

## **Techniki wirtualnej rzeczywistości w mechatronice**

### **Laboratorium nr 11-12**

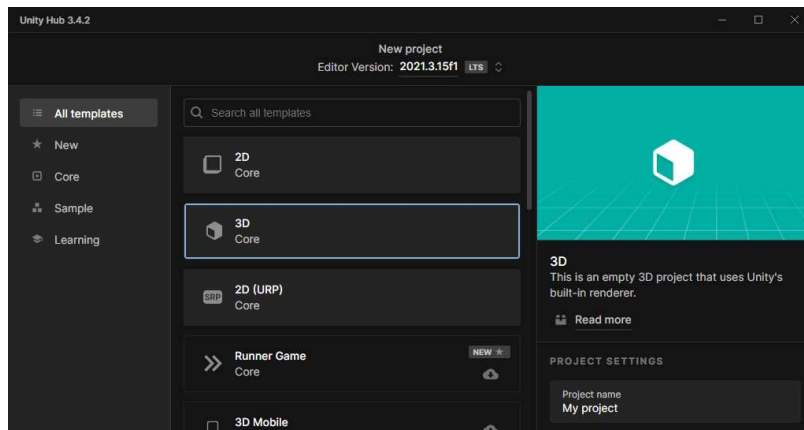
**Temat:** Unity Engine: sterowanie myszką i klawiaturą w środowisku 3D,  
pierwsza gra 3D

## 1. Wstęp

Instrukcja ma na celu zapoznanie studenta z podstawami sterowania myszką i klawiaturą w środowisku 3D oraz tworzeniem prostych gier.

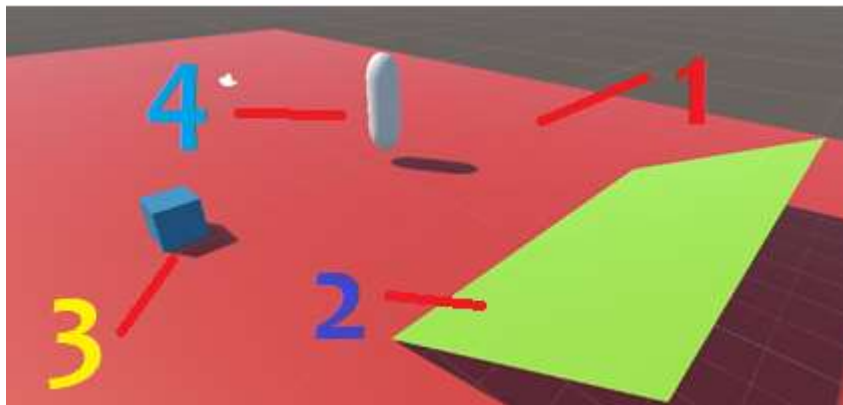
## 2. Przygotowanie sterowania i środowiska w Unity

- Uruchamiamy edytor Unity i wybieramy nazwę projektu oraz opcje 3D (rys. 1).

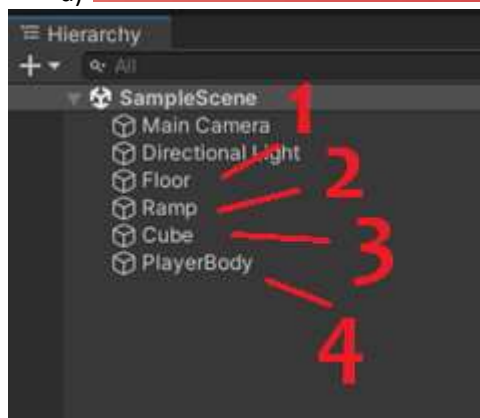


Rys. 1 Utworzenie projektu

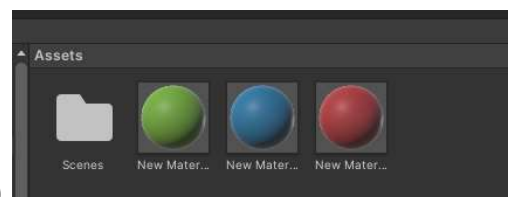
- W nowo utworzonym projekcie umieszczamy 4 bryły. Dwie płaszczyzny, kostkę i kapsułę oraz przypisujemy im różne materiały, tak by odróżniały się wizualnie od siebie (rys. 2a,b,c).



a)



b)



c)

Rys. 2 a) Rozstawienie brył, b) nadane nazwy, c) wybrane materiały

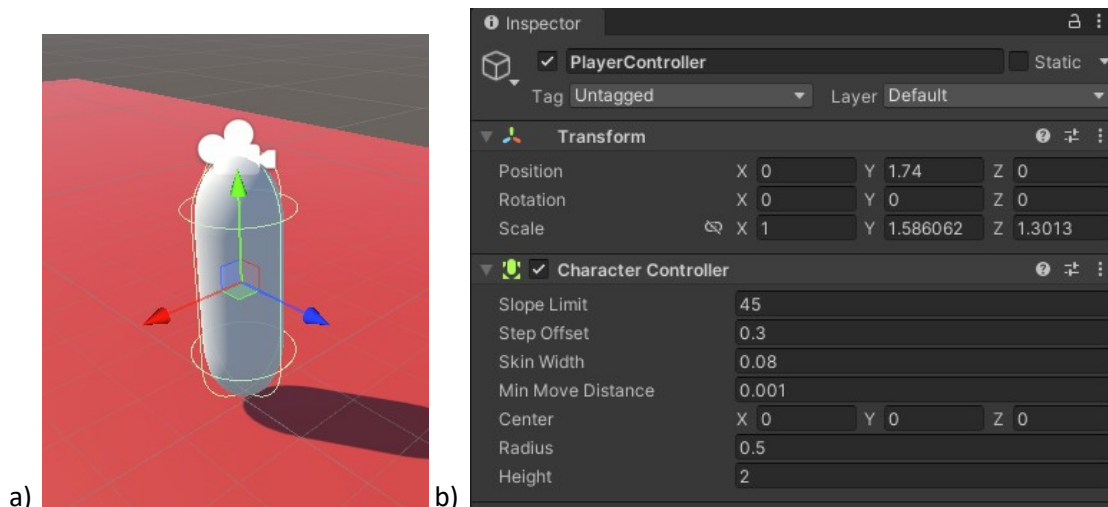
### Techniki wirtualnej rzeczywistości w mechatronice

Unity Engine: sterowanie myszką i klawiaturą w środowisku 3D, pierwsza gra 3D

## Katedra Mechaniki Stosowanej i Robotyki

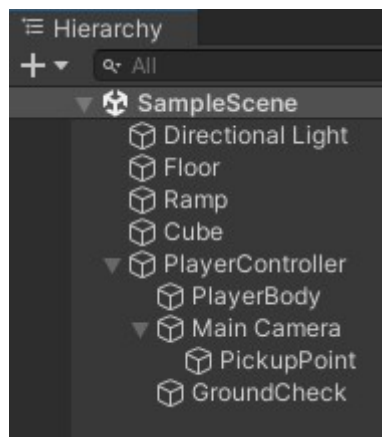
Wydział Budowy Maszyn i Lotnictwa, Politechnika Rzeszowska

- Utworzyć pusty „GameObject” (Drzewo hierarchii -> PPM -> „Create Empty”), zmienić nazwę na „PlayerController” oraz dołączyć komponent „Character Controller” (rys. 3 b). Po dołączeniu komponentu pojawi się zarys kapsuły na ekranie. Umieścić zarys na „PlayerBody” i zmienić jego rozmiar (rys. 3 a). Umieścić kamerę nad „PlayerBody” (rys. 3 a).



Rys. 3 a) Ustawienie zarysu i kamery, b) komponent „Character Controller”

- Utworzyć dwa puste „GameObject”. Zmienić nazwę jednego na „GroundCheck” i umieścić go poniżej postaci i ponad obiektem „floor” o około 0.2m (w „stopach” postaci). Drugi „GameObject” nazwać PickupPoint i umieścić go tam gdzie teoretycznie gracz ma wyciągnięte dłonie. Punkt ten będzie służył do podnoszenia obiektów.
- Przetawić w drzewku hierarchii elementy, tak jak na rys. 4.

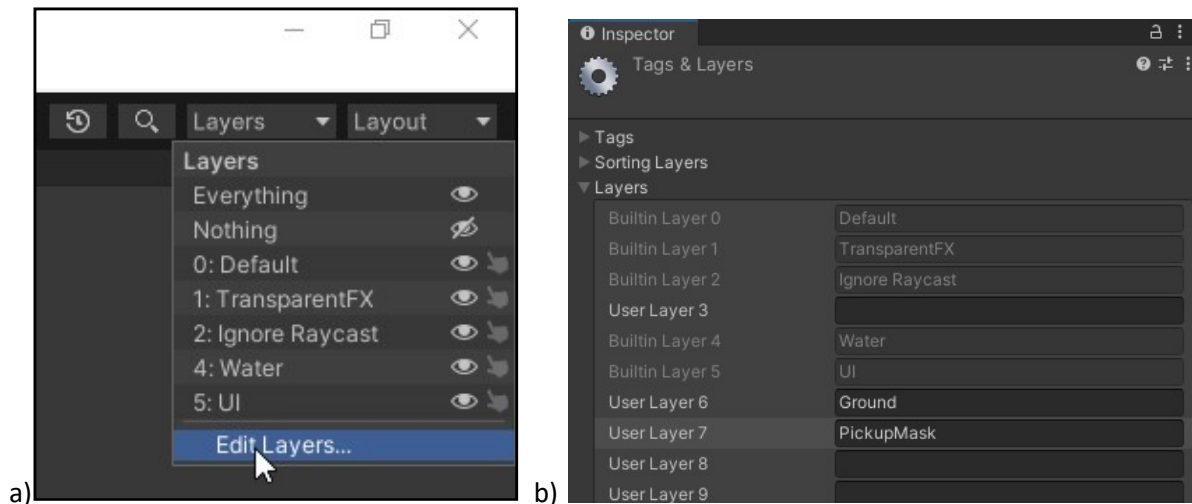


Rys. 4 Zmodyfikowane drzewko hierarchii

- Utworzyć dwie maski („Layers”) o nazwie „Ground” i „PickupMask” (rys. 5a,b)

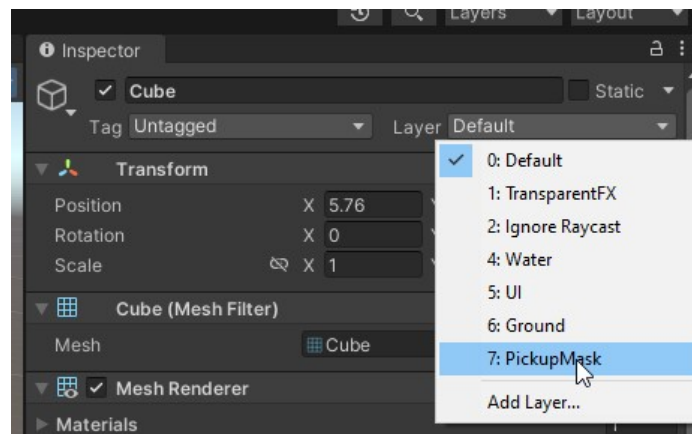
## Techniki wirtualnej rzeczywistości w mechatronice

Unity Engine: sterowanie myszką i klawiaturą w środowisku 3D, pierwsza gra 3D



Rys. 5 a,b) Tworzenie nowych masek

- Przypisać obiektom maskę „Ground”, po których będzie chodziła postać, a obiektom, które będzie podnosiła przypisać maskę „PickupMask” (rys. 6).



Rys. 6 Przypisanie maski „PickupMask” do obiektu „Cube”

- W obiekcie „PlayerBody” usunąć komponent „Capsule Collider” (PPM -> “Remove Component”).
- W obiekcie “Cube” dodać komponent “RigidBody” i włączyć interpolację (“Interpolate”).
- Utworzyć 4 skrypty C#: MouseLook, PhysicsPickup, PlayerMovement, PushObject.

### Skrypt MouseLook:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class MouseLook : MonoBehaviour
{
    public float mouseSensitivity = 100f; // czulosc myszki
    public Transform playerBody;
    float xRotation = 0f;

    void Start()
    {
```

## Katedra Mechaniki Stosowanej i Robotyki

Wydział Budowy Maszyn i Lotnictwa, Politechnika Rzeszowska

```
// Blokada wskaźnika myszki na środku pola widzenia kamery
Cursor.lockState = CursorLockMode.Locked;
transform.localRotation = Quaternion.identity;
}

void Update()
{
    // zczytanie wartosci z roznicy przemieszczenia sie myszki w osi x i y w ciagu
    jednej klatki
    float MouseX = Input.GetAxis("Mouse X") * mouseSensitivity * Time.deltaTime;
    float MouseY = Input.GetAxis("Mouse Y") * mouseSensitivity * Time.deltaTime;

    // zliczenie przesuniecie
    xRotation -= MouseY;
    // ograniczenie kata obrotu w osi x (ruch glowy gora-dol)
    xRotation = Mathf.Clamp(xRotation, -90f, 90f);
    // umozliwienie ruchu po 2 sekundach od startu symulacji
    if (Time.time < 2)
    {
        xRotation = 0f;
        MouseX = 0;
    }
    else{
        // zmiana kata obrotu kamery gora-dol
        transform.localRotation = Quaternion.Euler(xRotation, 0f, 0f);
        // zmiana kata obrotu kamery lewo-prawo
        playerBody.Rotate(Vector3.up * MouseX);
    }
}
}
```

### Skrypt PhysicsPickup:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PhysicsPickup : MonoBehaviour
{
    // stale do ktorych przypisujemy utworzone w unity maski i obiekty
    [SerializeField] private LayerMask PickupMask;
    [SerializeField] private Camera PlayerCam;
    [SerializeField] private Transform PickupTarget;
    [SerializeField] private float PickupRange = 8;
    // zmienne do obecnego trzymanego obiektu i jego danych z trafienia promieniem
    private Rigidbody CurrentObject;
    private RaycastHit hit;

    void Start()
    {}
    void Update()
    {
        // jezeli wcisnieto E (podniesienie obiektu) to
        if(Input.GetKeyDown(KeyCode.E))

```

### Techniki wirtualnej rzeczywistości w mechatronice

Unity Engine: sterowanie myszką i klawiaturą w środowisku 3D, pierwsza gra 3D

## Katedra Mechaniki Stosowanej i Robotyki

Wydział Budowy Maszyn i Lotnictwa, Politechnika Rzeszowska

```
{
    // jezeli trzymamy jakis obiekt CurrentObject
    if(CurrentObject)
    {
        //wlaczenie grawitacji wczesniej trzymanego obiektu
        CurrentObject.useGravity = true;
        // usuniecie ze zmiennej CurrentObject wczesniejszego obiektu
        CurrentObject = null;
        return;
    }
    // wytworzenie promienia przechodzacego przez pole widzenia kamery
    (wpulczyniki 0.5, 0.5 oznaczaja srodek)
    Ray CameraRay = PlayerCam.ViewportPointToRay(new Vector3(0.5f, 0.5f, 0f));
    // sprawdzenie czy promien CameraRay przecial sie z obiektem o masce
    PickupMask w odelgosci < PickupRange
    // jezeli trafiono jakis obiekt zwracany jest obiekt c# hit
    if (Physics.Raycast(CameraRay, out hit, PickupRange, PickupMask))
    {
        // przypisanie zmiennej CurrentObject komponentu Rigidbody z
        trafionego obiektu
        CurrentObject = hit.rigidbody;
        // wyłaczenie grawitacji trafionego obiektu
        CurrentObject.useGravity = false;
    }
}
}
// petla fizyki
void FixedUpdate()
{
    jezeli trzymamy jakis obiekt
    if(CurrentObject)
    {
        // wyznaczamy wektor z punktu Pickup do obiektu
        Vector3 DirectionToPoint = PickupTarget.position - CurrentObject.position;
        // wyznaczamy dlugosc tego wektora
        float DistanceToPoint = DirectionToPoint.magnitude;
        // ustalamy niezerowa predkosc tego obiektu tak by przemiescil sie do
        punktu Pickup
        // 30f to wspolczynnik projektowy
        CurrentObject.velocity = DirectionToPoint * 30f * DistanceToPoint;
    }
}
}
```

### Skrypt PlayerMovement:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

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

### Techniki wirtualnej rzeczywistości w mechatronice

Unity Engine: sterowanie myszką i klawiaturą w środowisku 3D, pierwsza gra 3D

## Katedra Mechaniki Stosowanej i Robotyki

Wydział Budowy Maszyn i Lotnictwa, Politechnika Rzeszowska

```
public float runSpeed = 20f; // predkosc biegania
public float walkSpeed = 6f; // predkosc chodzenia
public float speed; // zmienna predkosci gracza
public float gravity = -9.81f; // przyspieszenie grawitacyjne
public float jump = 1f; // wysokosc podskoku

public Transform groundCheck; // wspolrzedne srodka obiektu groundCheck z edytora
Unity
public float groundDistance = 0.4f; // odleglosc od srodka groundCheck do
powierzchni
public LayerMask groundMask; // zmienna maski Ground z edytora Unity

Vector3 velocity; // zmienna predkosci gracza
bool isGrounded; // zmienna sprawdzenia czy gracz stoi na chodliwej powierzchni

// Update is called once per frame
void Update()
{
    // sprawdzenie warunku czy gracz stoi na chodliwej powierzchni poprzez
    sprawdzenie
    // czy sfera o srodku w obiekcie groundCheck i promieniu groundDistance
    przecina sie z obiektem o masce groundMask
    isGrounded = Physics.CheckSphere(groundCheck.position, groundDistance,
    groundMask);

    // jezeli obiekt jest na powierzchni i jego predkosc pionowa jest <0 to
    zmniejszyc predkosc do -2.
    // predkosc bedzie malala poniewaz technicznie gracz znajduje sie nad
    // powierzchnia po ktorej chodzi a jedynie punkt obiekt groundCheck przecina
    sie
    // z ta powierzchnia

    if(isGrounded && velocity.y < 0)
    {
        velocity.y = -2f;
    }
    // uzyskanie wartosci naciśnięcia przycisków WASD
    float x = Input.GetAxis("Horizontal");
    float z = Input.GetAxis("Vertical");

    // utworzenie wektora przemieszczania i przemnożenie wartości położenia
    // gracza przez odpowiednie wartości z naciśniętych klawiszy WASD, które
    // mogą przyjmować jedynie wartości (-1,1)
    Vector3 move = transform.right * x + transform.forward * z;
    // jeżeli naciśnięto lewy shift to zmienić tryb z chodzenia na bieganie (czyli
    zmienić wsp. predkosci)

    if(Input.GetKey(KeyCode.LeftShift))
    { speed = runSpeed; } else { speed = walkSpeed; }
    // użycie polecenia Move dla komponentu Charakter Controller powoduje ruch
    gracza
    // o określony wektor predkosci
    controller.Move(move * speed * Time.deltaTime);
    // jeżeli naciśnięto spację
```

### Techniki wirtualnej rzeczywistości w mechatronice

Unity Engine: sterowanie myszką i klawiaturą w środowisku 3D, pierwsza gra 3D

## Katedra Mechaniki Stosowanej i Robotyki

Wydział Budowy Maszyn i Lotnictwa, Politechnika Rzeszowska

```
if(Input.GetButtonDown("Jump") && isGrounded)
{
    // wyznaczenie predkosci pocz. podskoku z rownowartosci energii kinet. i
    pot. mgh = mv^2/2
    velocity.y = Mathf.Sqrt(jump * -2f * gravity);
}
// pomniejszenie predkosci o przyciaganie grawitacyjne dzialajace w czasie
velocity.y += gravity * Time.deltaTime;
// zmiana ruchu gracza
controller.Move(velocity * Time.deltaTime);
}
}
```

### Skrypt PushObject:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PushObject : MonoBehaviour
{
    private float forceMagnitude = 2f; // amplituda sily

    void Start(){}

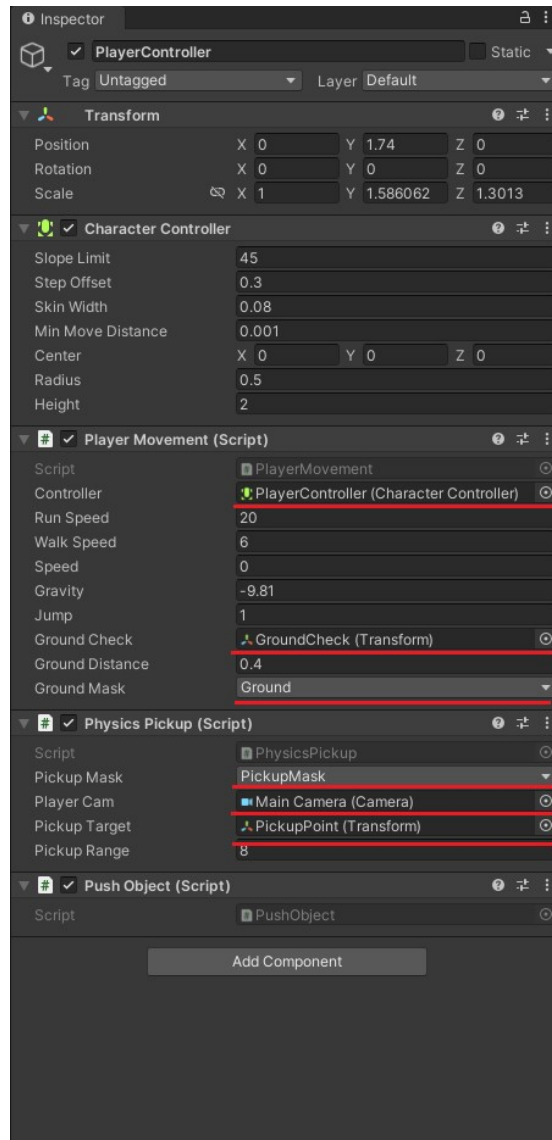
    void Update(){}
    // metoda wywoływana przez zdarzenie zetknięcia się dwóch ciał stałych (rigidbody)
    private void OnCollisionHit(Collision hit) {
        // przypisanie do zmiennej rigidbody obiektu ze zderzenia
        Rigidbody rigidbody = hit.collider.attachedRigidbody;
        // jeżeli istnieje ten obiekt
        if (rigidbody != null)
        {
            // wyznaczyć wektor styku od pozycji gracza do pozycji styku z obiektem
            Vector3 forceDirection = hit.gameObject.transform.position -
transform.position;
            // wyzerowanie pionowej składowej wektora (tak by nie odbić obiektu w gore
            lub w dol)
            forceDirection.y = 0;
            // normalizacja wektora do wartości 1 ale o zachowanym kierunku
            forceDirection.Normalize();
            // przyłożenie sily o amplitudzie forceMagnitude o charakterystyce
            wymuszenia impulsowego w punkcie styku do obiektu
            rigidbody.AddForceAtPosition(forceDirection * forceMagnitude,
            hit.gameObject.transform.position, ForceMode.Impulse);
        }
    }
}
```

- Przeciągnąć do obiektu "PlayerController" skrypty "Player Movement", "Push Object" i "Physics Pickup". Do odpowiednich skryptów w inspektorze przeciągnąć z drzewka hierarchii odpowiednie obiekty (rys. 7). Do pozycji „Controller” w skrypcie „Player Movement” przeciągnąć komponent „Character Controller” powyżej.

### Techniki wirtualnej rzeczywistości w mechatronice

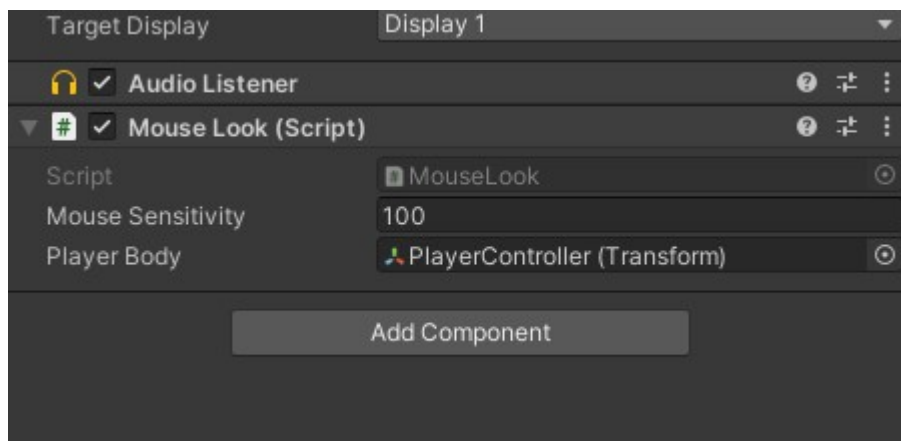
Unity Engine: sterowanie myszką i klawiaturą w środowisku 3D, pierwsza gra 3D





Rys. 7 Ustawienia w inspektorze obiektu „PlayerController”

Do obiektu „Main Camera” przeciągnąć skrypt „MouseLook” i w pozycji „Player Body” w komponencie skryptu przeciągnąć obiekt „PlayerController” (rys. 8).



Rys. 8 Ustawienia w inspektorze obiektu „Main Camera”

### Zadania do wykonania

Utworzyć grę w Jengę. Umieścić w grze co najmniej 10 klocków i stół, na którym będzie toczyła się gra.

- 1) Utworzyć projekt ze sterowaniem takim jak w instrukcji.
- 2) Zamodelować w Blenderze stół oraz jeden z klocków i ich proceduralne materiały.
- 3) Zaimportować obiekty z teksturami z Blendera do Unity oraz stworzyć grę Jenga.

Do zadania można wykorzystać przykłady z instrukcji i poprzednich laboratoriów.

### Student dostaje:

- ocenę 5 za prawidłowo wykonane punkty nr 1, 2 i 3
- ocenę 4 za prawidłowo wykonane punkty nr 1 i 2
- ocenę 3 za prawidłowo wykonany punkt nr 1

W rozwiązywaniu zadań można posłużyć się materiałami dydaktycznymi, przykładami z instrukcji i Internetu oraz dokumentacji oprogramowania Blender i Unity na stronach:

<https://docs.blender.org/manual/en/latest/modeling/index.html>

<https://docs.unity3d.com/Manual/index.html>