

1. Cel ćwiczenia:

Opracować grę używając ThirdPersonController jako gracza oraz Terrain jako podstawę. Celem jest zdobycie punktów w wyniku otwarcia ukrytych drzwi w ścianie. Przy trafianiu gracza w obiekt-"trigger" drzwi otwierają się w sposób (wg wariantu)

1. Obrót wokół osi OY
2. Przesunięcie wzdłuż ściany
3. Opuszczanie do dołu

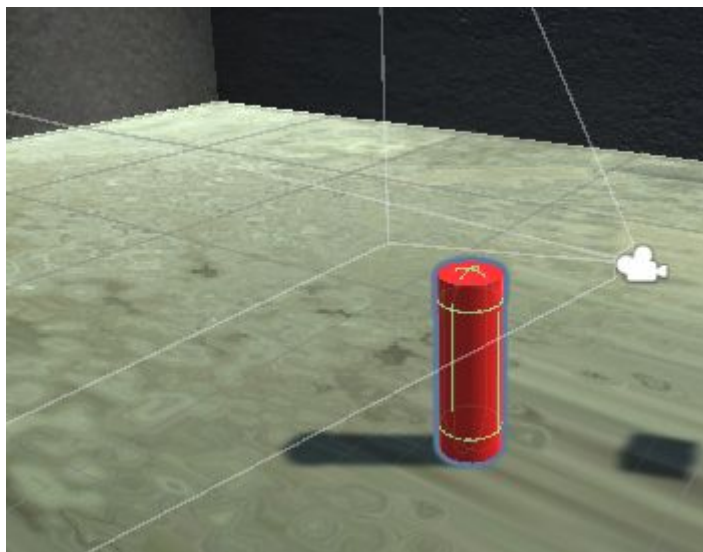
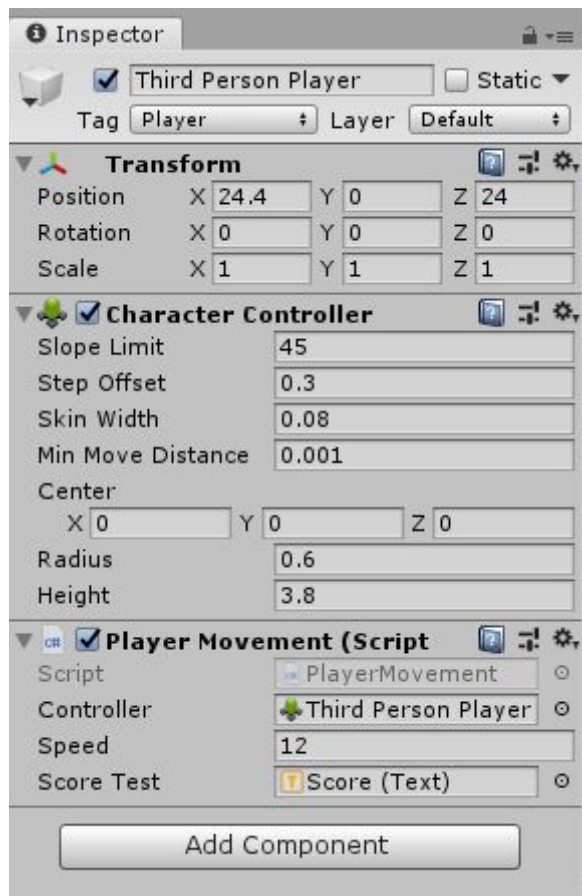
Zamykanie odbywa się automatycznie w sposób odpowiedni po wyjściu gracza przez drzwi.

Triggery dla otwarcia drzwi mają takie postaci (wg wariantu)

1. Sześcián
2. Kula
3. Kapsuła

2. Przebieg ćwiczenia:

Jako model gracza, zostanie użyty "Player" z poprzedniego ćwiczenia.



Kod gracza:

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

    public float speed = 12f;
    public Text ScoreText;
    private int score;

    // 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;

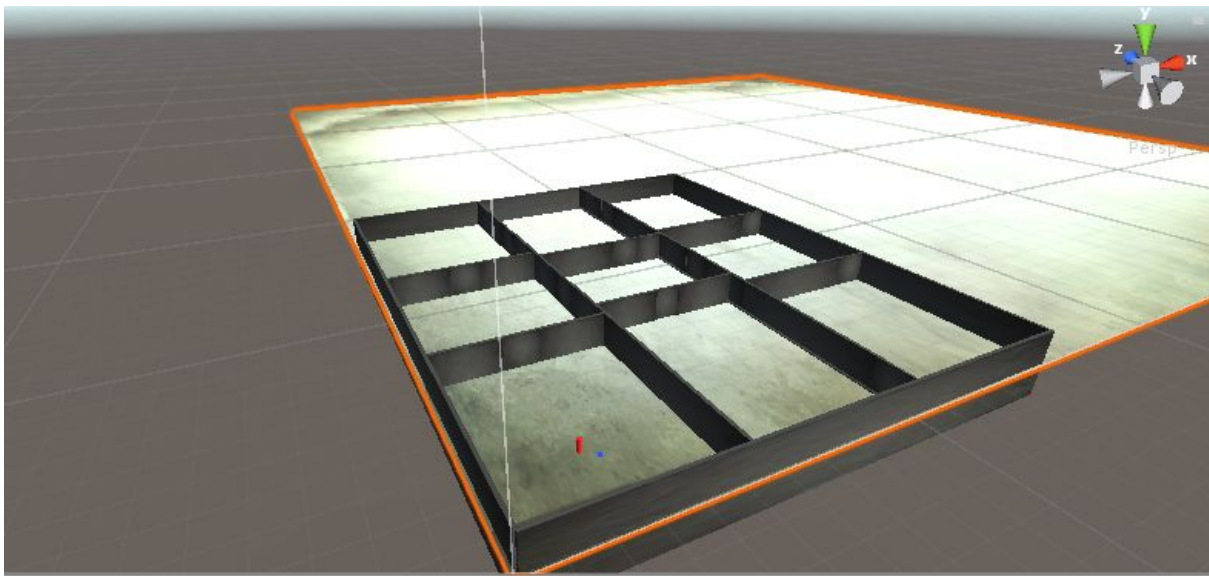
        controller.Move(move * speed * Time.deltaTime);
    }

    private void OnTriggerEnter(Collider other)
    {
        score += 1;
        SetScoreText();
    }

    void SetScoreText()
    {
        ScoreText.text = "Aktualny wynik: " + score;
    }
}
```

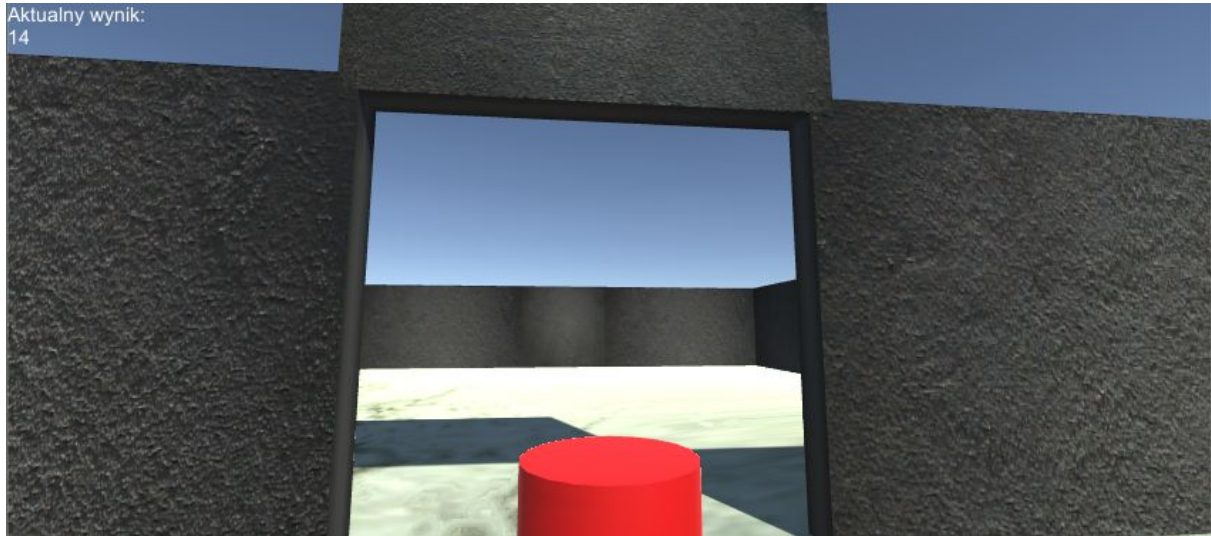
Kod ten umożliwia graczowi poruszanie się oraz zliczanie punktów poprzez wchodzenie na triggery. SetScoreText uaktualnia wynik zdobyty przez gracza.

Mapa świata:



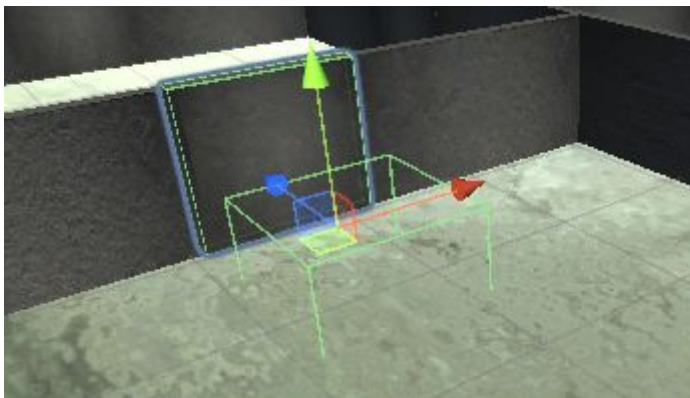
Świat składa się z 9 pokoi, między którymi znajdują się ruchome ściany. Ściany mogą zostać otwarte tylko z jednej strony. Oznacza to, że gracz nie może używać cały czas tego samego przejścia.

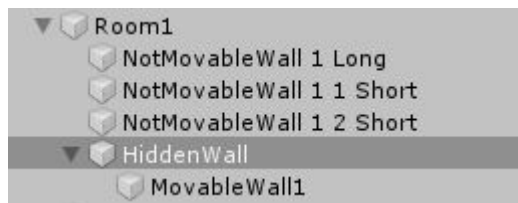
Przykładowe otwarcie drzwi:



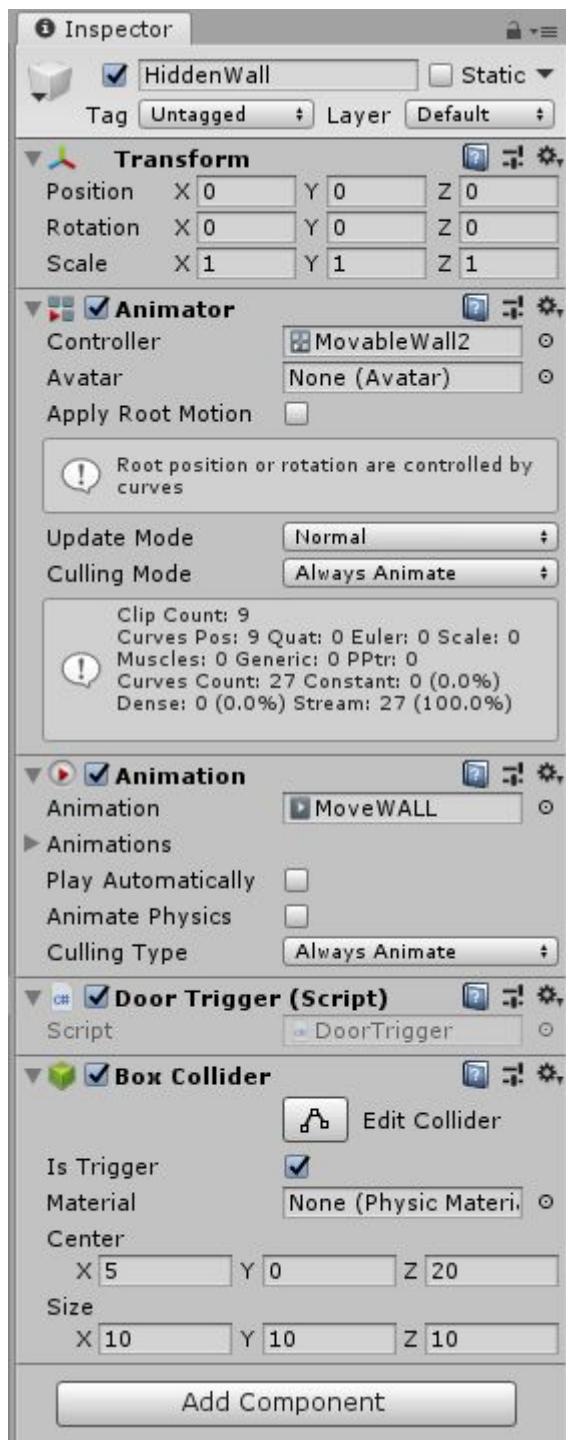
Zgodnie z wariantem zadania, drzwi wykorzystują oś Y w celu otwarcia.

Aby animacja była możliwa, wykorzystany został prostopadłościan, którego zadaniem jest wywołanie triggeru:

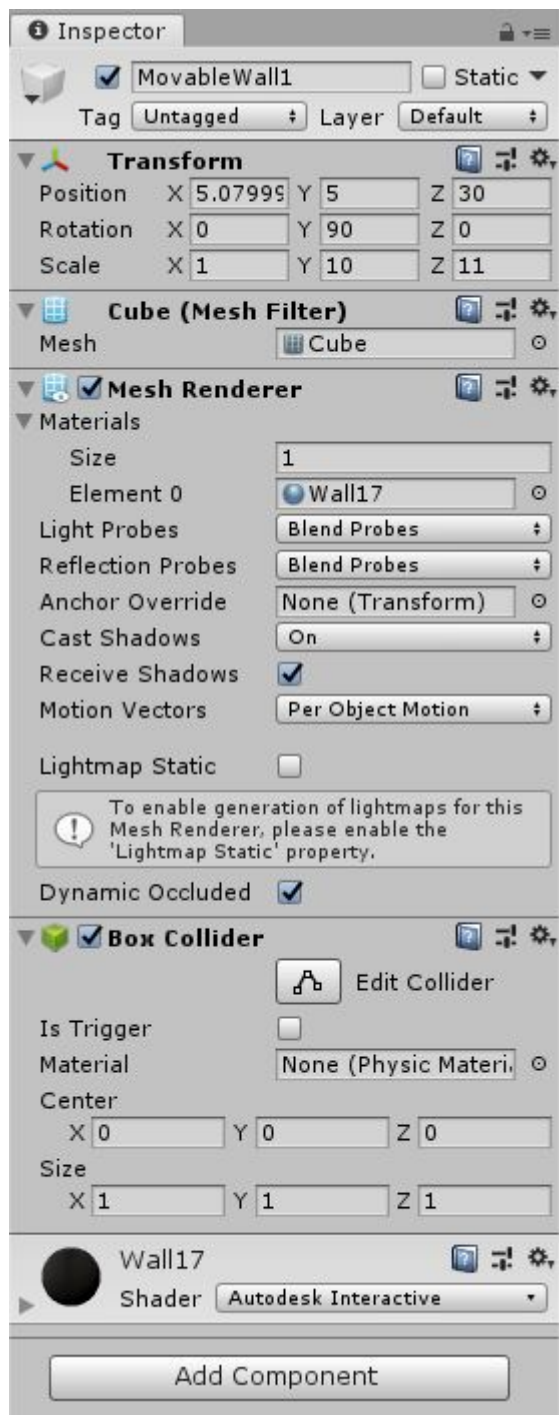




Pokój startowy posiada 1 drzwi oraz 1 ścianę, którą sąsiaduje z innym pokojem niepołączonym. HiddenWall w inspektorze:



Drzwi w inspektorze:



Po wejściu na trigger, zostaje wywołany odpowiedni kod:

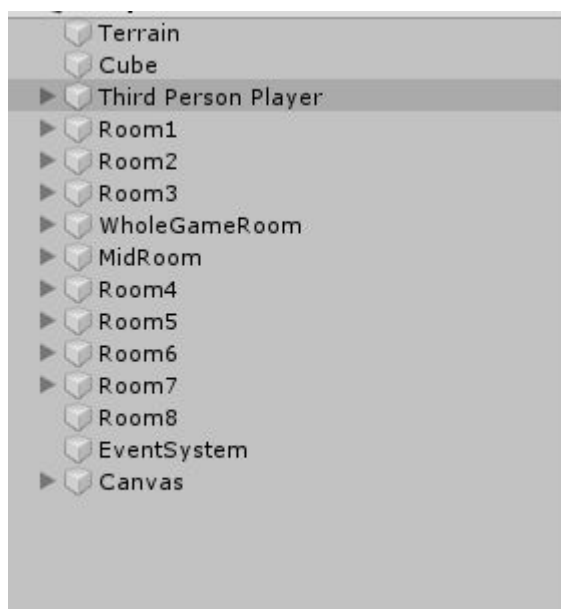
```
0 references
public class DoorTrigger : MonoBehaviour
{
    Animator animator;
    0 references
    void Start()
    {
        animator = GetComponent<Animator>();
    }

    // Update is called once per frame
    0 references
    private void OnTriggerEnter(Collider other)
    {
        if (other.tag == "Player")
        {
            animator.SetTrigger("MoveWallNow");
        }
    }
}
```

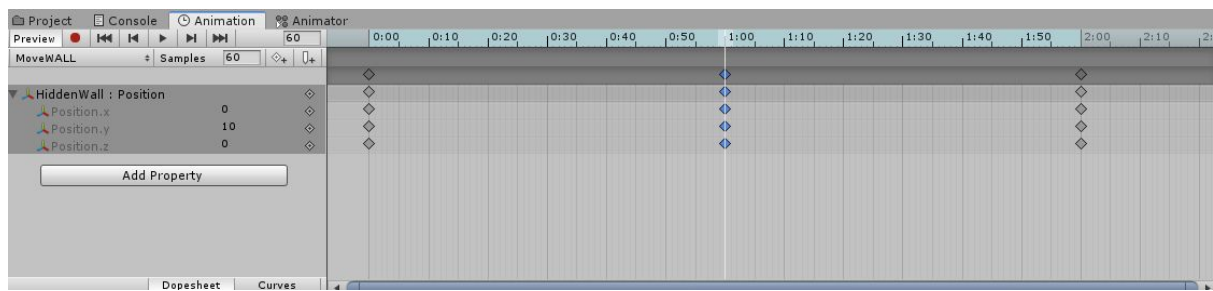
Zrzut z Animatora:



Struktura projektu:



Przykładowa animacja:



3. Wnioski:

Unity umożliwia tworzenie prostych gier wykorzystując tylko wiedzę związaną z “triggerami”. Dzięki wspomnianej funkcji, możliwe jest napisanie eventów w grze, zaczynając od prostego liczenia punktów, poprzez otwieranie drzwi lub dezaktywacji danych obiektów. Tworzenie prostych animacji jest intuicyjne i nie jest skomplikowane. Dzięki graficznemu edytorowi, programista nie musi pisać transformacji pozycji lub różnego rodzaju rotacji w kodzie. Zrobi to za niego Unity.