

1. Cel ćwiczenia:

Program 1. Opracować grę zajęcia 5 używając **Male3_ShortHair** jako gracza

Oprócz tego opracować takie animacji używane przez przyciski klawiatury (wg wariantu)

1. Obroty rąk

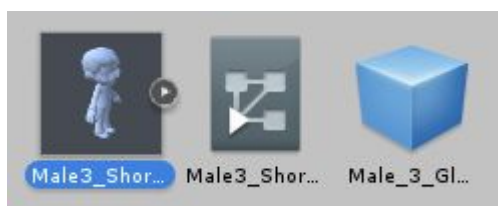
2. Obroty głowy

3. Opuszczanie do dołu

2. Przebieg ćwiczenia:

W celu wykonania ćwiczenia wymagane jest zaimportowanie paczki Male3.

W tym celu należy dodać folder do projektu.



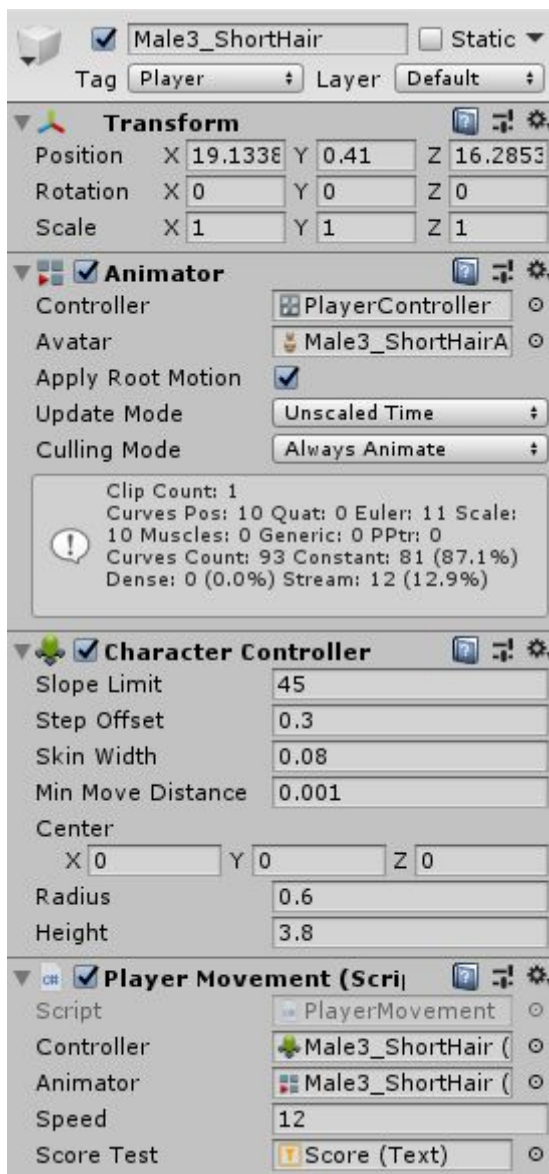
Następnie należy umieścić nowy element na scenie.



Przenosimy “Camere” tak aby była ona childem dla Male3, który od tej pory będzie miał Tag “Player”.



Kolejna część to dodanie “Character Controller” do Playera, ustawienie odpowiedniego Tagu oraz dodania skryptów poruszania Kamera.



Skrypt poruszania kamerą oraz obrotu gracza:

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

    float XRotation = 0;
    float YRotation = 0;

    0 references
    void Start()
    {
        Cursor.lockState = CursorLockMode.Locked;
    }

    0 references
    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);

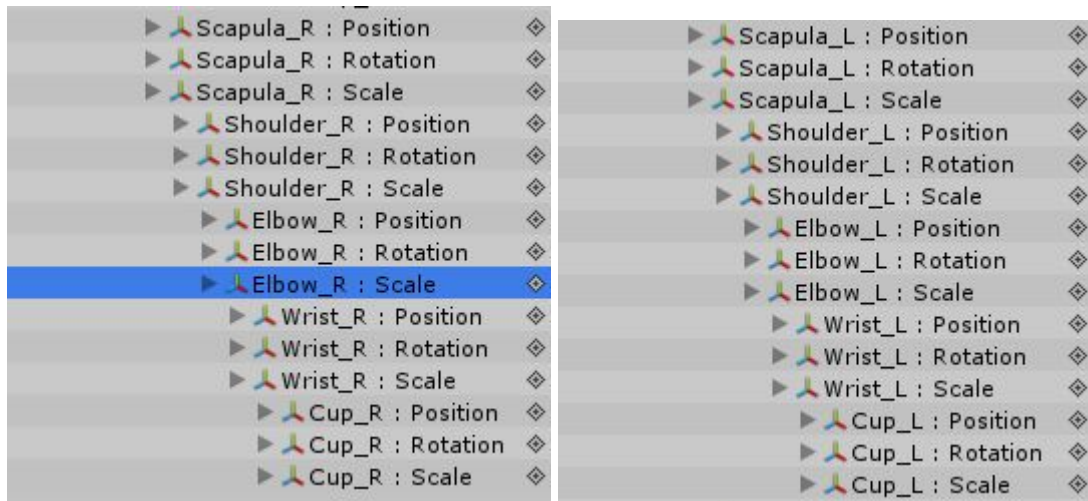
        transform.Rotate(Vector3.up * mouseY, Time.deltaTime * 90f);
        transform.localRotation = Quaternion.Euler(XRotation, 0f, 0f);
        playerBody.Rotate(Vector3.up * mouseX);
    }
}
```

Za pomocą `Mathf.Clamp`, blokujemy możliwość obrotu gracza do 90 stopni.

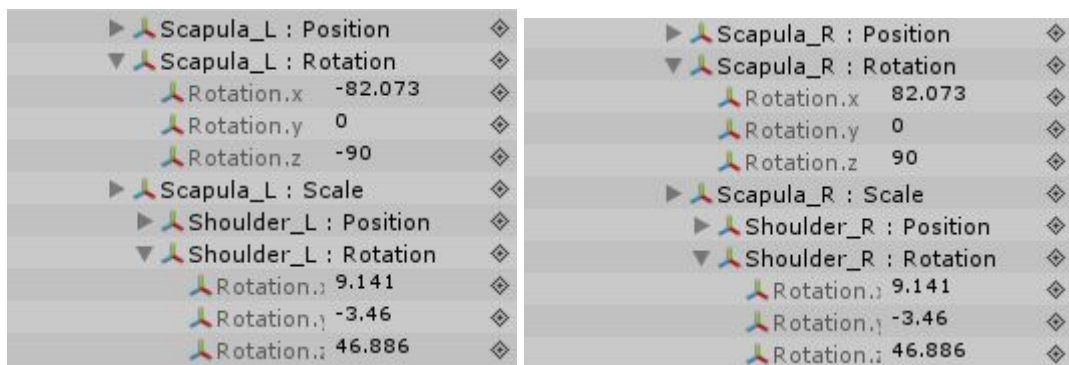
Kolejna część ćwiczenia dotyczy tworzenia animacji.

W tym celu należy otworzyć menu animacji. Wariantem zadania jest Obrót rąk. W tym celu w modelu postaci trzeba znaleźć elementy odpowiedzialne za pracę rąk.

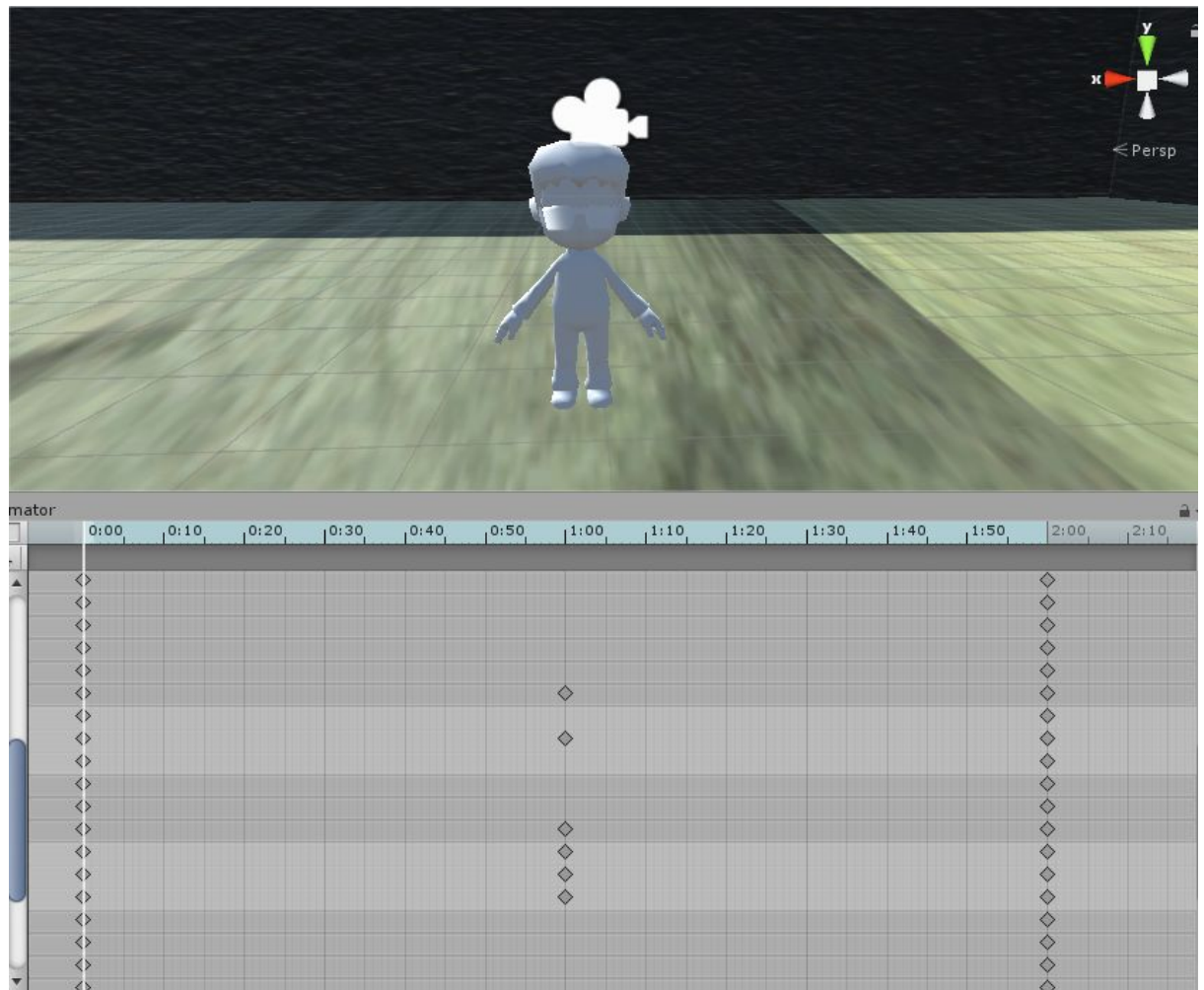
Będzie to łopátka prawa i lewa:



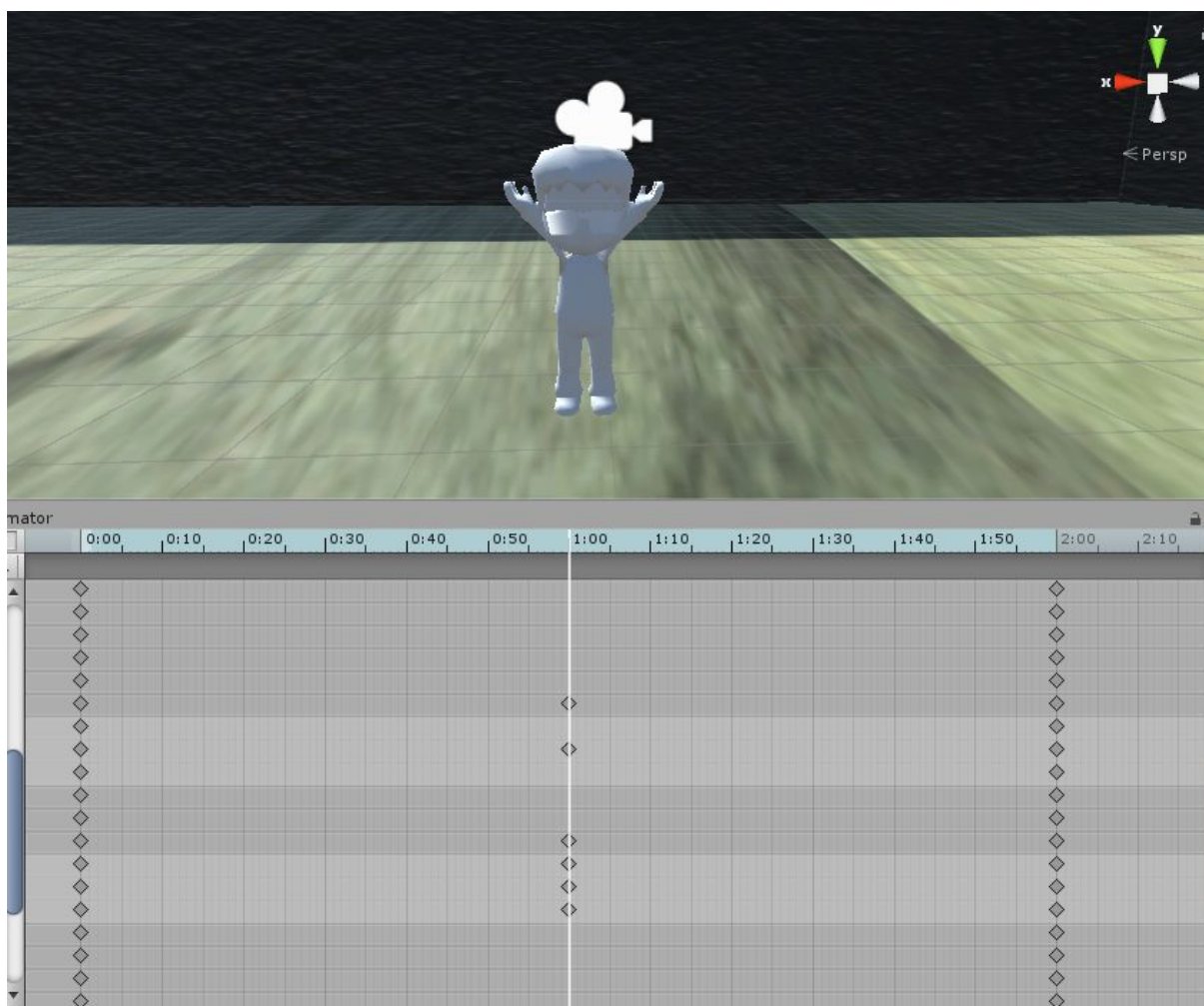
Następnie należy dodać właściwość Transform, gdzie zmianie będzie podlegać parametr "Rotation".



Animacja:



Pozycja startowa. Ręce opuszczone w dół. Animacja na początku (0:00) oraz po 2 sekundach (2:00) jest taka sama.



Pozycja w trakcie animacji (1:00). Ręce podniesione do góry, dłonie obrócone.

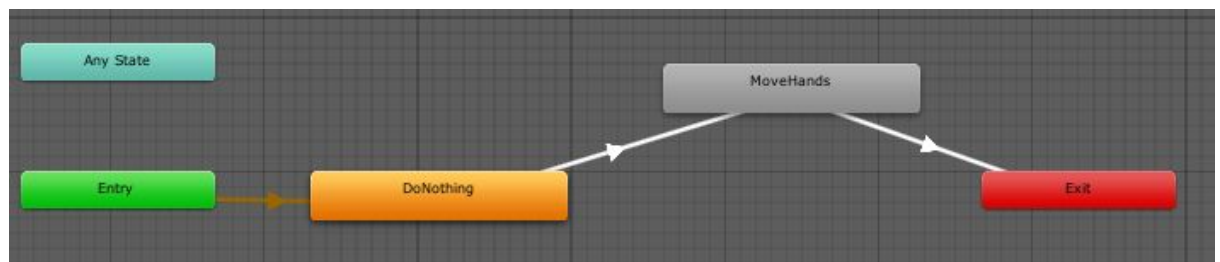
Kolejna część ćwiczenia dotyczy dodanie triggeru, który umożliwi wykonanie animacji. W tym celu należy dodać skrypt do "Playera":



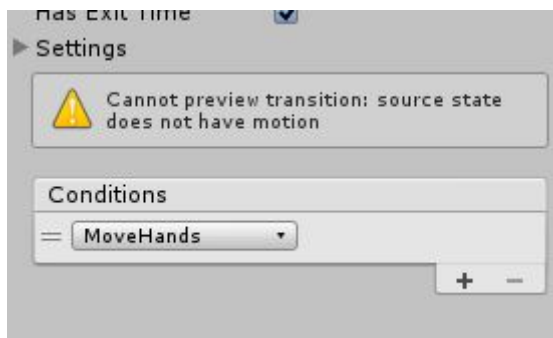
```
public class MoveHandsTrigger : MonoBehaviour
{
    public Animator animator;
    public CharacterController controller;
    // Start is called before the first frame update
    void Start()
    {
    }

    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space))
        {
            animator.SetTrigger("MoveHands");
        }
    }
}
```

Oraz dodać do Animatora odpowiednie elementy:

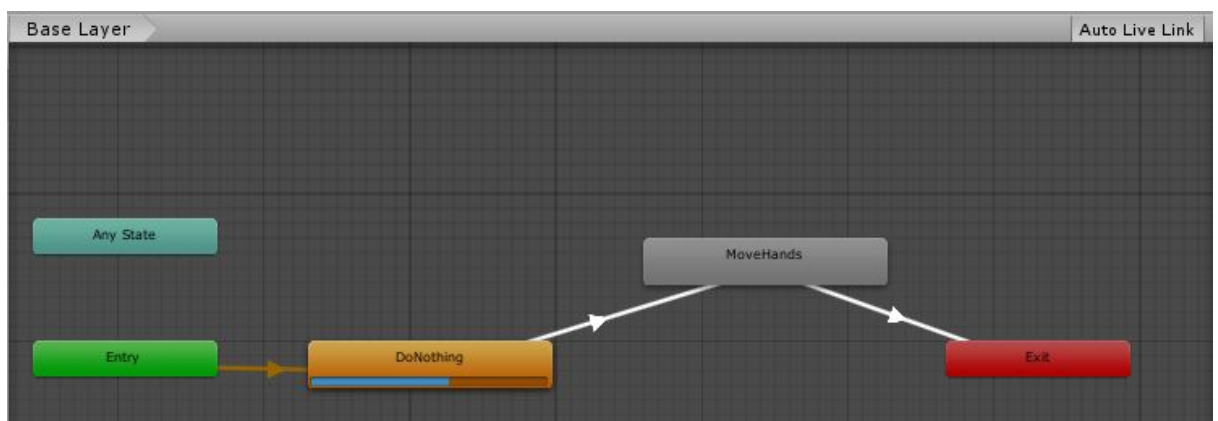


Pomiędzy stanem “DoNothing” oraz “MoveHands” dodany został warunek.

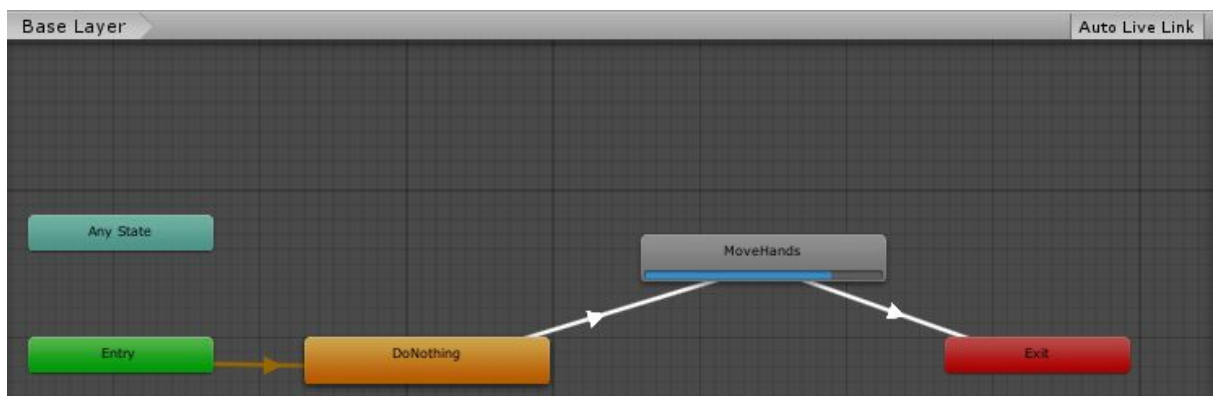


MoveHands jest Triggerem.

W ten sposób cały czas wykonywana jest akcja “DoNothing”



A przy naciśnięciu spacji aktywowany jest trigger, który przechodzi do stanu “MoveHands”



3. Wnioski:

Tworzenie prostych animacji w Unity jest przyjazne i proste. Przy wykorzystaniu gotowych modeli, należy tylko zaznajomić się z elementami, z których owy model został zrobiony. Czasochłonnym zajęciem przy animowaniu, byłoby tworzenie własnie modeli. Modele wykorzystywane w animacji muszą być wykonane dokładnie i powinny składać się z odpowiedniej ilości elementów (w tym ćwiczeniu, ręce bohatera składały się z łopatki, ramienia, łokcia, nadgarstka, 4 palców oraz kciuka).