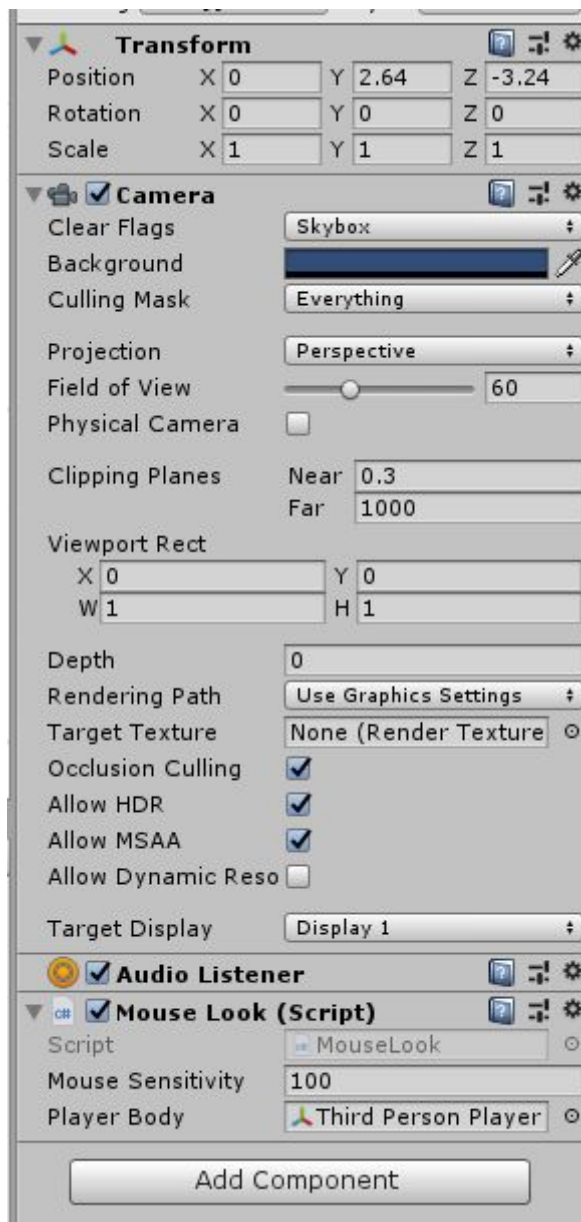
    // Update is called once per frame
    void Update()
    {
        float mouseX = Input.GetAxis("Mouse X") * mouseSensitivity * Time.deltaTime;
        float mouseY = Input.GetAxis("Mouse Y") * mouseSensitivity * Time.deltaTime;

        XRotation -= mouseY;
        XRotation = Mathf.Clamp(XRotation, -90f, 90f);

        transform.localRotation = Quaternion.Euler(XRotation, 0f, 0f);
        playerBody.Rotate(Vector3.up * mouseX);
    }
}
```

Gracz nie może obrócić kamery o więcej niż 90 stopni, na starcie kursor jest blokowany w obrębie okna.

Aby skrypt działał należy po prostu dodać go do kamery oraz ustawić parametr "Player Body" na naszego "Third Person Playera" czyli Character Controller.



Następnie zajmiemy się poruszaniem gracza, w tym celu dodajemy skrypt do "Third Person Playera" czyli Character Controller. Skrypt:

```
public class PlayerMovement : MonoBehaviour
{
    public CharacterController controller;

    public float speed = 12f;

    // Update is called once per frame
    void Update()
    {
        float x = Input.GetAxis("Horizontal");
        float z = Input.GetAxis("Vertical");

        Vector3 move = transform.right * x + transform.forward * z;

        controller.Move(move * speed * Time.deltaTime);
    }
}
```



W tym momencie posiadamy gracza, który może się poruszać oraz obracać wokół osi.

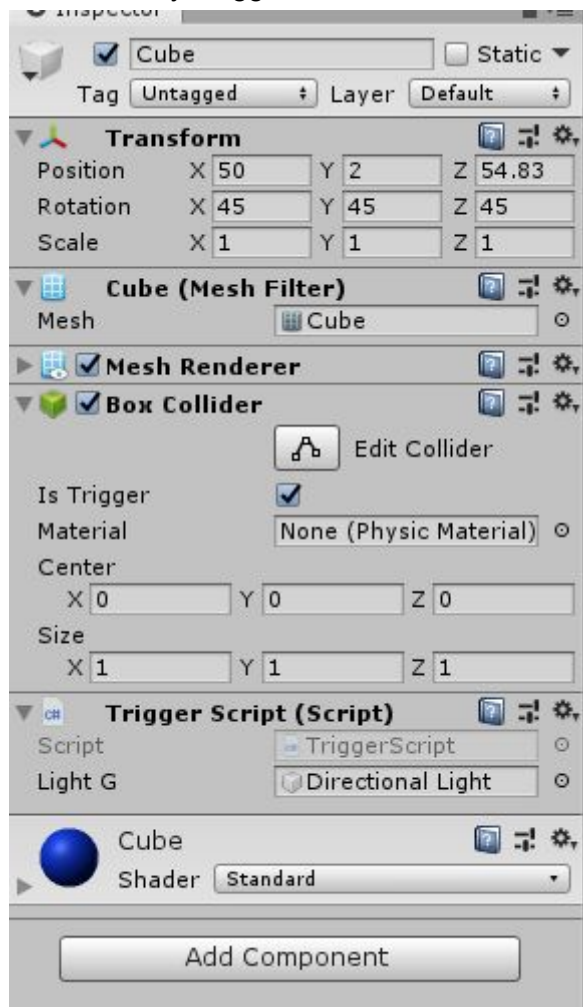
Kolejnym elementem jest stworzenie obiektu Cube (Sześcian zgodnie z wariantem zadania) oraz zrobienie z niego Triggeru, który zmienia kolor światła, gdy Player go dotknie.

W tym celu tworzymy Cube oraz dodajemy skrypt:

```
public class TriggerScript : MonoBehaviour
{
    public GameObject LightG;

    void OnTriggerEnter(Collider other)
    {
        if (other.GetComponent<Collider>().CompareTag("Player"))
        {
            LightG.GetComponent<Light>().color = Color.red;
        }
    }
}
```

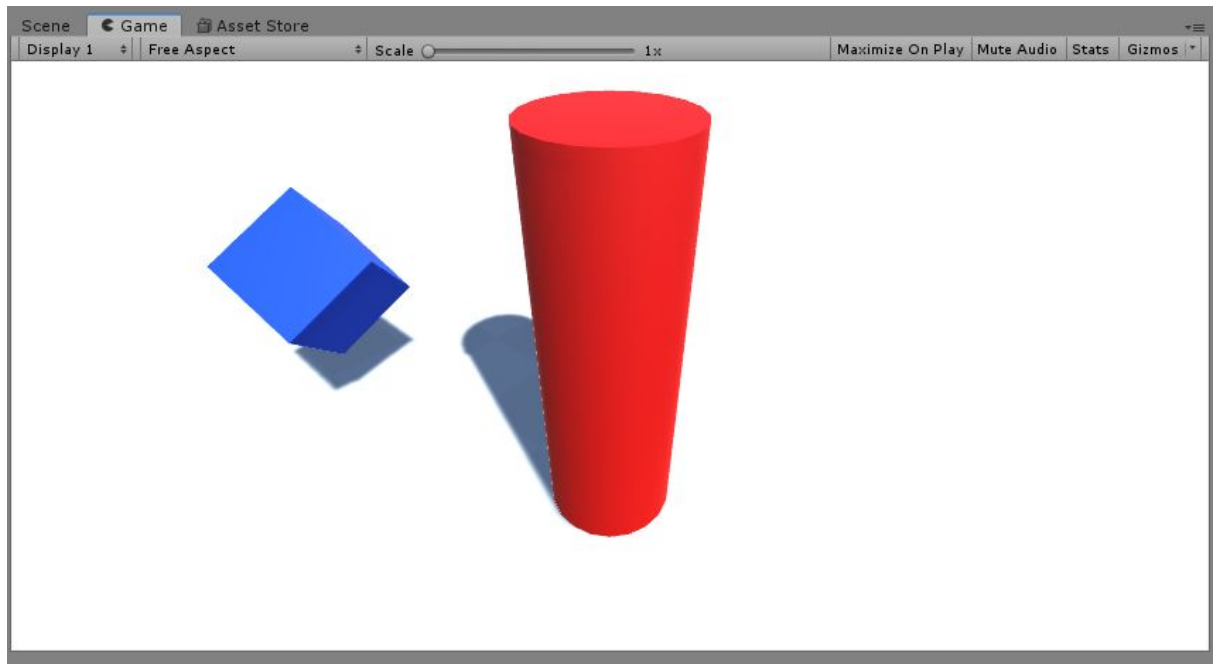
Cube musi być triggerem:



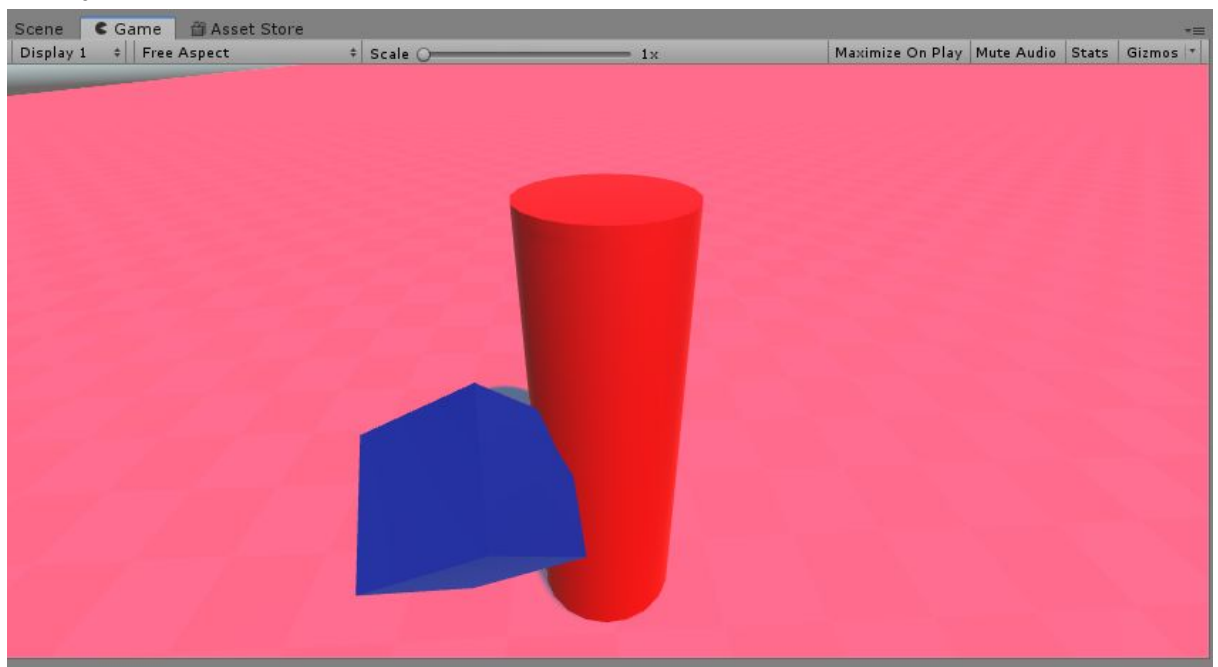
Ważne jest aby gracz posiadał Tag "Player".

Teraz gdy uruchomimy grę światło jest białe, lecz po wejściu na obiekt światło zmieni kolor na czerwony.

Przed wejściem:



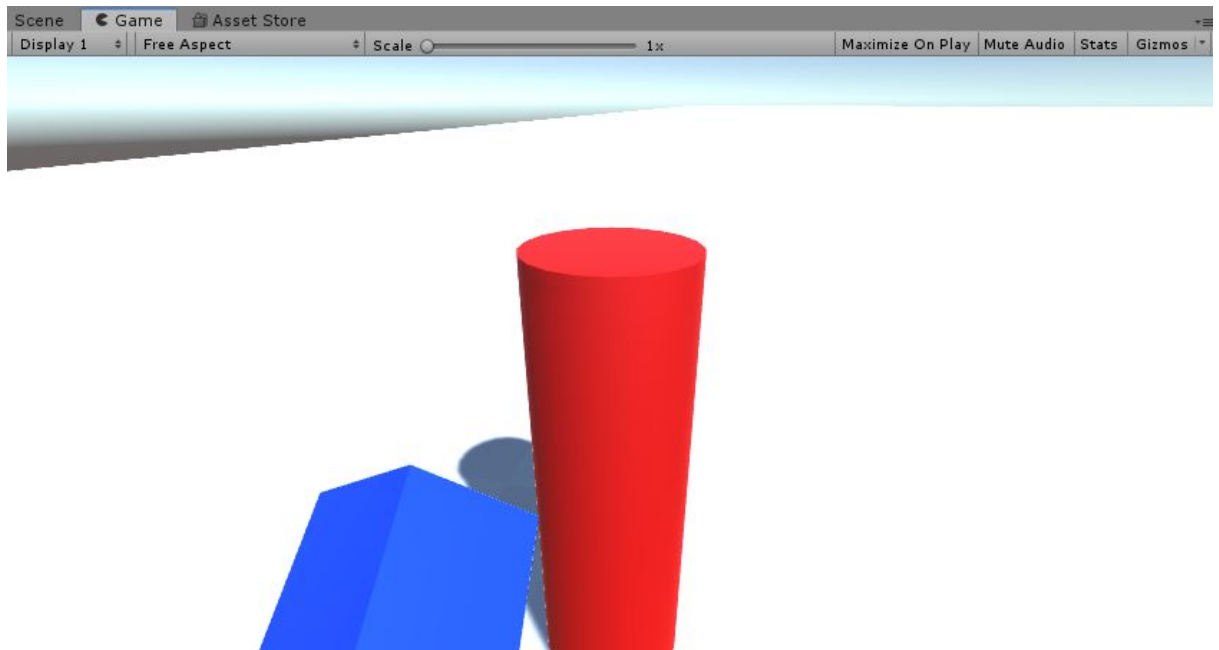
Po wejściu:



Następnie dodajemy skrypt, aby po zejściu z obiektu kolor powrócił na biały:

```
void OnTriggerExit(Collider other)
{
    if (other.GetComponent<Collider>().CompareTag("Player"))
    {
        LightG.GetComponent<Light>().color = Color.white;
    }
}
```

Po zejściu z obiektu:





### **3. Wnioski:**

Napisanie prostej gry Third Person, w Unity nie jest zadaniem skomplikowanym. Dzięki gotowym komponentom (takim jak Character Controller) programista oszczędza dużo czasu. komponenty tego typu posiadają np. wbudowane Collidery. Operacje na świetle za pomocą triggerów jest stosunkowo proste i przyjazne dla programisty. Dzięki funkcją OnTriggerEnter oraz OnTriggerExit, w łatwy sposób można utworzyć logikę gry oraz podstawowe zasady gry.