

**Vyšší odborná škola a Střední průmyslová škola, Šumperk,
Gen. Krátkého 1, 787 29 Šumperk**



Obor vzdělání: Informační technologie (18-20-M/01)

MATURITNÍ PRÁCE

z odborných předmětů

Školní rok: 2017/2018

Lukáš Linhart Třída: IT4A

**Vyšší odborná škola a Střední průmyslová škola, Šumperk,
Gen. Krátkého 1, 787 29 Šumperk**



Obor vzdělání: Informační technologie (18-20-M/01)

ZADÁNÍ MATURITNÍ PRÁCE

z odborných předmětů

Žák: Lukáš Linhart

Téma maturitní práce: Návrh a vývoj programu

Obsah maturitní práce:

- 1. Návrh**
- 2. Objektový návrh**
- 3. Vlastní vývoj programu v objektovém jazyce**
- 4. Návrh a tvorba GUI**
- 5. Ozvučení aplikace**
- 6. Testování**
- 7. Tvorba dokumentace**

Další náležitosti zadání: K:/sablony/maturitní práce

Datum zadání: 29. září 2017

Datum odevzdání: 28. března 2018

Vedoucí práce: Mgr. Zdeněk Přikryl

Podpis:

PROHLÁŠENÍ

Prohlašuji, že jsem maturitní práci z odborných předmětů vypracoval samostatně a k vypracování jsem použil pouze zdroje uvedené v seznamu literatury.

Datum: 28. března 2018

Žák: Lukáš Linhart

Podpis:

Vyšší odborná škola a Střední průmyslová škola, Šumperk,
Gen. Krátkého 1, 787 29 Šumperk

KONZULTAČNÍ LIST

Žák: Lukáš Linhart

Třída: IT4A

Téma maturitní práce: Návrh a vývoj programu

Datum kontroly: _____

Obsah kontroly: _____

Hodnocení kontroly: 1 2 3 4 5 **Podpis vedoucího práce:** _____

Datum kontroly: _____

Obsah kontroly: _____

Hodnocení kontroly: 1 2 3 4 5 **Podpis vedoucího práce:** _____

Datum kontroly: _____

Obsah kontroly: _____

Hodnocení kontroly: 1 2 3 4 5 **Podpis vedoucího práce:** _____

Datum kontroly: _____

Obsah kontroly: _____

Hodnocení kontroly: 1 2 3 4 5 **Podpis vedoucího práce:** _____

Datum kontroly: _____

Obsah kontroly: _____

Hodnocení kontroly: 1 2 3 4 5 **Podpis vedoucího práce:** _____

OBSAH

1	<i>Úvod</i>	6
2	<i>Objektový návrh</i>	7
2.1	Car	7
2.2	Movement	7
2.3	AiMovement	7
2.4	CameraControll	7
2.5	GameBoss	7
2.6	AI	8
2.7	Record point	8
3	<i>Grafika</i>	10
3.1	Auta	10
3.2	Závodní okruh	11
3.3	Ostatní grafika	12
3.4	GUI	13
4	<i>Zvuk</i>	16
4.1	Zvuk auta	16
4.2	Rádio	16
4.3	Ostatní zvuky	16
4.4	Mastering	16
5	<i>Kódování</i>	18
5.1	Rádio	18
5.2	Pohyb hráče	19
5.3	Umělá inteligence	19
5.4	GameBoss	20
5.5	Vítěz	21
6	<i>Testování</i>	22
7	<i>Chyby v unity</i>	23
8	<i>Výhody Unity</i>	24
9	<i>Seznam použité literatury a jiných zdrojů</i>	27
10	<i>Závěr</i>	28
11	<i>Resumé</i>	29
12	<i>Přílohy práce</i>	30

1 ÚVOD

Jako téma mé maturitní práce jsem si zvolil seminář z programování. Již na základní škole jsem se pokoušel učit programovat. Snažil jsem se vše naučit sám, za pomoci návodů a tvorby jednoduchých projektů. Na střední škole, hlavně v semináři programování v jazyce Java, jsem se v tomto předmětu začal více orientovat a chápat více principy programování. Cílem mé maturitní práce bylo vytvořit funkční závodní hru v herním enginu Unity, se kterým jsem již částečné zkušenosti měl, proto jsem si ho vybral. V enginu se používá programovací jazyk C#, JavaScript nebo Boo. Já jsem si zvolil C#, protože jsem s ním již měl zkušenost.

2 OBJEKTOVÝ NÁVRH

Celý projekt jsem se snažil využívat programovací konvence. Celý program jsem se snažil rozdělit do tříd. Tyto třídy zobrazují ostatním objektům pouze ty atributy, které jsou nutné. Gettery a Settery jsem netvořil, protože z Javy zvyklý na trošku jiné tvoření, než je v C# a potřeboval bych to ještě víc prostudovat. Dále jsem rozlišoval velikost začátečního písmene u public a private metod.

2.1 Car

Hlavní třída aut, obsahuje jen základní atributy, které má každé auto.

Mezi atributy patří:

- maxSpeed
- originalMaxSpeed - do kterého si ukládám původní hodnotu, protože maxSpeed v průběhu programu měním
- currentSpeed
- brake - true|false.

V metodě OnCollisionEnter a OnCollisionExit se prověřuje, zda auto najelo na trávu a případně se mění maximální rychlost na poloviční.

2.2 Movement

Třída, která je potomkem třídy Car, určuje pohyblivost auta, které ovládá hráč. Jak se auto chová, jak brzdí. Dále také udržuje auto na vozovce, aby se nepřevracelo.

2.3 AiMovement

Je potomek třídy Car, stará se o pohyblivost auta ovládané počítačem. Toto chování je jiné, protože AI auta načítají jednotlivé souřadnice ze souboru, po kterých se pohybují. Trasy jsou předem nahrány. Následně se z nich náhodně vybere, kterou trasu autíčko použije pro svoji jízdu.

2.4 CameraControll

V této třídě se provádí veškerá práce s kamerou. Spouští se zde počáteční a konečná animace, nastavuje auto, které má kamera zaměřit a sledovat.

2.5 GameBoss

Hlavní třída celé hry. Vytváří auta, volá spuštění animací, stará se o zvuky, a upravuje některé GUI texty.

2.6 AI

Tuto třídu jsem si vytvořil jako pomocnou, když jsem hru tvořil. Dokáže totiž nahrávat body, které mé auto právě projíždí, a ukládat je do samostatných tras, podle kterých později jezdí ostatní auta. Tento způsob nahrávání trasy mi připadal jako nejjednodušší řešení pro tuhle hru.

Zajímavou metodou této třídy je metoda ShowPointsOnMap. Tato metoda reaguje na stisk neviditelného tlačítka v levém dolním rohu obrazovky a zobrazí textový formulář, do kterého se napíše název trasy. Všechny body, které jsou zapsány v této trase se ukáží na vozovce. Dost mi to pomohlo při řešení komplikací.



Ukázka zobrazených bodů

2.7 Record point

Tato třída reprezentuje jednotlivé body, kterými auto projíždí. Kromě atributů souřadnic obsahuje také proměnné o rychlosti přidávání, a zda mají být použity brzdy.


```

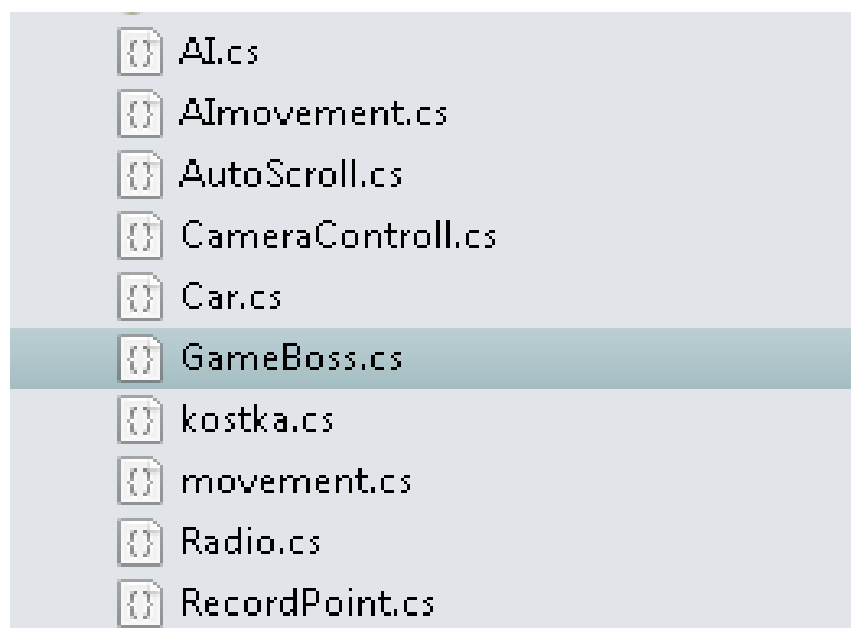
public class RecordPoint {
    public float throttle;
    public bool brake;
    public float x;
    public float y;
    public float z;

    public RecordPoint(float throttle, bool brake, float x, float y, float z)
    {
        this.throttle = throttle;
        this.brake = brake;
        this.x = x;
        this.y = y;
        this.z = z;
    }

    public RecordPoint(float throttle, bool brake, Vector3 position)
    {
        this.throttle = throttle;
        this.brake = brake;
        this.x = position.x;
        this.y = position.y;
        this.z = position.z;
    }

    override
    public string ToString()
    {
        return throttle+","+brake+","+x+","+y+","+z+","+ "\n";
    }
}

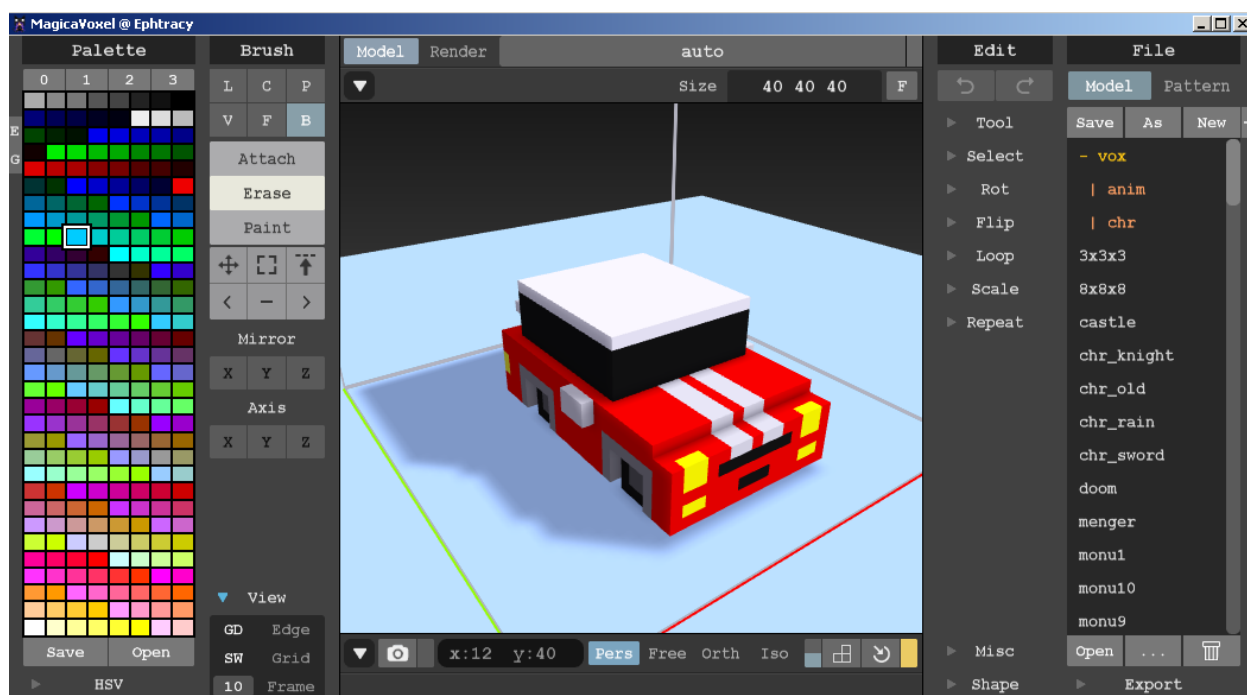
```



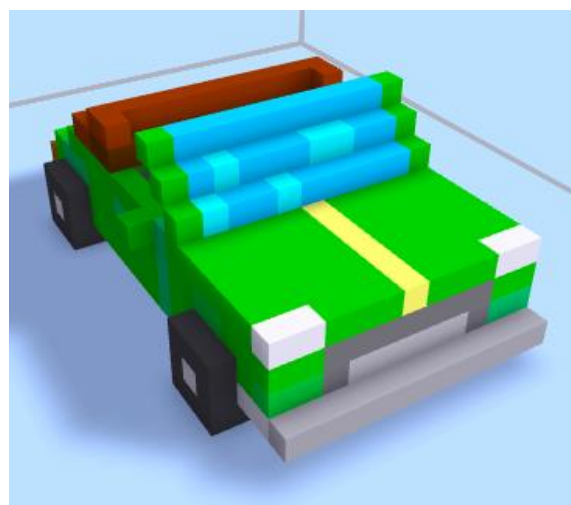
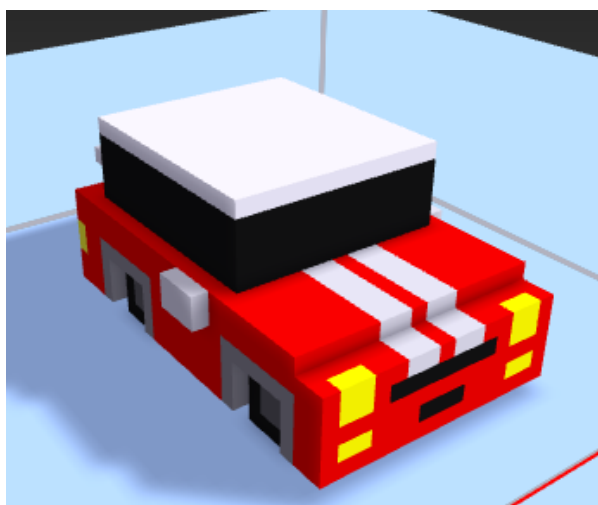
Všechny třídy

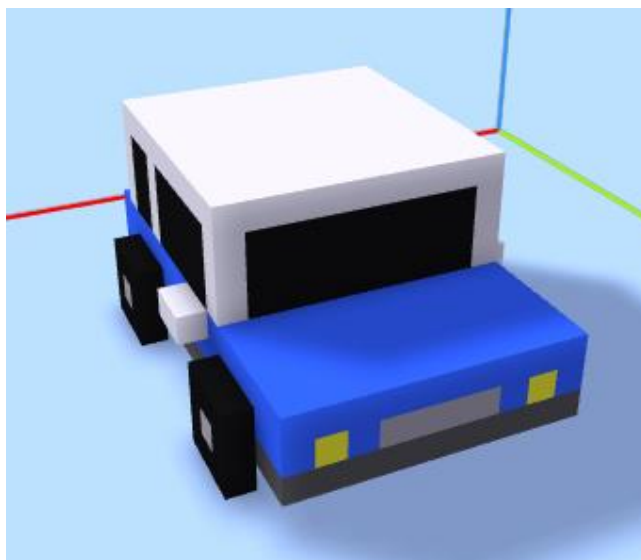
3 GRAFIKA

Všechny modely jsem vytvářel v programu MagicaVoxel. S tímto programem jsem se naučil pracovat teprve při tomto projektu. MagicaVoxel je jednoduchý 3D modelovací program, ve které se celý model vytváří pouze z kostek. Tento přístup mi vyhovuje, protože nemám moc velkou grafickou představivost.



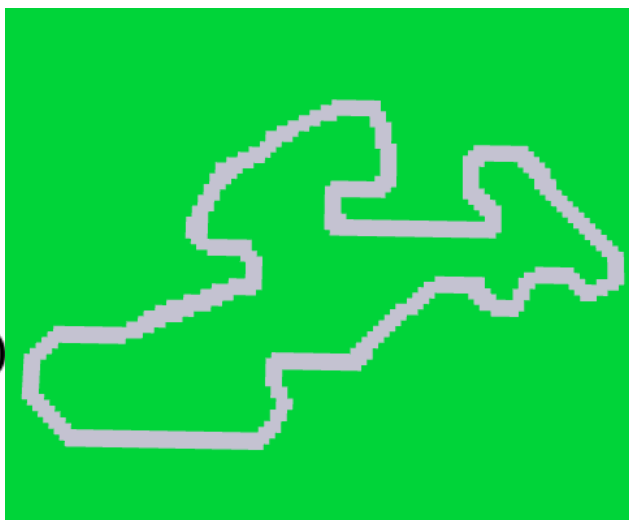
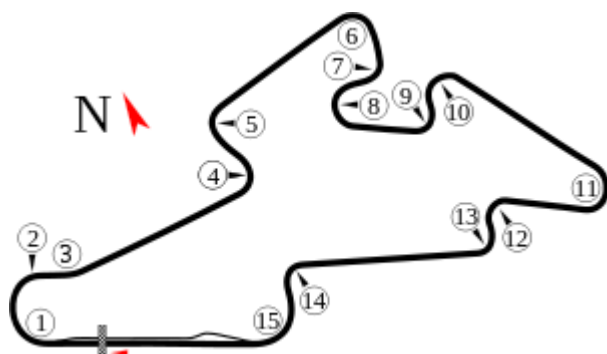
3.1 Auta

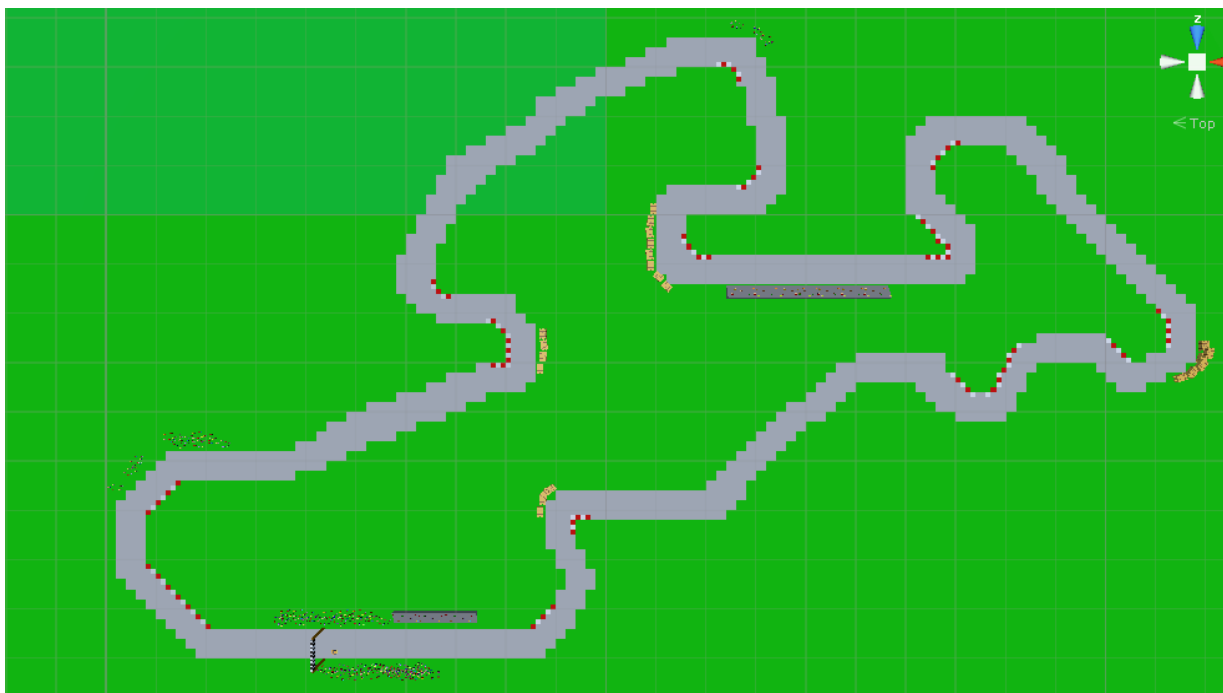




3.2 Závodní okruh

Závodní okruh je tvořen podle reálného brněnského okruhu. Z důvodu, že program MagicaVoxel je omezen na maximální velikost modelu na 126 pixelů do všech tří směrů, jsem musel mapu rozřezat na čtyři části. Každou rozřezanou část jsem musel rozdělit ještě na trávu a vozovku. Tento krok jsem udělal, protože se mi takhle lépe v kódu kontrolovalo vyjetí auta z trati na trávu. Tento program dokáže exportovat do nejpoužívanějších modelovacích formátů.

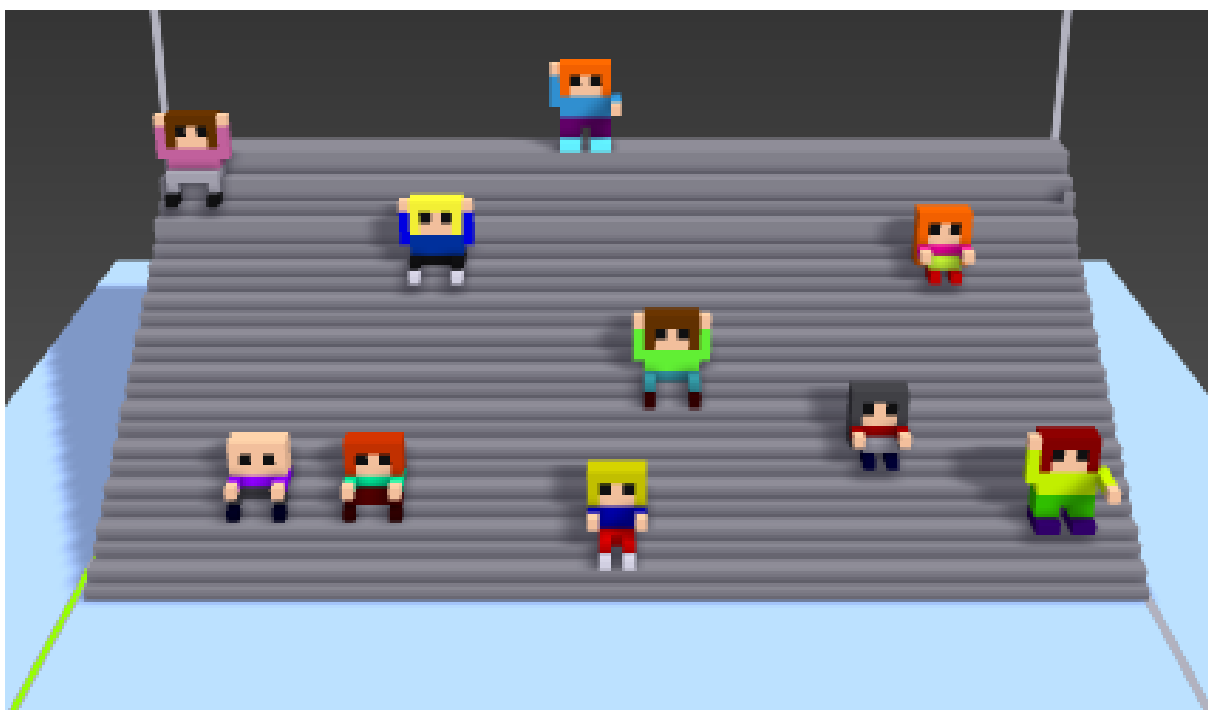


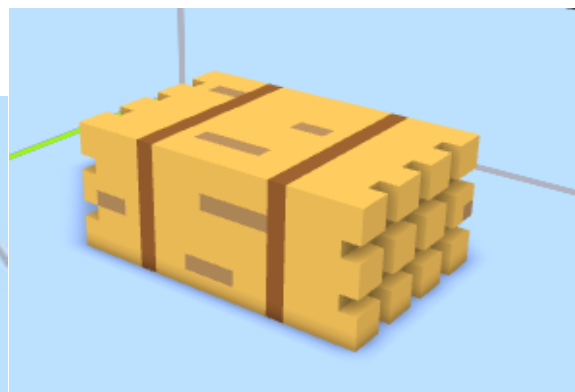


Trat' s vloženými objekty

3.3 Ostatní grafika

Do hry jsem přidal i diváky, tribuny, zátarasy a startovní ceduli.



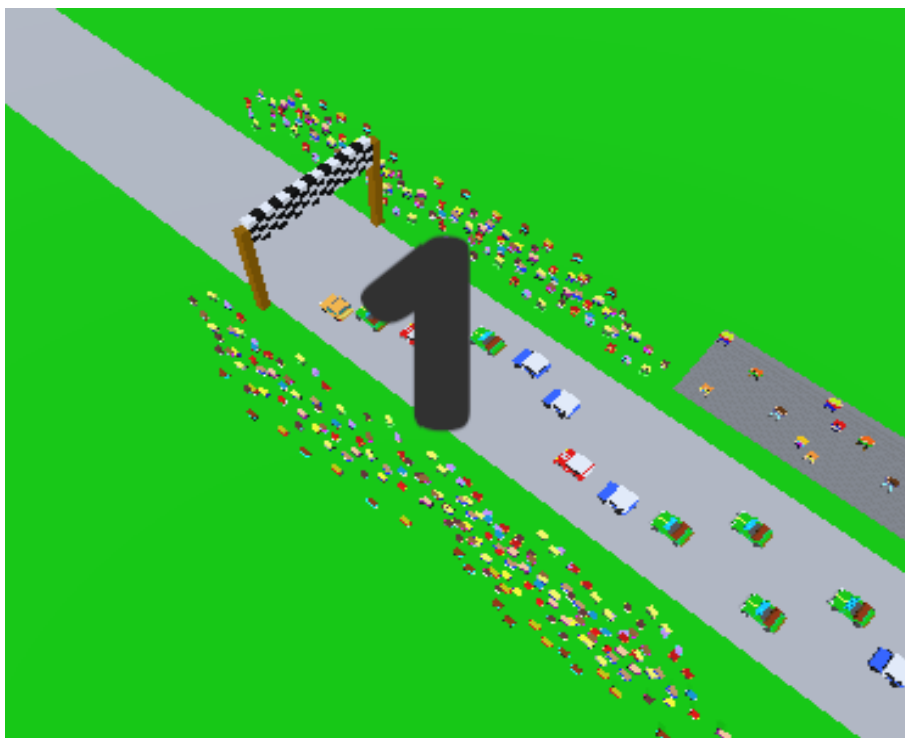


3.4 GUI

Na začátku hry se přehrává animace s přeletem kamery nad tratí. Tato animace se opakuje, dokud hráč nestiskne nějakou klávesu. Uprostřed se nachází logo hry a pod ním bliká text vyzývající k zahájení hry.



Po stisknutí klávesy se spustí odpočet, kamera přejde na animaci, která plynule přejde na pohled auta. Při animacích jsem využíval nástroj Animator, který se nachází přímo v Unity. V tomto nástroji se spustí nahrávání a při pohybu objektu se do klíčových snímků automaticky vkládá změněná hodnota pozice objektu.



Odpočet



Nástroj Animator

Při úspěšném dojetí závodu hra končí závěrečnou animací, která oddálí pohled z hráče na celé závodíště a uprostřed se zobrazí konečné umístění.



Závěrečná obrazovka

4 ZVUK

4.1 Zvuk auta

Zvuk auta jsem si našel volně ke stažení na internetu. Udělal jsem také to, že výška zvuku se mění v závislosti na aktuální rychlosti auta. Takto vzniká efekt, že auto zrychluje a zpomaluje.

```
void LateUpdate(){  
    GetComponentInParent<AudioSource> ().pitch = Mathf.Pow(currentSpeed/100,3);  
}
```

4.2 Rádio

Do hry jsem přidal také rádio, které náhodně přehrává skladby z nastavené složky. Skladby je možné přepínat tlačítka. Aktuální název skladby se zobrazuje mezi tlačítka, a pokud je text moc dlouhý, začne postupně rolovat. Na rolování nemá Unity žádnou knihovnu, proto jsem využil kód, který již někdo vytvořil na StackOverflow.

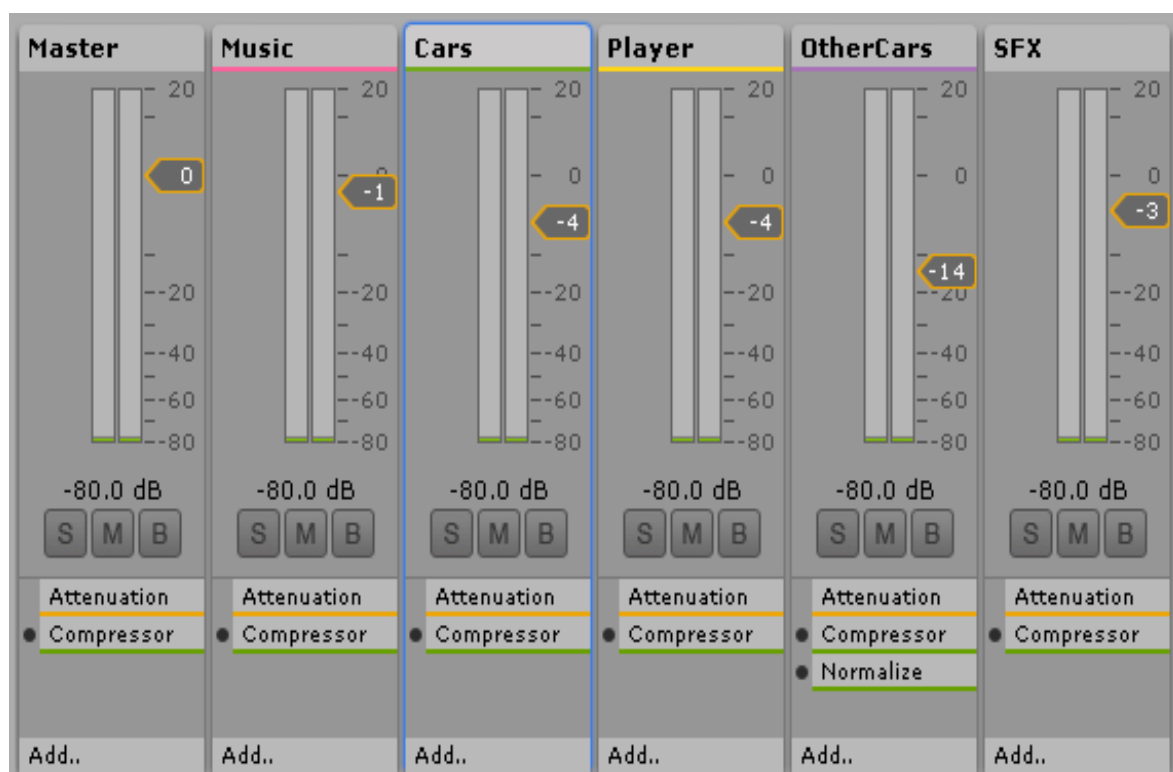


4.3 Ostatní zvuky

Mezi ostatní zvuky řadím například zvuk odpočítávání, který jsem také zdarma našel na internetu.

4.4 Mastering

Využil jsem nástroje Unity AudioMixer, do kterého jsem poslal výstupy všech zvuků. Zde jsem potom upravil jednotlivé úrovně skupin, aby nebyly zvuky auta moc hlasité. Přidal jsem i efekt kompresor, který vyrovnává hladinu zvuků.



Ukázka mixeru

5 KÓDOVÁNÍ

5.1 Rádio

Na začátku skript načte skladby ze složky „Songs“ do pole „myMusic“. Z tohoto pole se přehraje náhodná skladba. V metodě Update se zjišťuje, zda hraje nějaká písnička, a když ne, tak se změní text aktuální skladby a zapne další náhodná písnička.

```
void Awake () {
    audio=this.GetComponent<AudioSource>();
    myMusic = Resources.LoadAll("Songs",typeof(AudioClip));
    audio.clip = myMusic[0] as AudioClip;
}

void Update () {
    if (!audio.isPlaying) {
        text.GetComponent<AutoScroll> ().ChangeSong (audio.clip.name);
        playRandomMusic ();
    }
}

public void playNext(){
    index++;
    audio.clip = myMusic [index] as AudioClip;
    text.GetComponent<AutoScroll> ().ChangeSong (audio.clip.name);
    audio.Play ();
}

void playRandomMusic() {
    index = Random.Range (0, myMusic.Length);
    audio.clip = myMusic[index] as AudioClip;
    audio.Play();
}
```

5.2 Pohyb hráče

Hráč se ovládá pomocí klávesových šipek. Unity má pro ovládání vytvořenou speciální knihovnu, které umožní hrát i přes gamepad nebo touchpad bez větších zásahů do kódu.

```
void FixedUpdate()
{
    currentSpeed = rb.velocity.magnitude;
    float throttleForce = Input.GetAxis("Vertical");
    if (currentSpeed < maxSpeed)
    {
        rb.AddForce(transform.forward * maxSpeed * throttleForce * 200);
    }
    brake = (throttleForce < 0) ? true : false;
    if (brake)
    {
        rb.drag = 1.5f;
    }
    else rb.drag = 1;
    rb.transform.Rotate(0, Input.GetAxis("Horizontal") * 3, 0);
}
```

5.3 Umělá inteligence

Ve hře jsem vytvořil také umělou inteligenci. Autem hráče jsem projel pár kol a program si z mého průjezdu automaticky vytvořil body průjezdu. Souřadnice bodů a další veličiny se následně uloží do souboru. Z těchto souborů si potom každé auto náhodně vybere průjezdní trasu, kterou pojede. Kromě trasy si také vybere svůj vzhled ze tří předdefinovaných aut. Při závodu tedy auto postupně čte jednotlivé průjezdní body a snaží se jimi projet. Protože by auta jezdila hodně trhaně, musel jsem přidat i toleranci, která uzná, že auto průjezdním bodem projelo.

```
private bool equalPositions()
{
    Vector3 temp = transform.position;
    if(temp.x < actualRecord.x + maxTolerance && temp.x > actualRecord.x - maxTolerance)
    {
        if (temp.z < actualRecord.z + maxTolerance && temp.z > actualRecord.z - maxTolerance)
        {
            return true;
        }
    }
    return false;
}
```

```

public static List<RecordPoint> ReadFromFile(string path)
{
    List<RecordPoint> temp = new List<RecordPoint>();
    string[] lines = System.IO.File.ReadAllLines(path);
    int count = 0;
    foreach (string line in lines)
    {
        string[] splited = line.Split(',');

        float throttle = float.Parse(splited[0]);
        bool brake= bool.Parse(splited[1]);
        float x = float.Parse(splited[2]);
        float y = float.Parse(splited[3]);
        float z = float.Parse(splited[4]);
        count++;
        temp.Add(new RecordPoint(throttle, brake, x, y, z));
    }
    return temp;
}

```

5.4 GameBoss

Třída GameBoss se stará o kompletní fungování hry. Řídí jednotlivé kroky, jak mají jít za sebou. Jako například naslouchá stisknutí klávesy pro start hry a následně spouští odpočet. Třída také přehrává některé zvuky a spouští animace kamery. Dokáže měnit i důležité GUI texty ve hře, jako například vyhlášení vítěze.

```

if (intro) {
    if (Input.anyKey) {
        intro = false;
        countdown.enabled = true;
        logo.gameObject.SetActive (false);
        pressAnyKey.gameObject.SetActive (false);
        camControll.StartGame ();
        startCountdown ();
    }
}

```

```

void startCountdown(){
    player.GetComponent<Car> ().GetComponent<AudioSource> ().enabled = true;
    foreach (Car c in cars) {
        c.GetComponent<AudioSource> ().enabled = true;
    }
    InvokeRepeating ("Countdown", 1.0f, 1.0f);
    this.GetComponent<AudioSource>().Play ();
}
void Countdown () {
    AudioSource audioSource = this.GetComponent<AudioSource> ();
    if (--seconds == 0) {
        CancelInvoke ("Countdown");
        countdown.enabled = false;
        StartRace ();
        audioSource.clip = startSound;
        audioSource.Play ();
    }
    countdown.text = seconds.ToString();
    audioSource.Play ();
    increaseVolume ();
}

```

5.5 Vítěz

Dlouho jsem přemýšlel, jestli ve hře vytvořím ukazatel aktuální pozice hráče vůči ostatním. Po dlouhém zkoušení jsem však tento nápad opustil a přistoupil na metodu, kdy se zobrazí až pořadí v cíli. Vyřešil jsem to tak, že start obsahuje box collider, který má ze začátku tag start. V polovině závodiště je bod, který změní tag u startu na finish. Po této změně je možné závod dokončit, aniž by došlo k podvádění. Také jsem vytvořil neprůchozí stěny, aby nebylo možné si cestu zkracovat přes trávu.

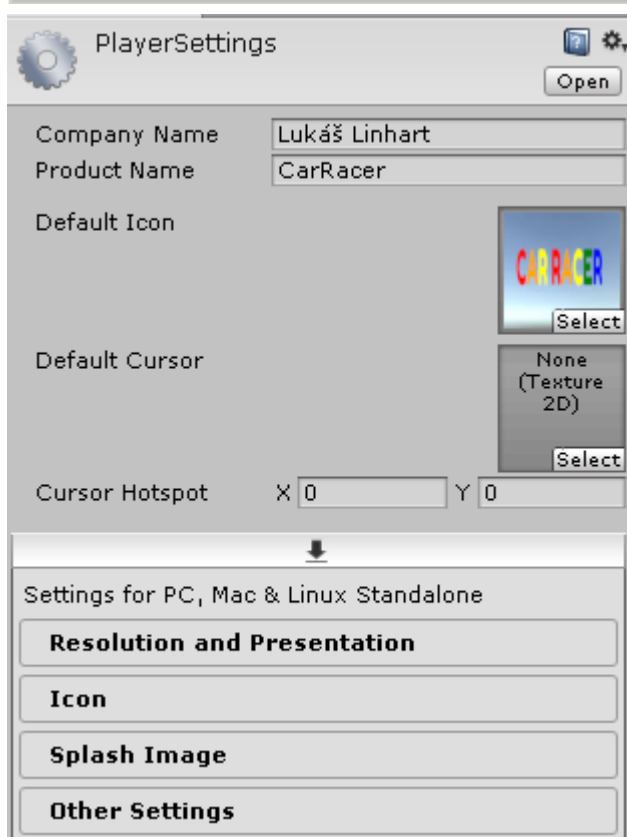
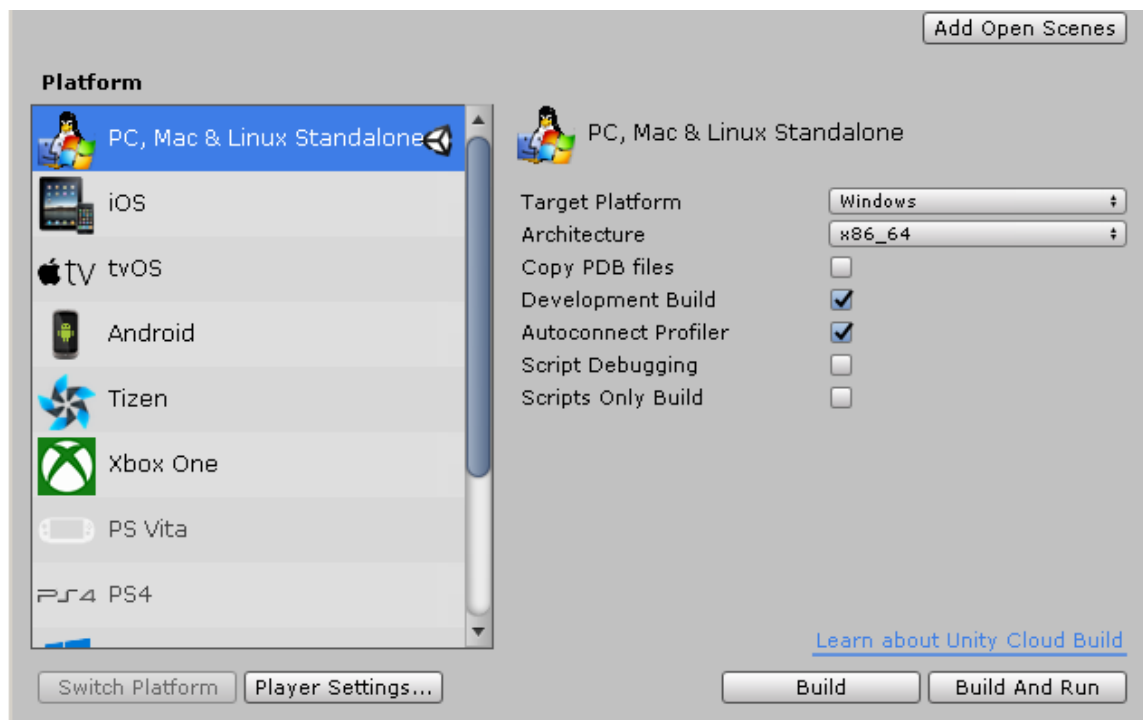
```

public void UpdatePosition(){
    position.SetActive (true);
    position.GetComponent<Text> ().text = "You finished <color=\"red\">"+countOfFinishedCars+"</color>";
    player.GetComponent<Car> ().GetComponent<movement>().enabled = false;
    camControll.EndGame ();
}
if (other.tag == "checkpoint") {
    GameObject.Find("start").transform.tag = "finish";
}
else if (other.tag == "finish")
{
    GameObject.Find("AI").GetComponent<AI>().StopRecording();
    GameBoss gb = GameObject.Find ("GameBoss").GetComponent<GameBoss>();
    gb.countOfFinishedCars++;
    gb.UpdatePosition ();
}

```

6 TESTOVÁNÍ

Pro vytvoření finální release verze programu má Unity nástroj. Zde jde vybrat, pro kterou platformu chcete výsledek vytvořit. Existuje zde i nastavení ikon a ostatní podrobná nastavení výstupního programu.



7 CHYBY V UNITY

Čím déle jsem na projektu pracoval, tím jsem nacházel víc chyb, které Unity obsahuje. Jednou z největších chyb, která mě provázela od začátku byly metasoubory. Tyto soubory Unity tvoří samo, když si načítá nové objekty a přetváří si je do čitelného formátu, aby s nimi mohlo rychleji pracovat. Avšak aby složka s metadaty zabírala víc místa na disku, než polovina projektu se mi zdá jako nevyřešený problém. Existuje zde však řešení, před zazipováním jsem smazal celou složku s metadaty. Takto se velikost projektu snížila na polovinu a nebyl pro mě problém nahrát jej na disk Google. Při následném otevření projektu se však všechny tyto metasoubory musely znovu vytvořit a projekt zase měl svoji dvojnásobnou velikost.
















Další problém, na který jsem narazil bylo špatné fyzické chování aut. Pro auta existuje v Unity vytvořená knihovna. Tato knihovna potřebuje pro svoje správné fungování rozdělení auta na kola a kapotu. Snažil jsem se tuto knihovnu použít, bohužel i přes důkladné dodržení dokumentace se auta chovala naprosto nerealisticky. Náhodně odskakovala do vzduchu a při kolizích se stávalo, že místo zpomalení ještě víc zrychlila.

8 VÝHODY UNITY

Když už jsem výše popsal chyby, na které jsem narazil, musím také sepsat výhody, které Unity nabízí. V Unity se mi líbí, že jde spoustu věcí „naklikat“ místo zdoluhavého psaní kódu. Také hodně věcí je zde již vytvořeno buď přímo od vývojářů, nebo komunitního trhu AssetStore. Dokonce se z trhu dají stáhnout přímo hotové projekty, které slouží jako tutoriály pro naučení se s programem.

SORT BY RELEVANCE / **POPULARITY** / NAME / PRICE / RATING / UPDATED

1 2 1 - 36 of 45

 2D Game Kit Unity Essentials/S... Unity Technologies ★★★★ (163) FREE	 Survival Shooter T... Unity Essentials/S... Unity Technologies ★★★★ (12103) FREE	 Space Shooter tut... Unity Essentials/S... Unity Technologies ★★★★ (12809) FREE
 2D Roguelike Unity Essentials/S... Unity Technologies ★★★★ (1944) FREE	 Tanks! Tutorial Unity Essentials/S... Unity Technologies ★★★★ (11110) FREE	 2D Platformer Unity Essentials/S... Unity Technologies ★★★★ (12725) FREE
 2D UFO Tutorial Unity Essentials/S... Unity Technologies ★★★★ (1356) FREE	 Unity ARKit Plugin Unity Essentials/S... Unity Technologies ★★★★ (1109) FREE	 VR Samples Unity Essentials/S... Unity Technologies ★★★★ (1477) FREE
 Tower Defense Te... Unity Essentials/S... Unity Technologies ★★★★ (124) FREE	 Adam Character P... Unity Essentials/S... Unity Technologies ★★★★ (1204) FREE	 Simple Mobile Pla... Unity Essentials/S... Unity Technologies ★★★★ (167) FREE
 Viking Village Unity Essentials/S... Unity Technologies ★★★★ (1819) FREE	 1/6 - Adventure Tu... Unity Essentials/S... Unity Technologies ★★★★ (174) FREE	 Endless Runner - S... Unity Essentials/S... Unity Technologies ★★★★ (196) FREE

Další výhodou je, že Unity dokáže vyexportovat projekt pro všechny populární platformy. Unity nově podporuje i rozšířenou realitu, pro kterou již existují mnohé projekty vysvětlující, jak s ní správně pracovat. Výsledné projekty běží ve všech zařízeních určených pro VR.



Cardboard



SAMSUNG
Gear VR



Microsoft HoloLens

htc VIVE

Podporované VR technologie

VR Samples

Unity Essentials/Sample Projects
Unity Technologies

★★★★ (1477)

FREE

Download

<http://unity3d.com/learn/tutorials/topics/virtual-reality>

The Unity VR Samples pack is a great way to get started in VR development. With this pack of a menu and 4 mini-games, you can build to Oculus DK2 and GearVR. See the learning material that accompanies this package to get started here -

Při práci mi také hodně pomohla perfektně napsaná dokumentace, z níž jsem hodně čerpal. Ke každé metodě, která je součástí Unity, zde existuje popis, vysvětlení i ukázka.

Transform.LookAt

[Leave feedback](#) [Other Versions](#)

SWITCH TO MANUAL

```
public void LookAt(Transform target, Vector3 worldUp = Vector3.up);
```

Parameters

target	Object to point towards.
worldUp	Vector specifying the upward direction.

Description

Rotates the transform so the forward vector points at /target/'s current position.

Then it rotates the transform to point its up direction vector in the direction hinted at by the worldUp vector. If you leave out the worldUp parameter, the function will use the world y axis. worldUp is only a hint vector. The up vector of the rotation will only match the worldUp vector if the forward direction is perpendicular to worldUp.

```
// This complete script can be attached to a camera to make it
// continuously point at another object.

// The target variable shows up as a property in the inspector.
// Drag another object onto it to make the camera look at it.
using UnityEngine;
using System.Collections;
```

Ukázka z dokumentace

9 SEZNAM POUŽITÉ LITERATURY A JINÝCH ZDROJŮ

Internet:

- *Unity Manual* [online]. [cit. 2018-03-24]. Dostupné z: <https://docs.unity3d.com/Manual/index.html>
- *Radio script* [online]. [cit. 2018-03-24]. Dostupné z: <https://answers.unity.com/questions/1026068/play-next-song.html>
- *StackOverflow* [online]. [cit. 2018-03-24]. Dostupné z: <https://stackoverflow.com/>
- *Fórum Unity* [online]. [cit. 2018-03-24]. Dostupné z: <https://answers.unity.com/index.html>
- *MagicaVoxel* [online]. [cit. 2018-03-24]. Dostupné z: <https://ephtracy.github.io/>
- *Hudebník Goblins From Mars* [online]. [cit. 2018-03-24]. Dostupné z: <https://soundcloud.com/goblinsfrommars>
- *GUI obrázky* [online]. [cit. 2018-03-24]. Dostupné z: <https://assetstore.unity.com/packages/2d/gui/icons/simple-ui-103969>

10 ZÁVĚR

S mojí závěrečnou prací jsem spokojen. Strávil jsem nad tímto projektem spoustu času, a myslím, že výsledek se mi povedl. Naučil jsem se některé nové věci, jako například pracovat v programu MagicaVoxel, používat animace v Unity, nebo třeba lépe pochopit kolize. Jsem rád, že jsem se vypořádal s umělou inteligencí, ze které jsem měl na začátku největší strach. Hojně jsem využíval dokumentaci a tutoriály, které mi hodně ulehčily práci.

11 RESUMÉ

Jako téma mojí maturitní práce jsem si zvolil naprogramování závodní hry v herním enginu Unity. Pro grafickou část hry jsem využil program MagicaVoxel. Také jsem vytvořil částečnou umělou inteligenci. Myslím, že tato práce mi otevřela cestu k vytvoření dalších her a projektů.

As theme of my graduating work I choosed to program the racing game in game engine Unity. For the graphical part of game I used program called MagicaVoxel. I also created a basic artificial intelligence. I think this project motivated me to create some next game.

12 PŘÍLOHY PRÁCE

Na CD se nachází dokumentace, zdrojový kód aplikace a výsledný exe soubor se hrou.