

UNITY- 3

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

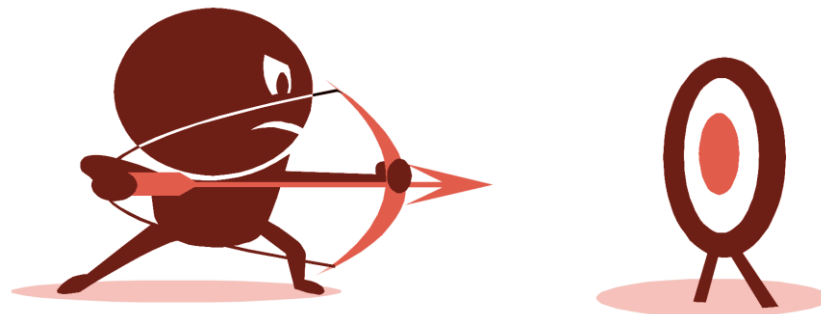
Katedra počítačov a informatiky, FEI TU v Košiciach

© 10

© 2026

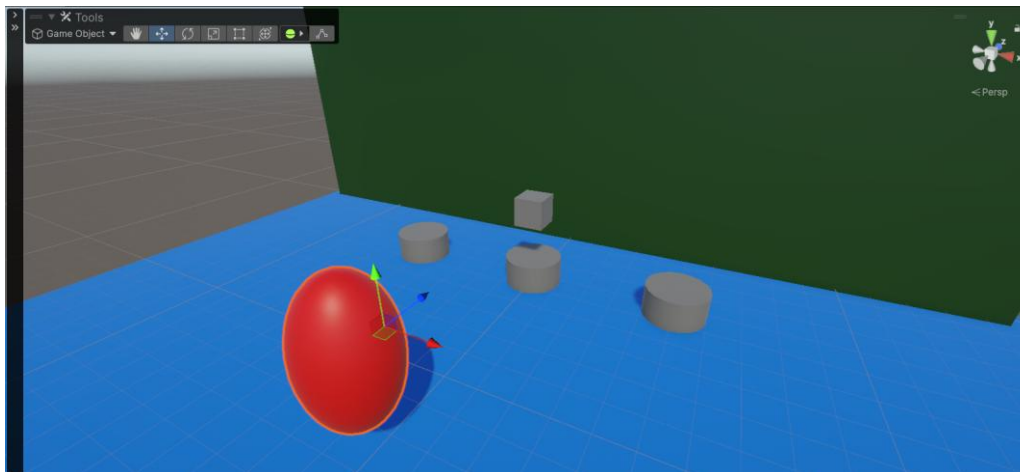
CIEĽ CVIČENIA

- Pohyb myši
- Look-At mechanika
- Vytvorenie VR aplikácie do telefónu



ÚLOHA

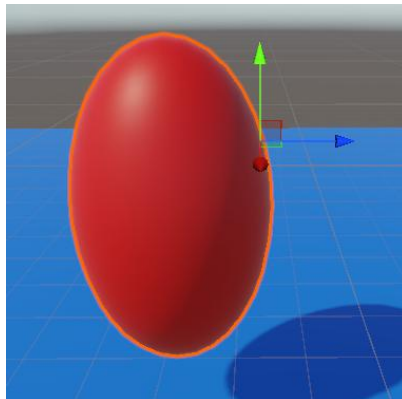
- Vytvorenie **novej scény**:
 - **Podlaha**
 - **Avatar**
 - **Rôzne objekty**



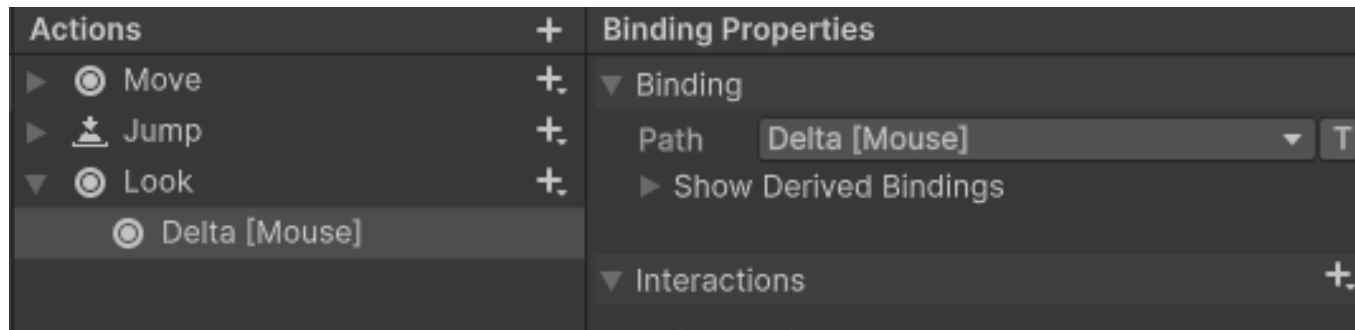
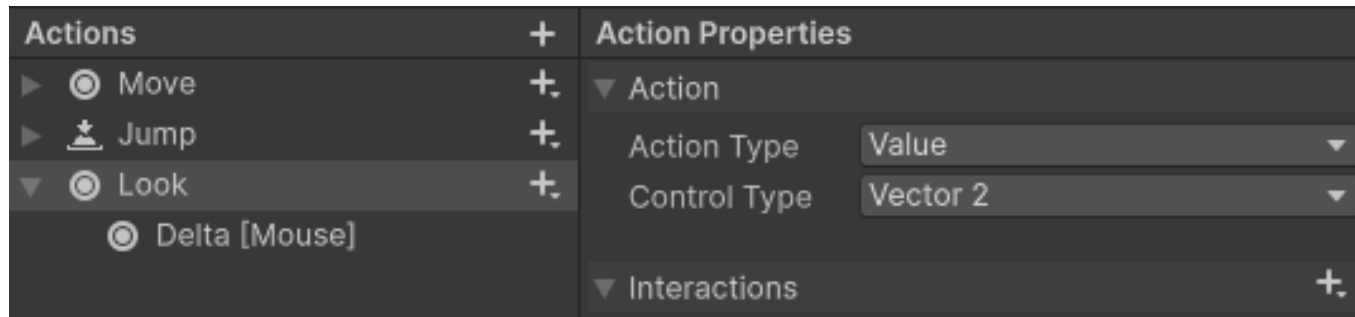
AVATAR / HRÁČ

Úloha:

1. Vytvorte objekt a nastavte mu **kameru** ako **child** objektu, s ktorým sa budeme pohybovať
2. Kameru nastavte na vhodné miesto
3. Pridajte **novú action** do vášho **Input Action**
4. Pridajte **script na pohyb myši**
5. Pridajte **script na pohyb avatara**



NOVÁ INPUT ACTION



MOUSE SCRIPT

```
using UnityEngine;
using UnityEngine.InputSystem;

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

    private Vector2 lookInput;
    private float xRotation = 0f;

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

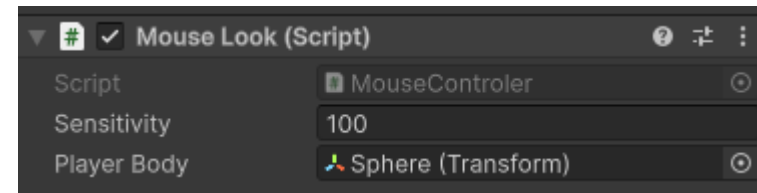
    public void OnLook(InputValue value)
    {
        lookInput = value.Get<Vector2>();
    }

    void Update()
    {
        float mouseX = lookInput.x * sensitivity * Time.deltaTime;
        float mouseY = lookInput.y * sensitivity * Time.deltaTime;

        // Vertical rotation (camera only)
        xRotation -= mouseY;
        xRotation = Mathf.Clamp(xRotation, -90f, 90f);
        transform.localRotation = Quaternion.Euler(xRotation, 0f, 0f);

        // Horizontal rotation (player body)
        if (playerBody != null)
        {
            playerBody.Rotate(Vector3.up * mouseX);
        }
    }
}
```

- Tento script pridajte na **Camera** object



- Nezabudnuť** pridať referenciu na **Parent** objekt

MOVEMENT SCRIPT

```
using UnityEngine;
using UnityEngine.InputSystem;

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

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

    public void OnMove(InputValue value)
    {
        moveInput = value.Get<Vector2>();
    }

    void FixedUpdate()
    {
        // get direction relative to player rotation
        Vector3 move = transform.forward * moveInput.y + transform.right *
moveInput.x;

        rb.MovePosition(rb.position + move * speed * Time.fixedDeltaTime);
    }
}
```

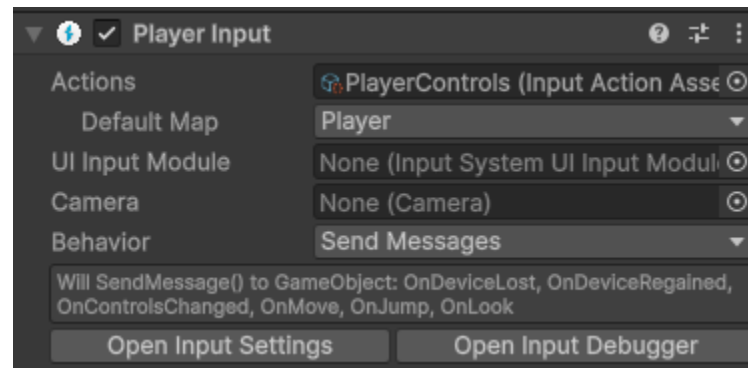
- Tento script pridajte na **Parent** objekt

SCRIPTY

- Pre oba parent a child objekty (Avatar a kamera) pridať komponent

Player Input

- **Actions** → vaše **Input Action**
- **Behavior** → **Send Messages**

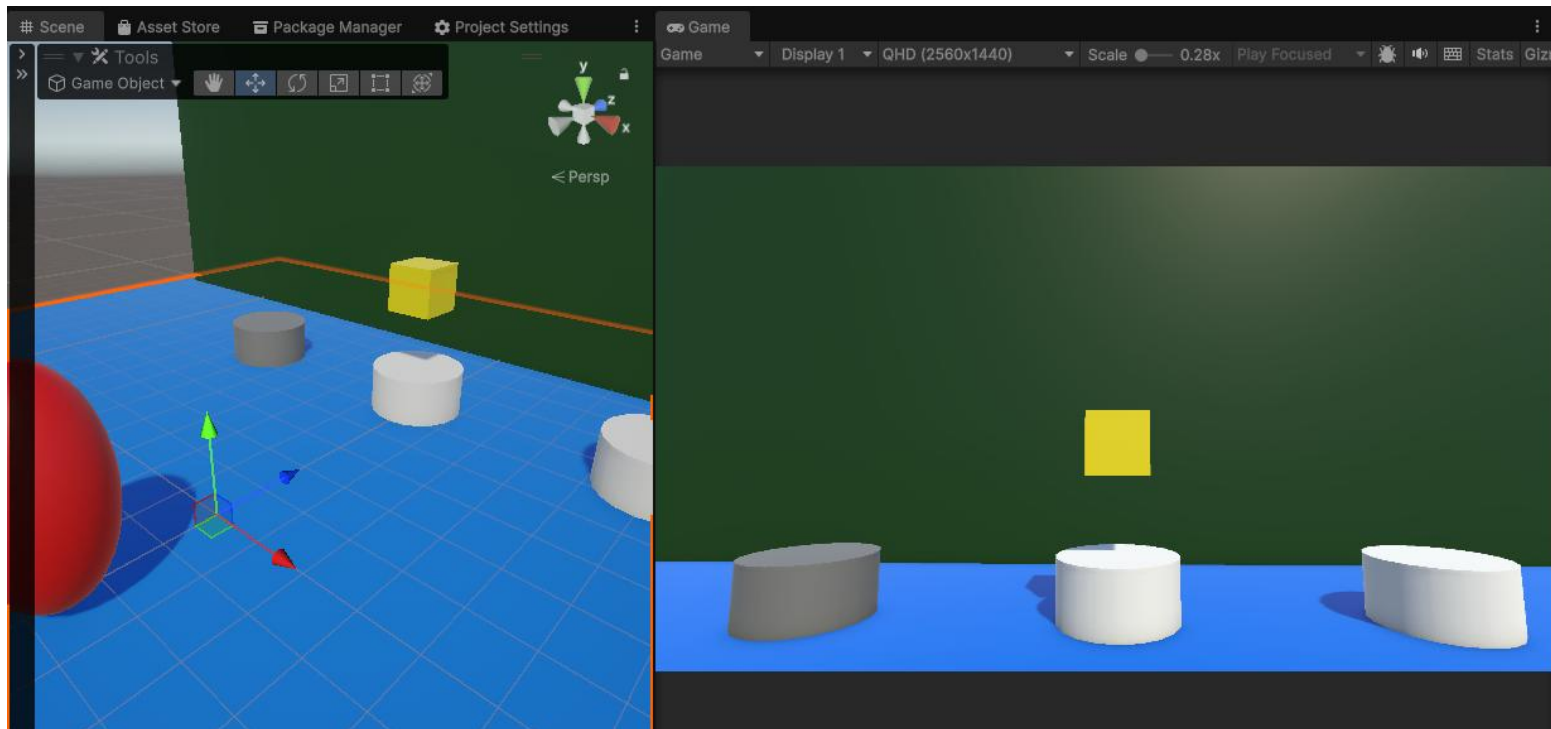


SCRIPTY

- Overte správanie scriptov

ÚLOHA

- Pridanie Look-At mechaniky



SCRIPT

- Tento script pridajte na **Camera** objekt
- Na objekt, ktorý chcete zvýrazniť pridajte tag:

LookedAtObject

- Objekty musia mať
 - Collider (pre raycast)
 - Renderer

```
using UnityEngine;

public class LookHighlight : MonoBehaviour
{
    public float distance = 10f;
    private GameObject lastObject;

    void Update()
    {
        Ray ray = new Ray(transform.position, transform.forward);
        RaycastHit hit;

        if (Physics.Raycast(ray, out hit, distance))
        {
            GameObject obj = hit.collider.gameObject;

            // ONLY objects with this tag
            if (obj.CompareTag("LookedAtObject"))
            {
                // reset previous object
                if (lastObject != null && lastObject != obj)
                {
                    ResetObject(lastObject);
                }
                HighlightObject(obj);
                lastObject = obj;
            }
            else
            {
                ResetLast();
            }
        }
        else
        {
            ResetLast();
        }
    }

    void ResetLast()
    {
        if (lastObject != null)
        {
            ResetObject(lastObject);
            lastObject = null;
        }
    }

    void HighlightObject(GameObject obj)
    {
        Renderer rend = obj.GetComponent<Renderer>();
        if (rend != null)
        {
            rend.material.color = Color.yellow;
        }
    }

    void ResetObject(GameObject obj)
    {
        Renderer rend = obj.GetComponent<Renderer>();
        if (rend != null)
        {
            rend.material.color = Color.white;
        }
    }
}
```

COLLISION SCRIPT

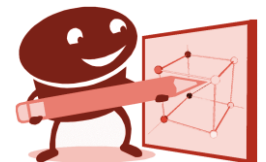
- Overté správanie scriptu



VYTVORENIE VR APLIKÁCIE DO TELEFÓNU

- Nainštalovanie Gitu

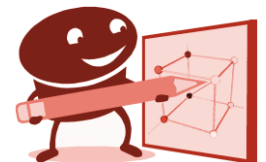
<https://git-scm.com/install/>



VYTVORENIE VR APLIKÁCIE DO TELEFÓNU

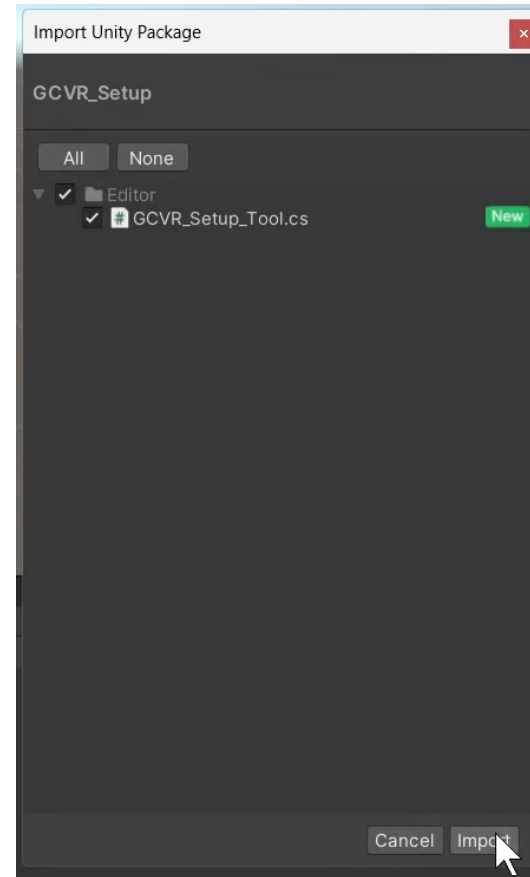
- Import Custom Package

<https://moodle.fei.tuke.sk/mod/resource/view.php?id=14522>



VYTVORENIE VR APLIKÁCIE DO TELEFÓNU

- Import v Unity:
 1. Assets
 2. RightClick
 3. Import Package
 4. Custom Package

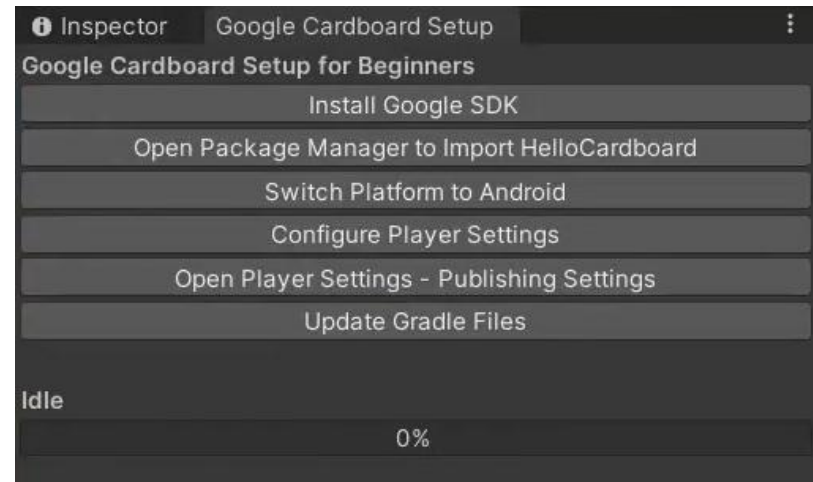
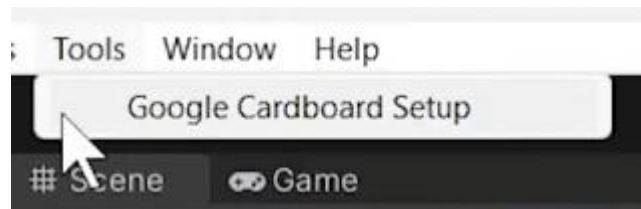


Ak nastanú errorý → restart a skúsiť znova

VYTVORENIE VR APLIKÁCIE DO TELEFÓNU

- Import v Unity:

1. Tools
2. Google Cardboard Setup
3. Install Google SDK

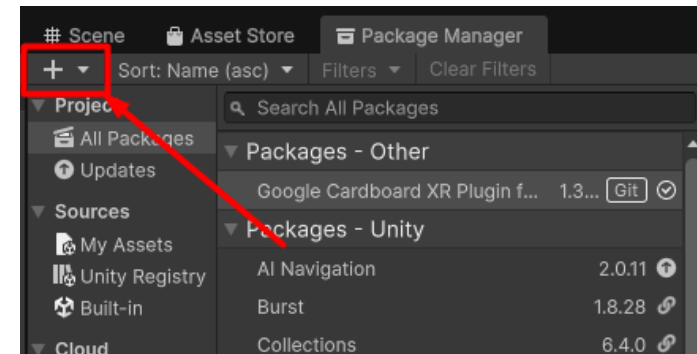


Ak nastanú errorý → restart a skúsiť znova

VYTVORENIE VR APLIKÁCIE DO TELEFÓNU

- Nainštalovanie Google Cardboard XR plugin

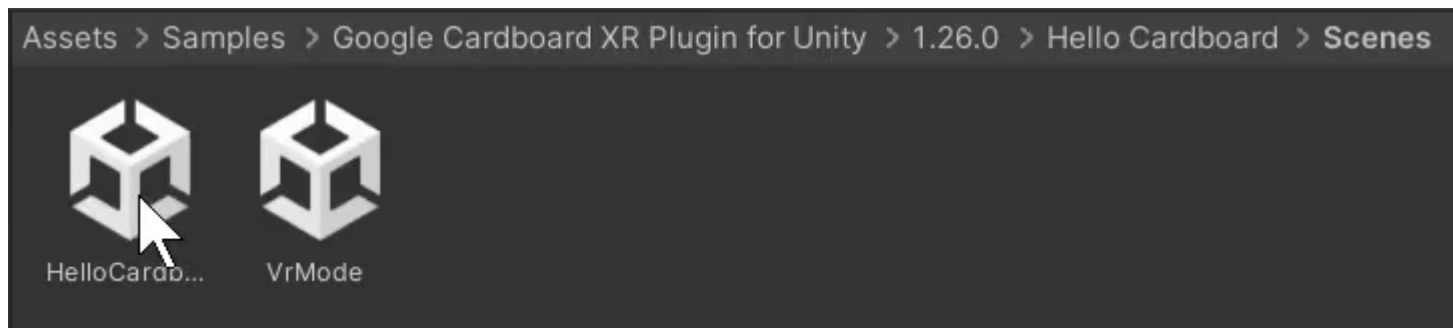
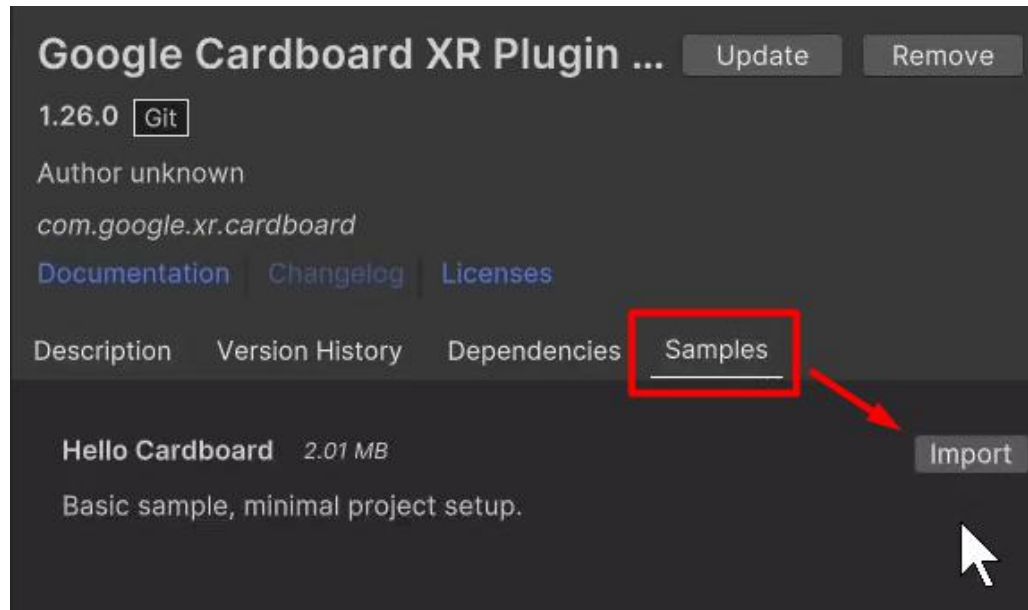
1. Window
2. Package Management
3. Package Manager
4. +
5. Install package from git URL
6. Install



<https://github.com/googlevr/cardboard-xr-plugin.git>

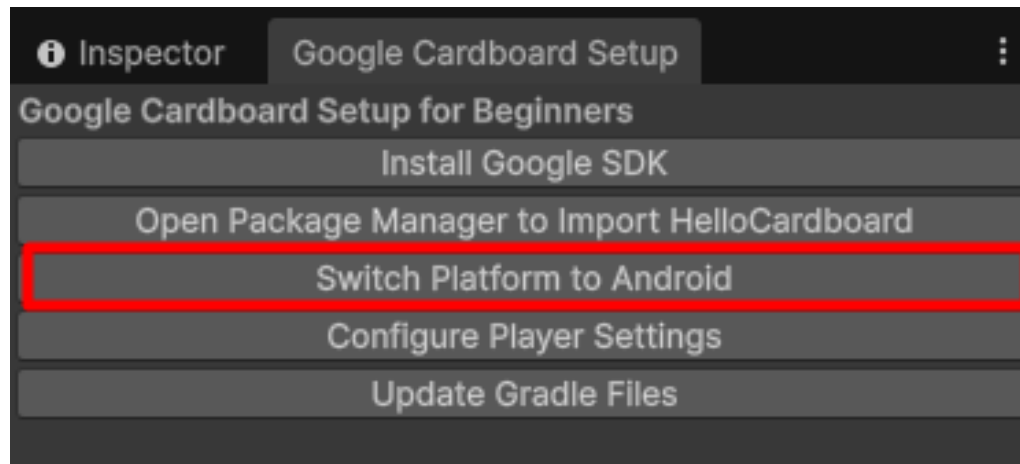
VYTVORENIE VR APLIKÁCIE DO TELEFÓNU

- Google Cardboard XR plugin Scéna



VYTVORENIE VR APLIKÁCIE DO TELEFÓNU

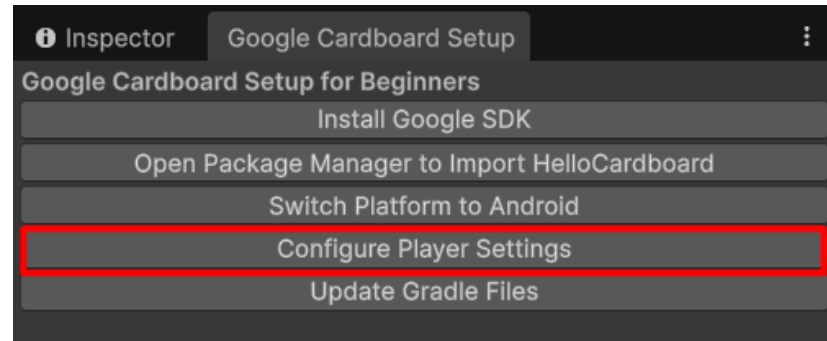
- Zmena platformy na Android



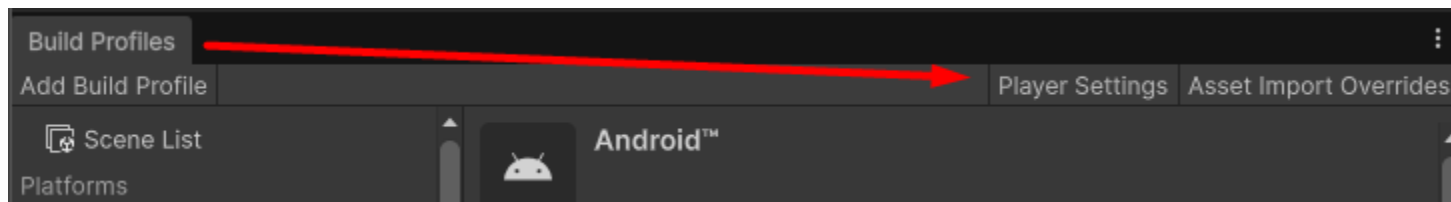
VYTVORENIE VR APLIKÁCIE DO TELEFÓNU

- Konfigurácia Player Settings

1. Google Cardboard:



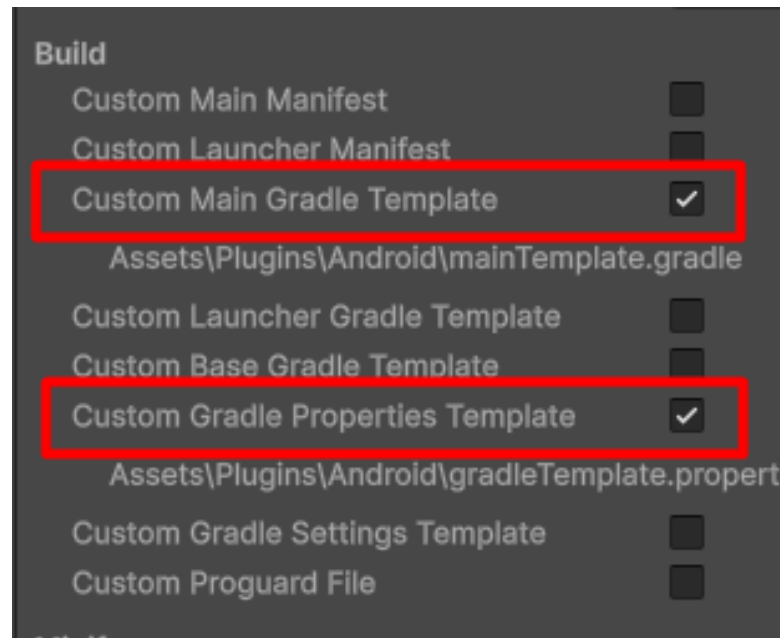
2. Build Settings/Profiles → Player Settings



VYTVORENIE VR APLIKÁCIE DO TELEFÓNU

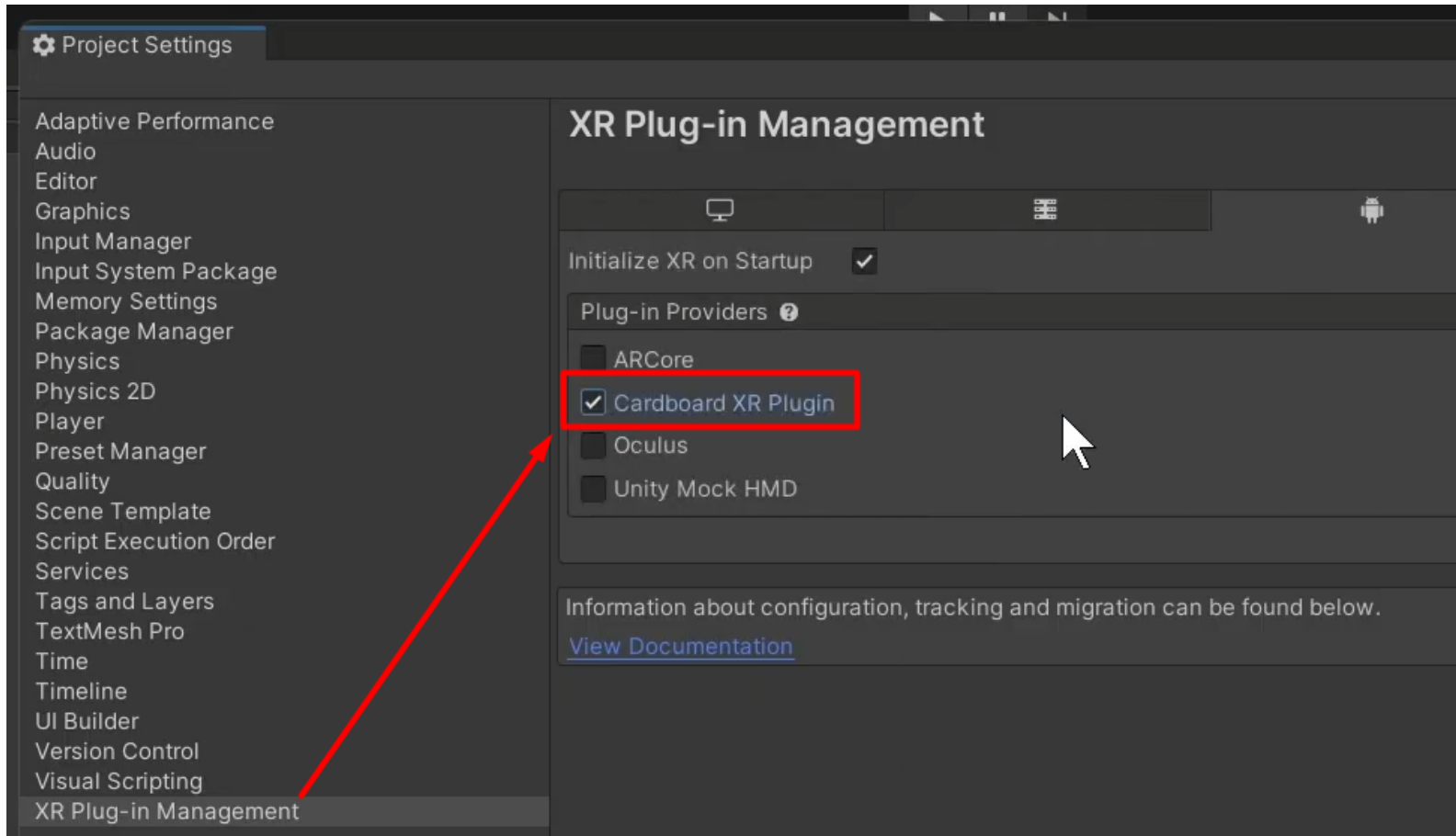
- Konfigurácia Player Settings

1. Player Settings
2. Publishing Settings
3. Build



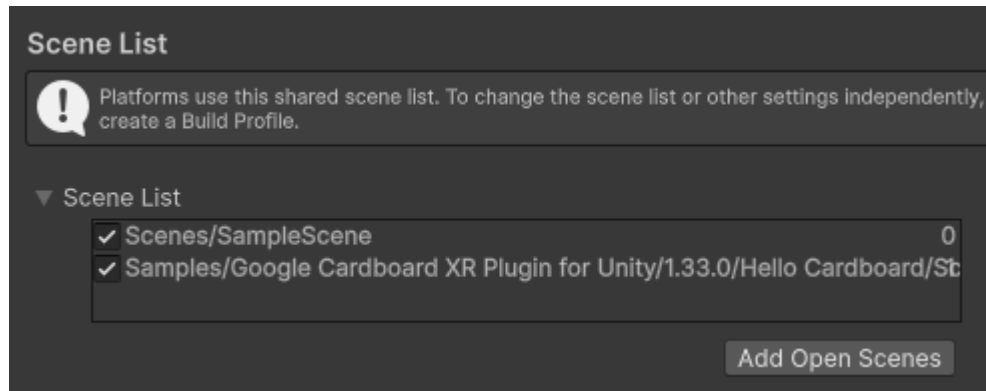
VYTVORENIE VR APLIKÁCIE DO TELEFÓNU

- XR Plug-in Management



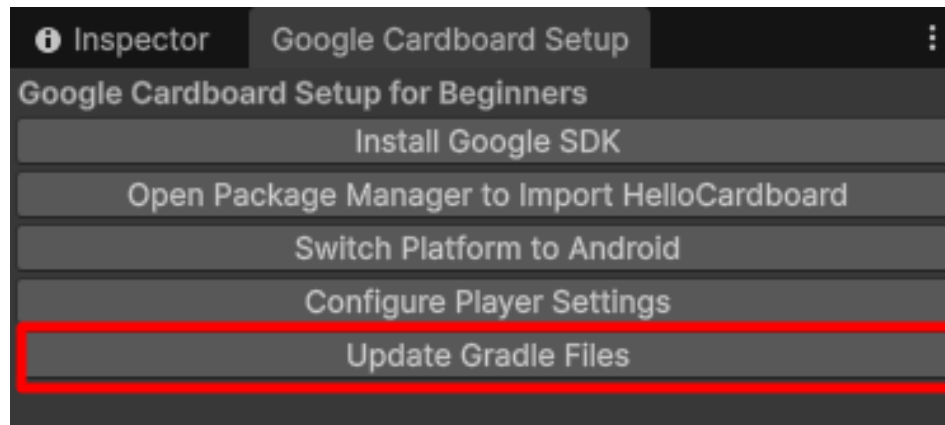
VYTVORENIE VR APLIKÁCIE DO TELEFÓNU

- Pridanie scény do Buildu
 1. File → Build Settings/Profiles
 2. Open Scene List
 3. Add Open Scenes



VYTVORENIE VR APLIKÁCIE DO TELEFÓNU

- Update Gradle Files

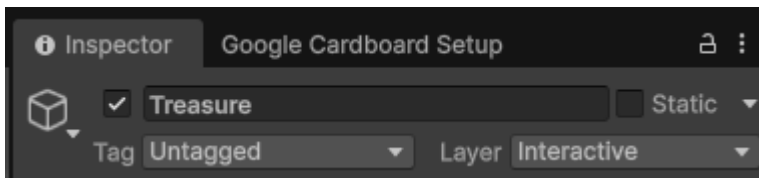


VYTVORENIE VR APLIKÁCIE DO TELEFÓNU

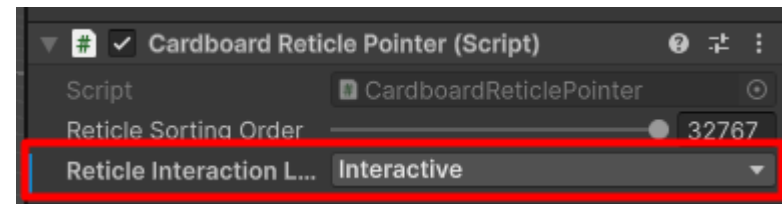
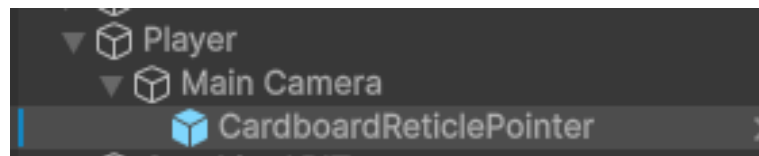
- Nastavenie Layerov na Interactive
- 1. Hierarchy → Treasure → Inspector
- 2. Layer → Add Layer
- 3. User Layer 6: Interactive



4.



5.



Q & A

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

Katedra počítačov a informatiky, FEI TU v Košiciach

© 2026