

Univerzita Pardubice
Dopravní fakulta Jana Pernera

Mobilní aplikace pro podporu propagace
studijních oborů Dopravní fakulty Jana Pernera

David Švarc

Bakalářská práce

2019

Univerzita Pardubice
Dopravní fakulta Jana Pernera
Akademický rok: 2018/2019

ZADÁNÍ BAKALÁŘSKÉ PRÁCE (PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **David Švarc**
Osobní číslo: **D16152**
Studijní program: **B3709 Dopravní technologie a spoje**
Studijní obor: **Aplikovaná informatika v dopravě**
Název tématu: **Mobilní aplikace pro podporu propagace studijních oborů
Dopravní fakulty Jana Pernera**
Zadávající katedra: **Katedra informatiky v dopravě**

Z á s a d y p r o v y p r a c o v á n í :

Cílem teoretické části bakalářské práce bude popsat současné možnosti tvorby mobilních aplikací, porovnání možností vybraných vývojových prostředí a detailní popis zvolené technologie. V praktické části bakalářské práce bude navržena sada aplikací pro podporu propagace studijních oborů DFJP. Cílovou skupinou budou zájemci o studium na DFJP, aplikace jim zábavnou formou přiblíží jednotlivé studijní obory. Budou navrženy a popsány vhodné datové struktury a algoritmy, které budou při implementaci aplikace využity.

Rozsah grafických prací:

Rozsah pracovní zprávy: 30 normostran

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:

Vedoucí bakalářské práce: Ing. Stanislav Machalík, Ph.D.
Katedra informatiky v dopravě

Datum zadání bakalářské práce: 11. prosince 2018

Termín odevzdání bakalářské práce: 24. května 2019


doc. Ing. Libor Švadlenka, Ph.D.
děkan

L.S.


doc. Ing. Karel Greiner, Ph.D.
vedoucí katedry

V Pardubicích dne 11. prosince 2018

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 20. 05. 2019

David Švarc

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu své bakalářské práce Ing. Stanislavu Machalíkovi, Ph.D. za jeho cenné rady, věcné připomínky a vstřícnost při konzultacích a vypracování bakalářské práce. Děkuji také své rodině a milované přítelkyni za podporu a trpělivost, kterou se mnou při psaní této práce měli.

ANOTACE

Bakalářská práce se zabývá problematikou vývoje aplikací pro mobilní zařízení. Shrnuje základní možnosti tvorby aplikace prostřednictvím různých vývojových prostředí a programovacích jazyků. V práci jsou popsány vybrané mobilní operační systémy a nejznámější herní enginy. Podrobně je popsán nástroj Unity, který byl využit i pro tvorbu aplikace, která představuje praktickou část bakalářské práce. Aplikace bude použita jako propagační materiál Dopravní fakulty Jana Pernera. Podstatná část textu je věnována i popisu návrhu, vývoje a implementace miniher v aplikaci.

KLÍČOVÁ SLOVA

Vývoj her, Unity, Android, iOS, C#, mobilní zařízení

TITLE

Mobile application designed to support the promotion of fields of study on Jan Perner Faculty of Transport Engineering

ANNOTATION

The thesis covers the issues of the development of applications for mobile devices. It summarizes the basic possibilities of application development through various development environments and programming languages. Selected mobile operating systems and the most renowned game engines are covered in the text. The thesis deals in detail with Unity tool, which was also used for creating an application that represents the practical part of the thesis. The application will be used as a promotional material of Jan Perner Faculty of Transport Engineering. A substantial part of the text is devoted to the description of design, development and implementation of mini-games featured in the application.

KEYWORDS

Game development, Unity, Android, iOS, C#, mobile device

OBSAH

| | |
|--|-----------|
| Seznam obrázků | 9 |
| Seznam zkratk | 10 |
| Úvod | 11 |
| 1 Současný stav tvorby mobilních aplikací | 12 |
| 1.1 Mobilní aplikace | 12 |
| 1.2 Mobilní operační systémy | 12 |
| 1.2.1 Android | 13 |
| 1.2.2 iOS | 13 |
| 1.3 Herní engine..... | 13 |
| 1.3.1 Unity | 14 |
| 1.3.2 Unreal Engine | 14 |
| 2 Vývojové prostředí Unity | 15 |
| 2.1 Instalace vývojového prostředí | 15 |
| 2.2 Editor Unity | 15 |
| 2.2.1 Scene view | 16 |
| 2.2.2 Game view | 17 |
| 2.2.3 Inspector view | 17 |
| 2.2.4 Hierarchy | 17 |
| 2.2.5 Project View | 18 |
| 2.3 Herní objekty | 18 |
| 2.3.1 Prefabs | 19 |
| 2.4 Scény v Unity..... | 19 |
| 2.5 Skripty v Unity..... | 19 |
| 2.6 Programovací jazyky | 19 |
| 2.6.1 C#..... | 20 |
| 2.6.2 JavaScript..... | 20 |
| 3 Popis a implementace součástí aplikace | 21 |
| 3.1 Představení aplikace | 21 |
| 3.1.1 Grafická stránka aplikace..... | 21 |

| | | |
|-------|---|-----------|
| 3.1.2 | UI aplikace | 22 |
| 3.1.3 | Ovládání aplikace | 22 |
| 3.2 | Návrhy her | 23 |
| 3.2.1 | Řízení dopravy | 23 |
| 3.2.2 | Kvíz..... | 23 |
| 3.2.3 | Simulace ovládání výhybek kolejiště | 24 |
| 3.2.4 | Sokoban klon | 25 |
| 3.3 | Hry v aplikaci..... | 25 |
| 3.3.1 | Řízení dopravy | 25 |
| 3.3.2 | Kvíz..... | 30 |
| 3.3.3 | Simulace ovládání výhybek kolejiště | 32 |
| 3.3.4 | Sokoban klon | 35 |
| 3.4 | Ovládání a instalace | 38 |
| 3.4.1 | Windows | 39 |
| 3.4.2 | Android | 39 |
| | Závěr | 40 |
| | Použitá literatura | 41 |

SEZNAM OBRÁZKŮ

| | |
|--|----|
| Obrázek 1: Prostředí programu Unity | 16 |
| Obrázek 2: Okno scény | 16 |
| Obrázek 3: Panel Inspektor | 17 |
| Obrázek 4: Panel Hierarchy s objekty..... | 18 |
| Obrázek 5: Grafická podoba hlavního menu aplikace | 22 |
| Obrázek 6: Grafická podoba UI prvků aplikace | 22 |
| Obrázek 7: Prostředí hry řízení dopravy | 23 |
| Obrázek 8: Prostředí minihry kvízu | 24 |
| Obrázek 9: Prostředí hry simulace ovládání výhybek kolejíště | 24 |
| Obrázek 10: Prostředí hry Sokoban klon | 25 |
| Obrázek 11: Tabulka zobrazující výsledek minihry | 26 |
| Obrázek 12: Ukázka assetů ve hře řízení dopravy | 27 |
| Obrázek 13: Varianty vlaků a stanic | 33 |
| Obrázek 14: Herní assety použité ve hře Sokoban klon | 36 |

SEZNAM ZKRATEK

| | |
|----|---------------------------------------|
| 2D | dvoudimenzionální |
| 3D | trojdimenzionální |
| OS | operační systém |
| UI | user interface (uživatelské rozhraní) |
| VR | virtuální realita |

ÚVOD

Novodobým trendem jsou bezesporu mobilní zařízení. Mnoho lidí si bez nich nedokáže představit každodenní život. Většina institucí a firem se snaží jít s dobou a vytvářejí nejrůznější aplikace, které moderní formou představí jejich nabízený produkt. I proto je cílem této bakalářské práce vytvořit mobilní aplikaci sloužící k propagaci studijních oborů Dopravní fakulty Jana Pernera. Ta bude obsahovat čtyři různé minihry, které uchazečům o studia zjednodušenou formou přiblíží charakter případného druhu studia. Myšlenkou bylo vytvořit interaktivní formu předávání informací využitelnou při propagačních akcích školy. Vzhledem k dnešnímu trendu moderním technologií a jejich dostupnosti bude mít projekt podobu mobilní aplikace, která studium daného oboru přiblíží zábavnou formou.

Pro vývoj aplikace je použit herní engine (software sloužící k vývoji videoher) Unity a jako platformy jsou zvoleny populární operační systémy Android a Windows. Cílem práce je vytvořit aplikaci primárně určenou pro dotykové zařízení, kterou by mohla škola využít na propagačních akcích.

Práce začíná rozbořem současného stavu tvorby mobilních aplikací (kapitola č. 1). Jsou zde popsány dva nejčastěji využívané mobilní operační systémy. Na konci této části jsou rozebrány nejběžnější nástroje sloužící k tvorbě her, jak pro počítač, tak pro mobilní zařízení.

Druhá část bakalářské práce (2. kapitola) důkladně analyzuje zvolený engine Unity. Je zde popsáno jeho prostředí, ovládání a práce s jednotlivými herními komponenty, které tvoří výslednou aplikaci.

Hlavní kapitolou práce je detailně charakterizována samotná aplikace (kapitola č. 3). Je představen její hlavní cíl a možnosti využití. Jsou zde detailně popsány grafické a funkční části programu, včetně jednoduchých popisů všech miniher, použité algoritmy a datové struktury, herní objekty a grafické prvky. Součástí je také jednoduchý návod ovládání každé minihry. Konec této kapitoly je určen zejména pro popis ovládání a instalaci výsledné aplikace na různé platformy.

Praktickou částí bakalářské práce je samotná aplikace propagující studium Dopravní fakulty Jana Pernera. Aplikace obsahuje čtyři minihry, přibližující jednotlivé obory na fakultě.

1 SOUČASNÝ STAV TVORBY MOBILNÍCH APLIKACÍ

Mobilní telefon vlastní v dnešní době téměř každý člověk. Stal se běžným společníkem našeho života. Mezi nejčastěji používané aplikace v telefonu patří prohlížeč internetu, aplikace fotoaparátu, emailový komunikátor či programy pro sledování sociálních sítí. Lidé stráví používáním těchto aplikací průměrně 3 hodiny denně. [1]

Tvorba aplikací vzrostla s používáním takzvaných „chytrých“ telefonů. Ty se vyznačují tím, že používají pokročilý operační systém s podporou instalace programů. Podle průzkumu společnosti Vodafone se četnost „chytrých telefonů“, neboli smartphonů, u jejich zákazníků pohybuje okolo 79 %. Zbytek uživatelů používá tzv. „hloupý“ mobilní telefon bez operačního systému. [2]

Aplikace jsou k dispozici pro běžného uživatele nejčastěji prostřednictvím předinstalovaných programů, které jsou určeny ke stahování a hodnocení jednotlivých aplikací. Pro systém android je to program Google Play a pro iOS ho nalezneme pod názvem AppStore. Na obě tyto platformy může svoji aplikaci nahrát téměř kdokoli. I díky tomu jsou tyto „obchody“ doslova přeplněny různými programy, které nemají žádné praktické využití a zmíněné obchody pouze zahlcují. Jenom na již zmíněném Google Play nalezneme přes 2 miliony aplikací. [3]

1.1 Mobilní aplikace

Mobilní aplikace je software, který je určen výhradně pro mobilní zařízení. Ty se často liší část od těch počítačových. Programátor musí počítat s tím, že mobilní zařízení má obvykle omezený výkon, funguje na baterii a má malý displej (často dotykový). Na všechny tyto faktory je potřeba myslet již při plánování samotné aplikace. Je nezbytné také dbát na jednoduchost a intuitivnost ovládání, neboť znalost v oblasti užívání mobilních telefonů se mezi uživateli zásadně liší.

1.2 Mobilní operační systémy

Mobilní operační systémy jsou speciálně navržené systémy pro mobilní zařízení, jako jsou smartphony, tablety ale i chytré hodinky či osobní počítače. Tyto systémy jsou většinou šity na míru hardwarovému zařízení, na kterém operační systém najdeme. Celosvětově nejčastěji

používaným OS je Android, následuje ho iOS. Tyto dva systémy představují přes 95 % světového trhu. [4] K dalším patří Windows Phone nebo Symbian.

1.2.1 Android

Historie tohoto operačního systému sahá do roku 2003, kdy Andy Rubin, Rich Miner, Nick Sears a Chris White založili společnost Android Inc. Firmu v roce 2005 koupil Google, který Android spravuje dodnes. První verze operačního systému vyšla v roce 2007. Platforma vychází z linuxového jádra a je volně dostupná jako otevřený systém. Roku 2008 vyšel první telefon HTC Dream, na kterém byl nainstalovaný systém OS Android 1.0. Nejnovější verzí je Android 9.0 „Pie“, který vyšel v srpnu roku 2018. [5]

1.2.2 iOS

iOS je operační systém vytvořený firmou Apple Inc. v roce 2007. První oficiální verze se jménem „iPhone OS 1.0.“ vyšla 6. března 2008. Po několika dalších vydaných verzích byl 7. června 2010 systém přejmenován na dnes již známé „iOS“. Tento operační systém je určen pouze pro zařízení firmy Apple, tedy iPhone, iPad a iPod Touch. iOS je odlehčená verze počítačového operačního systému firmy Apple MacOS. Aktuální verze systému je iOS 12. [6]

1.3 Herní engine

Herní engine je software, který pomáhá jeho uživateli s vývojem herních projektů. Je navržen tak, aby vývojáři ulehčil a urychlil práci. Ovlivňuje fyziku, vykreslování, skriptování a mnoho dalšího. Obsahuje vlastní sadu vývojových nástrojů, díky kterým je uživatel schopen vyvíjet aplikace na různé platformy (PC, herní konzole, mobilní telefony). Větší herní vývojová studia mají naprogramované vlastní herní enginey, menší studia si licencují cizí enginey. Mezi nejznámější patří Unreal Engine, Unity, Cry Engine a Game Maker. [7]

Pro nové mobilní hry je nejčastěji používaný herní engine Unity. Ten přímo obsahuje sadu vývojových nástrojů pro tvorbu mobilních her a aplikací, ať už ve 2D nebo 3D prostoru. [8] Unity je dobře známo svou jednoduchou portovatelností na mobilní platformy iOS a Android, i díky tomu je na obou herních obchodech (Google Play a AppStore) vytvořeno přes 30 % aplikací právě v tomto enginu. [9]

1.3.1 Unity

Unity je multiplatformní herní engine, který vyvíjí společnost Unity Technologies. První verze programu vyšla 8. června 2005. Unity aktuálně podporuje vývoj na více než 20 platformách, mezi které patří PC, herní konzole a mobilní telefony. V Unity je možné vytvořit prakticky jakýkoliv typ hry nebo aplikace. Skripty je možné psát v programovacím jazyce C# nebo UnityScript (podobný JavaScriptu). V engine je implementován obchod, ve kterém jsou nahrány všechny různé *assety* (položka reprezentující jakýkoliv přidaný objekt/model ve hře), které je možné zdarma nebo za poplatek přidat do svého projektu. Unity nabízí aktuálně pátou verzi svého programu, která je nabízena zdarma ke stažení kterémukoliv uživateli. [10]

1.3.2 Unreal Engine

Unreal Engine je herní engine společnosti Epic Games vytvořený v roce 1998. Jádro je napsáno v jazyce C++. Podporuje vývoj pro mnoho platforem, mimo jiné na PC, herní konzole, Mac OS nebo Linux. Díky implementovanému editoru *Blueprint*, který dovoluje vizuální programování pomocí vzájemného spojování jednotlivých programovacích bloků, vývojář nemusí ani umět programovat. V tomto engine je vyprodukováno mnoho velkých herních titulů, jejichž prodej přesáhl milion kusů. Je velmi populární v tvorbě VR (virtuální realita) her díky své schopnosti vykreslit velmi realistickou grafiku. Unreal Engine obsahuje svůj obchod s *assety* stejně jako konkurenční Unity. Ten nabízí modely, skripty, animace a mnoho dalšího ke stažení zdarma nebo za poplatek. Pro menší vývojáře je engine zdarma, při překročení určitého výtěžku ze hry je aplikován měsíční poplatek. [11]

2 VÝVOJOVÉ PROSTŘEDÍ UNITY

Pro vývoj aplikace byl použit herní engine Unity. Ten je známý svojí jednoduchou ovladatelností, přehledností a portovatelností výsledného programu na nejrůznější platformy. Díky integrovaným vývojovým nástrojům je ideální na podobný druh aplikací. Výhodou je také možnost psát kód v jazyce C#, který je velmi intuitivní a přehledný. Pro naprogramování projektu je využita bezplatná verze editoru Unity, která k vývoji této aplikace bohatě postačí.

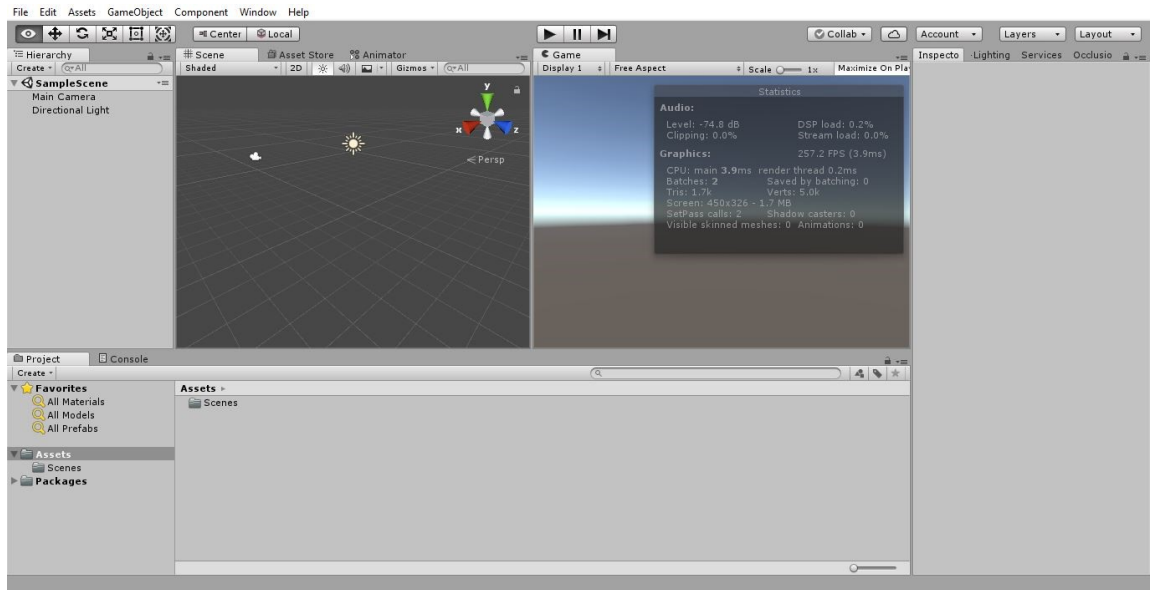
2.1 Instalace vývojového prostředí

Unity Technologies aktuálně nabízí tři edice svého programu. První je Unity Free, která nabízí zdarma omezené možnosti editoru. Zde je také podmínka, která dovoluje používat tuto verzi pouze v případě, kdy se výdělek projektu pohybuje pod 100 000 dolarů za jeden rok. Druhou verzí je Unity Plus. Tato edice je určena pro uživatele, kteří chtějí posunout své schopnosti nad rámec bezplatné verze Unity Free. Je placená sazbou 25 dolarů za měsíc. Třetí, poslední verze, se nazývá Unity Pro, která poskytuje veškeré funkce programu a je určena především pro větší společnosti. Je zpoplatněna 125 dolary za období jednoho měsíce [12].

Bezplatnou verzi programu Unity je možné stáhnout na oficiální webové stránce Unity. [12] Zde je potřeba zvolit platformu, na které budeme chtít aplikaci vyvíjet (možnosti jsou Windows nebo MacOS). Po zvolení platformy a odsouhlasení smluvních podmínek se automaticky spustí stažení instalačního balíčku Unity. Po dokončení instalace se nám spustí program Unity Hub, kde je možné vytvořit nový či editovat již vytvořený projekt.

2.2 Editor Unity

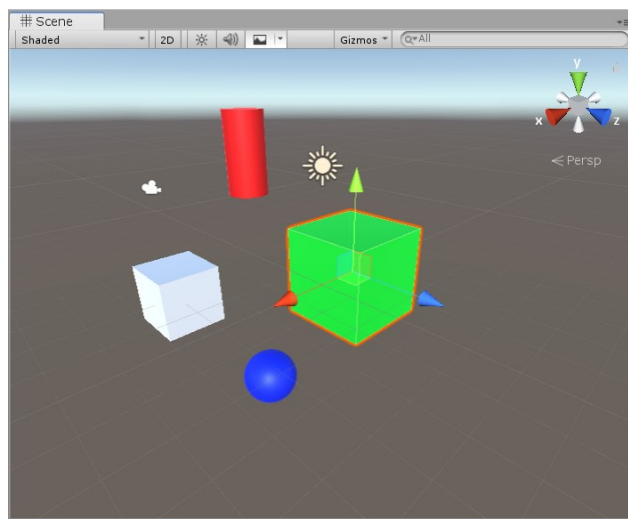
Po spuštění programu Unity Hub a vybrání či vytvoření projektu se zobrazí hlavní prostředí programu (obr. 1). To se skládá z menších oken (takzvaných pohledů). Každý z těchto pohledů má svoji důležitou funkci a může být uživatelem v okně programu uspořádán podle vlastního užití. Pohledy mohou být také skryty. Všechny tyto volby můžeme najít v položce menu *Window*.



Obrázek 1: Prostředí programu Unity

2.2.1 Scene view

Scene view je okno, kde je samotná hra tvořena. Nabízí perspektivní 3D pohled na celý projekt, který je možno změnit v pohled ortografický (z předu, shora, z leva apod.). Vše, co zde změníme, se promítne do obrazu hry.



Obrázek 2: Okno scény

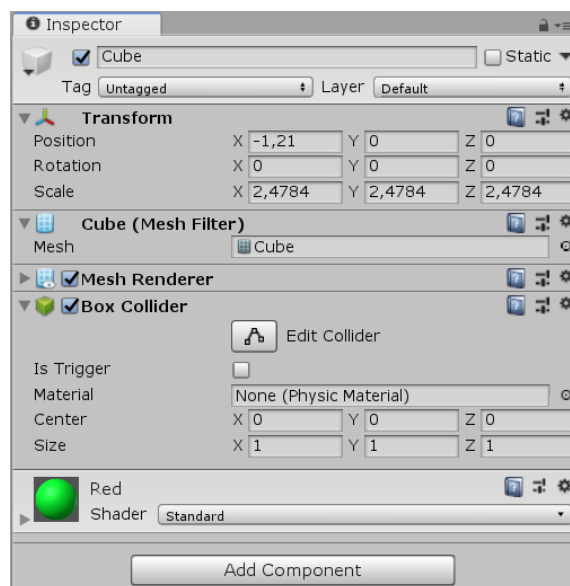
Jednotlivé objekty do tohoto pohledu přidáváme tak, že je sem z pohledu *Project* pomocí myši přetáhneme. S novými objekty je následně možné pohybovat, měnit jejich velikost či rotovat. Volby lze přepínat v panelu nad oknem scény nebo stisknutím přiřazené klávesové zkratky.

2.2.2 Game view

V tomto pohledu nalezneme finální vykreslení naší hry. Po spuštění tlačítka „Play“ se okno aktivuje a hra spustí. Můžeme v něm tedy postupně testovat naši aplikaci. Je zde také možnost změny rozlišení či maximalizování pohledu pro lepší testování hry. Všechny změny vlastností objektů jsou při aktivování tohoto okna pouze dočasné. Při opuštění aktivního pohledu se hodnoty vrací do původního stavu před spuštěním.

2.2.3 Inspector view

Dalším důležitým pohledem programu je *Inspector*. Slouží k změně jednotlivých parametrů každého herního objektu. Pokud v okně *Scene* nebo *Hierarchy* vybereme nějaký objekt, v panelu inspektor jsou jeho parametry zobrazeny. Může obsahovat textové pole, posuvníky či rozbalovací menu. Mezi tyto funkce patří i položka *Transform*, která udává polohu, rotaci a velikost každého objektu v daném herním světě. Klíčovou funkcí je možnost přístupu k proměnným v jednotlivých skriptech, které jsou k objektu přidány.

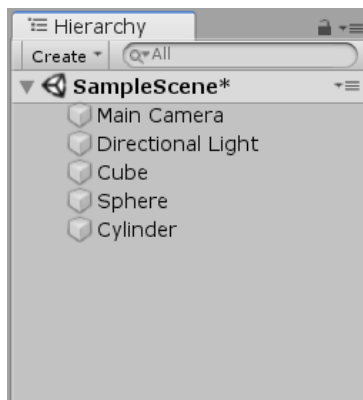


Obrázek 3: Panel Inspektor

2.2.4 Hierarchy

V tomto panelu nalezneme všechny objekty, které tvoří aktivní scénu. Unity podporuje možnost tyto objekty strukturovat jako rodiče a potomka, tato funkce pomáhá k rozlišení a práci s jednotlivými objekty. V okně je možné modely volně přesunovat či nechat Unity

objekty seřadit podle abecedy. Při dvojitým kliknutí na objekt v panelu *Hierarchy* se daný objekt vyhledá a zobrazí v okně *Scene*.



Obrázek 4: Panel Hierarchy s objekty

2.2.5 Project View

V tomto pohledu můžeme najít vše, co bylo v dosavadním projektu vytvořeno (složky, skripty, modely, textury a dále). Zde by se měl klást důraz na organizaci, neboť při větších projektech se v tomto pohledu nachází mnoho souborů a hledání konkrétního assetu může být obtížné. Jednotlivé herní objekty by měly být rozděleny do různých složek a podsložek nazvaných podle jejich funkce. Pokud chceme do projektu přidat asset vytvořený mimo program Unity, přesuneme ho právě do tohoto panelu. Unity už si s importem samo poradí.

Project view také obsahuje textové pole hledání, které při zadání klíčového slova vyhledá soubory s podobným jménem bez nutnosti manuálního procházení mnoha složek. Je možné zde vyhledat herní objekty podle typu souboru nebo jmenovky, kterou je možné každý objekt opatřit.

2.3 Herní objekty

Herní objekty (anglicky Game Objects) jsou základními prvky každého Unity projektu. Patří mezi ně modely, světla, kamery, ale i simulace částic a text. Každý z nich obsahuje jednotlivé komponenty, které je možné v panelu *Inspector* modifikovat. Základním je komponent *Transform*, který určuje polohu, rotaci a velikost daného objektu v prostoru herní scény.

2.3.1 Prefabs

Prefabs jsou speciální herní objekty, které v sobě mají napevno uloženy parametry a hodnoty. Jejich hlavní výhodou je znovupoužitelnost v dalších scénách projektu s uchovanými hodnotami.

2.4 Scény v Unity

Scény v Unity představují to, co známe v různých hrách jako jednotlivé úrovně, mapy a menu. Každá další scéna pak může obsahovat různé herní objekty a představovat tak naprosto odlišné herní mechaniky nežli scéna předchozí. Při vytváření nové úrovně vznikne prázdná scéna, se kterou můžeme nadále pracovat. Při dokončování hry nesmíme zapomenout přidat všechny aktivní scény do seznamu scén, které mají být vybudovány do finální podoby aplikace. Novější verze Unity podporuje úpravu více scén najednou.

2.5 Skripty v Unity

Skripty jsou jednou z klíčových částí vytváření her. Ty jsou přiřazeny jednotlivým objektům v editoru, které pak mohou měnit jejich chování. Pomocí panelu *Inspector* je možné nastavovat některé parametry a proměnné deklarované ve skriptu. Jedním kliknutím na skript v pohledu scény nebo projektu se nám skript v panelu *Inspector* vypíše pro čtení, při dvojitým kliknutí je skript otevřen v přednastaveném externím vývojovém prostředí (např. Visual Studio). Zde je možné kód upravovat.

Základní metodami, které se automaticky vygenerují při vytvoření nového skriptu, jsou `Start()` a `Update()`. Metoda `Start()` je volána pouze jednou při spuštění skriptu a metoda `Update()` je programem volána pro každý vykreslovaný snímek (frame). Dvnitř těchto dvou metod lze psát libovolný kód.

2.6 Programovací jazyky

Editor Unity umožňuje tvořit skripty ve dvou programovacích jazycích. Prvním jazykem, kterým Unity disponoval, byl jazyk Boo. Ten byl v páté verzi programu z důvodu zastaralosti odstraněn. Současnými programovacími jazyky jsou C# a obdoba JavaScriptu zvaná UnityScript. [13][14]

2.6.1 C#

Programovací jazyk C# byl vytvořený firmou Microsoft v roce 2000. Jedná se o vysokoúrovňový objektově orientovaný programovací jazyk. Je založený na jazycích C++ a Java. [15] Jeho základní charakteristikou je:

- Automatická správa paměti. O uvolnění paměti se stará *garbage collector*.
- Zpracování chyb pomocí výjimek.
- Podpora jednoduchá dědičnosti.
- Překladače C# jsou *Case sensitive* (nutnost dodržovat rozdíl malých a velkých písmen).

2.6.2 JavaScript

JavaScript je multiplatformní, objektově orientovaný programovací jazyk. Vytvořil ho Brendan Eich v roce 1995. Je často využíván při tvorbě interaktivních webových stránek. Všechny běžné prohlížeče dnes obsahují jeho jádro. Lidé ho často spojují s jazykem Java, který je ale od JavaScriptu odlišný a slovo „Java“ obsahuje pouze z marketingového hlediska. [16]

3 POPIS A IMPLEMENTACE SOUČÁSTÍ APLIKACE

Hlavním cílem bakalářské práce bylo vytvoření aplikace, která bude sloužit jako propagační nástroj Dopravní fakulty Jana Pernera. Součástí této kapitoly je nejdříve představení samotné aplikace s ukázkou grafických assetů a popisu ovládání. Dále jsou zde stručně popsány všechny minihry nacházející se v aplikaci. Poté následuje podrobný popis jednotlivých miniher včetně ukázky skriptů a použitých assetů.

3.1 Představení aplikace

Při vývoji aplikace byl kladen důraz především na jednoduchost obsluhy. Základním rysem je snadná ovladatelnost, která zajistí dostupnost širokému spektru uživatelů. Koncovým uživatelem může být stejně tak žák střední školy jako adept postgraduálního studia. Vzhledem k této skutečnosti bylo zapotřebí výslednou aplikaci necílit na konkrétní věkové skupiny, ale vybalancovat ji pro každou generaci.

Aplikace je tvořena čtyřmi jednoduchými minihrami, které mají za úkol reprezentovat podstatu studia vybraných oborů Dopravní fakulty Jana Pernera Univerzity Pardubice. Po spuštění aplikace je zobrazena hlavní nabídka, která představuje přehled jednotlivých miniher. Po zvolení konkrétní možnosti je odpovídající hra spuštěna.

Jednotlivé minihry jsou navrženy tak, aby upoutaly pozornost a podaly základní informace o studiu stručnou formou. Záměrem bylo nedělat hry zbytečně dlouhé, aby uživatele nenudily.

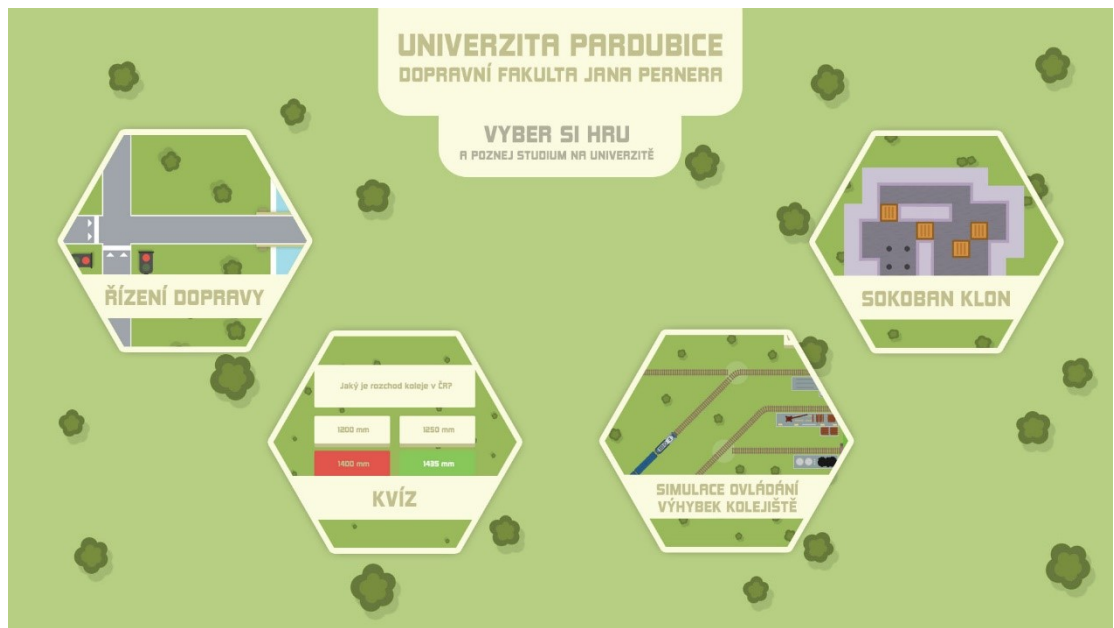
3.1.1 Grafická stránka aplikace

Vzhled aplikace se nese v minimalistickém duchu. Všechny minihry jsou vyvedené v podobné stylizaci, což tvoří ucelenější dojem celého projektu. Aplikace je tvořena v 2D grafické podobě. Textury (grafické prvky) byly nejdříve vytvořeny v grafickém editoru a poté exportovány jako bitmapové obrázky. Ty byly nahrány do editoru Unity, kde se jimi osadily jednotlivé herní objekty.

Důležitým krokem při vývoji bylo vždy podobu hry nejdříve v zjednodušené formě navrhnout a otestovat její ovládání na různých zařízeních. Poté, co byl návrh dostatečně otestován, bylo možné všechny herní komponenty osadit texturou tak, aby hra dostala podobu předem navržené šablony.

V podobných aplikacích je potřeba, aby bylo na první pohled zřejmé, které herní objekty fungují jako interaktivní tlačítka sloužící například pro změnu scény. To je často řešeno tím,

že se danému obrázku přidá vržený stín. Ten způsobí 3D efekt, a uživateli tak napoví, že se jedná o stisknutelný prvek v aplikaci.



Obrázek 5: Grafická podoba hlavního menu aplikace

3.1.2 UI aplikace

Uživatelské rozhraní je takové rozhraní, které je prostředníkem mezi uživatelem a systémem. Využívá se v situacích, kde dochází ke kontaktu člověka a zařízení. V tomto případě tlačítka menu, ukazatel skóre nebo tlačítka pro ukončení minihry. Při vytváření grafické podoby bylo zapotřebí zamyslet se nad co nejintuitivnějším ovládním aplikace. Zároveň nesmělo být zapomenáno na fakt, že by mohla být aplikace ovládána dotykově či kurzorem myši na obrazovce. S oběma těmito variantami bylo při vývoji počítáno. Stylizace všech UI prvků je v celé aplikaci stejná, aby změna uživatele zbytečně nemátla.



Obrázek 6: Grafická podoba UI prvků aplikace

3.1.3 Ovládání aplikace

Aplikace je připravena na ovládání pomocí dotyku na mobilním zařízení nebo pomocí myši na stolním počítači či notebooku. Ovládání je přizpůsobené tak, aby bylo možné se z každé

jednotlivé minihry vrátit do hlavního menu bez potřeby dohrání. Přejít na další hru pak probíhá právě v tomto menu.

Jednotlivé minihry jsou většinou rozděleny do dílčích úrovní. Z těch lze vyvoláním tlačítka „MENU“ zobrazit obrazovku s vypsáním všech možných úrovní minihry. Pokud je v minihře důležitý čas, je při stisknutí tlačítka „MENU“ pozastaven, dokud se uživatel stisknutím tlačítka „POKRAČOVAT“ do hry nevrátí.

3.2 Návrhy her

V této podkapitole se nachází stručný výpis jednotlivých miniher s obrázky jejich grafického prostředí. Nalezneme zde vypsání oborů, které mají dílčí minihry přibližovat.

3.2.1 Řízení dopravy

V této hře má uživatel za úkol řídit dopravu. Pomocí řízení semaforů se snaží o co největší plynulost v dopravě, zároveň však musí dát pozor, aby se vozidla v křižovatce nesrazila. Hra obsahuje pět úrovní, které jsou uspořádány podle obtížnosti. Tato minihra má blíže představit obor Technologie a řízení dopravy.



Obrázek 7: Prostředí hry řízení dopravy

3.2.2 Kvíz

Tato minihra má za úkol přiblížit obory Dopravní prostředky: Kolejová vozidla a Dopravní prostředky: Silniční vozidla. Uživateli se zobrazí 10 náhodně generovaných otázek z oblasti

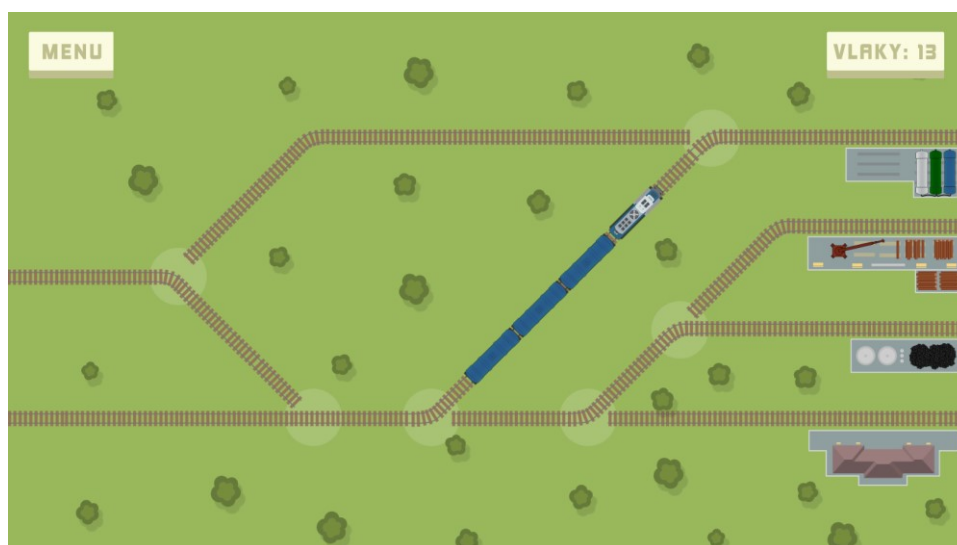
silničních a kolejových vozidel. Po zodpovězení všech se ukáže výsledné skóre správných odpovědí.



Obrázek 8: Prostředí minihry kvízu

3.2.3 Simulace ovládání výhybek kolejiště

Podstatou hry je seznámení se s řízením vlakové dopravy. Hráč pomocí odklápění výhybek určuje trasu vlaku, která se liší podle nákladu dané soupravy. Minihra se skládá ze čtyř různých kolejišť rozdělených podle obtížnosti ovládání. Má za úkol přiblížit studium oboru Elektrotechnické a elektronické systémy v dopravě, Technologie a řízení dopravy a Dopravní prostředky: Kolejová vozidla.



Obrázek 9: Prostředí hry simulace ovládání výhybek kolejiště

3.2.4 Sokoban klon

Poslední minihra si bere inspiraci v počítačové hře Sokoban z roku 1980. Sokoban je logická hra, kde má hráč za úkol posouvat bedny na vyznačená místa na mapě. Hra obsahuje pět úrovní seřazených od nejlehčích po ty nejtěžší. Má za úkol představit obor Dopravní stavitelství.



Obrázek 10: Prostředí hry Sokoban klon

3.3 Hry v aplikaci

V této podkapitole jsou detailně popsány všechny minihry nacházející se v aplikaci. V textu jsou vylíčeny herní mechaniky s ukázkou skriptů, které tvoří jádro každé minihry. Vyobrazené jsou zde i použité grafické assety.

3.3.1 Řízení dopravy

Úkolem této hry je řízení provozu na křižovatce pozemních komunikací pomocí ovládání semaforů. Hráč musí dbát na plynulost provozu a zároveň zabránit dopravním nehodám. Hra obsahuje pět různých úrovní obtížnosti. Ty se liší podle počtu pruhů dopravních cest. Při spuštění hry se ukáže krátký návod, který má za úkol představit ovládání hry. Poté už je uživatel uveden do samotné hry.

Herní mechanika

Hra je složena ze tří hlavních komponent. První jsou silniční pruhy, po kterém vozidla jezdí. Tyto pruhy se poté kříží. Druhou komponentou jsou semaforey, které jsou ovládány uživatelem.

Třetí komponentou jsou samotná vozidla. Ty jsou náhodně generována do jednotlivých jízdnic pruhů.

Důležitým elementem hry je skóre a čas. Za každé úspěšně dopravené vozidlo se přičte hráči určitý počet bodů. Ten závisí na době, kterou stráví vozidlo stáním v pruhu na semaforu a úrovni obtížnosti daného *levelu* (úrovně). Každá úroveň trvá určitý čas, který je odpočítáván v pravém horním rohu, hned vedle ukazatele hráčova skóre.

V levém horním rohu nalezneme tlačítko „MENU“, které nám hru nejdříve pozastaví a poté ukáže další tlačítka, kterými je možné vybrat jinou úroveň, hru ukončit nebo v rozehrané hře pokračovat.

Konec hry

První variantou skončení hry je vyčerpání časového limitu. Ten má každá úroveň nastavený jinak. S jeho skončením je zobrazena tabulka, která hráči ukazuje výsledek jeho snažení. Prvním určujícím měřítkem je obdržení počet hvězd. Ty jsou darovány podle počtu získaných bodů. Tři hvězdy se zobrazí, pokud je dosažené skóre větší než 95 % možných bodů a dvě se ukáží, pokud je procento získaných bodů větší než 85. Ve všech ostatních případech získá hráč pouze jednu hvězdu. V tabulce je nadále zobrazen počet dosažených bodů a maximální počet bodů, které mohl hráč získat, při správném řízení dopravy. Pod tabulkou pak nalezneme tlačítko „POKRAČOVAT“, které hráče přesune na další úroveň.

Druhou variantou ukončení hry může být dopravní nehoda. Ta nastane v případě srážky dvou vozidel. Uživateli je zobrazena hláška dopravní nehody a tlačítko „RESTART“, které danou úroveň restartuje. Hráč má tedy další pokus na zvládnutí úrovně.



Obrázek 11: Tabulka zobrazující výsledek minihry

Assety

Hra obsahuje přes 20 různých herních assetů. Mezi nimi je i 9 různých dopravních prostředků a dva obrázky semaforu. Hru tvoří celkem šest scén. Pět představuje jednotlivé úrovně a šestá slouží pro výběr těchto úrovní.



Obrázek 12: Ukázka assetů ve hře řízení dopravy

Skripty

Tato minihra obsahuje celkem devět skriptů. V této podkapitole budou vypsány nejdůležitější metody a funkce, které tvoří jádro hry.

```
if (moving){
    transform.position =
Vector3.MoveTowards(transform.position, transform.parent.position, step);
}
```

Tento kód je součástí prefabu dopravního vozidla (skript *Moving.cs*). Má za úkol vozidlo rozpohybovat správným směrem. Vidíme zde podmínku, která je volána v metodě `Update()`. Pokud je splněna podmínka `moving`, která závisí například na tom, jestli nám zbývá nějaký čas nebo je spuštěné menu, metoda `MoveTowards()` nám dokáže rozpohybovat herní objekt (v našem případě vozidlo). Parametrem metody jsou aktuální pozice objektu, cílová pozice a rychlost pohybu. V podmínce se nachází příkaz, který určuje umístění objektu v herním světě.

```
if (!moving)
{
    if (
!GameObject.Find("GameControll").GetComponent<GameControll>().pause &&
!GameObject.Find("GameControll").GetComponent<GameControll>().endTime &&
!accident)
    {
```

```

        t = t + Time.deltaTime;
    }
    if (t >= scoreTime01)
    {
        score = 5;
        warning01.SetActive(true);
    }
    if (t >= scoreTime02)
    {
        score = 1;
        warning01.SetActive(false);
        warning02.SetActive(true);
    }
}

```

Část tohoto kódu patří opět k prefabu jednotlivých vozidel (skript *Moving.cs*). Stará se o vypočítání stráveného času při stání vozidla na křižovatce. Do proměnné *t* se díky metodě *Update()*, ve které se nachází, přičítá čas od posledního načteného snímku. Můžeme zde také vidět, že čas je počítán pouze v době, kdy hra není pozastavena, ukončena či nedošlo k dopravní nehodě. Všechny tyto určující parametry jsou předávány ze skriptu *GameControll.cs*. Dále v kódu můžeme vidět určení skóre, které je za jednotlivé vozidlo přičteno v závislosti na strávené době čekání na semaforu.

```

void FixedUpdate() {
    if (time >= spawnTime && !pause &&
        aktObsazenostPruhu < maxObsazenostPruhu &&
        levelTime > 0 && !endAcc && levelTime >= noCarAtEnd) {
        SpawnObject();
        SetRandomTime();
        spawned++;

        GameObject.Find("GameControll").GetComponent<GameControll>().spawnedCar++;
        aktObsazenostPruhu++;
    }
}

void SpawnObject()
{
    int randomCar = Random.Range(0,
        GameObject.Find("GameControll").GetComponent<GameControll>().carList.Count)
;

    (Instantiate(GameObject.Find("GameControll").GetComponent<GameControll>().carList[randomCar], start.transform.position, start.transform.rotation) as
        GameObject).transform.parent = end.transform;
}

void SetRandomTime()
{

```

```

        spawnTime = Random.Range(minTime, maxTime);
    }

```

Kód zobrazený výše je součástí skriptu *Spawner.cs*. Jeho úkolem je náhodně generovat vozidla do jednotlivých pruhů. V metodě `FixedUpdate()`, se nejdříve zjistí, zda je možné další auto vytvořit. To závisí na obsazenosti pruhu, času do konce limitu úrovně a dalších parametrů.

Metoda `SpawnObject()` nejdříve vygeneruje náhodné celé číslo od nuly do počtu nasazených vozidel v daném levelu. Díky tomu je pak vybráno z pole objektů aut přiřazených v jednotlivé úrovni. Poté je vytvořena instance objektu daného vozidla, která se nachází na místě, kde má dopravní pruh začátek.

Díky metodě `SetRandomTime()` nejsou vozidla generována lineárně ale v závislosti na náhodě. Do proměnné `spawnTime`, která určuje čas k dalšímu vytvoření instance vozidla, je náhodně přiřazena hodnota určená z rozmezí minimálního a maximálního času určeného ke generování.

```

void Start()
{
    red = GetComponent<SpriteRenderer>().sprite;
}
void OnMouseDown()
{
    if (Close)
    {
        GetComponent<SpriteRenderer>().sprite = green;
        Open = true;
        Close = false;
    }
    else if (Open)
    {
        GetComponent<SpriteRenderer>().sprite = red;
        Close = true;
        Open = false;
    }
}

```

Úryvek tohoto kódu nalezneme ve skriptu s názvem *Controller.cs*, jeho hlavní funkcí je měnit signalizaci semaforu a tím určovat volnost průjezdu jednotlivých vozidel.

Při metodě `Start()` je automaticky nastaven *SpriteRenderer* (renderer sloužící pro vykreslení 2D či 3D obrázku) na obrázek semaforu s červeným světlem, který signalizuje, aby vozidla na křižovatce zastavila.

K změně signálu slouží metoda `OnMouseDown()`, která rozpozná, zda bylo na objekt kliknuto myší nebo prstem. Uvnitř metody se poté nachází podmínky, které daný semafor změní na signál určující možnost průjezdnost pruhu.

3.3.2 Kvíz

Cílem této minihry je pomocí deseti otázek představit uživateli obor Dopravní prostředky: Kolejová vozidla a Dopravní prostředky: Silniční vozidla. Otázky jsou hráči náhodně generovány, tak aby při druhotném spuštění byly zobrazeny úplně jiné a zároveň se ve kvízu neopakovaly.

Herní mechanika

Hráči je při spuštění aplikace zobrazena otázka se čtyřmi možnostmi odpovědí. Při kliknutí na odpověď se projeví správnost odpovědi změním barvy stisknutého tlačítka. Pokud je odpověď správná, tlačítko zezelená a hráči se přidá bod. V druhém případě stisknuté tlačítko zčervená a tlačítko se správnou odpovědí zezelená. Hráči se v tomto případě bod nepřidá. Poté se zobrazí otázka další.

Minihra nabízí dva typy otázek. Prvním typem je klasická textová otázka, druhým je otázka, při které se s textem zobrazí taky obrázek. Po zodpovězení 10 otázek se uživateli zobrazí výsledný počet bodů.

Hra nabízí tlačítko „KONEC“, které uživatele přesune zpět do hlavního menu aplikace.

Skripty

Minihra obsahuje dva skripty. Jedním z nich je *QuizGameControll.cs*, který se stará o chod celé minihry. Uchovává v sobě počet správných a špatných odpovědí a vybírá následující otázky.

Hra obsahuje pouze jedinou scénu, ve které jsou již všechny otázky vytvořeny ale jsou skryty. Při prvním spuštění hry je skript aktivován. Ten v závislosti na celkovému počtu otázek vygeneruje deset náhodných otázek. Ty jsou poté zobrazovány, podle již daného vygenerovaného pořadí uloženého v proměnné `numbers`. Tato metoda je v následujícím kódu předvedena.

```
void Start()
{
    Random random = new Random();
    HashSet<int> numbers = new HashSet<int>();
    while (numbers.Count < questions.Length)
```

```

    {
        numbers.Add(Random.Range(0, questions.Length));
    }

    foreach (var item in numbers)
    {
        numbers2.Add(item);
    }

    questions[numbers2.IndexOf(index)].gameObject.SetActive(true);
}

```

Druhý skript, který se v kvízu nachází je *QuizAnswer.cs*. Ten je přiřazen každému tlačítku možné odpovědi. Má za úkol zjistit, zda uživatel odpověděl správně či špatně. Zjištění probíhá tak, že jednotlivá tlačítka mají od vývojáře nastavený *tag* (značku), která určuje, zda je odpověď obsažená v tlačítku správná či špatná. Tento zjišťovací systém je pod tímto odstavcem zjednodušeně demonstrován.

```

void OnMouseDown()
{
    if (gameObject.tag == "QuizRight")
    {
        //správná odpověď
        GameObject.Find("Main
Camera").GetComponent<QuizGameControll>().right++;
        gameObject.GetComponent<SpriteRenderer>().color = new
Color(0.545f, 0.792f, 0.399f, 1);
    }

    else
    {
        //špatná odpověď
        GameObject.Find("Main
Camera").GetComponent<QuizGameControll>().bad++;
        gameObject.GetComponent<SpriteRenderer>().color = new Color(0.905f,
0.337f, 0.337f, 1);
    }
}

```

Další ukázkou kódu, která se v tomto skriptu nachází je vyobrazený proces změny otázky uvedený níže. Po stisknutí tlačítka odpovědi program vyčká dvě vteřiny. Poté zavolá metodu z hlavního skriptu *QuizGameControll.cs* s názvem `ChangeQuestion()`, která danou otázku změní.

```

yield return new WaitForSeconds(2f);
GameObject.Find("Main
Camera").GetComponent<QuizGameControll>().ChangeQuestion();

```

3.3.3 Simulace ovládání výhybek kolejiště

Tato minihra má za cíl ukázat uživateli zjednodušenou verzi ovládání rozvětveného kolejiště. Hráč pomocí přehazování výhybek staví vlakovou cestu různým vlakům, které mají za úkol dorazit do správné stanice. Hra obsahuje celkem čtyři různé kolejiště, ty jsou uspořádány podle obtížnosti. Pokud je hra spuštěna z hlavního menu, objeví se nejdříve malý návod, který uživateli předvede, jak se hra ovládá.

Herní mechanika

Hra je složena ze tří hlavních komponent, které tvoří jádro hry. Prvním komponentem je kolejiště. Po něm se pohybuje vlaková souprava. Druhým jsou výhybky, které umožní změnit směr jízdy vlaku. Ty jsou možné ovládat uživatelem, pomocí kliknutí nebo dotyku na obrazovku umístěné výhybky. A třetím je samotná vlaková souprava. Ta je náhodně generována ze čtyř různých variant popsaných v podkapitole níže.

Minihra má v levém horním rohu tlačítko „MENU“, které hru pozastaví a zobrazí tlačítka s možnostmi změnit kolejiště, ukončit hru a v rozehrané hře pokračovat. V pravém horním rohu nalezneme počet správně dopravených vlaků.

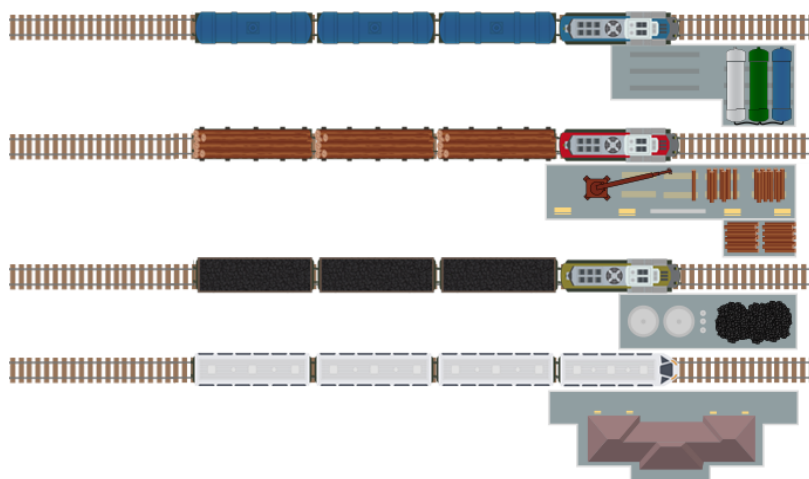
Vlakové soupravy

Součástí hry jsou čtyři různé varianty vlakových souprav. Ty jsou na vstupní koleje umístěny náhodně. Program poté ověřuje, zda byly pomocí výhybek dopraveny do správných stanic.

Varianty vlaků jsou:

- Nákladní cisternový vlak
- Nákladní vlak převážející dřevo
- Nákladní vlak převážející uhlí
- Osobní vlak

Vlaky a stanice jsou od sebe rozlišeny texturou, tak aby uživatel jednoznačně určil, kam má vlak pomocí výhybek dopravit.



Obrázek 13: Varianty vlaků a stanic

Konec hry

Minihra nemá žádný časový limit. Ukončit se dá jen pomocí chyby ovládní výhybek samotným hráčem. Zpočátku jsou vlaky vysílány malou rychlostí, která se v průběhu hry zvětšuje až do té doby, než uživatel udělá chybu. Čím větší rychlost vlaky mají, tím musí uživatel rychleji reagovat na změnu postavené vlakové cesty.

Prohrát je tedy možné jen dvěma variantami. První variantou je vykolejení vlaku. To proběhne v okamžiku, kdy hráč nestihne výhybku přehodit a vlak tak vykolejí. Druhou variantou je dopravení vlaku do nesprávné stanice. Pokud nastane jedna z těchto variant zobrazí se uživateli tabulka s výsledným počtem úspěšně dopravených vlaků. Pod tabulkou se nachází tlačítka, která dovolují úroveň opakovat, změnit mapu kolejiště či hru ukončit.

Princip jízdy vlaku

Herní objekt, který ve finální podobě představuje vlak je potřeba rozpohybovat a udat mu správný směr. Tento pohyb je určován body rozmístěnými po mapě, které jsou objektem (vlakem) následovány. Tyto body jsou uchovány v poli objektů. Uživatel pak měněním pozic výhybek určuje, který bod bude vlak následovat.

Skripty

O chod hry se stará celkem šest skriptů. V textu níže budou popsány nejdůležitější úryvky kódu, které tvoří jádro hry.

```

IEnumerator SpawnObject()
{
    numberOfTrains++;
    trains = new GameObject("trains");

    startWaypoint = Random.Range(0, homes.Length);
    endWaypoint = Random.Range(0, ends.Length - 1);

    Instantiate(train, trains.transform);
}

```

Kód výše nalezneme v bezesporu nedůležitější skriptu minihry nazvaném *TrainGameControl.cs*. Ten se stará o chod celé minihry. Tato část kódu má za úkol vygenerovat a umístit vlakovou soupravu na kolejiště. Pomocí náhody je vybráno, na jaké koleji vlak začne a do které stanice je ho potřeba nechat dovézt.

V tomto skriptu nalezneme metodu, která určuje rychlost vlaků. Proměnná `numberOfTrains` určuje, kolik vlaků již bylo vytvořeno. V závislosti na této proměnné pak roste i rychlost pohybu vlaků.

```

private void OnMouseDown()
{
    if (!Changed)
    {
        transform.GetChild(0).gameObject.SetActive(false);
        transform.GetChild(1).gameObject.SetActive(true);
        Changed = true;
    }
    else
    {
        transform.GetChild(0).gameObject.SetActive(true);
        transform.GetChild(1).gameObject.SetActive(false);
        Changed = false;
    }
}

```

Úryvek kódu výše se nachází ve skriptu *ChangeSwitch.cs*. Ten je obsažen v herním objektu, jehož potomkem jsou dva další objekty. Ty představují jednotlivou pozici výhybek. Pomocí kódu je pak jeden z nich aktivován a druhý deaktivován. O tuto změnu se stará bezparametrová metoda `OnMouseDown()`, která zjišťuje, zda bylo na objekt kliknuto.

```

void Move()
{
    if (transform.position == waypoints[0].transform.position)
    {

```

```

        if (waypoints[0].GetComponent<ChangeSwitch>().Changed == false)
        {
            nextWaypoint = 4;
            beforeWaypoint = 0;
        }
        else
        {
            nextWaypoint = 5;
            beforeWaypoint = 1;
        }
    }
}

```

Poslední ukázkou kódu z této minihry je zjednodušený případ pohybu vlaku mezi jednotlivými body (výhybkami). Tento kód je obsažen ve skriptu *TrainControll.cs*, který je komponentem každého vytvořeného vlaku.

Metoda `Move()` se stará o pohyb vlaku po kolejišti. V jejím těle nalezneme dvě vnořené podmínky. První podmínka zjišťuje, zda se vlak nachází na pozici umístěného bodu. Pokud ano, zjišťuje se podmínka druhá. Ta ověřuje, zda byla následující výhybka přehozena či nikoliv. Podle výsledku této podmínky je určen další bod, kam bude vlak nasměrován. Tento obdobný princip se opakuje u všech výhybek, které jsou ve scéně obsažené.

3.3.4 Sokoban klon

Tato minihra je inspirována velmi populární hrou Sokoban. Přebírá její herní mechaniky a je předělána, tak aby mohla být ovládána i na mobilním zařízení. Jedná se o logickou hru z roku 1980 vymyšlenou japonským vývojářem Hiroyukim Imabayshim. Zveřejněna pak byla o dva roky později firmou Thinking Rabbit. [17]

Hráč má za úkol pomocí svého panáčka posouvat bedny v bludišti na vyznačená místa. Při splnění této podmínky je mu umožněno vstoupit do další úrovně. Hra má dvě základní pravidla:

- Bednu je po mapě možné posouvat pouze tlačáním před sebou
- Není možné posouvat více beden naráz

Minihra obsahuje pět různých map, které jsou seřazeny podle obtížnosti.

Herní mechanika

Při spuštění hry se uživateli zobrazí první jednoduchý level, na kterém si může vyzkoušet ovládání hry. Herní objekt, který ve hře slouží jako hráč je ovládán šipkami umístěnými

v pravém spodním rohu aplikace. Ty se aktivují pomocí kliknutí myši nebo dotyku prstu na obrazovku. Pohyb je realizován na čtvercové mřížce, po které jsou posouvány stejným způsobem i bedny. Po stisknutí šipky se tedy hráč posune vždy o stejnou část do dalšího čtverce v mřížce.

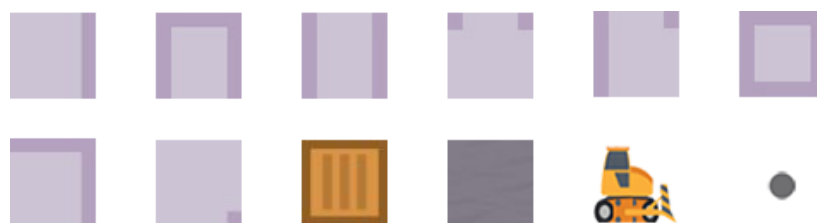
Minihra kromě šipek obsahuje tři základní tlačítka, z nichž viditelná jsou na začátku pouze dvě. Prvním je tlačítko „MENU“, které uživateli zobrazí možnosti pokračovat ve hře, přesunout se do jiného levelu nebo ukončit hru a vrátit se do hlavního menu. Druhým tlačítkem, které nalezneme hned vedle, je tlačítko „RESTART“. To po stisknutí úroveň zrestartuje. Tato situace může nastat v případě, kdy uživatel přesune bednu do takové pozice, ze které již není možné minihru dokončit. Třetí tlačítko se zobrazí až poté co jsou všechny bedny umístěny na správném místě. Objeví se v pravém horním rohu s nápisem „POKRAČOVAT“. Po stisknutí je zobrazena další úroveň.

Pokud uživatel přesune bednu na správné místo, její textura změní barvu, tak aby dala hráči opticky najevo, že je umístěna správně.

Assety

Hra je složena z celkem 12 assetů, které tvoří celou hru. V této hře jsou použity čtvercové textury a pomocí těch je mapa seskládána. Pro vizualizace bylo potřeba vytvořit všechny možné varianty zdí (rohy, strany a jejich kombinace).

Tvořena je šesti scénami, z nichž pět představuje jednotlivé úrovně. Zbývající scéna slouží k výběru úrovní.



Obrázek 14: Herní assety použité ve hře Sokoban klon

Skripty

Minihra obsahuje tři skripty. Jádro celé hry tvoří skript *SokobanGameController.cs*. Pod tímto textem je ukázán úryvek kódu operující ve hře.

```
bool IsLevelComplete() {
    Box[] boxes = FindObjectsOfType<Box>();
    foreach (var box in boxes)
    {
        if (!box.onPosition) return false;
    }
    return true;
}
```

Tato metoda slouží k zjištění, zda jsou všechny bedny umístěny na správném místě. Vrací hodnotu `true` (pravda) nebo `false` (nepravda). Nejdříve jsou načteny všechny herní objekty, které mají typ `Box` (bedna) do pole typu `Box`. Cyklus `foreach` má pak pomocí procházení tohoto pole za úkol zjistit, zda jsou všechny bedny umístěny na správném místě. Pokud ano je vrácena hodnota `true`. V opačném případě je vrácena hodnota `false`. Navracená hodnota této metody je poté ve stejném skriptu ověřována. Při úspěšném ověření vrací hodnota `true` je úroveň splněna a zobrazí se tlačítko „POKRAČOVAT“.

```
public bool Move(Vector2 direction)
{
    if (Mathf.Abs(direction.x) < 0.5)
    {
        direction.x = 0;
    }
    else
    {
        direction.y = 0;
    }
    if (Blocked(transform.position, direction))
    {
        return false;
    }
    else
    {
        return true;
    }
}
```

Část kódu uvedená výše se nachází ve skriptu *Player.cs*. Jejím úkolem je řídit pohyb hráče po mapě ve čtvercové síti (první podmínka). Zobrazená metoda `Move` vrací hodnotu `true` nebo `false` a zároveň zaručuje pohyb hráče po čtvercové mřížce (druhá podmínka).

```
void TestOnPosition()
{
    GameObject[] crosses = GameObject.FindGameObjectsWithTag("Cross");
    foreach (var cross in crosses)
    {
        if (transform.position.x == cross.transform.position.x &&
transform.position.y == cross.transform.position.y)
        {
            GetComponent<SpriteRenderer>().color = new
Color(0.4f,0.7f,0.4f);
            onPosition = true;
            return;
        }
    }
    GetComponent<SpriteRenderer>().color= Color.white;
    onPosition = false;
}
```

Poslední ukázkou kódu z této minihry pochází ze skriptu *Box.cs*. Tento skript mají implementovaný všechny herní objekty sloužící ve hře jako bedny. Metoda testuje, zda se bedna nachází na správném místě. Nejdříve jsou načteny do pole všechny herní objekty, které určují polohu správného umístění bedny. Pokud se na jednom z nich bedna nachází je nastaven parametr `onPosition` na hodnotu `true` a je změněna její barva. V opačném případě se hodnota `onPosition` nastaví na `false` a barva se nemění.

3.4 Ovládání a instalace

Aplikace je vytvořena pro platformy Windows a Android. Proto je potřeba, aby ji bylo možné na těchto dvou odlišných zařízeních ovládat. Program Unity se na obě tyto platformy specializuje, a tak není potřeba měnit funkčnost či ovladatelnost pro každou z nich.

Ovládání je možné pomocí počítačové myši nebo dotykové obrazovky. Přejít mezi různými minihrami je řešen pomocí tlačítek. Řízení samotných her probíhá většinou stiskem vyznačených herních objektů. V aplikaci je kladen důraz na jednoduchost a intuitivnost ovládání bez zdržujících návodů.

Nainstalování aplikace je možné pouze na zařízení s operačním systémem Windows a mobilní zařízení se systémem Android. V následujících podkapitolách je popsán návod instalace.

3.4.1 Windows

Unity při budování aplikace vyexportuje data projektu a spustitelný soubor ve formátu *exe*. Ten je možné ihned spustit bez nutnosti instalace. K spuštění tedy stačí přenést zmiňované soubory z příloženého CD do počítače.

3.4.2 Android

Při exportu aplikace pro android Unity vybuduje soubor s příponou *apk*. Tu je potřeba přenést do mobilního zařízení. Zde je nutné nejdříve povolit instalaci z neznámých zdrojů, která se nachází v nastavení zařízení. Poté aplikaci standardně nainstalovat. Po úspěšné instalaci se hra zobrazí na jedné z ploch mobilního zařízení. Kliknutím na ikonu aplikaci se hra spustí.

ZÁVĚR

Hlavním cílem bakalářské práce bylo vytvořit mobilní aplikaci sloužící jako propagační materiál Dopravní fakulty Jana Pernera. Vzhledem k dostupným typům zařízení, kterými fakulta disponuje, byl vývoj aplikace cílen na operační systém Android a Windows. K vytvoření programu byl použit herní engine Unity, který byl na základě průzkumu trhu nabízených herních enginů vybrán.

Při spuštění aplikace se uživateli zobrazí hlavní menu, které umožňuje výběr z jednotlivých miniher. Součástí aplikace jsou čtyři minihry, jež představují jednotlivé studijní obory fakulty.

První minihra představuje obor Technologie a řízení dopravy a její náplní je řízení silničního provozu na různých typech křižovatek. Má za úkol přiblížit jednu z možností uplatnění budoucího absolventa tohoto oboru.

Ve druhé hře má uživatel za úkol odpovědět na deset otázek týkajících se oborů Dopravní prostředky: Kolejová vozidla a Dopravní prostředky: Silniční vozidla. Prostřednictvím tohoto kvízu získá uchazeč o studium představu o náplni studia.

Třetí hra simuluje úlohu výpravčího, který musí zajistit správnou cestu vlaku pomocí manipulace s výhybkami. Aby byl hráč úspěšný, musí vzít v potaz i náklad vlaku, díky němuž určí, do které stanice vlak patří. Minihra reprezentuje obory Elektrotechnické a elektronické systémy v dopravě, Technologie a řízení dopravy a Dopravní prostředky: Kolejová vozidla.

Poslední částí je klon celosvětově známe hry Sokoban, která má za úkol prověřit hráčovo logické myšlení. Pointa spočívá v přesouvání krabic s materiálem na předem určená místa v bludišti. Tato hra přibližuje uchazeči obor Dopravního stavitelství.

Teoretická část práce se zabývá problematikou vývoje tvorby aplikací na mobilní zařízení. První část je věnována mobilním operačním systémům a podrobnému popisu vybraných herních enginů, které se v dnešní době nejvíce používají. V druhé části se práce zaměřuje na detailní popis použitého herního enginu Unity. Jsou zde zmíněny například informace o instalaci a ovládání programu, vývojové prostředí a pohledy nacházející se v tomto editoru. Další stránky jsou věnovány představení aplikace, která byla vytvořena v praktické části práce. Je zde popsána grafická stránka hry, její ovládání a stručný přehled dílčích miniher. Následující úsek se dopodrobna zabývá jednotlivými minihrami z pohledu vývojáře. Jsou zde uvedeny informace o použitých assetech, popis herních mechanik a úryvky nejdůležitějších skriptů. V poslední části práce se nachází popis ovládání aplikace a návod na instalaci ke dvěma využitým platformám.

Cíle stanovené v zadání bakalářské práce byly splněny včetně samotné vytvořené aplikace.

POUŽITÁ LITERATURA

- [1] *Deník.cz* [online]. VLTAVA LABE MEDIA, a.s, 2017 Dostupné z: <https://www.denik.cz/ekonomika/lide-travi-na-mobilu-tri-hodiny-denne-mohou-zpusobovat-neurotickeho-chovani-20171018.html>
- [2] *SMARTmania* [online]. SMARTmania.cz, 2019 Dostupné z: <https://smartmania.cz/uhadnete-jak-si-vedou-hloupe-telefony-vodafone-se-pochlubil-zajimavymi-cisly/>
- [3] *Statista* [online]. Statista, 2019 Dostupné z: <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>
- [4] *StatCounter* [online]. StatCounter, 2019 Dostupné z: <http://gs.statcounter.com/os-market-share/mobile/worldwide>
- [5] Historie mobilního operačního systému Android. O technice [online]. otechnice.cz, 2017 [cit. 2019-05-08]. Dostupné z: <https://otechnice.cz/historie-mobilniho-operacniho-systemu-android/>
- [6] iOS (Apple) – Wikipedie. [online]. Dostupné z: [https://cs.wikipedia.org/wiki/IOS_\(Apple\)](https://cs.wikipedia.org/wiki/IOS_(Apple))
- [7] *Tři nejlepší herní enginy, se kterými nikdy neuděláte chybu* [online]. tvorbaher.cz, 2017 [cit. 2019-05-08]. Dostupné z: <http://tvorbaher.cz/tri-nejlepsi-herni-enginy-se-kterymi-nikdy-neudelate-chybu/>
- [8] *Mobile Games* [online]. unity.com, 2019 [cit. 2019-05-08]. Dostupné z: <https://unity.com/solutions/mobile>
- [9] *iOS Developers Ship 29 % Fewer Apps In 2017, The First Ever Decline – And More Trends To Watch* [online]. appfigure.com, 2019 [cit. 2019-05-08]. Dostupné z: <https://blog.appfigures.com/ios-developers-ship-less-apps-for-first-time/>
- [10] Unity (herní engine). In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-05-08]. Dostupné z: [https://cs.wikipedia.org/wiki/Unity_\(hern%C3%AD_engine\)](https://cs.wikipedia.org/wiki/Unity_(hern%C3%AD_engine))

- [11] Unreal Engine. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-05-08]. Dostupné z: https://cs.wikipedia.org/wiki/Unreal_Engine
- [12] Unity Store. *Unity* [online]. [cit. 2019-05-21]. Dostupné z: <https://store.unity.com/>
- [13] *Unity Game Development Essentials*. Will Goldstone: Packt Publishing, 2009. ISBN 184719818X.
- [14] GEIG, Mike. Sams teach yourself unity 2018 game development in 24 hours. 3rd edition. Indianapolis, IN: Sams Pub., 2018. ISBN 978-0134998138.
- [15] C Sharp. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-05-09]. Dostupné z: https://cs.wikipedia.org/wiki/C_Sharp
- [16] JavaScript. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-05-09]. Dostupné z: <https://cs.wikipedia.org/wiki/JavaScript>
- [17] Sokoban. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-05-17]. Dostupné z: <https://cs.wikipedia.org/wiki/Sokoban>