

UNITY- 2

Ing. Timotej Sobota
doc. Ing. Branislav Sobota, PhD.

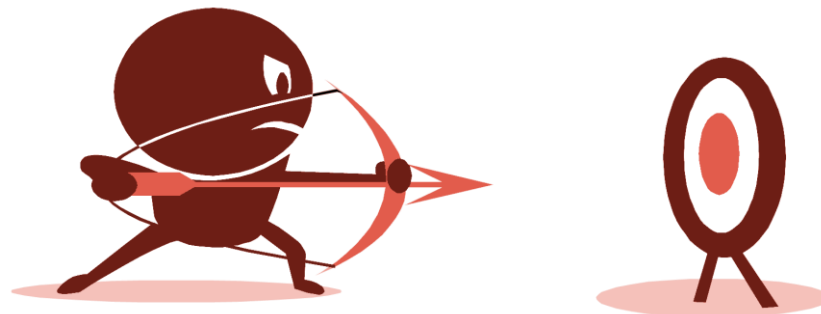
Katedra počítačov a informatiky, FEI TU v Košiciach

© 09

© 2026

CIEĽ CVIČENIA

- Zadanie v Unity
- skriptovanie C#
- Input Actions
- Pridávanie tagu
- Vytvorenie jednoduchkej aplikácie



ZADANIE - UNITY

- **Naimportované modely (SketchUp / Blender)**
 - Vlastné modely sa hodnotia lepšie ako iba stiahnuté
- **Dynamika prostredia / pohyb**
- **Pohyb hráča / kamery**
- **Box colliders**
- **Vyplnené prostredie**
- **Formát odovzdania:**
 - Desktop appka
 - Splitscreen appka na telefóne
 - VR headset appka

SKRIPTOVANIE V UNITY

- **Definuje správanie objektov**
- Používa **jazyk C#**
- Skripty sú **komponenty**
- Pridávajú sa na objekty
- **Funkcie:**
 - **Start()** – spustí sa raz
 - **Update()** – opakuje sa



- Je potrebné si doinštalovať aj správne Unity balíčky

VYTVORENIE SKRIPTU

- v **Project**

1. RightClick
2. Create
3. Scripting

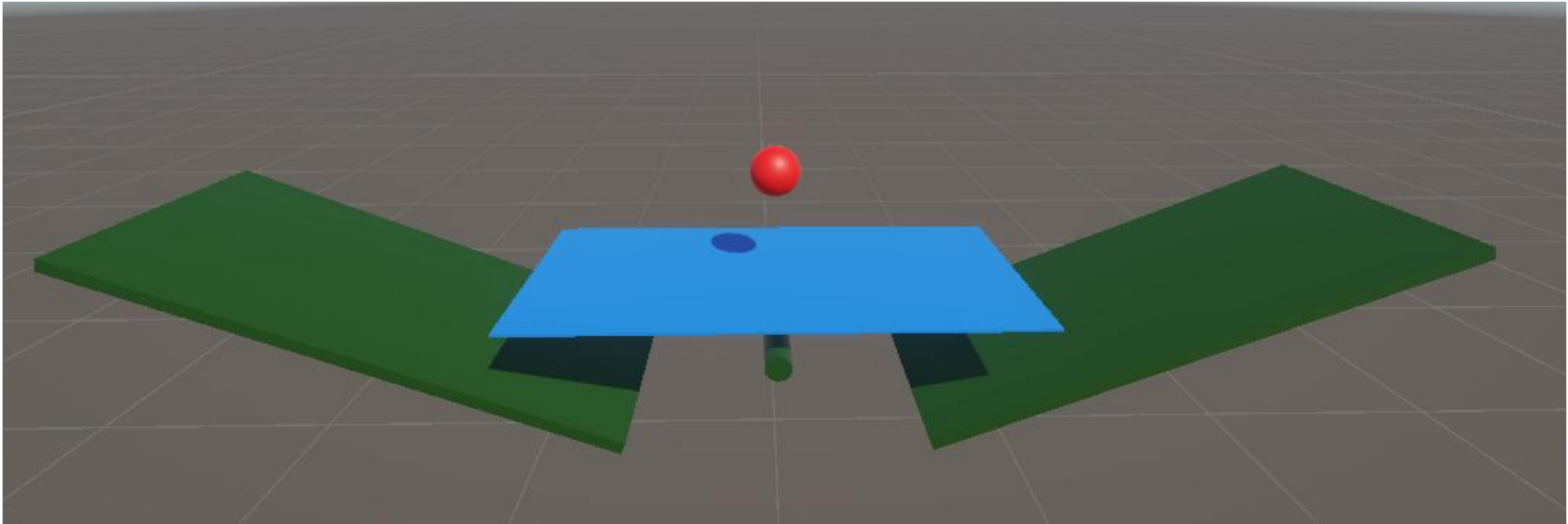
- v **Inspector-e** objektu

1. Add Component
2. Vypísať názov svojho scriptu
3. Enter

- Drag & Drop -> Inspector objektu -> Add Component

ÚLOHA

- Vytvorenie **novej scény**

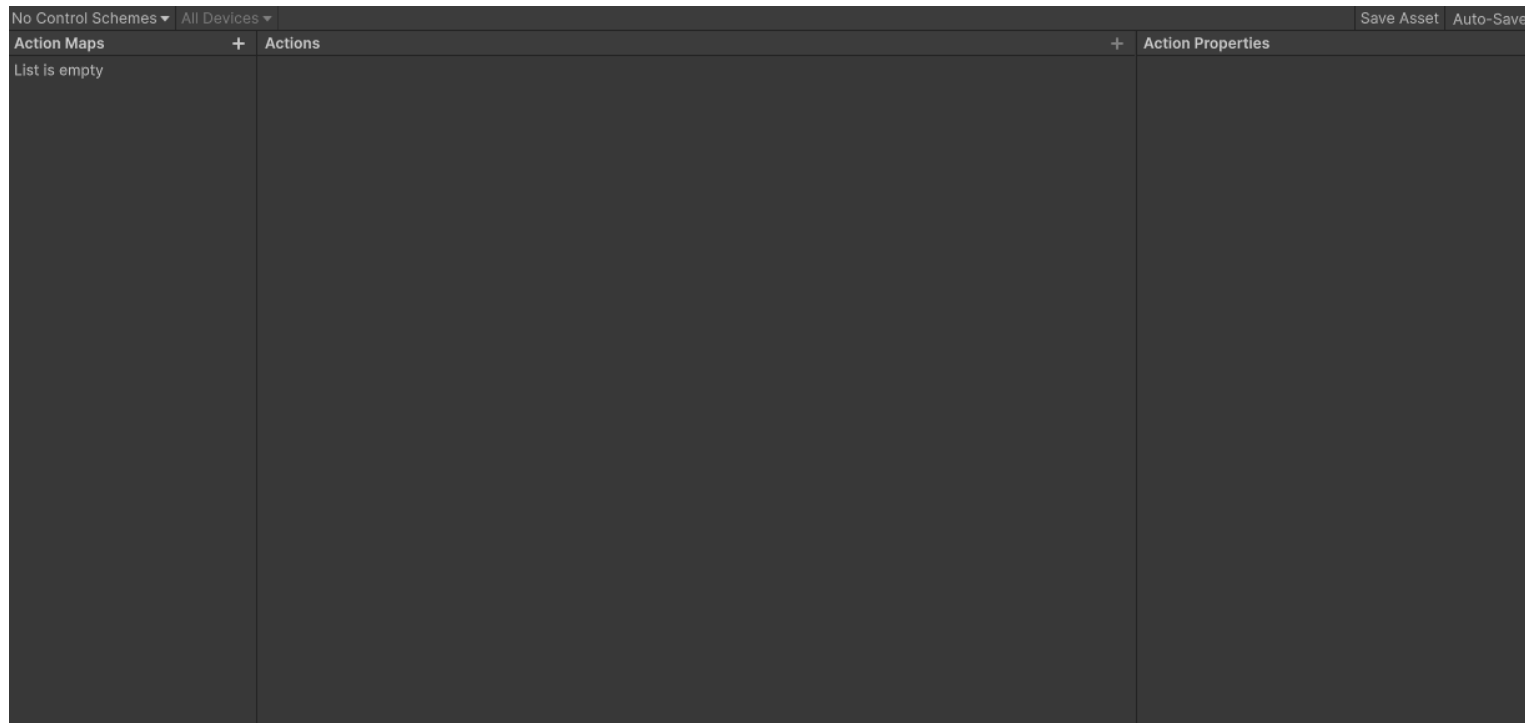
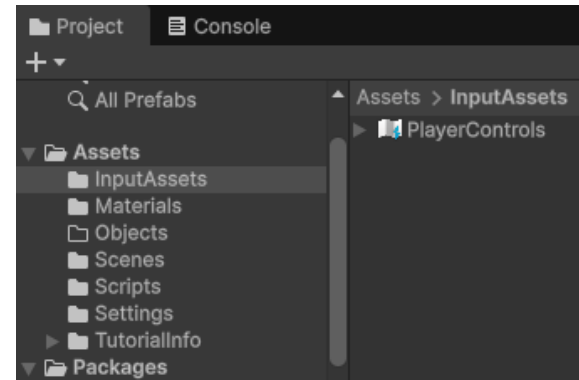


INPUT ACTIONS (INPUT SYSTEM)

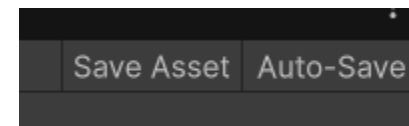
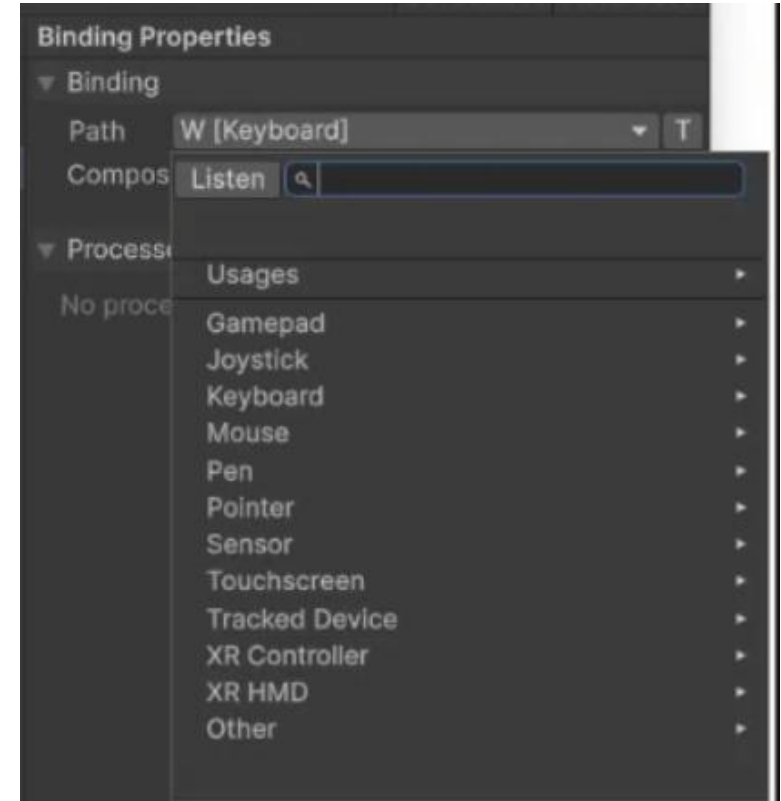
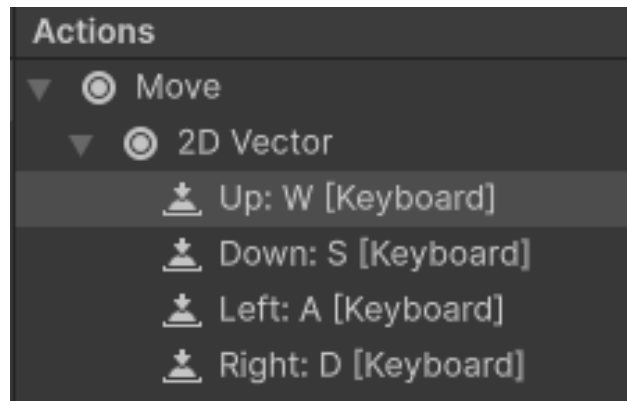
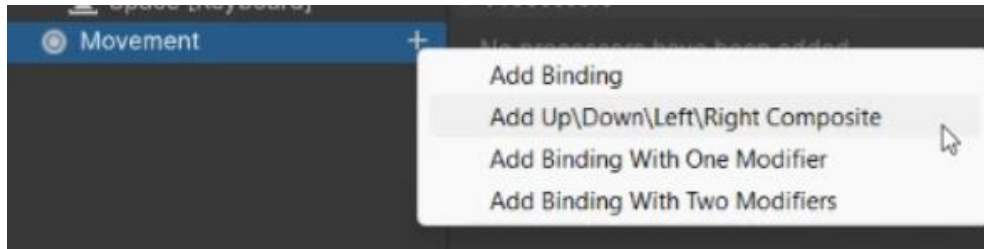
- **Nový systém vstupu v Unity**
- Nahrádza starý Input Manager
- **Definovanie vstupov:**
 - **Klávesnica**
 - **Myš**
 - **Gamepad**
- Používa Actions a Bindings
- Ukladá sa ako `.inputactions` súbor

INPUT ACTIONS (INPUT SYSTEM)

1. RightClick
2. Create
3. Input Actions



INPUT ACTIONS (INPUT SYSTEM)



MOVEMENT SCRIPT

```
using UnityEngine;
using UnityEngine.InputSystem;

public class SphereMovement : MonoBehaviour
{
    public float speed = 5f;
    private Rigidbody rb;
    private Vector2 moveInput;

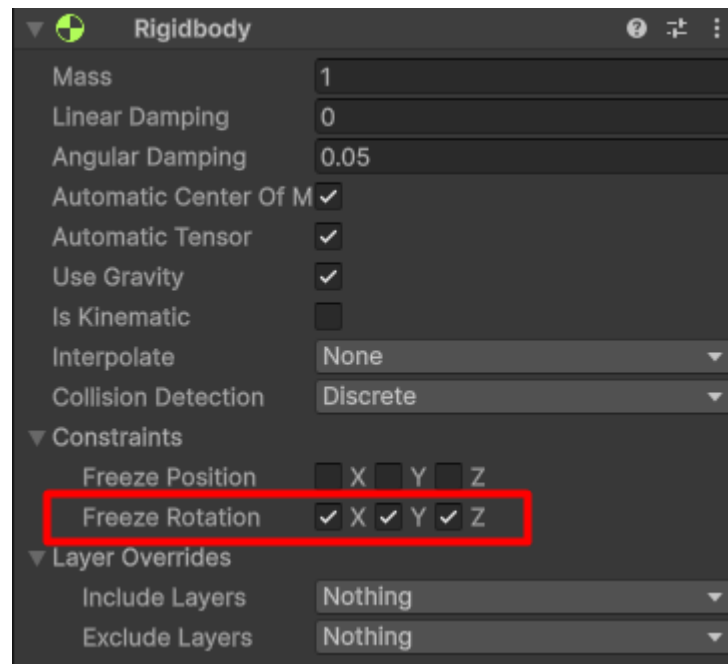
    void Start()
    {
        rb = GetComponent<Rigidbody>();
    }

    void FixedUpdate()
    {
        Vector3 movement = new Vector3(moveInput.x, 0f, moveInput.y);
        rb.MovePosition(rb.position + movement * speed * Time.fixedDeltaTime);
    }

    // This MUST match the action name: "Move"
    public void OnMove(InputValue value)
    {
        moveInput = value.Get<Vector2>();
    }
}
```

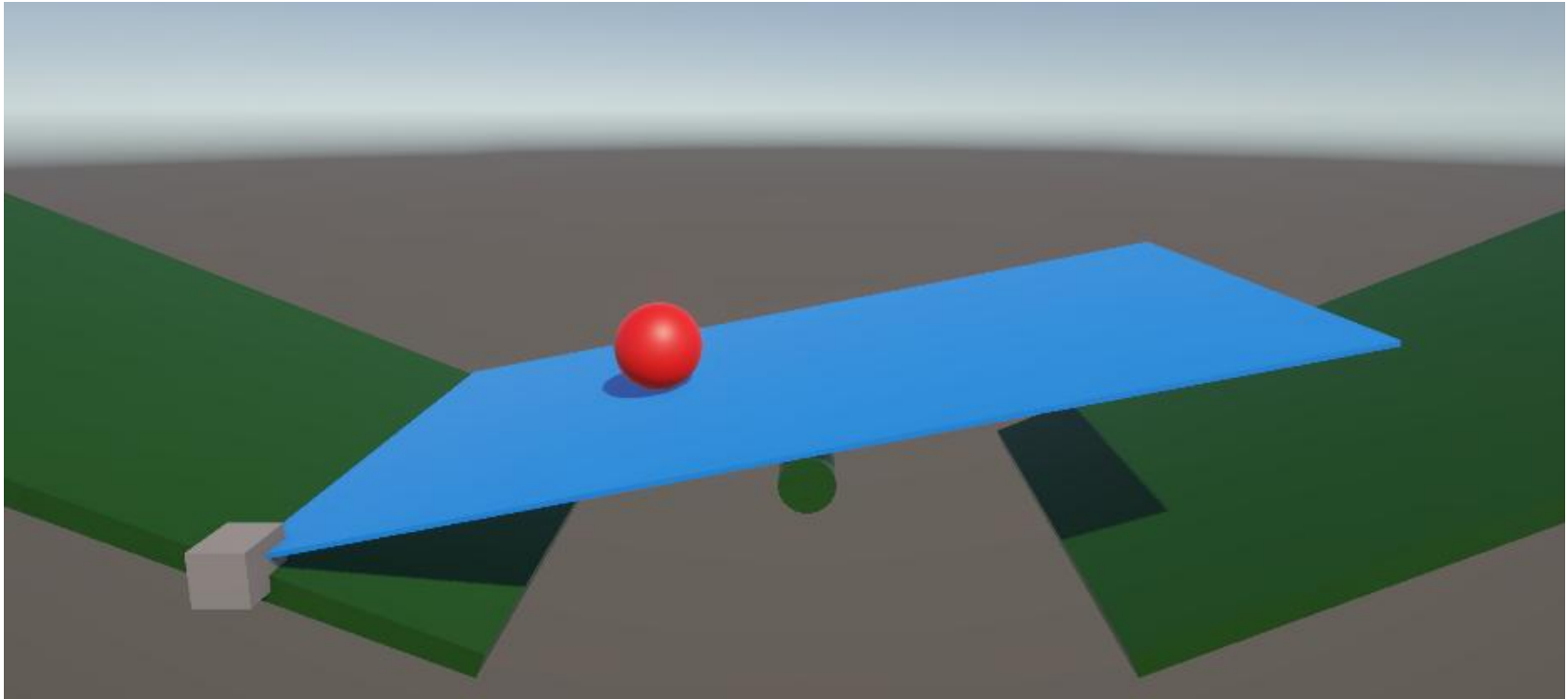
MOVEMENT SCRIPT

- Overté správanie scriptu



ÚLOHA

- Pridanie detekcie kolízií platforiem

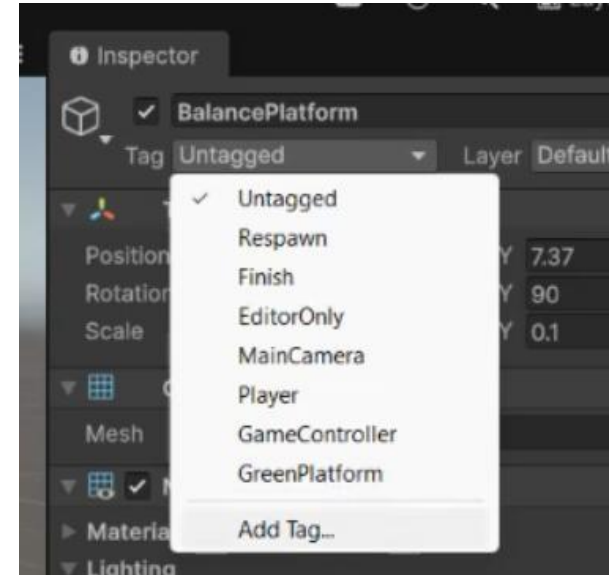


TAGY

- Tag je jednoduchý spôsob, ako si označiť objekty v scéne.
- Tagy používame hlavne v skriptoch, keď chceme:
 - zistiť, **s akým objektom sme sa stretli**
 - reagovať rôzne podľa typu objektu
 - filtrovať objekty (napr. iba nepriatelía)

TAGY

- Ako nastaviť tag v Unity
 - Vyber objekt v scéne
 - V hornej časti **Inspectoru** zvol' **Tag**
 - Klikni **Add Tag...**
 - Vytvor nový tag
 - Prirad' ho objektu
- názov tagu musí byť **presný** (veľké/malé písmená)
- objekt môže mať **iba jeden tag**



ÚLOHA

- Zeleným platformám nastavte tag:

GreenPlatform



ÚLOHA

- Modrej platforme vytvorte skrip pre detekciu kolízie s platformou, ktorá má tag:

GreenPlatform



COLLISION SCRIPT

```
using UnityEngine;

public class SpawnCubeOnCollision : MonoBehaviour
{
    public GameObject cubePrefab;

    void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.CompareTag("GreenPlatform"))
        {
            // Get contact point (where they touched)
            ContactPoint contact = collision.contacts[0];
            Vector3 spawnPosition = contact.point;

            // Spawn cube
            Instantiate(cubePrefab, spawnPosition, Quaternion.identity);
        }
    }
}
```

ÚLOHA

1. Vytvorte na scéne cube objekt
2. Presunte ho z **Hierarchy** do **Project**
3. **Vymažte** objekt cube zo scény (voliteľné)
4. Vyberte si modrú platformu a presuňte prefab kocky do poľa „Cube Prefab“



COLLISION SCRIPT

- Overté správanie scriptu



PRIDANIE DYNAMIKY DO SCÉNY

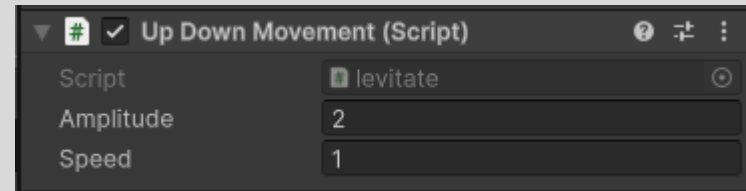
```
using UnityEngine;
```

```
public class UpDownMovement : MonoBehaviour
{
    public float amplitude = 25f; // how high it goes
    public float speed = 2f;      // how fast it moves

    private Vector3 startPos;

    void Start()
    {
        startPos = transform.position;
    }

    void Update()
    {
        float y = Mathf.Sin(Time.time * speed) * amplitude;
        transform.position = startPos + new Vector3(0, y, 0);
    }
}
```



PRIDANIE DYNAMIKY DO SCÉNY

```
using UnityEngine;
```

```
public class Rotate : MonoBehaviour
```

```
{
```

```
    [Range(1,50)]
```

```
    public float speed = 1;
```

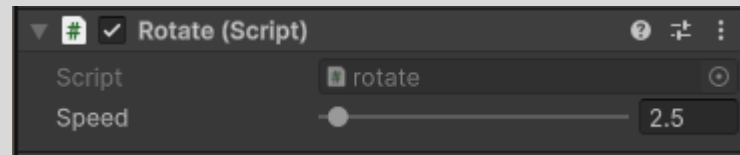
```
    void Update()
```

```
    {
```

```
        transform.Rotate(0, 50 * Time.deltaTime * speed, 0);
```

```
    }
```

```
}
```



ÚLOHY NA SAMOSTATNÉ RIEŠENIE

1. Pridajte možnosť skoku (Jump)
2. Nastavte aby sa rotujúce kocky po určitom čase zničili
3. Zmeňte jednu platformu na inú farbu, pridajte jej iný tag a nastavte aby sa pri kolízii s ňou objavil iný objekt



Q & A

branislav.sobota@tuke.sk
timotej.sobota@tuke.sk

Katedra počítačov a informatiky, FEI TU v Košiciach

© 2026