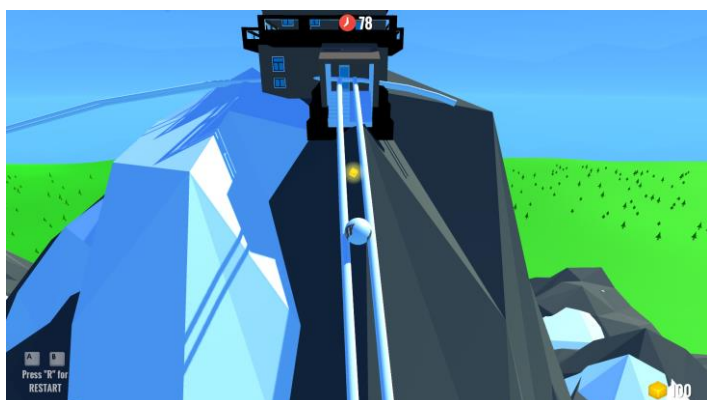
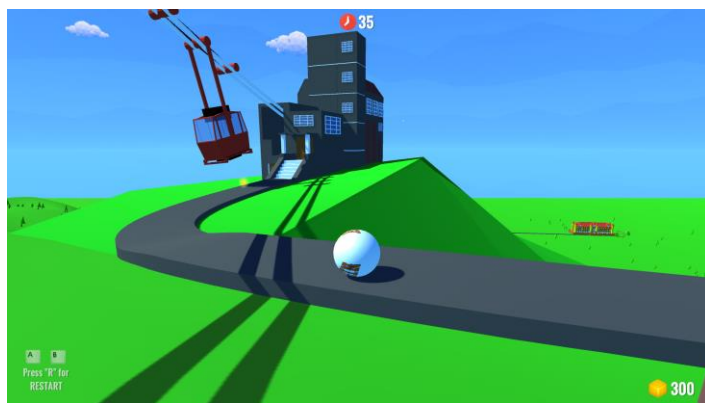


dokumentácia  
Rastislav Bvoc

## Vysoké Tatry

Jednoduchá low poly hra s vysokou náročnosťou, ktorá slúži ako funkčný prototyp s možnosťou neskoršieho rozširovania a vylepšovania.



Stiahnutie hry



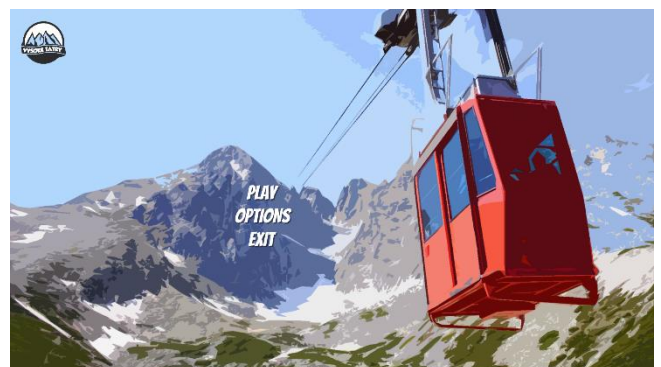
Jednotlivé úlohy:

- **Jadro hry / 11.Mar.2018**

Prenesenie jadra hry z minulého projektu. Konkrétne nanovo vytvorený objekt lopty, prenesený skript kamery, ktorá sa môže otáčať pomocou A D a spolu s ňou aj otáčanie pohybu lopty. Prenesený bol taktiež skript samotného pohybu lopty.

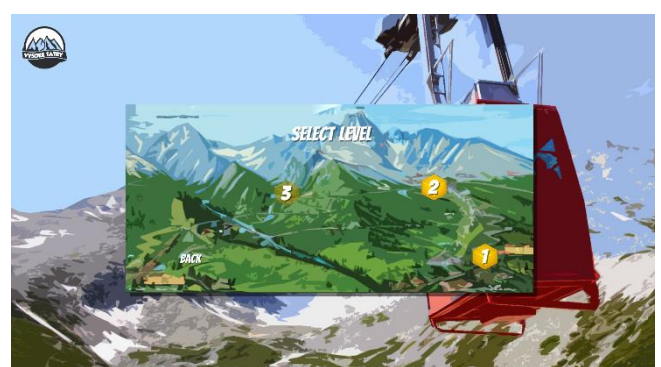
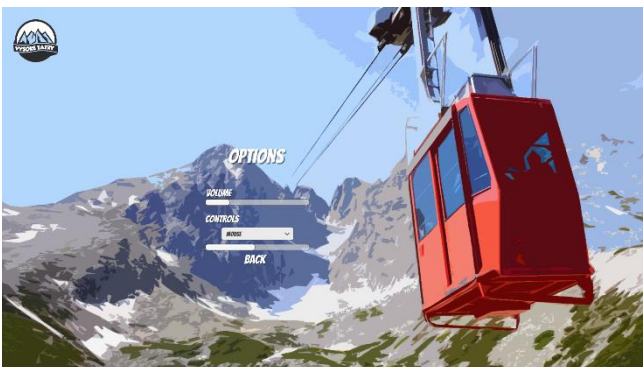
- **Menu hry / 11.Mar.2018**

Cieľ vytvoriť jednoduché menu pre spustenie hry. Menu bolo potrebné vytvoriť nanovo pre neskorší level management. Dizajn menu je v prostredí UNITY responzívny. Časom som pridal do menu aj logo hry.



- **Options menu a level selector / 25.Mar.2018**

Pre hru som vytvoril jednoduché option menu kde sa dá nastaviť napríklad zvuk hudby a efektov samotnej hry. Taktiež level selector pre výber levelu (scény). V menu sú využité aj animácie a taktiež vylepšený textový editor Text Mesh Pro.



- **Logika hry / 10.Apr.2018**

Hru sme nadizajnovali tak aby bola ťažká na prejdienie. Preto som vymyslel systém, kde sa level reštartuje keď loptička spadne mimo cesty. V podstate loptička spadne pod textúru kde keď sa stretne s colliderom pod povrchom automaticky sa reštartuje. Taktiež je možné začať odznovu pomocou klávesy R.

- **Collectibles a User Interface / 24.Mar.2018**

Do hry som pridal akési predmety na zbieranie (pomocou box collideru, taktiež pridaná dynamická fyzika miesto statickej kvôli optimalizácii), ktoré budú určovať úspešnosť hráča a bez nazbierania určitého počtu nebude možné úroveň prejsť. Taktiež som pridal časovač, ktorý sa zobrazuje aj na obrazovke a ak nestihneme nazbierať dostatočný počet bodov a dostať sa na finálne miesto, level bude neúspešný a reštartuje sa. Do UI som pridal aj zobrazenie samotného skóre a pomôcku pre hráča, ktorá ukazuje ostatné klávesy pomocou ktorých sa dá hra ovládať. Taktiež bolo potrebné naprogramovať že pri ukončení levelu sa počítadlo prestane odpočítavať, to bolo asi najnáročnejšie pretože som potreboval prísť na to ako získam premennú z jednej triedy do druhej.



- **Skybox a úpravy prvého levelu / 24.Mar.2018**

Pridal som pekný skybox, ktorý som našiel v asset store a dokonale sa hodil do našej hry. Taktiež som pridal nejaké detaily ako zábradlia stromy a podobne pre krajší dizajn prostredia hry. Názorná ukážka na obrázku vyššie.

- **Pridanie zvukov a efektov / 24.Mar.2018**

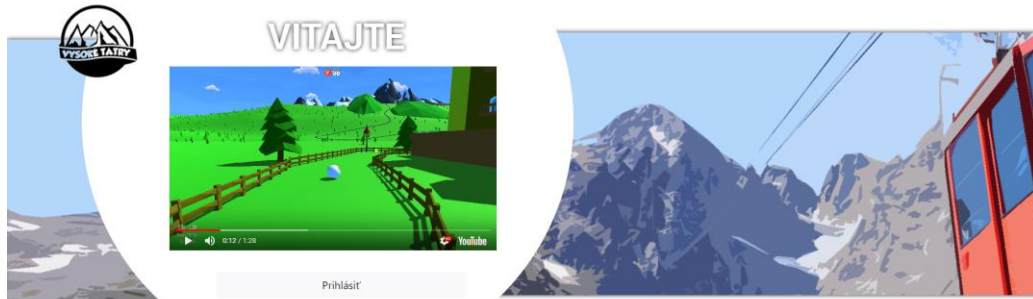
Aby som hru oživil pridal som do pozadia veselú hudbu a taktiež pridal zvuky efektov pri pokročení do nového levelu, zrodení loptičky alebo po pozbieraní bodov. Celé to bolo zrealizované pomocou SoundManageru.

- **Level design / 10.Apr.2018**

Vytvoril som druhý level, kde som využil dynamické prostredie, padajúce a rotujúce objekty. Taktiež boli doriešené existujúce bugy a vznikol tak funkčný prototyp hry s 2 levelmi, do ktorého by už nebol problém jednoducho pridávať ďalšie nové úrovne.

- **Webstránka / 28.Apr.2018**

V rámci predmetu Vývoj WEB aplikácií som vytvoril pre našu hru webstránku na ktorej sa môže hráč pomocou alpha kľúča registrovať do systému kde môže nahráť svoje nahrané skóre a porovnávať sa tak s ostatnými hráčmi.



### VYSOKÉ TATRY SÚ SKVELÁ HRA ZDARMA

Vyskúšaj svoje schopnosti v multiplatformovej hre z nášho krásneho horského prostredia. Hru je možné ovládať na myške , gamepade ale aj mobilných telefónoch či tabletoch.

[Stiahni si našu hru ZDARMA!](#)



#### Features



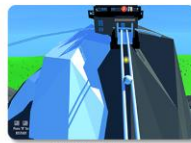
##### COMBAT

There is plenty of combat options in our game. Combat or fight is a purposeful violent conflict meant to weaken, establish dominance over, or kill the opposition, or to drive the opposition away from a location where it is not wanted or needed. The term combat (French for fight) typically refers to armed conflict between opposing military forces in warfare, whereas the more general term "fighting" can refer to any physical or verbal conflict between individuals or nations.



##### ADVENTURES

There is a big story in Ravenport, hear stories and join your side! Geralt is a bounty hunter, an apex predator in a world gone wrong. In a monster-infested world he is a necessary counter-balance. His main task is to eliminate creatures that pose a threat to society. Hunting monsters is his business -- and there's always plenty of work. Monsters come in all shapes and sizes, and while some are members of broader species, others are one-of-a-kind.

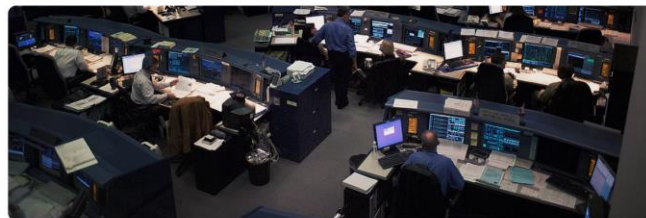


##### EXPLORATION

Explore our great world with dynamic zones, caves and lot more. The world of Valglory beckons to be explored -- just pick a point on the horizon and venture off into the unknown. You'll have your trusty horse to call on whenever you need. Load it with trophies taken from the massive beasts you hunt down or gallop it into combat to crush those standing in your way. You'll also go where no horse can follow, descending deep into sprawling caves, diving beneath .

#### System Requirements

Minimum Requirements	Recommended Requirements
500 MHz CPU	700 MHz CPU
96 MB RAM	128 MB RAM
16 MB GPU	32 MB GPU
1.8 GB HDD	1.8 GB HDD
16x CD-ROM	32x CD-ROM
DirectX 8.1	DirectX 11



#### SUPPORT US

Please if u like our project please give us some tip we are young developers , most of us are students so if you want see our future projects give us some love :) We are fresh company working on all kind of projects but without your help we can t realize nothing so thanks for every donate, every penny counts. You can support us with paypal or your VISA-Electron credit card. THANKS!



- Dokumentácia / 3.Jun.2018

Finálny kód použitý v našej hre.

## OVLÁDANIE A LOGIKA HRY

Hlavný skript, ktorý sa stará o väčšinu logiky a obsahuje mnoho hlavných premenných.

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

public class player : MonoBehaviour
{
    public Rigidbody rb;
    public float rychlost;
    public Text collectibleText;
    public Text winText;
    public float terminalRotationSpeed = 30.0f;
    private Transform camTransform;
    public float collectible;

    private void Start()
    {
        rb = GetComponent<Rigidbody>();
        rb.maxAngularVelocity = terminalRotationSpeed;
        camTransform = Camera.main.transform;
        Cursor.visible = false;
        collectible = 0 ;
        SetCollectibleText();
        winText.text = "";
    }

    void FixedUpdate()
    {
        float pohybHorizontal = Input.GetAxis("Mouse X"); //Horizontal
        float pohybVertical = Input.GetAxis("Mouse Y"); //Vertical

        Vector3 pohyb = new Vector3(x: pohybHorizontal, y: 0.0f, z: -pohybVertical);
        // -pohybVertical mouse/ball

        //Kamera dircetion
```

```

    Vector3 rotatedDir = camTransform.TransformDirection(pohyb);
    rotatedDir = new Vector3(rotatedDir.x, 0, rotatedDir.z);
    rotatedDir = rotatedDir.normalized * rychlost;

    rb.AddForce(rotatedDir * Time.deltaTime, ForceMode.Force); // time deltatime
- optimalizacia
}
//ZBIERANIE COLLECTIBLES
private void OnTriggerEnter(Collider other)
{
    if (other.gameObject.CompareTag("pickup"))
    {
        other.gameObject.SetActive(false);
        collectible = collectible + 100;
        SetCollectibleText();
        SoundManagerScript.PlaySound("collect");
    }

    if (other.gameObject.CompareTag("fail"))
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
        Cursor.visible = false;
    }

    if (other.gameObject.CompareTag("levelup") && (collectible >= 400))
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
        Cursor.visible = false;
    }

    if (other.gameObject.CompareTag("platform") && (collectible >= 400))
    {
        SoundManagerScript.PlaySound("collect");
    }
}

// VYTVORENIE FUNKCIE PRE ZAPIS SKORE DO UI
void SetCollectibleText() {

    collectibleText.text = collectible.ToString();
    if (collectible >= 400) {
        winText.text = "NEXT LEVEL";
        SoundManagerScript.PlaySound("levelup");
    }
}
}

```

# KAMERA

Skript zabezpečujúci správnu funkciu kamery a jej ovládania.

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

public class camerascript : MonoBehaviour
{

    private Vector3 desiredPosition;
    public GameObject Player;
    public float smoothSpeed = 4f;
    private Vector3 offset;
    public Transform lookAt;

    // initialization
    void Start()
    {
        offset = transform.position - Player.transform.position;
    }

    // Update once per frame
    void Update()
    {

        if (Input.GetKeyDown(KeyCode.A))    //|||(Input.GetButton("Fire1"))
            SlideCamera(true);
        else if (Input.GetKeyDown(KeyCode.D))    //||| (Input.GetButton("Fire2"))
            SlideCamera(false);

    }

    private void FixedUpdate()
    {
        desiredPosition = Player.transform.position + offset;
        transform.position = Vector3.Lerp(transform.position, desiredPosition,
smoothSpeed * Time.deltaTime);
        transform.LookAt(lookAt.position + Vector3.up);

    }

    public void SlideCamera(bool left)
    {
        if (left)
            offset = Quaternion.Euler(0, 90, 0) * offset;
        else
            offset = Quaternion.Euler(0, -90, 0) * offset;

    }
}
```

# MENU A SCENE MANAGER

Pomocou menu skriptu sa zvolí vybraný level a aj ho môžeme reštartovať.

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

public class MainMenu : MonoBehaviour {

    public void PlayGame() {

        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);

    }
    public void Level1() {

        SceneManager.LoadScene("scene1");
    }
    public void Level2()
    {

        SceneManager.LoadScene("scene2");
    }

    public void QuitGame() {
        Debug.Log("QUIT");
        Application.Quit();
    }
}
void Start()
{
    if (Instance != null) // ak existuje znicí
    {
        GameObject.Destroy(gameObject);
    }

    else
    {
        GameObject.DontDestroyOnLoad(gameObject);
        Instance = this;
    } }

void Update()
{
    if (Input.GetKeyUp(KeyCode.Escape))
    {
        Application.LoadLevel("menu");
        Cursor.visible = true;
    }

    if (Input.GetKeyUp(KeyCode.R))
    {

        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
        Cursor.visible = false;
    }
}}
```



# SOUND MANAGER

Tento skript zvolí správny zvuk pri kolízií napríklad so zbierateľnými predmetmi a iný zvuk pri prejení levelu.

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

public class SoundManagerScript : MonoBehaviour {

    public static AudioClip collect, endgame, levelup;
    static AudioSource audioSrc;

    // Use this for initialization
    void Start () {

        collect = Resources.Load<AudioClip>("collect");
        endgame = Resources.Load<AudioClip>("endgame");
        levelup = Resources.Load<AudioClip>("levelup");

        audioSrc = GetComponent<AudioSource>();

    }

    // Update is called once per frame
    void Update () {

    }

    public static void PlaySound(string clip) {
        switch (clip) {
            case "collect":
                audioSrc.PlayOneShot (collect);
                break;

            case "endgame":
                audioSrc.PlayOneShot(endgame);
                break;

            case "levelup":
                audioSrc.PlayOneShot(levelup);
                break;

        }

    }

}
```

# TIMER

Skript časovaču, ktorý odpočítava čas na dokončenie levelu a pri prejdení levelu prestane odpočítavať.

```
using UnityEngine;
using System.Collections;
using UnityEngine.UI;

public class Timer : MonoBehaviour
{
    public int timeLeft = 100;
    public Text countdownText;
    public float collectible1;

    // Use this for initialization
    void Start()
    {
        StartCoroutine("LoseTime");

        GameObject.Find("Player").GetComponent<player>().collectible = collectible1;
    }

    // Update is called once per frame
    void Update()
    {
        countdownText.text = ("" + timeLeft);

        if (GameObject.Find("Player").GetComponent<player>().collectible == 400) {
            StopCoroutine("LoseTime");
        }

        if (timeLeft <= 0)
        {
            SoundManagerScript.PlaySound("endgame");
            StopCoroutine("LoseTime");
            countdownText.text = "0";
            Application.LoadLevel("scene1");
            Cursor.visible = false;
        }
    }

    IEnumerator LoseTime()
    {
        while (true)
        {
            yield return new WaitForSeconds(1);
            timeLeft--;
        }
    }
}
```