PROGRAMOWANIE GIER

WSPÓŁPRACA Z UŻYTKOWNIKIEM

FORMALIZACJA POJĘĆ

1. Przestrzenią akcji nazywamy zbiór dostępnych do wykonania akcji w pewnej chwili *t*.
2. Instrukcją nazywamy akcję, którą uczestnik ma wykonać w swoim następnym ruchu.
3. Sterowaniem nazywamy uszeregowany zbiór instrukcji.
4. Stan gry w pewnej chwili *t* zależy wyłącznie od stanu gry w chwili *t-1* oraz zbioru sterowań wszystkich uczestników.
5. Kontrolerem nazywamy oprogramowanie lub sprzęt, którego wyjściem jest sterowanie.

PYTANIE:

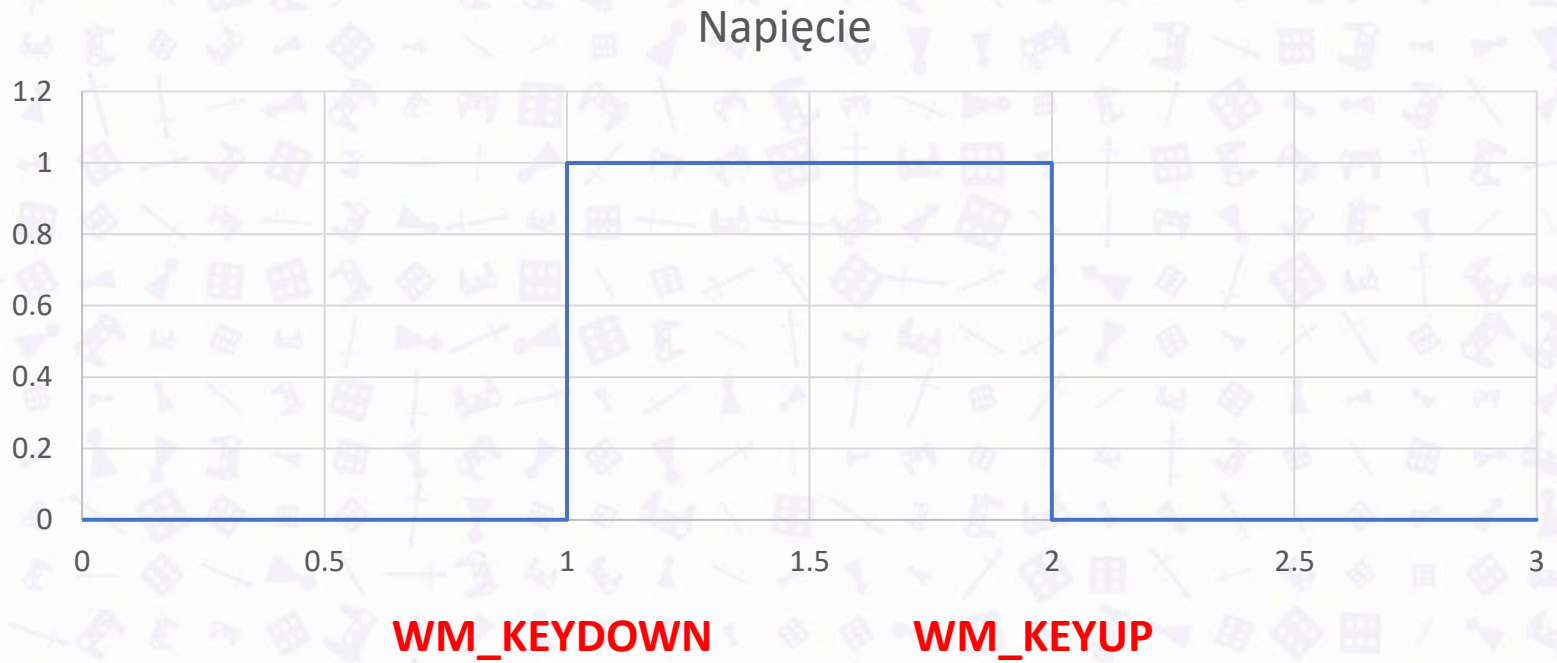
Jakie kontrolery sprzętowe się współcześnie wykorzystuje?

KONTROLERY SPRZĘTOWE

- Klawiatura
- Mysz
- Joypad
- Joystick
- Kierownica
- Touchpad
- Ekran dotykowy
- Żyroskop
- Urządzenia interpretujące ruch ciała

PRZYCISKI I KLAWISZE

- W momencie naciśnięcia klawisza lub przycisku, do systemu wysłany zostaje odpowiedni komunikat.
- Kontekstem tego komunikatu jest klawisz lub przycisk, który został wciśnięty
- Analogiczny komunikat jest wysyłany w momencie puszczenia klawisza lub przycisku.

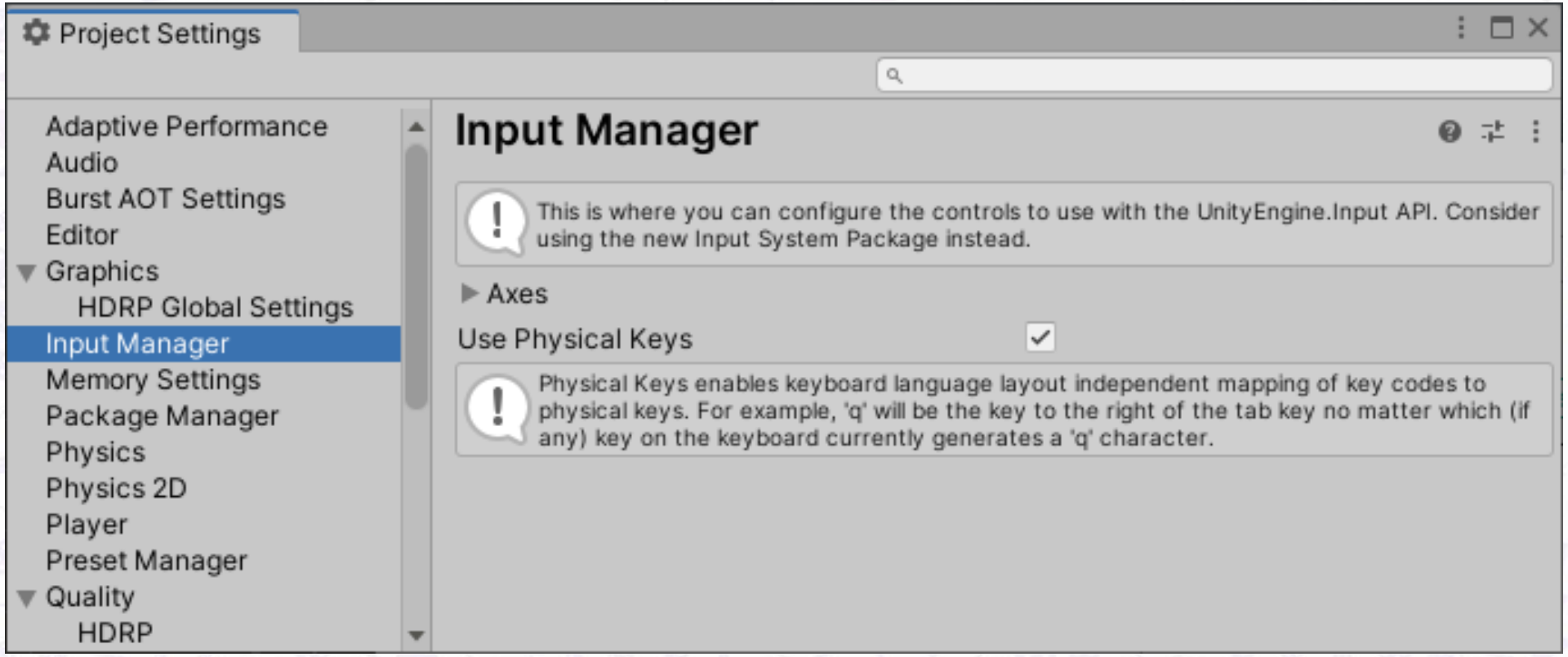


Wejścia liczbowe i wektorowe

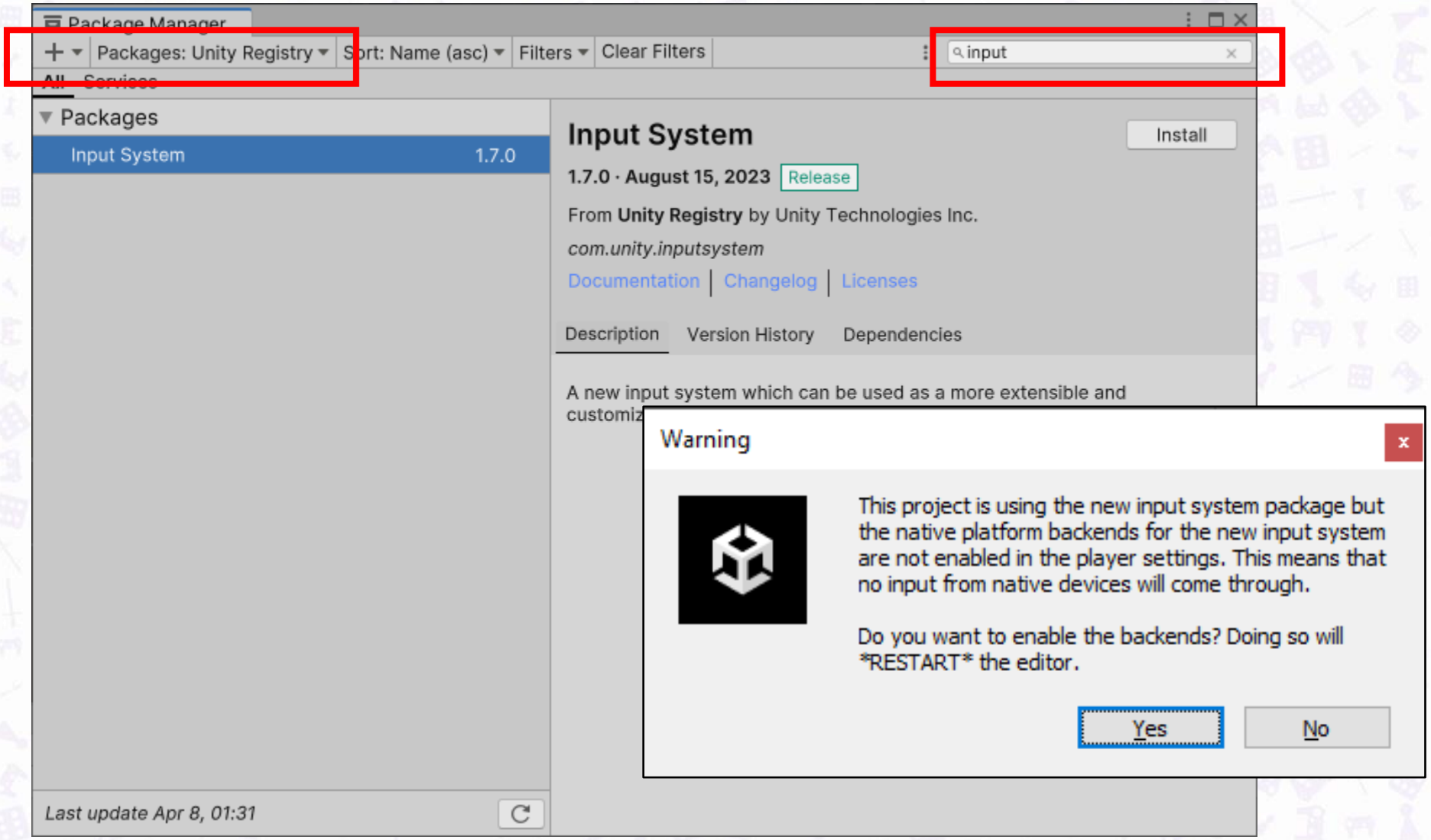
- Do systemu przesyłana jest informacja w postaci pojedynczej liczby lub wektora liczbowego
- Wektor zazwyczaj jest dwuelementowy
- Występują dwie powszechnie występujące kategorie takich wejść:

1. Posiadające wartości w zakresie od 0 do pewnego MAX, w postaci liczb całkowitych. Przenoszą one informacje o pozycji. Przykładem takich wartości jest pozycja myszki lub pozycja punktu na ekranie dotykowym.
2. Posiadające wartości w zakresie od –MAX do MAX, w postaci liczb zmiennoprzecinkowych. Przenosza one informacje o odchyleniu. Wartości takie nazywa się osiami (ang. „*axis*”). Przykładami są odchylenie joysticka analogowego lub wychylenie „grzybka” na joypadzie.

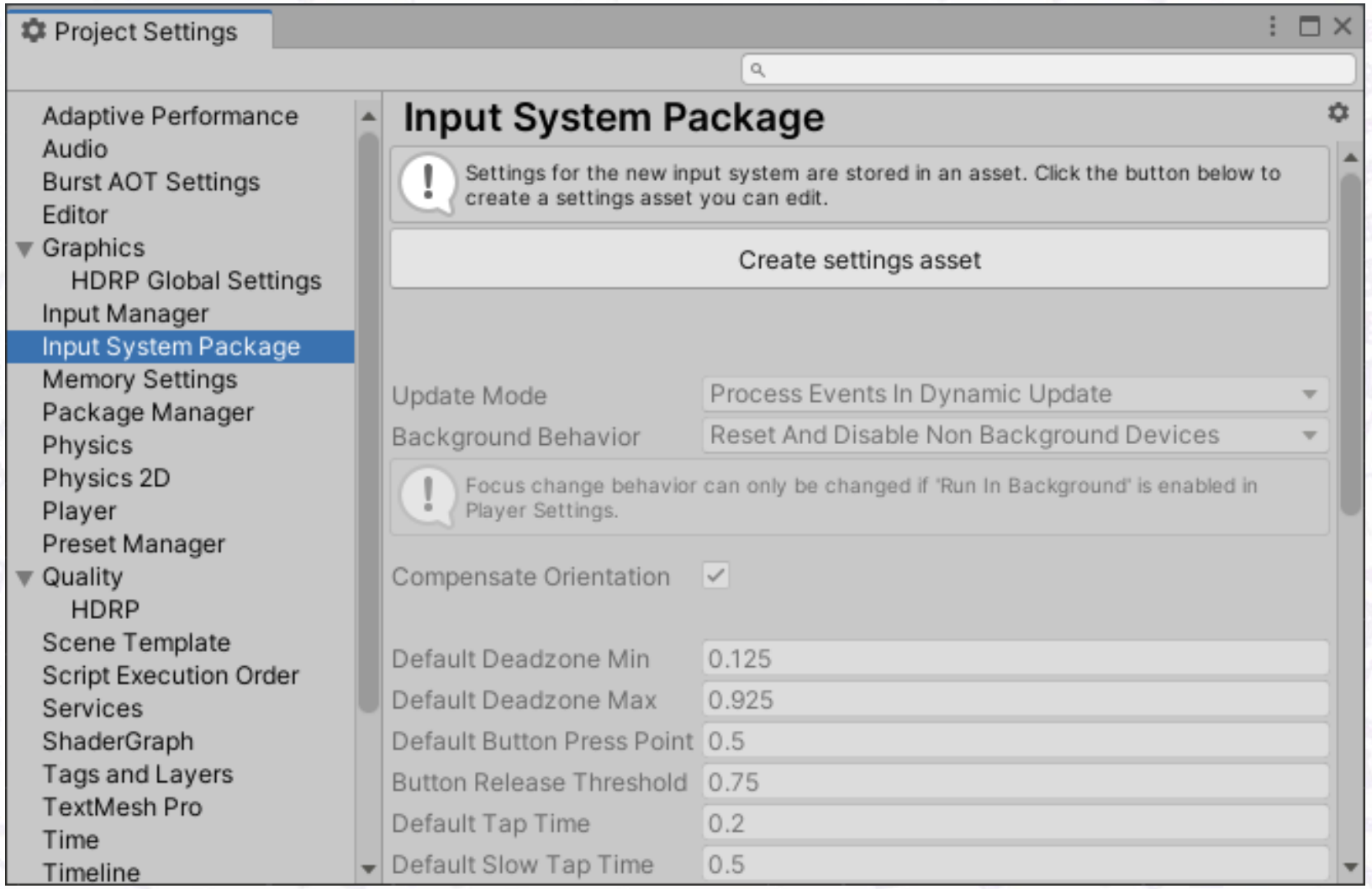
Obsługa wejścia w Unity



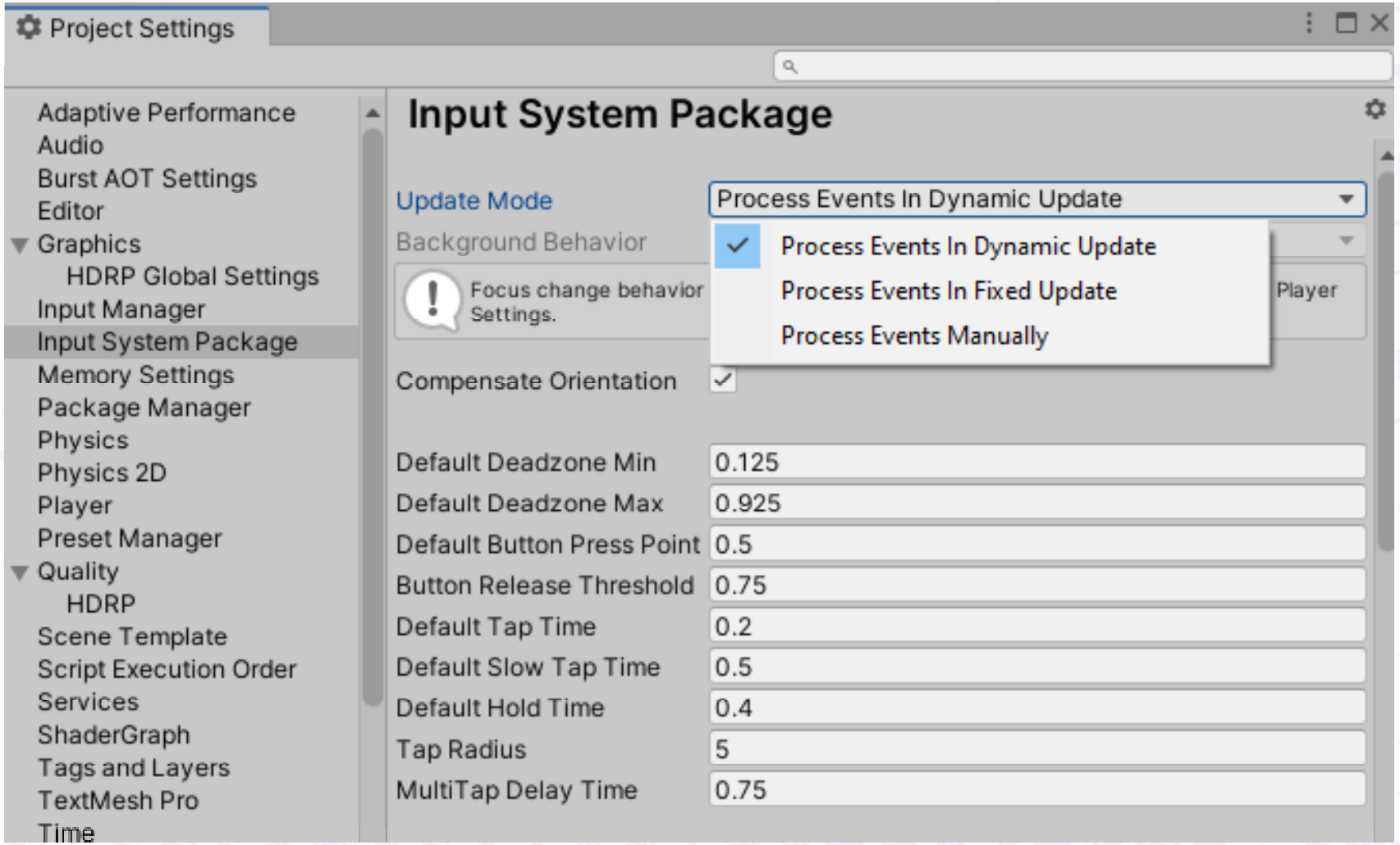
ZARZĄDCA PAKIETÓW



USTAWIENIA PROJEKTU

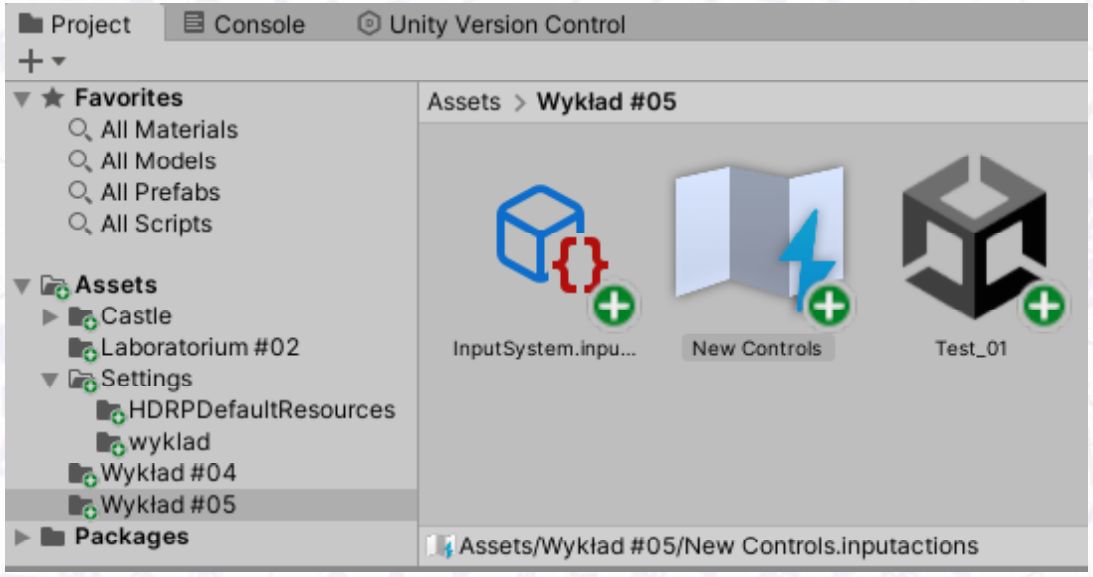


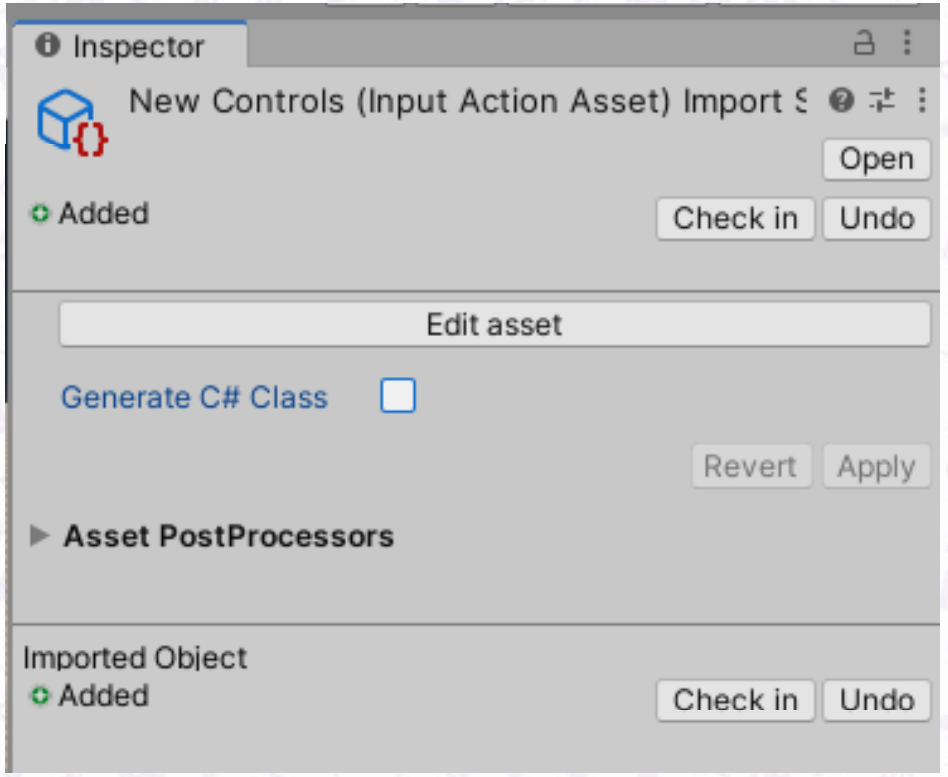
USTAWIENIA PODSTAWOWE



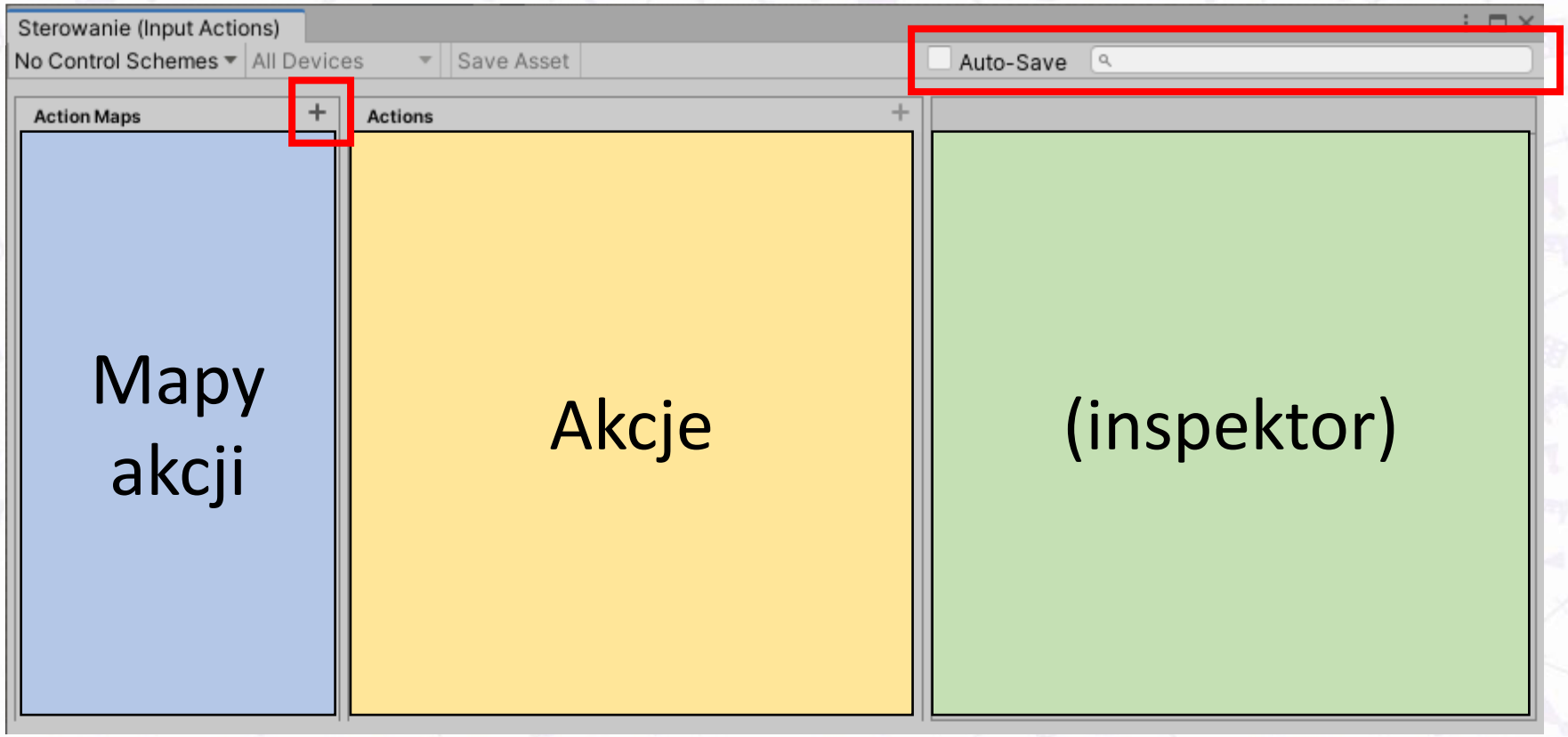
Zasób „Input Actions”

Create → Input Actions (na samym dole menu kontekstowego)

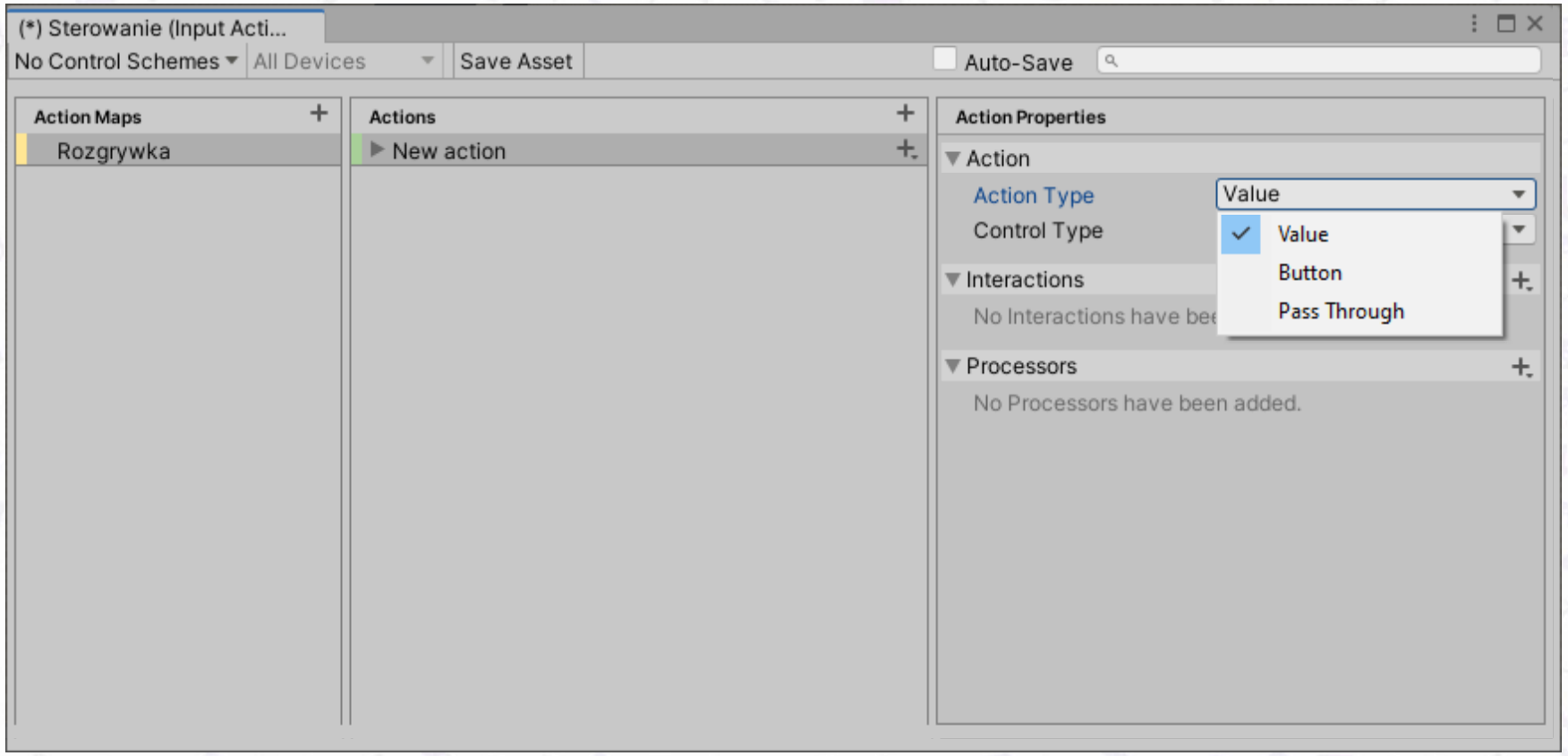




Okno edycji sterowania



TWORZENIE MAPY AKCJI



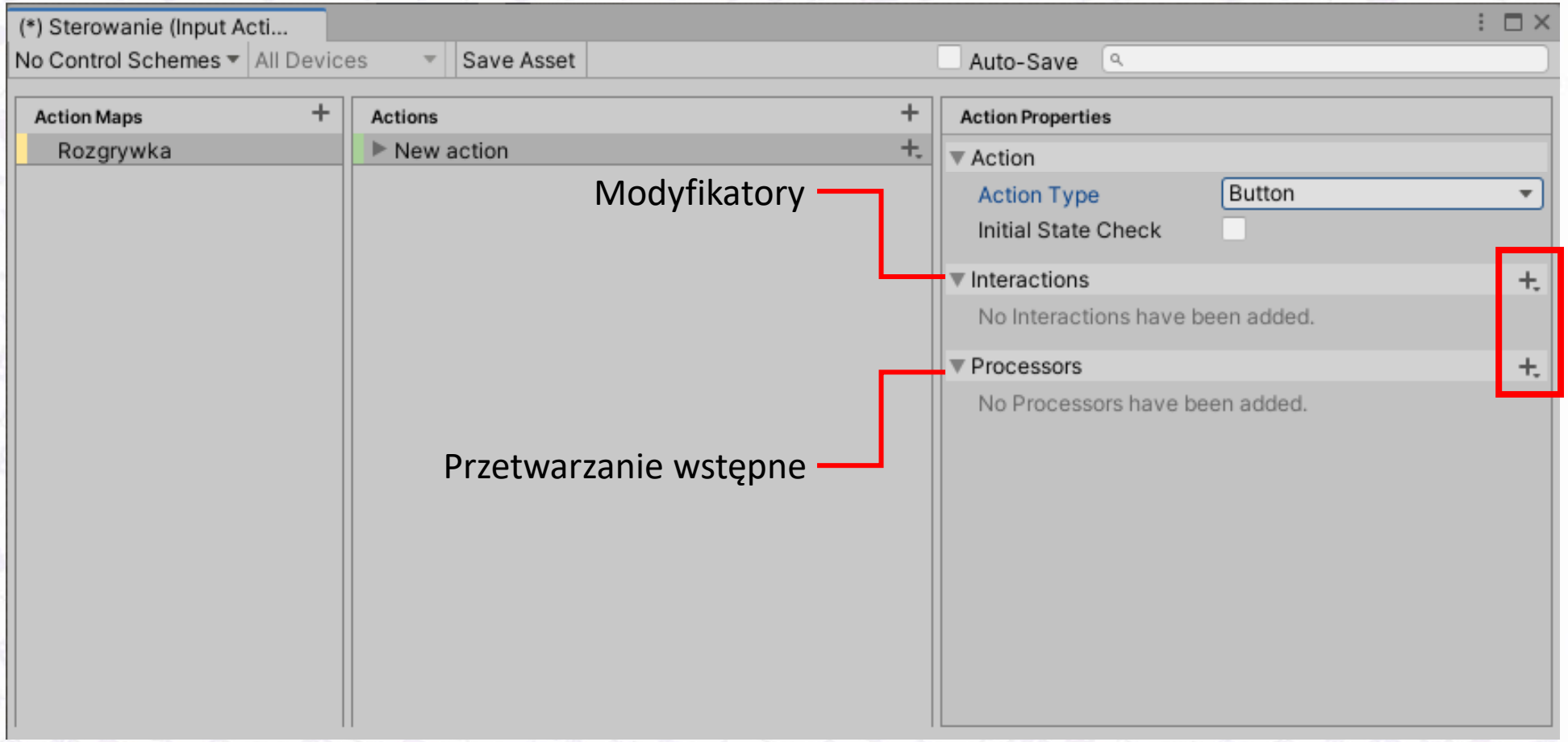
Typy akcji

Typ	Zachowanie domyślne
Button	1. Traktuje każde podpięte urządzenie jako jedno źródło, 2. Ma tylko dwa stany – aktywny i nieaktywny, 3. Stan zmienia się na aktywny, gdy wartość na wejściu przekroczy zadany próg, 4. Stan zmienia się na nieaktywny, gdy wartość na wejściu spadnie poniżej progu
Value	1. Traktuje każde podpięte urządzenie jako jedno źródło, 2. Informuje o każdej zmianie wartości, 3. Aktywuje się natychmiast, gdy wartość odbiega od domyślnej
Pass Through	1. Traktuje każde podpięte urządzenie indywidualnie, 2. Informuje o każdej zmianie wartości, 3. Ignoruje kierunek zmian

https://docs.unity3d.com/Packages/com.unity.inputsystem@1.0/api/UnityEngine.InputSystem.InputActionType.html

https://docs.unity3d.com/Packages/com.unity.inputsystem@1.0/manual/Actions.html#action-types

MODYFIKATORY I PRZETWARZANIE WSTĘPNE



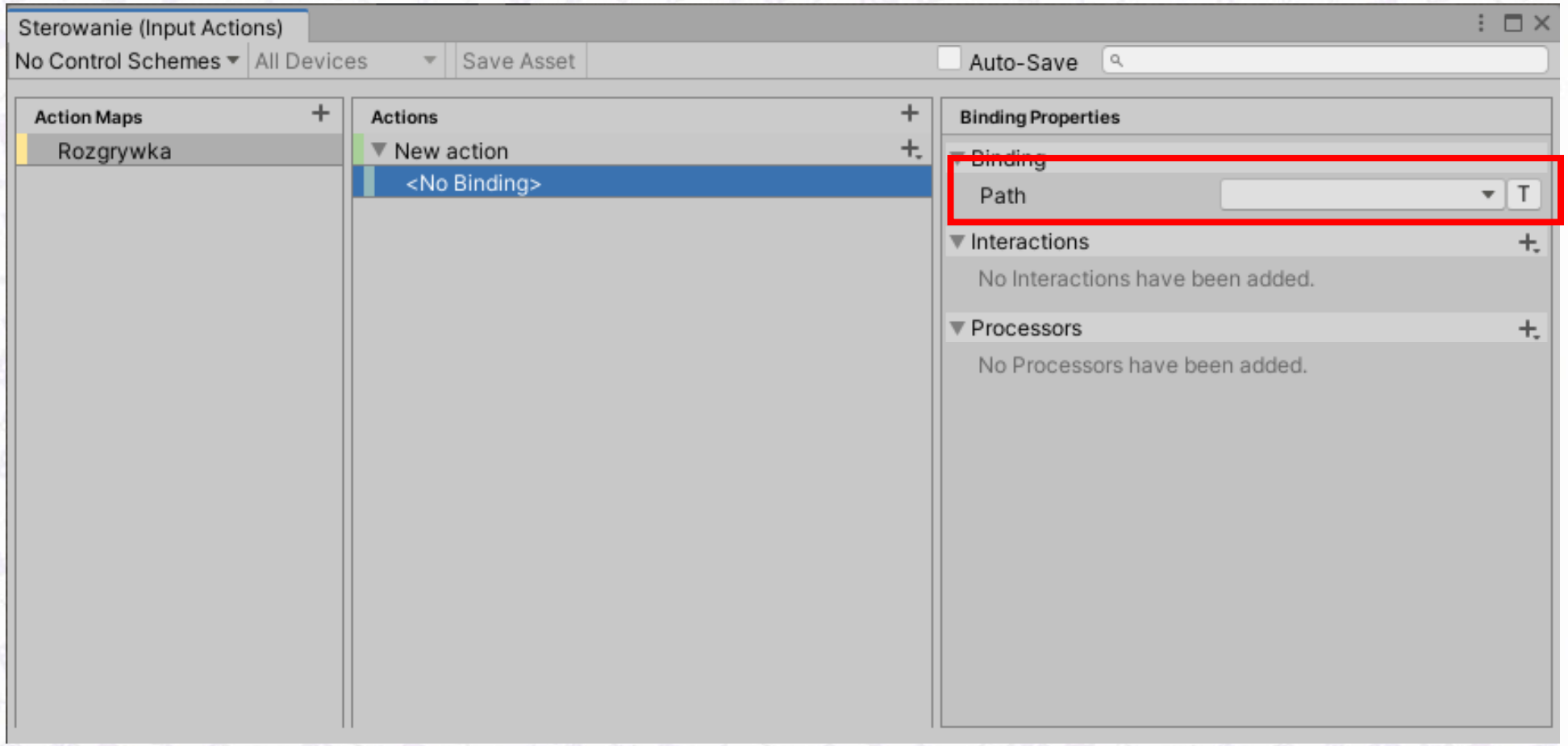
MODYFIKATORY

Modyfikator	Warunki
Tap	Akcja jest w stanie aktywnym nie dłużej niż określony czas. Przekroczenie czasu skutkuje niepowodzeniem wywołania akcji.
Multi Tap	Warunki analogiczne jak przy Tap, ale muszą być spełnione określoną liczbę razy, zanim akcja zostanie wywołana.
Slow Tap	Akcja jest w stanie aktywnym powyżej określonego czasu. Każde przejście do stanu nieaktywnego resetuje licznik czasu.
Hold	Akcja jest w stanie aktywnym powyżej określonego czasu. Każde przejście do stanu nieaktywnego resetuje licznik czasu. Jeżeli stan aktywny trwał powyżej progu czasowego, akcja jest wywoływana stale, aż do przejścia w stan nieaktywny.
Press	Sprawia, że akcja wywołuje zdarzenia tak, jak gdyby była przyciskiem.

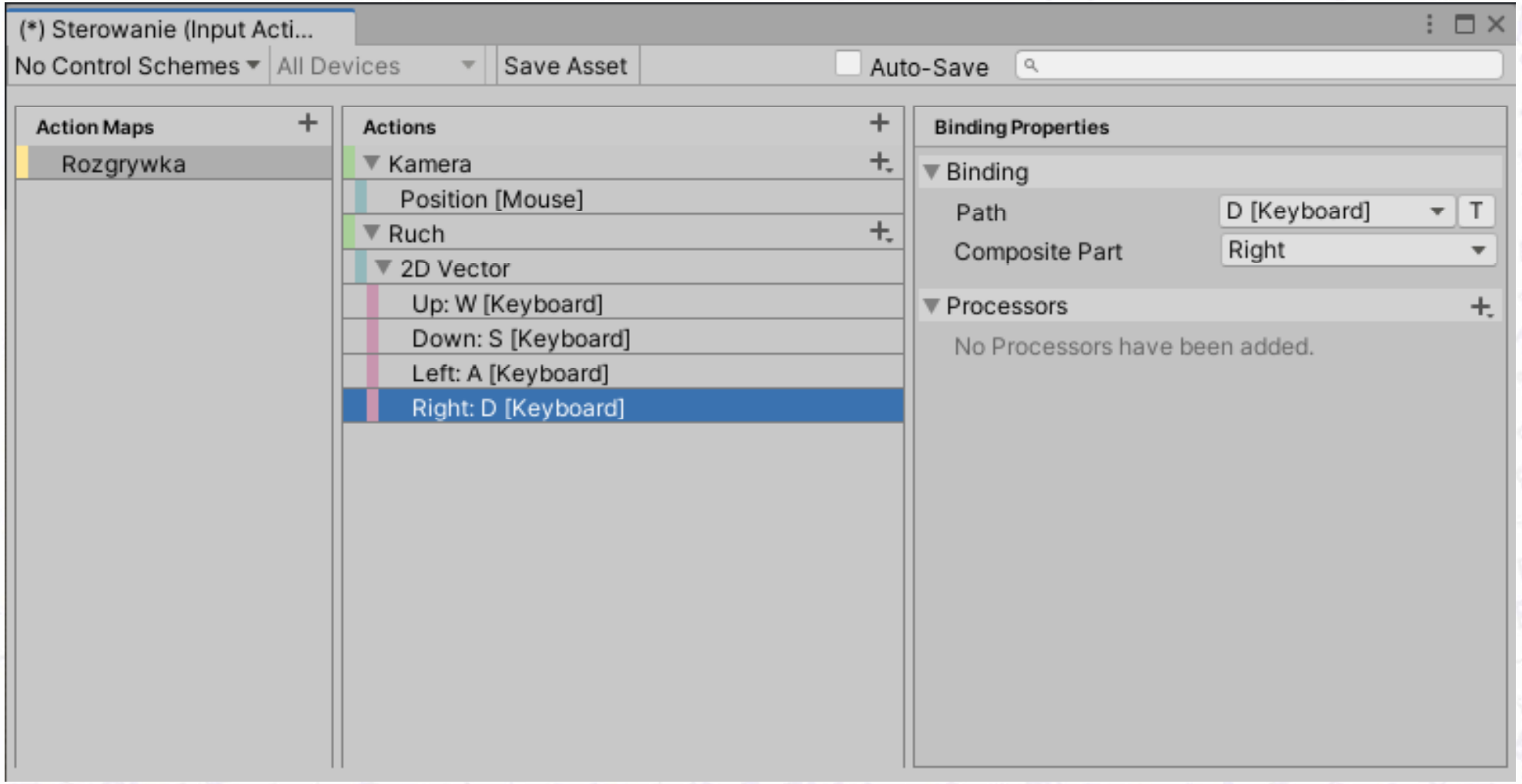
PRZETWARZANIE WSTĘPNE

Grupa	Działanie
Deadzone	Wartości poniżej *min* ustawiane są na 0. Wartości, które są bezwzględnie większe od *max*, ustawiane są na -1 lub 1.
Clamp	Wartości poniżej *min* ustawiane są na *min*. Wartości powyżej *max*, ustawiane są na *max*.
Invert	Mnoży wartość na wejściu przez -1.
Normalize	Sprowadza wartości do zakresu [0..1] lub [-1..1], zależnie od tego czy wartość minimalna jest większa, czy mniejsza od zera.
Scale	Mnoży wartość na wejściu przez zadaną wartość.

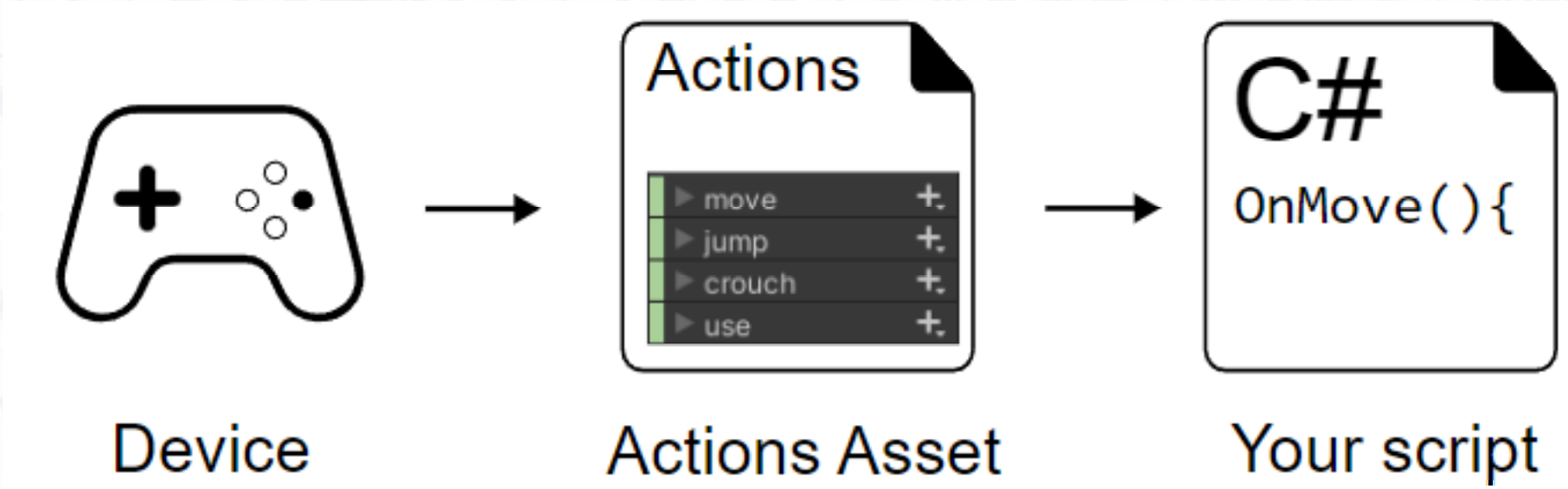
PRZYPISYWANIE KLAWISZY



RZECZYWISTY SCENARIUSZ



ZALECANY POTOK PRACY



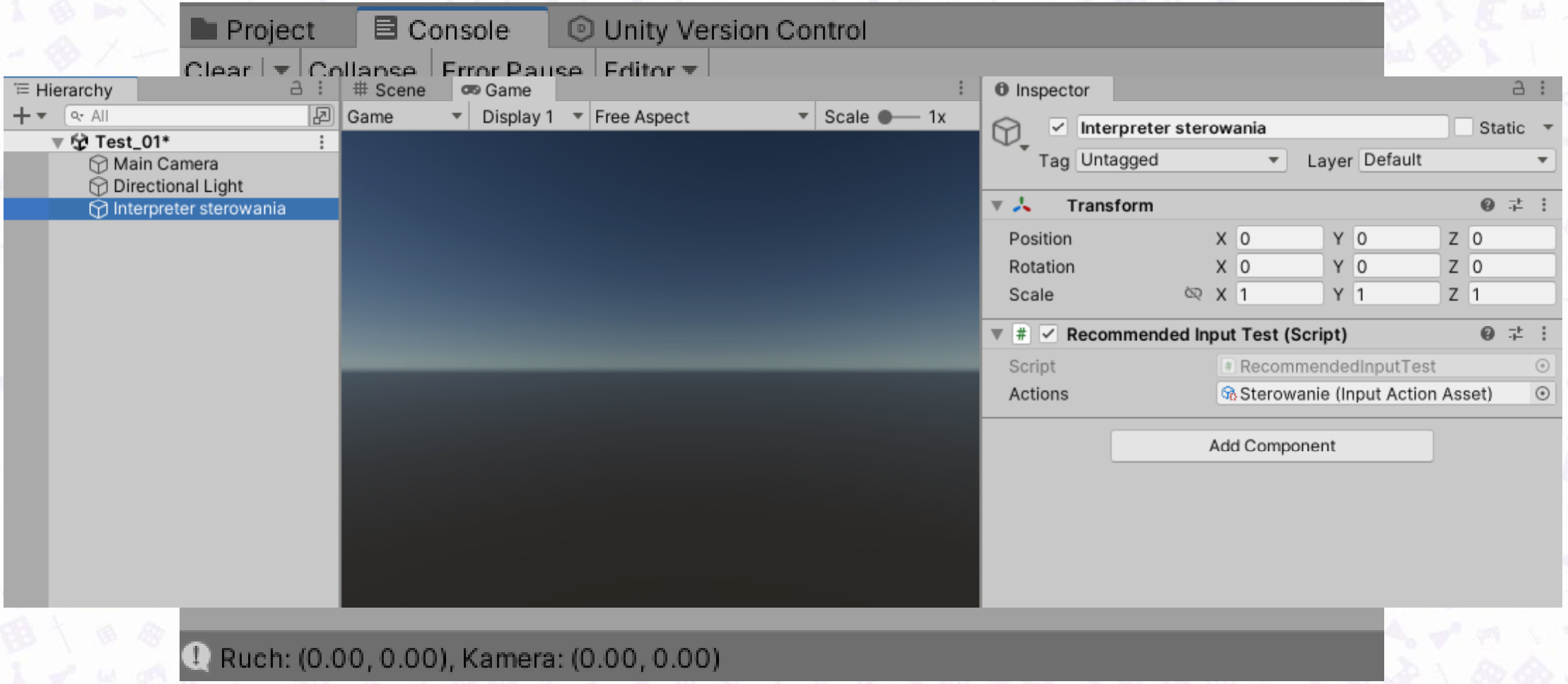
https://docs.unity3d.com/Packages/com.unity.inputsystem@1.7/manual/Workflow-ActionsAsset.html

Informacje dotyczące potoków pracy są podane dla wersji 1.7 pakietu.

POTOK ZALECANY: SKRYPT

```
1. using UnityEngine;
2. using UnityEngine.InputSystem; // Krok 1: uzycie pakietu
3.
4. public class RecommendedInputTest : MonoBehaviour
5. {
6.     public InputActionAsset actions; // Krok 2: referencja do zasobu
7.
8.     private InputAction akcjaRuchu; // Krok 3: zmienne pomocnicze
9.     private InputAction akcjaWidoku;
10.
11.    void Start()
12.    {
13.        // Krok 4: wyszukanie referencji do przypisania
14.        akcjaRuchu = actions.FindActionMap("Rozgrywka").FindAction("Ruch");
15.        akcjaWidoku = actions.FindActionMap("Rozgrywka").FindAction("Kamera");
16.    }
17.    void Update()
18.    {
19.        // Krok 5: pobranie danych
20.        Vector2 wektorRuchu = akcjaRuchu.ReadValue<Vector2>();
21.        Vector2 wektorKamery = akcjaWidoku.ReadValue<Vector2>();
22.
23.        // Krok 6: wypisanie danych
24.        Debug.Log(string.Format("Ruch: {0}, Kamera: {1}",
25.            wektorRuchu, wektorKamery));
26.    }
27.}
```

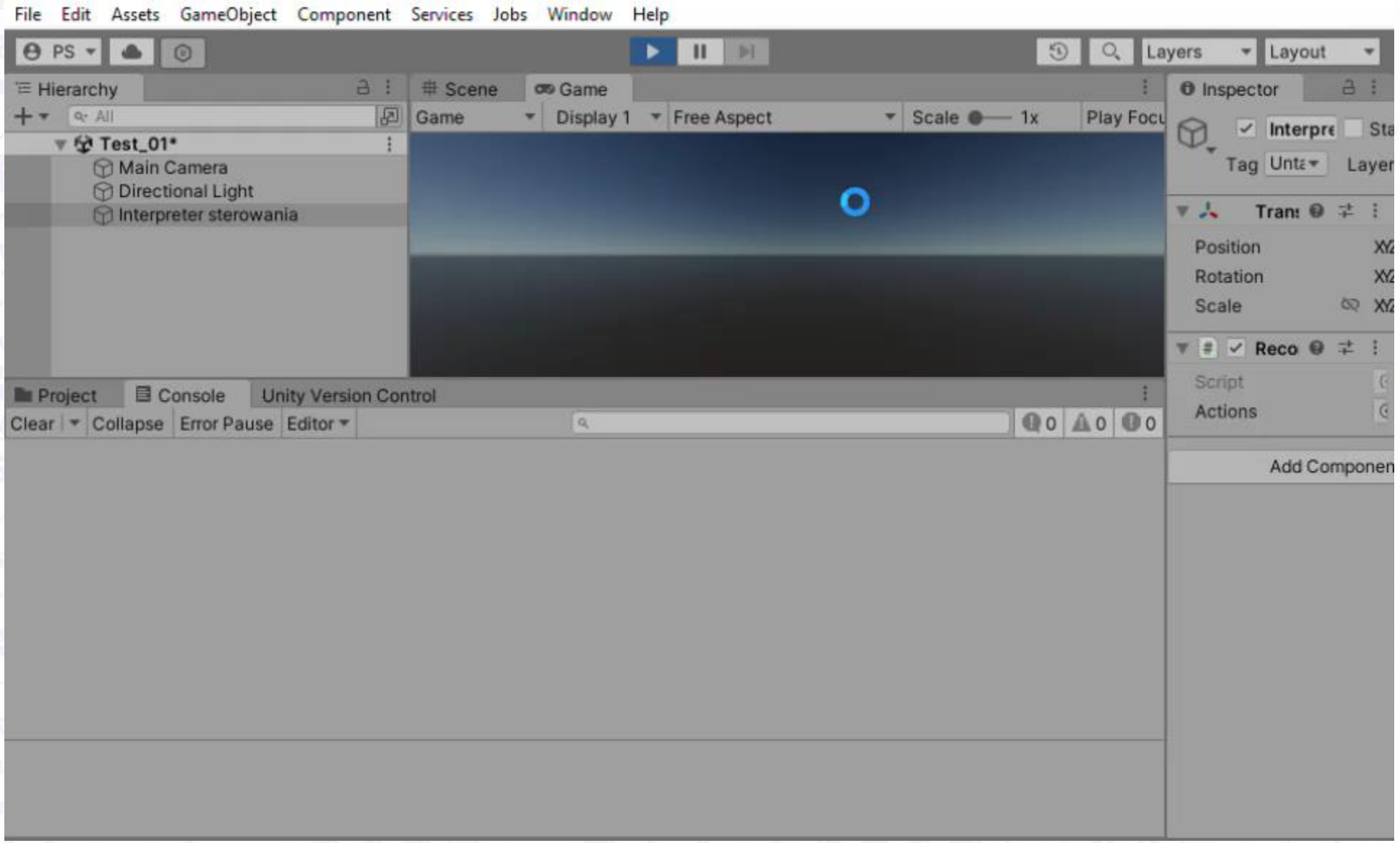
SKRYPT: DZIAŁANIE



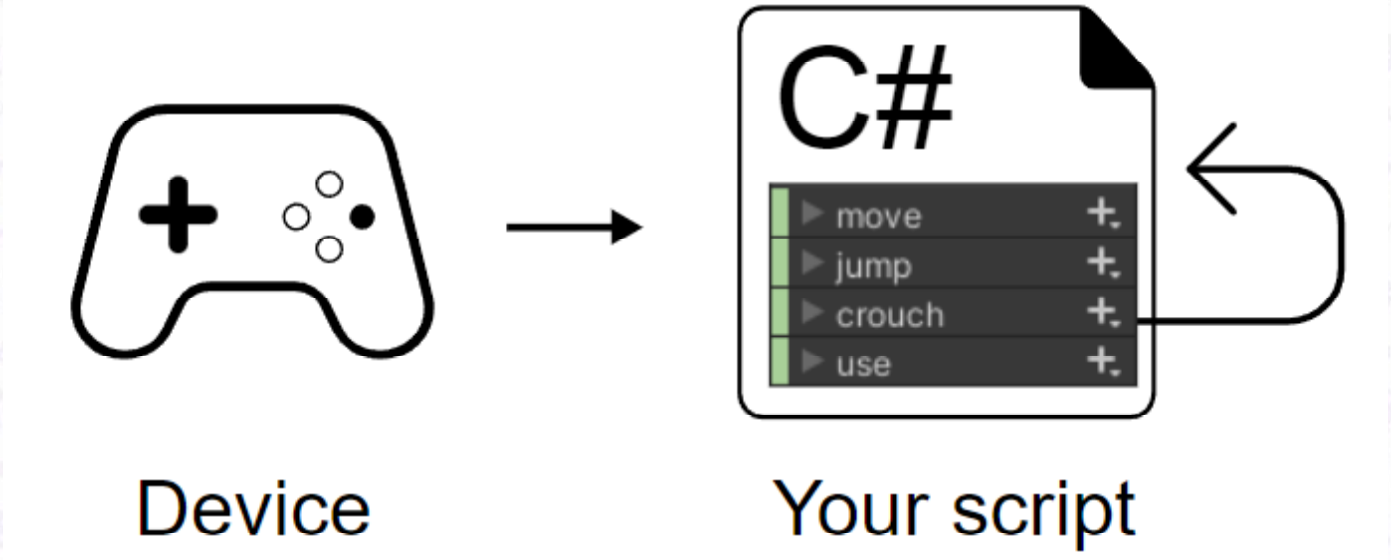
POTOK ZALECANY: SKRYPT POPRAWIONY

```
1.  using UnityEngine;
2.  using UnityEngine.InputSystem; // Krok 1: uzycie pakietu
3.
4.  public class RecommendedInputTest : MonoBehaviour
5.  {
6.      public InputActionAsset actions; // Krok 2: referencja do zasobu
7.
8.      private InputAction akcjaRuchu; // Krok 3: zmienne pomocnicze
9.      private InputAction akcjaWidoku;
10.
11.     void Start()
12.     {
13.         // Krok 4: wyszukanie referencji do przypisania
14.         akcjaRuchu = actions.FindActionMap("Rozgrywka").FindAction("Ruch");
15.         akcjaWidoku = actions.FindActionMap("Rozgrywka").FindAction("Kamera");
16.
17.         // Krok 4.5: aktywacja przestrzeni akcji
18.         actions.FindActionMap("Rozgrywka").Enable();
19.     }
20.     void Update()
21.     {
22.         // Krok 5: pobranie danych
23.         Vector2 wektorRuchu = akcjaRuchu.ReadValue<Vector2>();
24.         Vector2 wektorKamery = akcjaWidoku.ReadValue<Vector2>();
25.
26.         // Krok 6: wypisanie danych
27.         Debug.Log(string.Format("Ruch: {0}, Kamera: {1}",
28.             wektorRuchu, wektorKamery));
29.     }
30. }
```

PREZENTACJA DZIAŁANIA



OSADZONY POTOK PRACY



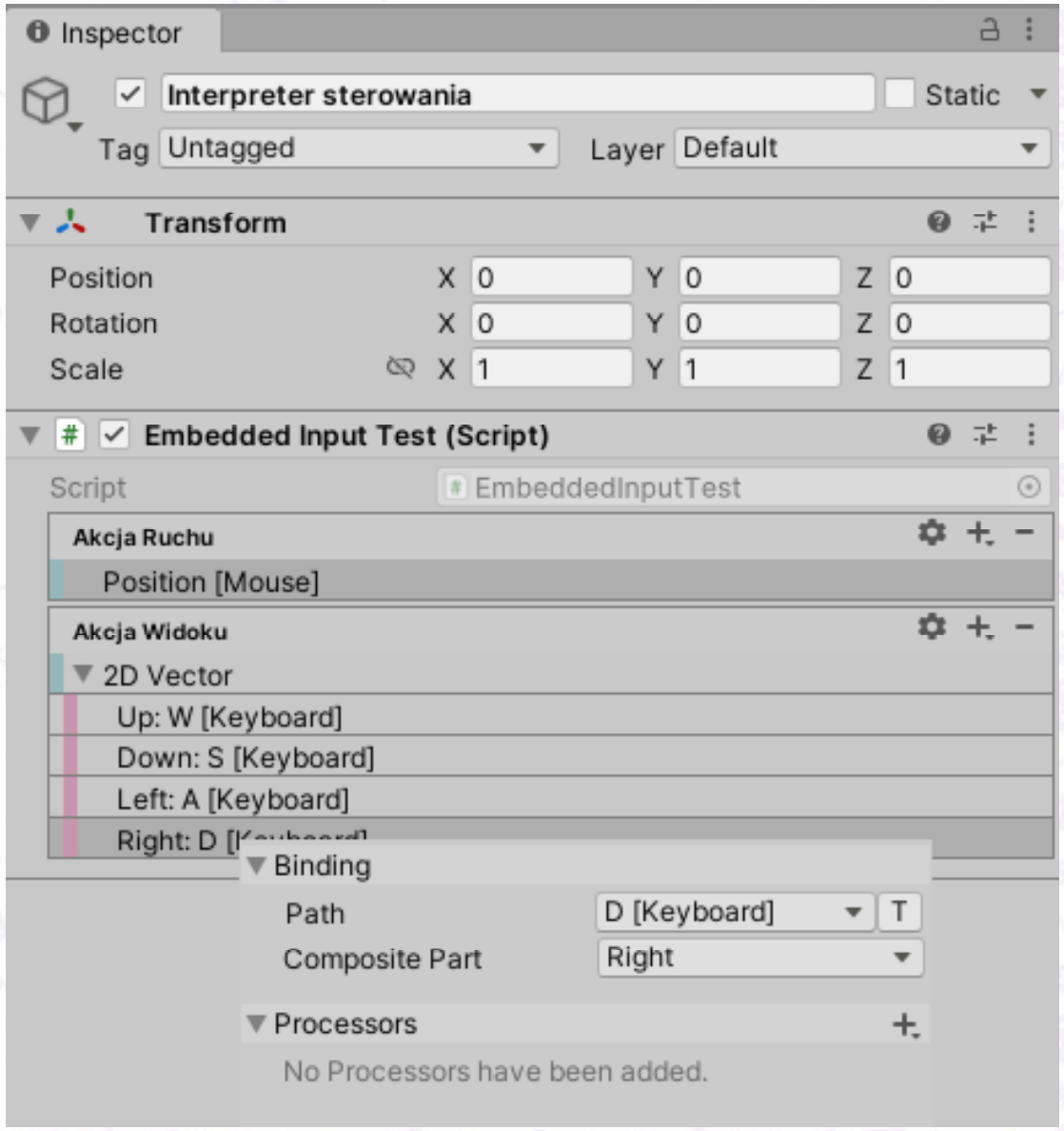
https://docs.unity3d.com/Packages/com.unity.inputsystem@1.7/manual/Workflow-Embedded.html

POTOK USUNIĘTY W WERSJI 1.8 PAKIETU INPUT SYSTEM

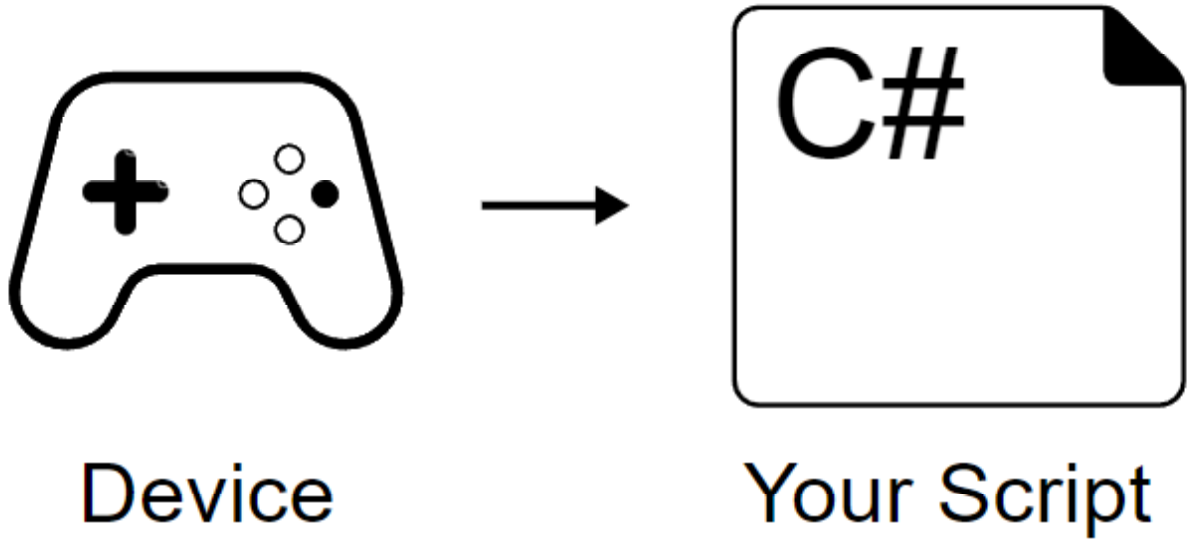
POTOK OSADZONY: SKRYPT

```
1. using UnityEngine;
2. using UnityEngine.InputSystem; // Krok 1: uzycie pakietu
3.
4. public class EmbeddedInputTest : MonoBehaviour
5. {
6.     public InputAction akcjaRuchu; // Krok 2: zmienne pomocnicze
7.     public InputAction akcjaWidoku
8.         ;
9.     private void Start()
10.    {
11.        // Krok 3: aktywacja wszystkich akcji
12.        akcjaRuchu.Enable();
13.        akcjaWidoku.Enable();
14.    }
15.    void Update()
16.    {
17.        // Krok 4: pobranie danych
18.        Vector2 wektorRuchu = akcjaRuchu.ReadValue<Vector2>();
19.        Vector2 wektorKamery = akcjaWidoku.ReadValue<Vector2>();
20.
21.        // Krok 5: wypisanie danych
22.        Debug.Log(string.Format("Ruch: {0}, Kamera: {1}",
23.            wektorRuchu, wektorKamery));
24.    }
25.}
```

OSADZONY POTOK PRACY: EDYCJA



Bezpośredni potok pracy



https://docs.unity3d.com/Packages/com.unity.inputsystem@1.7/manual/Workflow-Direct.html

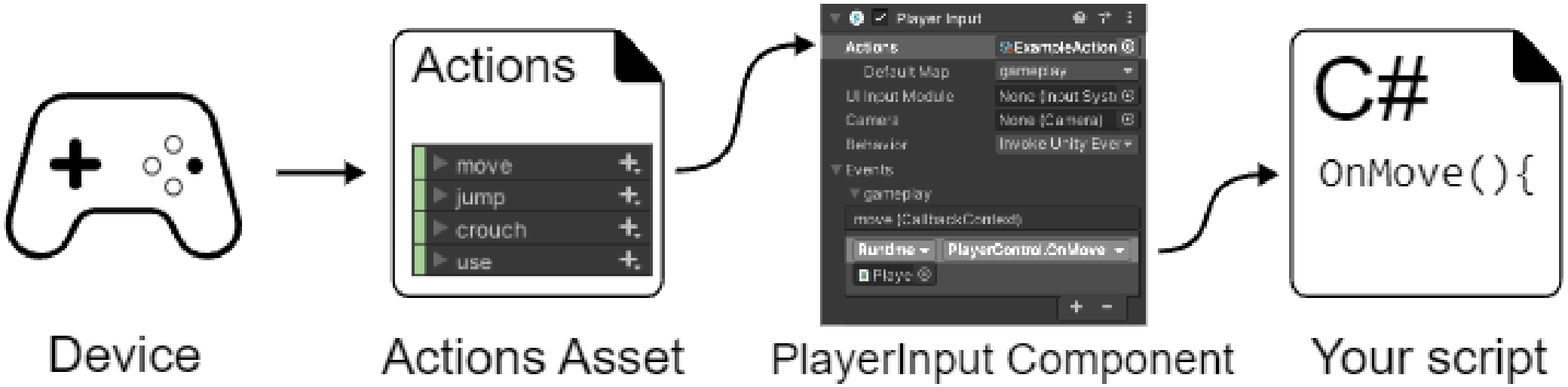
POTOK BEZPOŚREDNI: SKRYPT

```
1.  using System;
2.  using UnityEngine;
3.  using UnityEngine.InputSystem; // Krok 1: uzycie pakietu
4.
5.  public class DirectInputTest : MonoBehaviour
6.  {
7.      Keyboard klawiatura = Keyboard.current;     // Krok 2: pobieranie referencji urzadzen
8.      Mouse mysz = Mouse.current;
9.
10.     // Update is called once per frame
11.     void Update()
12.     {
13.         // Krok 3: pobieranie stanu klawiszy
14.         int gora = Convert.ToInt32(klawiatura.wKey.isPressed);
15.         int dol = Convert.ToInt32(klawiatura.sKey.isPressed);
16.         int lewo = Convert.ToInt32(klawiatura.aKey.isPressed);
17.         int prawo = Convert.ToInt32(klawiatura.dKey.isPressed);
18.
19.         // Krok 4: tworzenie wektora ruchu
20.         Vector2 wektorRuchu = new Vector2(gora - dol, prawo-lewo);
21.
22.         // Krok 5: pobranie pozycji myszy
23.         Vector2 wektorKamery = mysz.position.value;
24.
25.         // Krok 6: wypisanie danych
26.         Debug.Log(string.Format("Ruch: {0}, Kamera: {1}",
27.             wektorRuchu, wektorKamery));
28.     }
29. }
```

KONTROLERY POTOKU BEZPOŚREDNIEGO

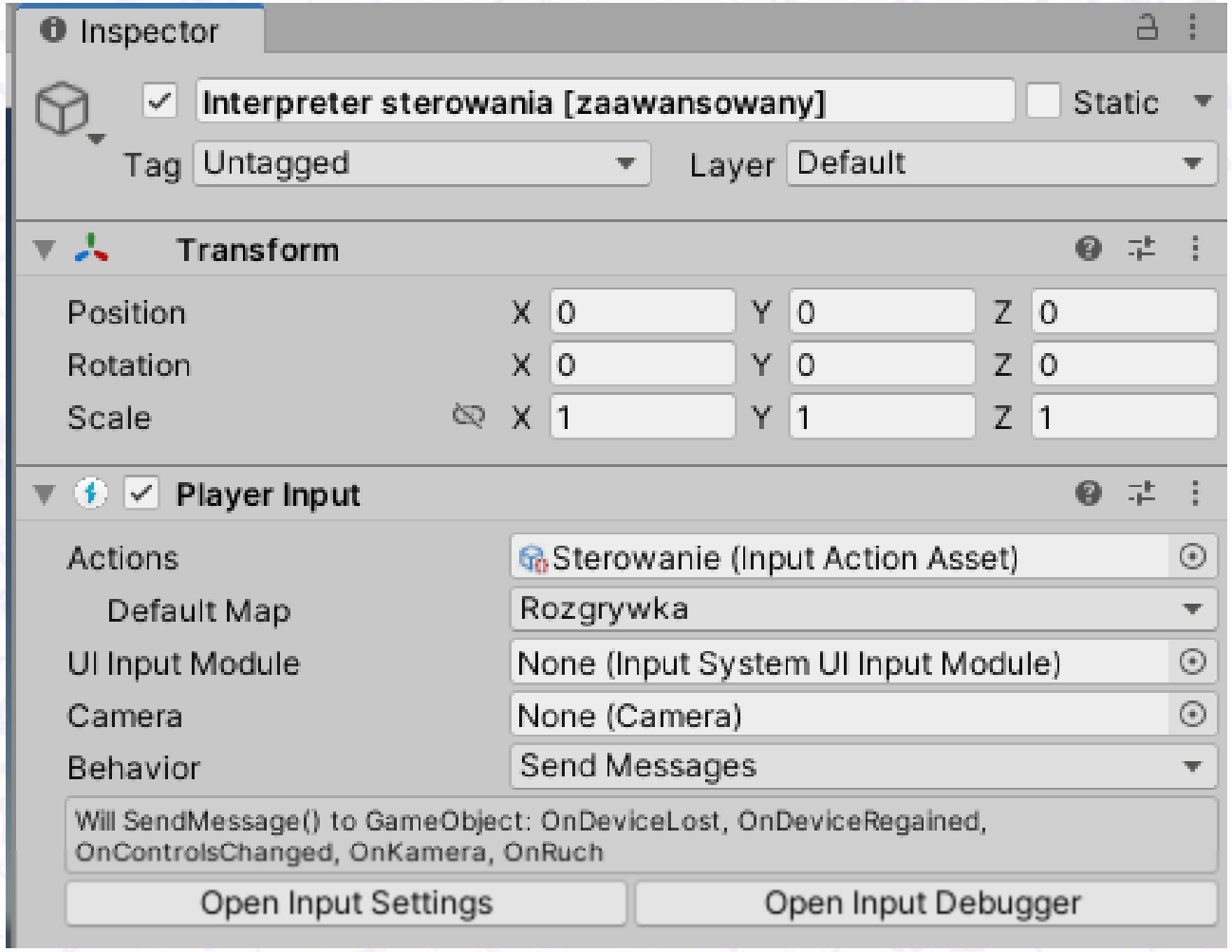
Klasa	Podklasy	Urządzenia
Gamepad	AndroidGamepad, DualShockGamepad, iOSGameController, NimbusGamepadHid, SwitchProControllerHID, WebGLGamepad, XInputController	Joypady
HID	BRAK	Nieznane urządzenia wejścia
Joystick	AndroidJoystick, WebGLJoystick	Joysticki
Keyboard	BRAK	Klawiatury
Pointer	Mouse, Pen, Touchscreen	Urządzenia wskazujące: myszy, ekrany dotykowe, rysiki
Sensor	Accelerometer, AmbientTemperatureSensor, AttitudeSensor, GravitySensor, Gyroscope, HumiditySensor, LightSensor, LinearAccelerationSensor, MagneticFieldSensor, PressureSensor, ProximitySensor, StepCounter	Czujniki: żyroskop, licznik kroków, termometr, akcelerometr…
TrackedDevice	XRController, XRHMD	Urządzenia rozszerzonej rzeczywistości

ZAAWANSOWANY POTOK PRACY



https://docs.unity3d.com/Packages/com.unity.inputsystem@1.7/manual/Workflow-PlayerInput.html

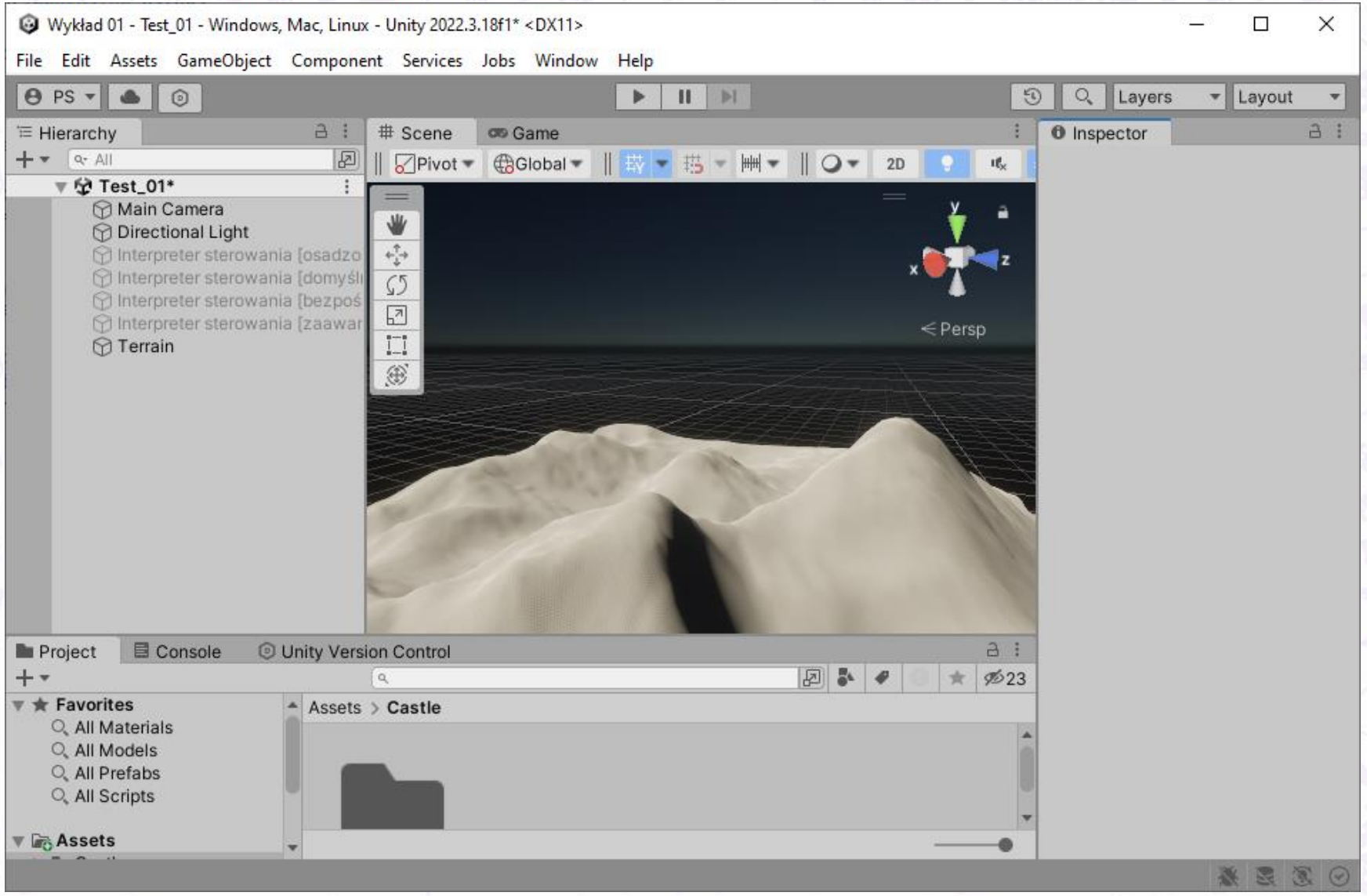
KOMPONENT PLAYERINPUT



POTOK ZAAWANSOWANY: SKRYPT

```
1. using UnityEngine;
2. using UnityEngine.InputSystem; // Krok 1: uzycie pakietu
3.
4. public class AdvancedInputTest : MonoBehaviour
5. {
6.     private Vector2 wektorRuchu = Vector2.zero; // Krok 2: zmienne pomocnicze
7.     private Vector2 wektorKamery = Vector2.zero;
8.
9.     public void OnRuch(InputValue value)
10.    {
11.        // Krok 3: odczytanie wektora ruchu w funkcji OnRuch
12.        wektorRuchu = value.Get<Vector2>();
13.    }
14.    public void OnKamera(InputValue value)
15.    {
16.        // Krok 4: odczytanie wektora kamery w funkcji OnKamera
17.        wektorKamery = value.Get<Vector2>();
18.    }
19.
20.    public void Update()
21.    {
22.        // Krok 5: wypisanie danych
23.        Debug.Log(string.Format("Ruch: {0}, Kamera: {1}",
24.            wektorRuchu, wektorKamery));
25.    }
26.}
```

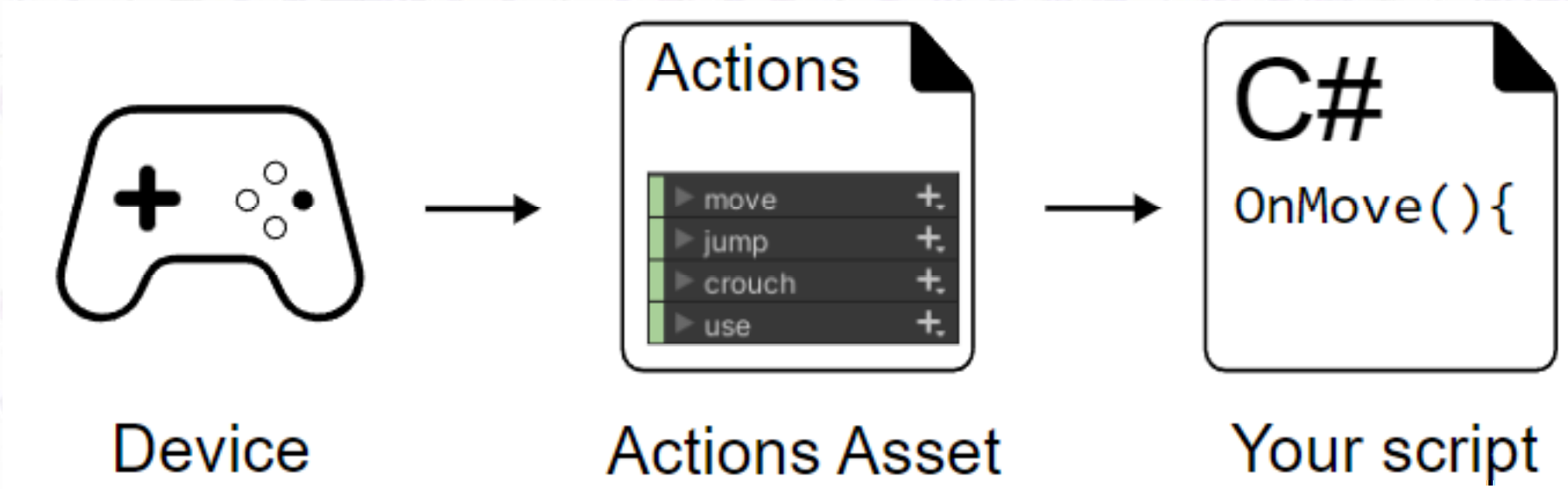
SCENA TESTOWA



PROBLEMY PRAKTYCZNE

Problem 1: sterowanie kamerą w czterech kierunkach

ZALECANY POTOK PRACY: PRZYPOMNIENIE



https://docs.unity3d.com/Packages/com.unity.inputsystem@1.7/manual/Workflow-ActionsAsset.html

POTOK ZALECANY: SKRYPT

```
1. using UnityEngine;
2. using UnityEngine.InputSystem; // Krok 1: uzycie pakietu
3.
4. public class RecommendedInputTest : MonoBehaviour
5. {
6.     public InputActionAsset actions; // Krok 2: referencja do zasobu
7.     private InputAction akcjaRuchu; // Krok 3: zmienne pomocnicze
8.     private InputAction akcjaWidoku;
9.
10.    void Start()
11.    {
12.        // Krok 4: wyszukanie referencji do przypisania
13.        akcjaRuchu = actions.FindActionMap("Rozgrywka").FindAction("Ruch");
14.        akcjaWidoku = actions.FindActionMap("Rozgrywka").FindAction("Kamera");
15.    }
16.    void Update()
17.    {
18.        // Krok 5: pobranie danych
19.        Vector2 wektorRuchu = akcjaRuchu.ReadValue<Vector2>();
20.        Vector2 wektorKamery = akcjaWidoku.ReadValue<Vector2>();
21.
22.        // Krok 6: wypisanie danych
23.        Debug.Log(string.Format("Ruch: {0}, Kamera: {1}",
24.            wektorRuchu, wektorKamery));
25.    }
26.}
```

RUCH KAMERĄ: SKRYPT SKRÓCONY

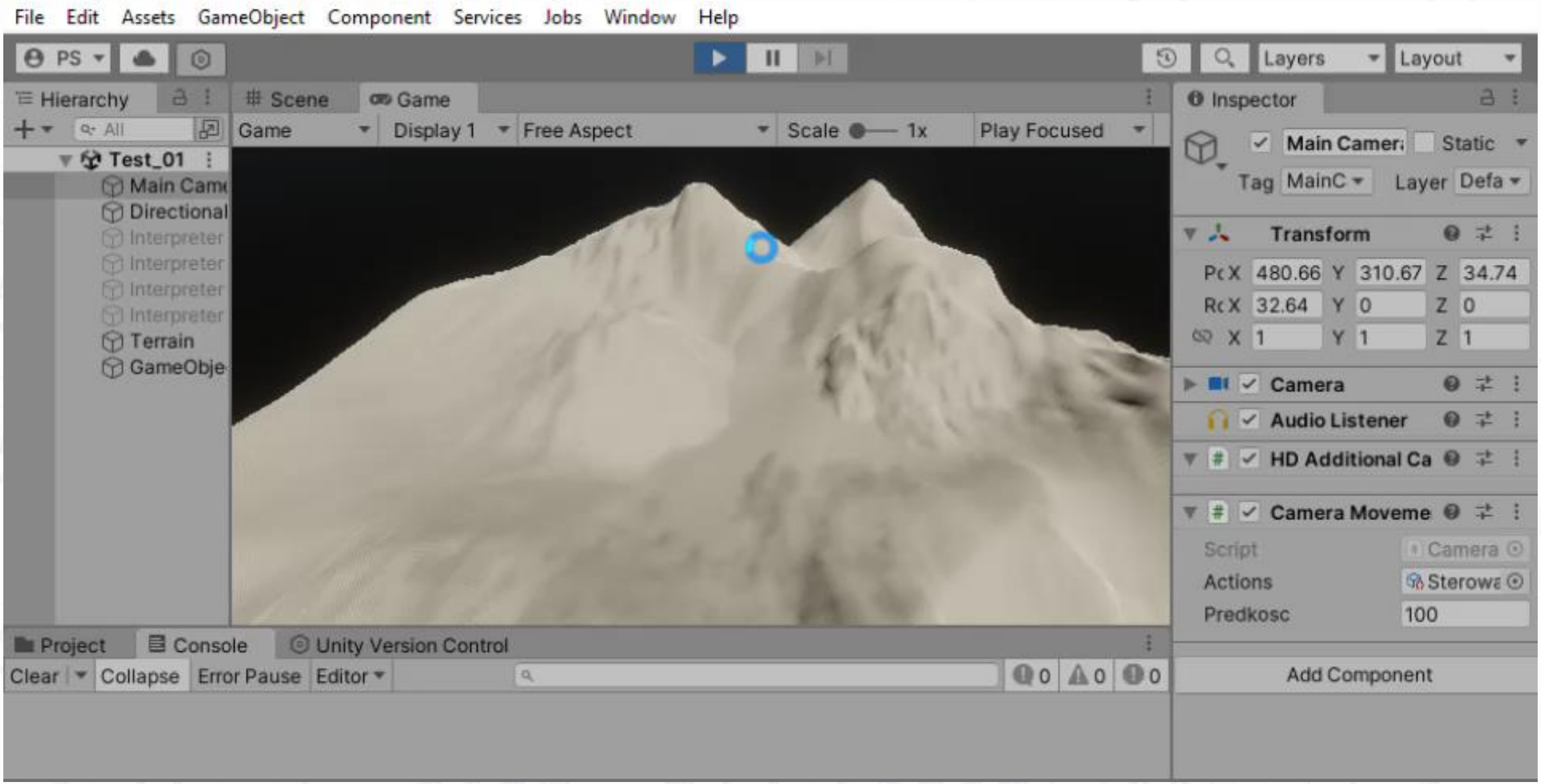
```
1. using UnityEngine;
2. using UnityEngine.InputSystem; // Krok 1: uzycie pakietu
3.
4. public class CameraMovement : MonoBehaviour
5. {
6.     public InputActionAsset actions; // Krok 2: referencja do zasobu
7.     private InputAction akcjaRuchu; // Krok 3: zmienne pomocnicze
8.     public float predkosc = 100f;
9.
10.    void Start()
11.    {
12.        // Krok 4: wyszukanie referencji do przypisania
13.        akcjaRuchu = actions.FindActionMap("Rozgrywka").FindAction("Ruch");
14.
15.        // Krok 5: aktywacja przestrzeni akcji
16.        actions.FindActionMap("Rozgrywka").Enable();
17.    }
18.    void Update()
19.    {
20.        // Krok 6: pobranie danych
21.        Vector2 wektorRuchu = akcjaRuchu.ReadValue<Vector2>();
22.
23.        // Krok 7: ruch kamery
24.        // ???
25.    }
26.}
```

RUCH KAMERĄ: SKRYPT UKOŃCZONY

```
1. using UnityEngine;
2. using UnityEngine.InputSystem; // Krok 1: uzycie pakietu
3. using UnityEngine.InputSystem.Controls;

4. public class CameraMovement : MonoBehaviour
5. {
6.     public InputActionAsset actions; // Krok 2: referencja do zasobu
7.     private InputAction akcjaRuchu; // Krok 3: zmienne pomocnicze
8.     public float predkosc = 100f;
9.
10.    void Start()
11.    {
12.        // Krok 4: wyszukanie referencji do przypisania
13.        akcjaRuchu = actions.FindActionMap("Rozgrywka").FindAction("Ruch");
14.
15.        // Krok 5: aktywacja przestrzeni akcji
16.        actions.FindActionMap("Rozgrywka").Enable();
17.    }
18.    void Update()
19.    {
20.        // Krok 6: pobranie danych
21.        Vector2 wektorRuchu = akcjaRuchu.ReadValue<Vector2>();
22.
23.        // Krok 7: ruch kamery
24.        Vector3 zmiana = new Vector3(wektorRuchu.x, 0, wektorRuchu.y);
25.        zmiana *= Time.deltaTime * predkosc;
26.        transform.position += zmiana;
27.    }
28. }
```

RUCH KAMERĄ W CZTERECH KIERUNKACH



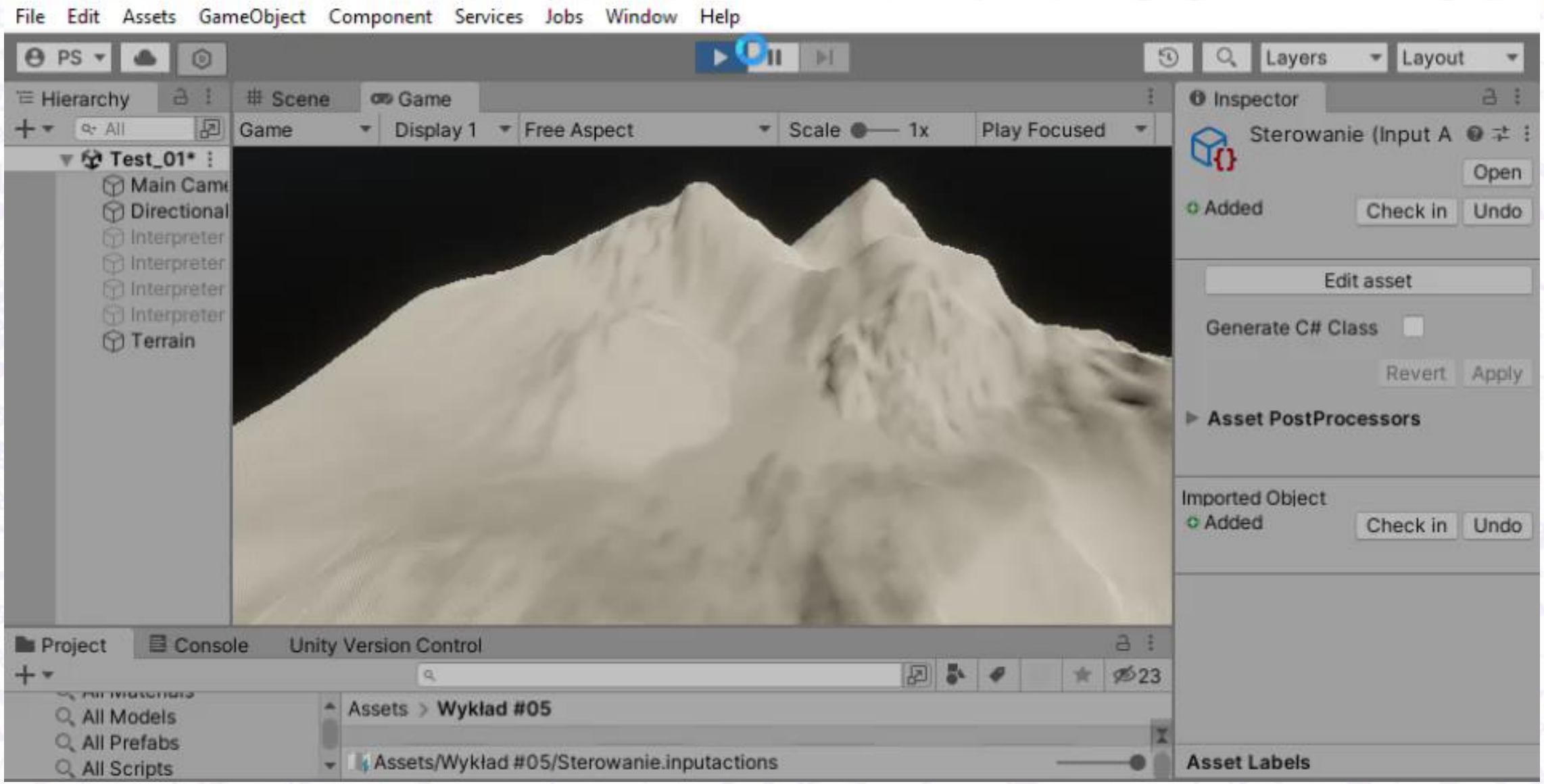
PROBLEMY PRAKTYCZNE

Problem 2: rozglądanie się kamerą

OBRÓT KAMERĄ: SKRYPT

```
1. using UnityEngine;
2. using UnityEngine.InputSystem; // Krok 1: uzycie pakietu
3.
4. public class CameraRotation : MonoBehaviour
5. {
6.     public InputActionAsset actions; // Krok 2: referencja do zasobu
7.     private InputAction akcjaKamery; // Krok 3: zmienne pomocnicze
8.     public float predkosc = 5f;
9.
10.    void Start()
11.    {
12.        // Krok 4: wyszukanie referencji do przypisania
13.        akcjaKamery = actions.FindActionMap("Rozgrywka").FindAction("Kamera");
14.
15.        // Krok 5: aktywacja przestrzeni akcji
16.        actions.FindActionMap("Rozgrywka").Enable();
17.    }
18.    void Update()
19.    {
20.        // Krok 6: pobranie danych
21.        Vector2 wektorObrotu = akcjaKamery.ReadValue<Vector2>();
22.
23.        // Krok 7: ruch kamery
24.        Vector3 zmiana = new Vector3(wektorObrotu.y, wektorObrotu.x, 0);
25.        zmiana *= Time.deltaTime * predkosc;
26.        transform.eulerAngles += zmiana;
27.    }
28. }
```

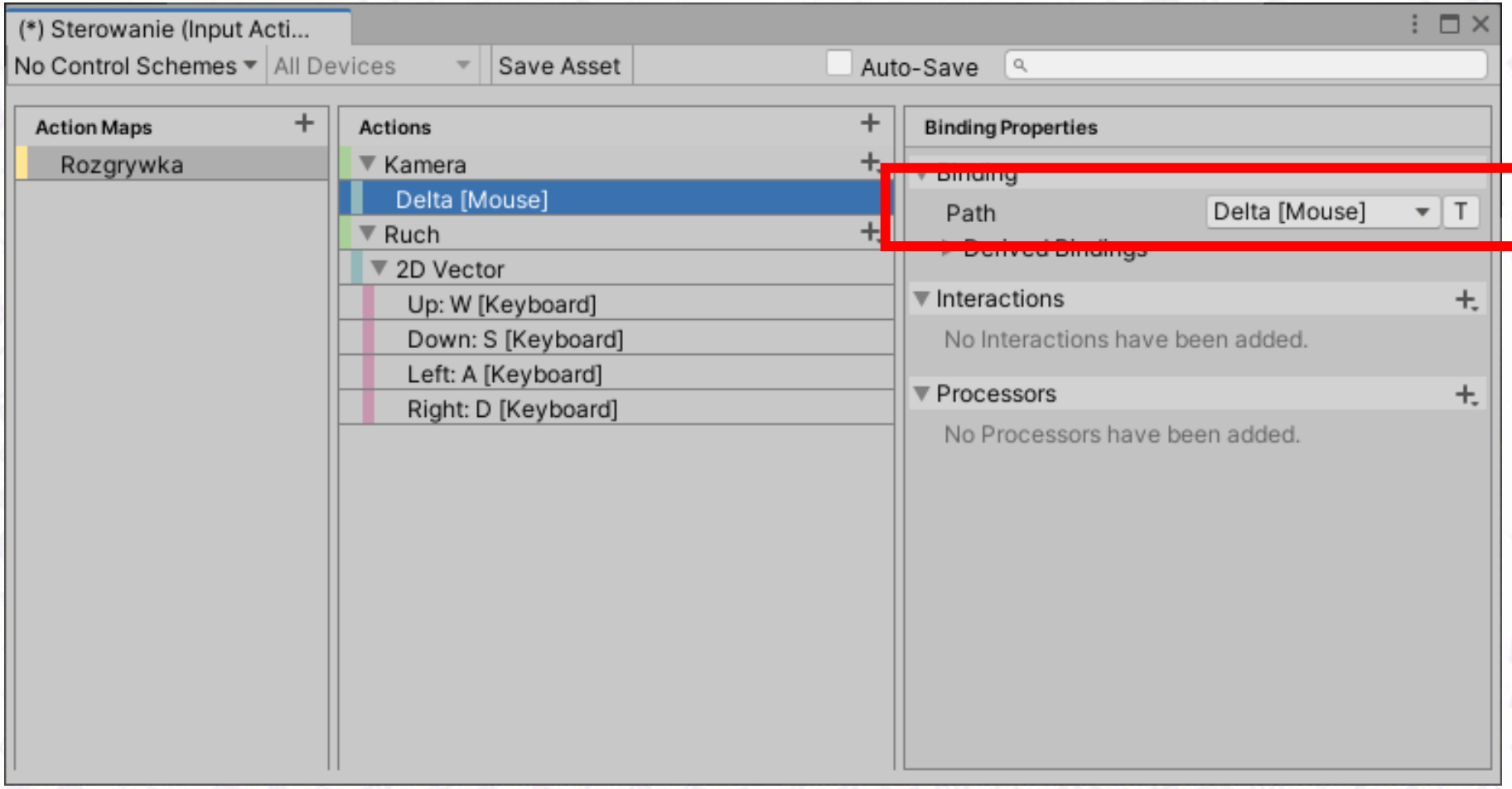
RUCH KAMERĄ W CZTERECH KIERUNKACH



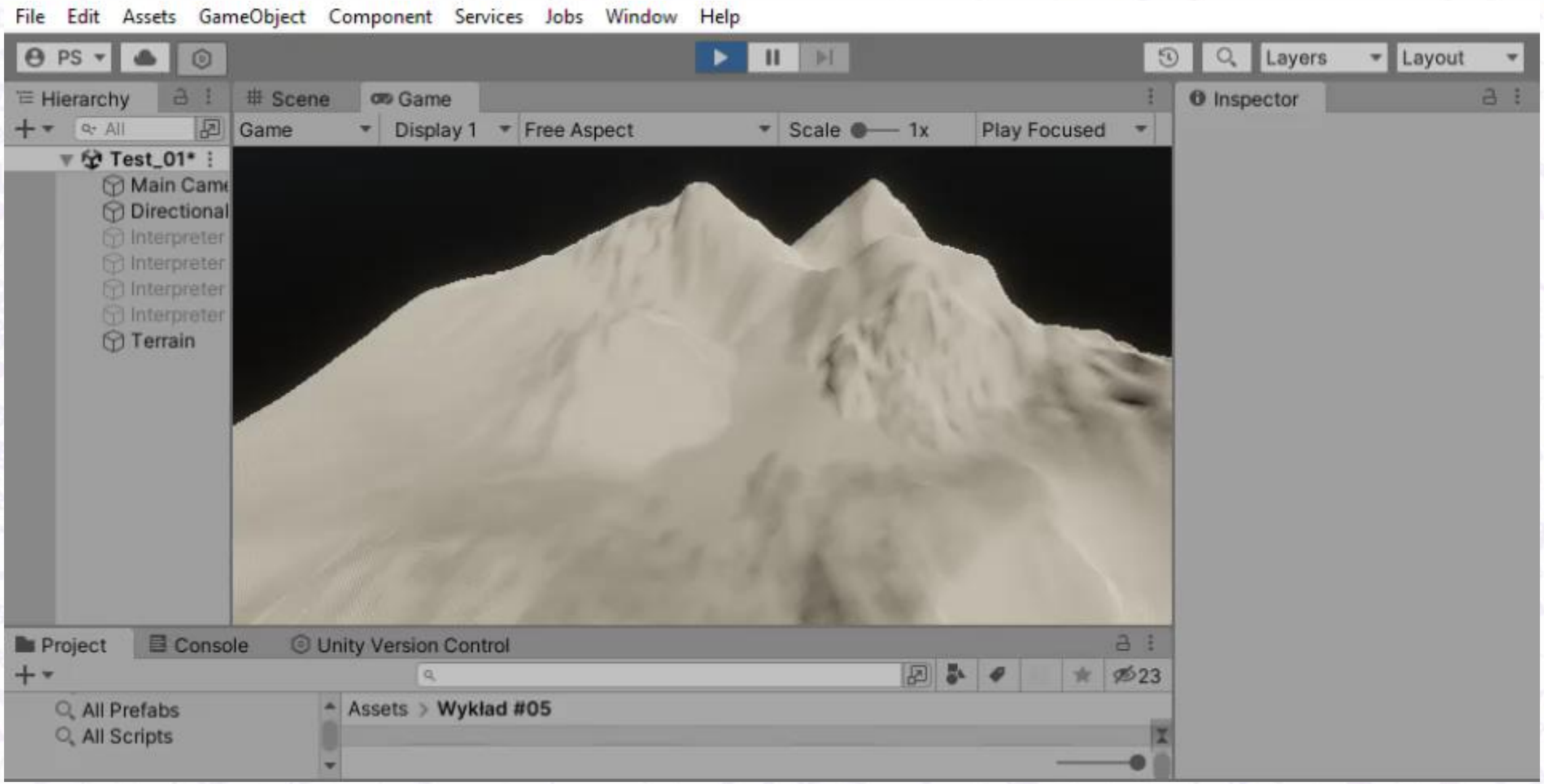
OBRÓT KAMERĄ: SKRYPT ???

```
1. using UnityEngine;
2. using UnityEngine.InputSystem; // Krok 1: uzycie pakietu
3.
4. public class CameraRotation : MonoBehaviour
5. {
6.     public InputActionAsset actions; // Krok 2: referencja do zasobu
7.     private InputAction akcjaKamery; // Krok 3: zmienne pomocnicze
8.     public float predkosc = 5f;
9.
10.    void Start()
11.    {
12.        // Krok 4: wyszukanie referencji do przypisania
13.        akcjaKamery = actions.FindActionMap("Rozgrywka").FindAction("Kamera");
14.
15.        // Krok 5: aktywacja przestrzeni akcji
16.        actions.FindActionMap("Rozgrywka").Enable();
17.    }
18.    void Update()
19.    {
20.        // Krok 6: pobranie danych
21.        Vector2 wektorObrotu = akcjaKamery.ReadValue<Vector2>();
22.
23.        // Krok 7: ruch kamery
24.        Vector3 zmiana = new Vector3(wektorObrotu.y, wektorObrotu.x, 0);
25.        zmiana *= Time.deltaTime * predkosc;
26.        transform.eulerAngles += zmiana;
27.    }
28. }
```

Mapa akcji


(*) Sterowanie (Input Acti...
No Control Schemes
All Devices
Save Asset
Auto-Save
Action Maps
Rozgrywka
Actions
Kamera
Delta [Mouse]
Ruch
2D Vector
Up: W [Keyboard]
Down: S [Keyboard]
Left: A [Keyboard]
Right: D [Keyboard]
Binding Properties
Path
Delta [Mouse]
T
Interactions
No Interactions have been added.
Processors
No Processors have been added.

Ruch kamerą w czterech kierunkach



Dziękuję za uwagę

Proszę o pytania