



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

MOBILNÍ APLIKACE

"PEJSEK ZÁCHRANÁŘ" – CÉVNÍ MOZKOVÁ PŘÍHODA

MOBILE APPLICATION "PARAMEDIC DOGGY" – BRAIN STROKE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MONIKA ROSINSKÁ

VEDOUCÍ PRÁCE

SUPERVISOR

Prof. Dr. Ing. PAVEL ZEMČÍK,

BRNO 2021

Zadání bakalářské práce



Studentka: **Rosinská Monika**
Program: Informační technologie
Název: **Mobilní aplikace "Pejsek záchranář" - cévní mozková příhoda**
Mobile Application "Paramedic doggy" - Brain Stroke
Kategorie: Uživatelská rozhraní

Zadání:

1. Seznamte se s prostředím pro vývoj multimediálních mobilních aplikací (Unity) a seznamte se také s činností záchranářů při cévní mozkové příhodě.
2. Navrhněte rozhraní a strukturu vzdělávací (mobilní) aplikace pro děti na výše uvedené téma.
3. Aplikaci implementujte a diskutujte vlastnosti.
4. Proveďte uživatelskou studii pro implementované výsledky.
5. Vytvořte video reprezentující výsledky vaší práce a zhodnoťte možnosti pokračování.

Literatura:

- Dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 2 zadání

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Zemčík Pavel, prof. Dr. Ing.**
Konzultant: Chlubna Tomáš, Ing., UPGM FIT VUT
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2020
Datum odevzdání: 12. května 2021
Datum schválení: 30. října 2020

Abstrakt

Cílem této práce bylo vytvořit vzdělávací hru pro děti ve věku 7–10 let na téma první pomoc při cévní mozkové příhodě. Aplikace byla vyvíjena pro operační systém Android ve vývojovém prostředí Unity. Při vývoji byla zohledněna psychologie, emoce a vzdělávání dětí a byla aplikována gamifikace. Děti, které aplikaci vyzkoušely, byly poté schopny popsat příznaky mrtvice a věděly, co mají v takové situaci dělat. Přínosem této práce je, že děti umějí podat první pomoc a kontaktovat zdravotnickou záchrannou službu, pokud je to potřeba. Umějí také rozpoznat příznaky cévní mozkové příhody neboli mrtvice.

Abstract

The aim of the thesis was to create educational application for kids in age 7–10 years on the topic first aid for brain stroke. Application was developed for operating system Android by using Unity game engine. The game was developed while taking into account psychology, emotions and educating children, and it's using gamification. Children who played the game are able to recognize the symptoms of brain stroke and know what to do in similar situations. Benefits of this application are that children can give first aid and call ambulance if needed. They can also recognize the symptoms of brain stroke.

Klíčová slova

vzdělávací aplikace, hra pro děti, vzdělávací hra pro děti, Pejsek Záchranář, Defík, mobilní aplikace, první pomoc, gamifikace, vývoj hry

Keywords

educational application, game for children, educational game for children, Paramedic Doggy, Defík, mobile application, first aid, gamification, game development

Citace

ROSINSKÁ, Monika. *Mobilní aplikace "Pejsek záchranář" – cévní mozková příhoda*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Prof. Dr. Ing. Pavel Zemčík,

Mobilní aplikace "Pejsek záchranář" – cévní mozková příhoda

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením pana prof. Dr. Ing. Pavla Zemčíka. Další informace mi poskytla Dagmar Šimáková. Uvedla jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpala.

.....
Monika Rosinská
8. května 2021

Poděkování

Tímto bych chtěla poděkovat panu prof. Dr. Ing. Pavlu Zemčíkovi za vedení a pomoc při tvorbě této bakalářské práce. Dále děkuji paní Dagmar Šimákové ze ZZSJMK za její vstřícnost, pomoc a poskytnutí informací a panu Daliborovi Krchovi za návrh grafického provedení pejska záchranáře. Děkuji také všem, kteří mi pomohli s testováním aplikace, poskytli cenné rady a byli mi velkou oporou.

Obsah

1	Úvod	2
2	Vývoj aplikací pro mobilní telefony	3
2.1	Nativní VS hybridní vývoj aplikací	3
2.2	Mobilní operační systémy	4
2.3	Nástroje pro vývoj her	5
2.4	Vývojové prostředí Unity	6
2.5	Distribuce aplikace pomocí Google Play Store	7
2.6	Ukládání dat v souborech	9
3	Vytváření vzdělávacích her pro děti	12
3.1	Psychologie, emoce a učení u dětí ve věku 7–10 let	12
3.2	První pomoc při cévní mozkové příhodě	13
3.3	Základní principy gamifikace	15
3.4	Existující aplikace o první pomoci	17
4	Návrh struktury a vzhledu aplikace	22
4.1	Analýza současného stavu	22
4.2	Návrh hry pro děti	23
4.3	Využití gamifikace	24
4.4	Návrh vzhledu aplikace	24
4.5	Návrh jednotlivých scén aplikace	26
4.6	Potřebná data a jejich uložení	27
5	Implementace aplikace Pejsek Záchranář	28
5.1	Rozdělení aplikace na scény	28
5.2	Uživatelské rozhraní a vzhled aplikace	29
5.3	Zadávání otázek a práce s nimi	33
5.4	Třídy a s nimi spojené skripty	35
5.5	Přepínání animací objektů	41
5.6	Testování a vydání výsledné aplikace	42
6	Závěr	44
	Literatura	45

Kapitola 1

Úvod

Dnešní děti vyrůstají obklopeny technikou. Sedmileté děti ovládají tablety, telefony a počítače lépe než mnozí dospělí. A není divu – není výjimkou, že roční dítě v kočárku drží telefon, pětileté dítě usíná u televizních pohádek, a při cestování v autě rodiče místo hraní různých slovních her dají dítěti tablet, ať se zabaví hrami. Rodiče si často podobným způsobem zjednodušují hlídání svých vlastních dětí, a proto je pro dnešní děti elektronika tak samozřejmá, že je čím dál obtížnější najít cokoli jiného, co by je nadchlo. Dnešní děti žijí v jiném světě, než jsme žili my, a s tím je třeba počítat a zachovat se podle toho – a to i co se týče vzdělávání. Proč tedy děti učit první pomoc pro ně nudnými přednáškami, když lze vytvořit zábavnou hru, která je první pomoc naučí? Hra o první pomoci by pro děti mohla být zábavná, a přitom přínosná – pokud by se někdy něco stalo jejich rodičům či prarodičům, děti by byly schopny jim pomoci. A právě to je cílem této bakalářské práce.

A proč vlastně učit děti zachraňovat životy? Není vždy okolo nějaký dospělý, který může poskytnout rychlejší a lepší pomoc? Ačkoli na to málokterý rodič pomyslí, děti jsou v takové situaci poměrně často. V létě tráví spoustu času u prarodičů, neboť jsou rodiče v práci – a kdyby se jejich babičce či dědovi něco stalo, pomoc možná neseženou. Ve škole, obzvláště na prvním stupni, často podnikají výlety či procházky do přírody – i zde se může paní učitelce či panu učiteli cokoli stát.

Místo aby byly děti v takových chvílích bezmocné a nevěděly co dělat, můžeme je naučit základy první pomoci – zavolání záchranné služby, správný popis toho co se stalo a kde se dítě nachází. A proč to neudělat pomocí něčeho, co je baví?

Motivací pro vytvoření této aplikace je tedy vzdělat děti v první pomoci, v tomto případě konkrétně první pomoci při cévní mozkové příhodě neboli mrtvičce. Cílem je poskytnout dětem zábavný způsob, jak se něco takového naučit – první pomoc patří do kategorie věcí, které děti příliš nezajímají, a proto by pojetí podávání první pomoci jako hry mohlo děti motivovat a něčemu naučit. Cévní mozková příhoda je druhá nejčastější příčina úmrtí v ČR a přitom 70 % lidí v Česku nepozná její příznaky, a proto považuji za důležité naučit děti (i dospělé) tyto příznaky rozpoznat a co nejdříve zavolat pomoc.

Práce je členěna do čtyř kapitol. První kapitola se zabývá informacemi potřebnými k vývoji aplikace pro mobilní telefon – operační systémy, nástroje pro vývoj a distribuci aplikací, ukládání dat. Druhá kapitola pojednává o vzdělávání dětí, popisuje rozpoznání příznaků cévní mozkové příhody, seznamuje čtenáře se základními principy gamifikace a s již existujícími aplikacemi podobného zaměření. Třetí kapitola obsahuje návrh aplikace spolu s grafikou, a poslední kapitola popisuje implementaci aplikace, její testování a uveřejnění na trhu.

Kapitola 2

Vývoj aplikací pro mobilní telefony

Tato kapitola seznamuje čtenáře s problematikou vývoje her pro mobilní telefony – zaměřuje se na volbu operačního systému, výběr nástroje pro vývoj her, distribuci aplikace a možnosti ukládání dat. Nejedná se o podrobný popis uvedených problematik, ale jen o seznámení se základními fakty potřebnými v této práci. Kapitola obsahuje jen informace potřebné pro tuto práci.

2.1 Nativní VS hybridní vývoj aplikací

Existují dva způsoby, jak vyvíjet aplikaci – buď může být aplikace vyvíjena přímo pro konkrétní operační systém (nativní vývoj), nebo může být vyvíjena tak, aby ji bylo možné spustit na více operačních systémech (hybridní vývoj). Zatímco při nativním vývoji se používají jazyky typické pro danou platformu, při hybridním vývoji se používají webové technologie jako HTML, CSS nebo JavaScript. Mimo tyto dva typy vývoje existuje také webový, který není třeba uvažovat, neboť aplikace má být pro mobilní telefony a její distribuce má probíhat pomocí Google Play Store (viz kapitola 2.5). Další možností je tzv. multiplatformní vývoj (cross-platform), který ale v dnešní době stále často používá webové služby, podobá se tedy hybridnímu vývoji. Porovnání těchto přístupů je uvedeno v následující tabulce 2.1. Informace pro tuto kapitolu byly čerpány ze článku o vývoji mobilních aplikací [6] a ze článku od Peko Studia¹

¹<https://peko-studio.cz/rozdil-mezi-nativni-a-hybridni-mobilni-aplikaci/>

Nativní vývoj	Hybridní vývoj
- kód je potřeba psát a udržovat pro každý operační systém zvlášť a je potřeba více znalostí o systémech	+ kód stačí napsat jen jednou → snazší údržba a úpravy
- aplikaci nelze spustit v prohlížeči jako webovou stránku	+ aplikaci lze vytvořit včetně webové stránky a lze ji spustit v prohlížeči
+ méně závislostí na knihovnách, díky čemuž se v aplikaci vyskytuje méně chyb	- má závislosti na knihovnách a frameworkcích
+ i náročné aplikace jsou výkonné a rychlé	- náročnější aplikace jsou pomalejší
+ nativní aplikace nepotřebují pluginy pro přístup k funkcím zařízení	- přístup k funkcím zařízení je závislé na pluginech

Tabulka 2.1: Porovnání nativního a hybridního vývoje aplikací (+ označuje výhodu, – označuje nevýhodu)

Nativní vývoj je jednodušší, pokud nemáme za cíl rozšířit aplikaci ve všech platformách (tj. operačních systémech). V takovém případě je potřeba zvolit operační systém, pro který bude aplikace vyvíjena. Hybridní vývoj je vhodný pokud chceme aplikace zpřístupnit všem lidem. Pokud je aplikace náročná, je lepší zvážit možnost více vývojových týmů, přičemž každý bude vyvíjet stejnou aplikaci, ale pro jiný operační systém.

2.2 Mobilní operační systémy

Při vývoji mobilní aplikace je potřeba zvolit operační systém (dále jako OS), na kterém aplikace poběží, či vyvíjet aplikaci pro více OS. Vývoj aplikace pro více OS je časově náročnější, a v případě operačního systému iOS je vývoj i finančně náročný. Zatímco aplikaci pro Android lze vyvíjet na téměř jakémkoli OS a na libovolném zařízení, aplikace pro iOS lze vyvíjet pouze na zařízení od firmy Apple, případně pomocí cloudových služeb. Pro představu – nejlevnější MacBook (notebook od firmy Apple) stojí okolo 26 000 Kč (k roku 2021, zjištěno z²). Vývoj na cloudových službách stojí v přepočtu 22 Kč za hodinu, v případě využití služby s jistými omezeními od 650 Kč měsíčně (k roku 2021 z oficiálních stránek macincloud³⁴).

Poměr mobilních operačních systémů v ČR

Poměr OS na mobilních telefonech v ČR je uveden v následující tabulce 2.2 a jsou vykreslena v grafu 2.1 (data pocházejí z průzkumu GlobalStats⁵):

²<https://www.alza.cz/levne-macbook/18854758.htm>

³<https://checkout.macincloud.com/>

⁴<https://checkout.macincloud.com/select/payg>

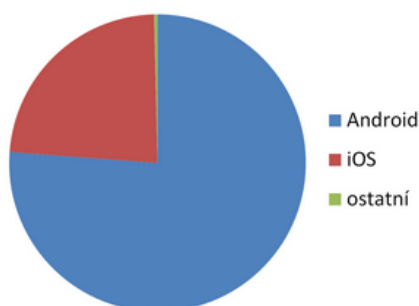
⁵<https://gs.statcounter.com/os-market-share/mobile/czech-republic>

Android	76.18 %
iOS	23.44 %
Samsung	0.24 %
Windows	0.04 %
Nokia	0.02 %
SymbianOS	0.02 %
Ostatní	0.06 %

Tabulka 2.2: Poměr mobilních operačních systémů v ČR

Tato data odpovídají stavu v prosinci roku 2020.

Lepší představu o poměru OS v České republice poskytne tento graf:



Obrázek 2.1: Graf podílu mobilních operačních systémů v ČR

Z uvedených informací je patrné, že většina obyvatel České republiky vlastní mobilní telefon s operačním systémem Android. Při vývoji aplikace určené na český trh je tedy téměř nutné, aby byla aplikace funkční právě na Androidu.

2.3 Nástroje pro vývoj her

Pro celou tuto podkapitolu jsou použity informace ze stránek GameDesigning [5] a Game Carrier Guide [11] kapitola 1.6. Nástroj pro vývoj her (tzv. herní engine) je software, který usnadňuje vývoj počítačových a mobilních her – zajišťuje renderování grafických prvků, načítání a přehrávání zvukových souborů a simulaci fyzikálních zákonů (jako detekce kolize, vzájemné působení těles, deformace). Některé nástroje nabízejí i další funkce. Mezi tyto nástroje patří Unity, Unreal Engine, Source, GameMaker Studio 2, CryEngine, a další.

Pro každou aplikaci či hru je potřeba zvolit vhodný herní engine – každý nástroj umí něco jiného, jsou jinak složité či technicky náročné, některé nástroje podporují jen určitý operační systém, a také se liší cenou.

Hry na mobilní telefon lze vyvíjet pomocí Unity a Unreal Engine. Pro vývoj Pejska Záchranáře byl zvolen nástroj Unity, proto je rozebrán podrobněji. Jejich porovnání je uvedeno v tabulce 2.3.

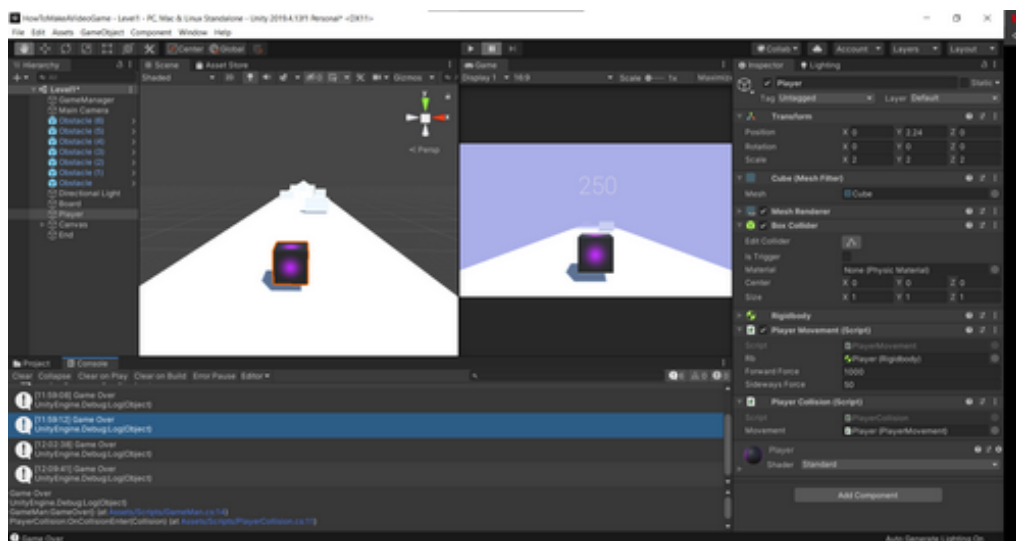
Unity	Unreal Engine
podpora více jak 25 platformem	podporuje méně platformem
používá jazyk C#	používá jazyk C++
rychlejší vývoj	lepší výkon
zdarma do tržeb 2.5 milionu Kč	zdarma do tržeb 21.5 milionu Kč
lepší pro vývoj 2D (ale zvládne i 3D)	modernější 3D funkce
umožňuje jednoduché testování na zařízení s OS Android	celkově méně vhodný na Android

Tabulka 2.3: Porovnání vývojových prostředí Unity a Unreal Engine

2.4 Vývojové prostředí Unity

Unity je herní engine, který umožňuje vývoj 2D i 3D her pro mobilní telefony i počítače, nabízí dokonce možnosti i pro specifické aplikace (např. aplikace pro virtuální realitu). Pro psaní kódu v Unity se používá programovací jazyk C#.

Práce v Unity editoru je jednoduchá a intuitivní, aplikaci lze spustit přímo v editoru. Upozornění, varování i chyby se také zobrazují přímo v editoru, a pokud je Unity provázáno s konkrétním vývojovým prostředím, které daný programátor používá pro psaní kódu, lze aplikaci i debugovat (debugging – způsob hledání chyby v programu pomocí postupného krokování programu, hlídání hodnot proměnných apod.). Vzhled vývojového prostředí je ilustrován obrázkem 2.2.



Obrázek 2.2: Vzhled vývojového prostředí Unity (snímek obrazovky počítače)

Výhody použití Unity

Informace pro tuto kapitolu byly čerpány ze stránky softwarové firmy Potenza Global Solutions⁶. Velkou výhodou je, že Unity je multiplatformní – díky tomu je možné vyvíjet

⁶<https://www.potenzaglobalsolutions.com/blogs/5-rarely-known-advantages-and-disadvantages-of-unity-game-development>

aplikaci pro více operačních systémů pomocí jediného nástroje. V případě mobilní aplikace zejména pro Android a iOS. Unity podporuje přes 25 platforem.

Další výhodou je, že vývoj aplikací pomocí Unity je zdarma, pokud jsou roční příjmy z aplikace nižší než 100 000\$ (v přepočtu cca 2,5 milionu Kč). Po přesáhnutí této částky je potřeba zaplatit odpovídající částku (nejméně 399\$, tj. 8 600 Kč za rok)⁷.

Také současně nabízí dobrý výkon aplikace a příjemnou grafiku a kvalitní vizuální efekty, a přitom je vývoj pomocí tohoto nástroje poměrně snadný.

Pro začátečníky je připravena jednoduchá, ale rozsáhlá dokumentace přímo na stránkách Unity⁸. Pokud by začínajícímu vývojáři nestačily tyto návody, díky rozšířenosti Unity lze na internetu nalézt nespočet videonávodů, které začátečníkovi pomohou vytvořit svou první hru. Vývojové prostředí navíc obsahuje vestavěný nástroj na debugging.

Nevýhody použití Unity

Mezi nevýhody Unity patří zejména skutečnost, že pro použití nejlepších grafických nástrojů je potřeba si koupit licenci.

Další nevýhodou je, že chybí některé složitější fyzikální funkce a je potřeba je případně přidat. Tato nevýhoda se ale projeví jen při vývoji náročné aplikace. Unity ale skýtá další dvě nevýhody, které vývojář nemusí zpočátku zaznamenat. Jednou z nich je skutečnost, že je velmi náročné nalézt a opravit problémy s výkonností – zdrojový kód je složitější právě kvůli kvalitním grafickým a fyzikálním funkcím, které Unity nabízí. Druhou skrytou nevýhodou je náročnost vývojového prostředí na paměť, kvůli čemuž mohou nastávat problémy s laděním.

Testování aplikace pomocí Unity Remote

Během vývoje je možné aplikaci spustit ve vývojovém prostředí Unity, vždy je ale vhodné vyzkoušet aplikaci na samotném telefonu. Nejjednodušším řešením je využití Unity Remote.

Unity Remote je aplikace, která je dostupná na Google Play Store (na OS Android). Vývojář si na telefon stáhne tuto aplikaci, přepne telefon do režimu vývojáře a propojí telefon s počítačem, na kterém je aplikace vyvíjena. Pomocí nástrojů ve vývojovém prostředí Unity lze poté aplikaci jednoduše nahrát na daný mobilní telefon, kde ji lze ihned spustit. Mobilní telefon současně odesílá podrobné informace o průběhu hry do počítače, díky čemuž lze například odhalit chyby nebo nalézt vhodné místo pro optimalizaci.

2.5 Distribuce aplikace pomocí Google Play Store

Google Play Store je služba pro distribuci aplikací na zařízeních s operačním systémem Android. Uživatelé mohou stahovat nabízené aplikace a hry, vývojáři pomocí něj mohou šířit své aplikace a jejich aktualizace, a jednoduše na ně získávají zpětnou vazbu. Aplikace se nahrávají ve formátu APK, které jsou limitovány na 50 – 100MB v závislosti na verzi Androidu, na kterém má být aplikace přístupná (k roku 2021⁹). [7]

Vývojář je při registraci povinen zaplatit jednorázový poplatek 25\$ (v přepočtu 540 Kč)¹⁰. Aby mohla být aplikace zveřejněna na Google Play Store, musí splňovat určité podmínky, blíže uvedené v distribuční smlouvě pro vývojáře Google Play¹¹.

⁷<https://store.unity.com/compare-plans>

⁸<https://learn.unity.com/>

⁹<https://support.google.com/googleplay/android-developer/answer/9859152?hl=en>

¹⁰<https://play.google.com/console/signup>

¹¹<https://play.google.com/about/developer-distribution-agreement.html>

V případě, že je aplikace placená, vývojář obdrží jen 70 % z celkové částky. Zbýlých 30 % činí servisní poplatek, který obdrží distribuční partner – jedná se o jednu z hlavních nevýhod využití této služby¹².

Výhody použití Google Play Store

Google Play Store upozorňuje uživatele na vydání aktualizace aplikace, vývojář díky tomu nemusí řešit, jak tuto informaci uživatelům sdělit. Google Play Store se současně může starat o propagaci dané aplikace – její umístění na hlavní stránku Play Store, ve výběru aplikací či v doporučených. Další výhodou využití této služby je, že lze nabízet i beta verze aplikace – uživatelé si mohou vyzkoušet plánované funkce a zároveň na ně poskytnout zpětnou vazbu. Vývojář tak může doladit nové funkce ještě před uvedením na trh.

Proč nedistribuovat aplikaci jinde?

Hlavním důvodem, proč nabízet svou aplikaci na Google Play Store je fakt, že tato služba má vestavěnou funkci Google Play Protect, která uživatele chrání před viry a podobnými hrozbami [8]. Je tedy pravděpodobnější, že uživatel stáhne aplikaci právě zde, než z neověřeného zdroje. Dalším důvodem je upozornění uživatele na aktualizace, které bylo zmíněno výše.

APK soubor

Tato kapitola čerpá ze zdroje [9]. Mobilní aplikace se často poskytují ve formátu APK, který používá i Google Play Store pro Android, a umožňuje snadnou instalaci aplikace do telefonu. APK je zkratka od Android Package Kit, a jedná se o soubory zabalené do balíčku (můžeme si ho představit jako složitější a nerozbalitý ZIP formát). APK soubor obsahuje zdrojové soubory, obrázky a zvukové soubory použité v projektu, zdroje, certifikáty.

APK soubor obsahuje 7 základních složek [18]:

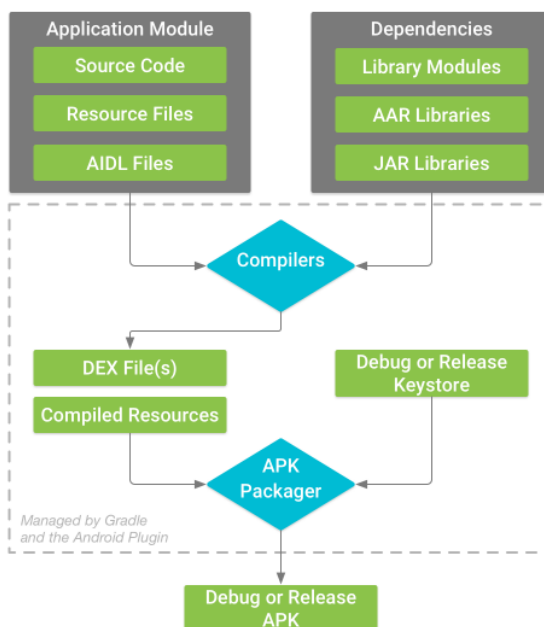
- META-INF – obsahuje Manifest soubor, certifikáty aplikace a seznam zdrojů (resources) odpovídající
- lib – adresář obsahující zkompileovaný kód pro různé typy procesorů
- res – adresář obsahující nezkompilované zdroje (resources)
- assets – adresář obsahující obrázky, zvukové soubory apod., které jsou načítány AssetManagerem
- AndroidManifest.xml – další manifest soubor, který obsahuje jméno, verzi, práva přístupu a knihovny aplikace
- classes.dex – třídy aplikace kompilované do dex souboru, který používá Dalvik virtual machine a Android Runtime (virtuální stroje, které vytváří prostředí pro běh aplikací na androidu¹³)
- resources.arsc – soubor obsahující kompilované zdroje (resources)

¹²<https://support.google.com/googleplay/android-developer/answer/112622?hl=cs>

¹³<https://source.android.com/devices/tech/dalvik>

Kompilace zdrojových souborů a dalších součástí aplikace do APK souboru probíhá následovně [10] a pro snazší pochopení je popsán i na obrázku 2.3:

1. kompilace veškerých souborů do DEX souborů (Dalvik Executable)
2. kompilace zdrojů a dalších souborů (formát arsc)
3. tzv. APK Packager zkombinuje DEX soubory a arsc soubory a vytvoří z nich jeden APK soubor
4. APK Packager podepíše vytvořený APK soubor klíčem, který lze získat registrací aplikace do Android Studia
5. pomocí nástroje zipalign je aplikace optimalizována a je vygenerován finální APK soubor



Obrázek 2.3: Průběh kompilace zdrojových souborů aplikace do APK souboru¹⁴

použitý obrázek je převzat ze stránky developer.android [10]

Unity umožňuje vytvoření souboru ve formátu APK přímo v editoru, a aplikaci lze jednoduše instalovat na připojený telefon. Telefon musí být připojen k počítači kabelem, který umožňuje i přenos dat, a uživatel musí povolit instalaci souborů přes USB debugging (funkce dostupná ve vývojářském módu telefonu).

2.6 Ukládání dat v souborech

Serializační formát je způsob uložení serializovaných dat. Serializace dat je proces převedení např. objektu v programu do jednoduché podoby – z pohledu počítače do sekvence bitů,

¹⁴

z pohledu člověka do prostého textu, kdy jsou jednotlivé položky seřazeny za sebou do nějakého seznamu.¹⁵ Slouží pro práci s daty, jejich posílání po síti, načítání dat do programů apod. Mezi nejznámější serializační formáty patří XML, JSON a YAML. Jejich velkou výhodou je, že jsou čitelné lidmi, a lidé tedy mohou vytvářet soubory v jejich formátu. Hlavní rozdíl mezi nimi jsou patrné na obrázku 2.4. [14]

XML formát

XML, neboli Extensible Markup Language, je serializační formát s hierarchickým zanořením elementů. Má jeden kořenový element, všechny elementy jsou párové, mají jméno a atributy, jejichž hodnoty jsou uvedeny v uvozovkách. Hodnota elementu je vložena mezi počáteční a koncovou značkou. Element má formát `<nazev elementu atribut="hodnota atributu">hodnota elementu</nazev elementu>`. Elementy lze do sebe libovolně zanořovat.¹⁶ XML formát je snadno zpracovatelný počítačem, a je poměrně čitelný i pro lidi. Při rozsáhlém souboru a složité struktuře souboru se ale text stává pro člověka téměř nečitelným. V programovacím jazyce C#, které je v projektu použito, existuje knihovna pro snadnou práci s XML formátem, `System.Xml`. [15]

JSON formát

JSON (JavaScript Object Notation) je serializační formát, kde se data ukládají ve formátu `název/klíč: hodnota`, přičemž hodnota může nabývat typů string, integer, bool, array, object a null. Objekty jsou vloženy mezi složené závorky, pole jsou vloženy mezi hranaté závorky. Jednotlivé položky jsou odděleny čárkou. Výhodou je, že je poměrně snadno čitelný pro člověka, a zároveň jej počítače umí jednoduše parsovat a mapovat na objekty, či je naopak vytvořit z dat v aplikaci.¹⁶ V Unity, které je pro tvorbu aplikace použito, lze použít knihovnu `Newtonsoft.Json`, která velmi usnadňuje práci s JSON soubory. Mimo jiné dovede z textu v JSON souboru vytvořit objekt libovolného typu, či naopak vytvořit z jakéhokoli objektu JSON formát, a to včetně dat použitých v aplikaci (tato konverze probíhá pomocí reflexe, kdy za běhu programu zjišťuje informace o typu a jménu proměnných v daném objektu). [3]

YAML formát

YAML, plným názvem YAML Ain't Markup Language, je serializační formát snadno čitelný lidmi, ale počítače mají s některými konverzemi problém. Čtení YAML souborů, či převádění objektu z programu do YAML formátu je jednoduché, ale mohou se vyskytnout problémy při snaze naplnit objekt daty z YAML souboru (především co se týče typu dané proměnné, resp. hodnoty).¹⁷ Pro použitý programovací jazyk C# sice existují knihovny pro práci s YAML soubory, ale neumožňují jednoduché načtení dat z YAML souboru a naplnění objektu těmito daty. Oproti tomu se tento formát jeví jako nejčitelnější pro lidi, protože pro znázornění hierarchie používá odsazení, a např. pro seznam lze použít odrážky, což je pro člověka přirozené. [1]

¹⁵<https://isocpp.org/wiki/faq/serialization>

¹⁶přednáška: Burget R., Informační systémy – Vizualizace a serializace dat, VUT FIT, 2020 str. 18–47

¹⁷<https://www.cloudbees.com/blog/yaml-tutorial-everything-you-need-get-started/>

XML

```
<Servers>
  <Server>
    <name>Server1</name>
    <owner>Fouzi</owner>
    <created>28092019</
created>
    <status>active</status>
  </Server>
</Servers>
```

JSON

```
{
  Servers: [
    {
      name : Server1,
      owner : Fouzi,
      created : 28092019,
      status : active,
    }
  ]
}
```

YAML

```
Servers :
- name : Server1
  owner : Fouzi
  created : 28092019
  status : active
```

Obrázek 2.4: Formát serializačních formátů XML, JSON a YAML, všechny reprezentují stejná data totožného objektu¹⁹

Jaký serializační formát zvolit závisí na účelu a použití daných dat, co přesně se s daty bude dělat (zda se budou načítat do programu či i přenášet), zda se soubory mají pracovat i lidé, a na dalších aspektech. Často také rozhodují i preference konkrétního člověka. Všechny serializační formáty mají své výhody i nevýhody a je potřeba zvolit serializační formát s vhodnými vlastnostmi.

¹⁹obrázek převzat z facebookové stránky Network Devices & Programmability <https://www.facebook.com/NDACisco/>

Kapitola 3

Vytváření vzdělávacích her pro děti

Tato kapitola obsahuje základní informace o psychologii dětí, jejich emocích a učení. Jelikož se jedná o vzdělávací aplikaci pro děti, bylo potřeba tyto informace nastudovat, aby aplikace splnila svůj cíl. Dále je v kapitole popsána gamifikace a první pomoc při cévní mozkové příhodě, na kterou je tato práce zaměřena. Uvedeny jsou i existující aplikace o první pomoci. Jsou zde probrána témata potřebná pro tuto práci, ale nejedná se o jejich encyklopedický přehled (z důvodu omezeného rozsahu práce).

3.1 Psychologie, emoce a učení u dětí ve věku 7–10 let

Tato kapitola využívá poznatků z nastudované literatury [12] str. 21–24 a [2] kapitoly 1 a 2. Děti ve věku, na které je hra zaměřena, tedy cca 7 – 10 let, se dostávají do fáze, kdy jsou pilní a snaží se v čemkoli, co jej potká. Děti touží pochopit okolní svět a rády se učí. Dokonce si rády vyhledávají informace samy, i když by je mohli poskytnout blízcí dospělí. V tomto věku se dětem zdokonaluje vnímání, a získávají schopnost věnovat pozornost i podrobnostem a detailům, ne jen celku. Nejlépe si pamatují to, co samy prožily či si samy vyzkoušely. Dle Petra Štípka, psychologa a psychoterapeuta s mnohaletou praxí, je jednou z potřeb dítěte je vědět že něco umí nebo dokáže udělat, a s tím související chvála a ocenění za dobře odvedenou práci. Děti potřebují občas povzbudit, či poradit takovým způsobem, aby dítě problém vyřešilo samo. Chvála je občas potřeba i za samozřejmé a jednoduché věci. To by mělo platit i ve vzdělávání, a tedy i v této vzdělávací aplikaci. Například když dítě dostane odmítavé gesto od pejska za chybnou odpověď, mělo by získat i pochvalu za správnou odpověď. [17]

Děti často potřebují motivaci, aby něco dělaly. Základní dělení motivace dle Linhartova uvažuje 4 typy [13] (str. 542–543):

1. pozitivní motivace – dítě má tendenci dosáhnout žádoucího cíle, které si samo vytyčilo, či jej potřebuje
2. negativní motivace – dítě se chce vzdálit od něčeho nepříjemného, např. od trestu; proto dítě vykoná nějakou činnost, kterou dělat nechce, aby nebylo potrestáno
3. zájem o danou činnost – dítě má o danou věc zájem, činnost ho baví a chce se v ní zlepšovat
4. sociální motivy – dítě cítí povinnost danou věc udělat, cítí odpovědnost, či touží po pochvale od rodiče či učitele

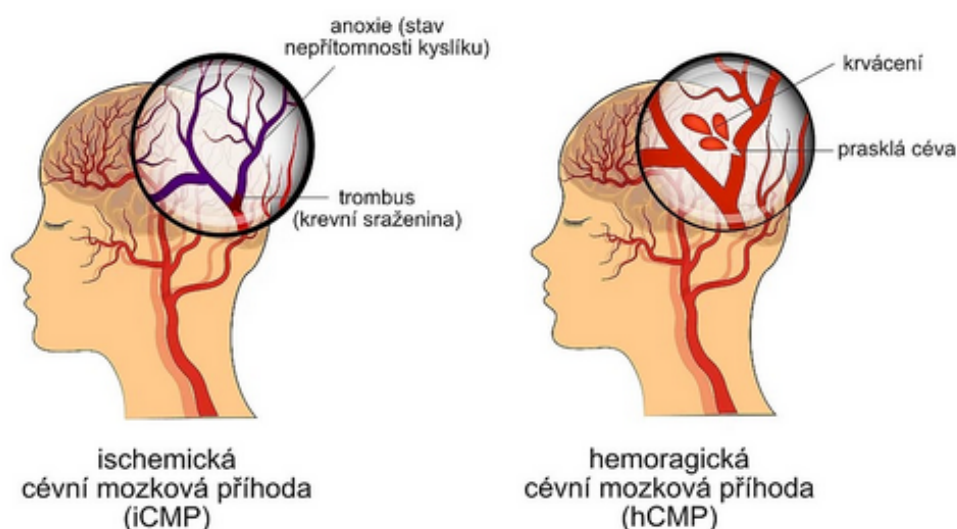
Aby dítě nějakou věc provádělo rádo a co nejvíce se přitom naučilo, měl by mu být jasný cíl dané činnosti, a mělo by tento cíl přijmout za svůj. Cíl by měl být adekvátní k schopnostem dítěte – neměl by být příliš těžký, ani příliš lehký. Jelikož se děti nejlépe učí z toho co samy vyzkoušely, měly by být schopny cíle dosáhnout samy, bez cizí pomoci. Zároveň ale potřebuje včas získat informaci o tom, zda jeho odpověď byla správná či špatná. Pokud jsou tyto požadavky splněny, dítě získá velké množství vědomostí a dovedností, může si vytvořit i nové postoje a jimi ovlivnit svůj charakter a osobnost.

3.2 První pomoc při cévní mozkové příhodě

Tato kapitola čerpá z knihy Neurologie [16] a ze stránek Ministerstva zdravotnictví ČR¹ a fyziokliniky Praha².

Jelikož je zadání bakalářské práce zaměřeno právě na cévní mozkovou příhodu, je potřeba přiblížit jak se projevuje a jaká je správná první pomoc v této situaci.

Cévní mozková příhoda, známá pod pojmem mrtvice, nastává při přerušení zásobování mozku kyslíkem. Tato situace je způsobena ucpaním tepny trombem (krevní sraženinou), případně prasknutím cévy v mozku, jak je ilustrováno na obrázku 3.1.



Obrázek 3.1: Příčina cévní mozkové příhody³

Příznaky

Mezi hlavní příznaky patří ochabnutí jedné strany těla, pokles ústního koutku a koutku oka na dané straně těla, a omezená schopnost mluvit. Může se objevit i ztráta vědomí či necitlivost ochablé části těla. Tyto příznaky jsou vypořádány na obrázcích 3.2, 3.3 a 3.4.

Mezi vedlejší příznaky patří porucha vědomí, výpadek poloviny zorného pole nebo dvojitě vidění, atypická bolest hlavy, ztuhlost šíje, případně závratě s pocitem na zvracení.

¹<https://www.nzip.cz/clanek/980-cevni-mozkova-prihoda-co-to-je>

²<https://www.fyzioklinika.cz/clanky-o-zdravi/cevni-mozkova-prihoda-cmp>

³obrázek převzat ze stránek Ministerstva zdravotnictví České republiky <https://www.nzip.cz/clanek/980-cevni-mozkova-prihoda-co-to-je>

První pomoc

V případě cévní mozkové příhody je potřeba jednat rychle a co nejdříve zavolat zdravotní záchrannou službu. Jedině včasná lékařská pomoc dokáže člověku v této situaci zachránit život.

Nejdůležitější je zjistit, kdy nastaly první příznaky mrtvice, a to co nejpřesněji. Pokud u člověka nastala cévní mozková příhoda, je potřeba jej do 3 hodin dopravit do nemocnice na odpovídající oddělení.

Pro včasné rozpoznání mrtvice se používá metoda FAST (obrázky byly staženy ze stránek kampaně Čas je mozek⁴):

1. Face (obličej) – ochrnutí obličeje, které se projevuje poklesem koutku úst a oka, zpravidla jen na jedné polovině; mohou se objevit tekoucí sliny



Obrázek 3.2: Příznak cévní mozkové příhody – pokleslý koutek⁵

2. Arm (ruka) – pacient má problém zvednout jednu ruku, nemá v ní cit, nedokáže udržet předmět



Obrázek 3.3: Příznak cévní mozkové příhody – ochrnutí končetin⁶

⁴<https://www.casjemozek.cz/>

⁵obrázek převzat ze stránek kampaně Čas je mozek<https://www.casjemozek.cz/>

⁶obrázek převzat ze stránek kampaně Čas je mozek<https://www.casjemozek.cz/>

3. Speech (řeč) – daný člověk má problémy s mluvením, popřípadě s pochopením smyslu toho, co mu říkáte



Obrázek 3.4: Příznak cévní mozkové příhody – problémy s řečí⁷

4. Time (čas) – pokud se objeví některý z těchto příznaků, je potřeba co nejdříve zavolat zdravotníky pomocí tel. čísla 155

Po kontaktování zdravotní záchranné služby lze člověku pomoci tím, že se jej pokusíme uklidnit, položíme jej na záda a podložíme vrchní část těla. Pokud má člověk těsné oblečení, je vhodné jej uvolnit. Pokud má člověk zubní protézu, můžeme se ji pokusit vyndat – ovšem jen za předpokladu, že je uvolněna. Poslední věcí, kterou můžeme pro daného člověka udělat, je zvlhčovat mu oko s povislým koutkem, buď pomocí kapek, nebo jemným mnutím.⁸

Léčba cévní mozkové příhody

Po převezení pacienta do nemocnice je provedeno vyšetření, jsou provedeny a zanalyzovány krevní odběry a je provedeno neurologické vyšetření, kdy se zjistí širší nálezu na škále NIHSS (National Institutes of Health Stroke Scale), která se provádí dle tabulky s příznaky.⁹ Poté proběhne uzavření tepny nebo je rozpuštěn trombus (obojím lze provést podáním léku nebo mechanicky) – dle příčiny vzniku cévní mozkové příhody. Poté je potřeba zahájit prevenci,

3.3 Základní principy gamifikace

Tato kapitola čerpá z knihy Encyclopedia of Information Science and Technology [19] .

Jaký je důvod vyvíjet hru o první pomoci, když už existují aplikace na dané téma, a děti často absolvují kurz první pomoci během studia na základní škole? Důvodem je, že děti mnohem více zaujme hra, než kurz nebo instruktážní videa. A právě převedení neherních prvků (tj. první pomoc) na prvky herní se nazývá gamifikace.

Gamifikace využívá přirozené soutěživosti lidí, přání zvítězit a zvědavosti, jak příběh či hra skončí. Mezi techniky gamifikace patří¹⁰ :

- získávání bodů
- získávání odznaků
- žebříčky

⁷obrázek převzat ze stránek kampaně Čas je mozek <https://www.casjemozek.cz/>

⁸https://www.youtube.com/watch?v=nKZa8c28RU4&ab_channel=Ba%C5%A1trngMichalKubov%C4%8D%C3%ADk

ADk

⁹tabulka viz Neurologie od E. Růžičky [16], str. 230

¹⁰<https://gamification21.wordpress.com/content/1-what-is-gamification/>

- grafy postupu uživatele
- postupné odemykání obsahu
- příběhy
- týmy
- získávání herních předmětů
- a další

Cílem gamifikace je udělat produkt více poutavým a zábavnějším. Často má za cíl lidi motivovat k vykonávání stále stejné aktivity, a toho dosahuje pomocí vizualizace postupu – grafů, získávání bodů a následné zvyšování úrovně, a podobně. Gamifikace se uplatňuje i v případě, že je potřeba motivovat lidi k nějaké činnosti nebo k podávání co nejlepších výkonů¹¹. V takovém případě se použijí žebříčky a týmy.¹²

Dle Yu-kai Chou, který se věnoval výzkumu gamifikace pro řadu let, je základem gamifikace 8 lidských vlastností [4] str. 23–24 (hovořil o tom i na TedTalks)¹³ a jsou zmíněny i v článku Gamifikace [19] :

- pocit důležitosti – využito například u aplikace Pain Squad, která motivuje děti, které trpí rakovinou, aby zaznamenávaly své bolesti; děti jsou zde tajnými agenty, kteří se snaží chytit zloducha bolest
- vědomí že se člověk posouvá ke svému cíli, ocenění
- projevení kreativity – například stavení výtvorů z lega, i když pomocí návodu; využito ve hře FoldIt, která pomohla vědcům odhalit strukturu proteinů v buňce AIDS – vědci nedokázali problém vyřešit po 15 let, díky této aplikaci jej nějaký hráč vyřešil za 10 dnů
- vlastnictví – díky tomu, že má člověk pocit, že něco vlastní, snaží se to vylepšit, ochránit a chce získat více věcí, např. herní měnu; příkladem je aplikace DragonBox, která motivuje děti k učení se algebry pomocí dráčka, který je schovaný v krabici a vyleze, pokud venku nic není – proto je například potřeba převést rovnici do tvaru $x = 2*y + 5$ (aby na straně s x nic nebylo)
- sociální vliv – založeno na starosti o to, co si o nás ostatní myslí – například SAS posílalo jejich zákazníkům spolu s účtem za energie i informace o jejich umístění v jejich městě dle spotřeby energie, průměrnou spotřebu, nejlepší a nejhorší spotřebu – za rok lidé utratili za energie o 250 milionů dolarů méně než předchozí roky
- nedostatek a nedočkavost – lidé (často děti) něco chtějí jen z důvodu, že to momentálně mít nemohou; typicky se používá u televizních reklamních prodejí – „Pokud koupíte ihned, dostanete jedinečnou slevu 2 000 Kč.“

¹¹<https://www.youtube.com/watch?v=9vJRopau0g0&t=1s>

¹²<https://gamification21.wordpress.com/learning-content-2/5-why-gamify/o>

¹³https://www.youtube.com/watch?v=v5Qjuegtiyc&ab_channel=TEDxTalks

- nepředvídatelnost a zvědavost
- tendence vyhnout se prohře – například aplikace *ZombieRun*, která motivuje uživatele běhat, aby nebyli snězeni zombie

Využití gamifikace může být prosté – v jednom městě se rozhodli motivovat lidi pro použití schodů místo eskalátorů, a na schody přidělali keyboardové klávesy, které hrály hudbu, když po nich někdo šel. Počet lidí využívající schody se zvedl o 66 %¹⁴.

Příkladem využití gamifikace k učení je výuková aplikace *Duolingo* – uživatel získává body za každou úspěšně dokončenou úroveň, za splnění úkolů získává odznaky a dle svého týdenního postupu se umísťuje v žebříčcích. Postupně se odemykají náročnější úrovně. Hra motivuje uživatele opakovat i starší lekce tím, že po dokončení lekce její ikona zezlátne, a časem začne praskat. Pokud si uživatel lekci zopakuje, ikona se opět zacelí. Dále je zde možnost vsadit herní měnu na to, že následující týden uživatel získá každý den určitý počet bodů – pokud to uživatel dokáže, získá dvojnásobek toho, co vsadil, jinak o peníze přijde.

3.4 Existující aplikace o první pomoci

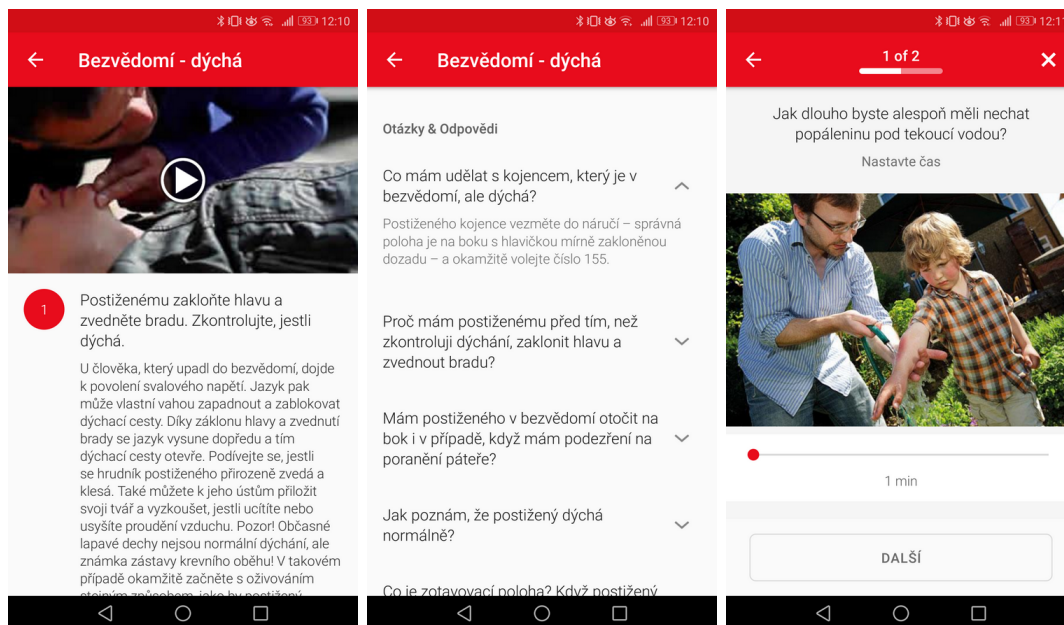
Na Google Play Store i na internetových stránkách je velké množství aplikací, které se se věnují první pomoci. Často jsou ale v cizích jazycích či se věnují spíše vzdělávání dospělých, než dětí. Osobně jsem vyzkoušela několik z nich, a níže jsou podrobně popsány 4 nejlepší aplikace, které jsem našla.

Aplikace První pomoc

Tato aplikace od Českého červeného kříže má na Google Play¹⁵ velmi dobré hodnocení – 4.7 hvězdiček z 5. Aplikace je přehledná (viz obrázek 3.5), obsahuje všechny důležité informace, které jsou často doplněny o instruktážní videa. Ke každé situaci jsou uvedeny nejčastější otázky a odpovědi, a situace samy na sebe často odkazují, pokud je to relevantní – například bodnutí hmyzem obsahuje odkaz na alergickou reakci. Po nastudování informací může uživatel dokonce otestovat své znalosti. Video jsou krátká a stručná a člověk si je tedy může pustit i přímo při vykonávání první pomoci.

¹⁴https://beyond-social.org/wiki/index.php/Review_Piano_Stairs

¹⁵<https://play.google.com/store/apps/details?id=com.cube.gdpc.cze>



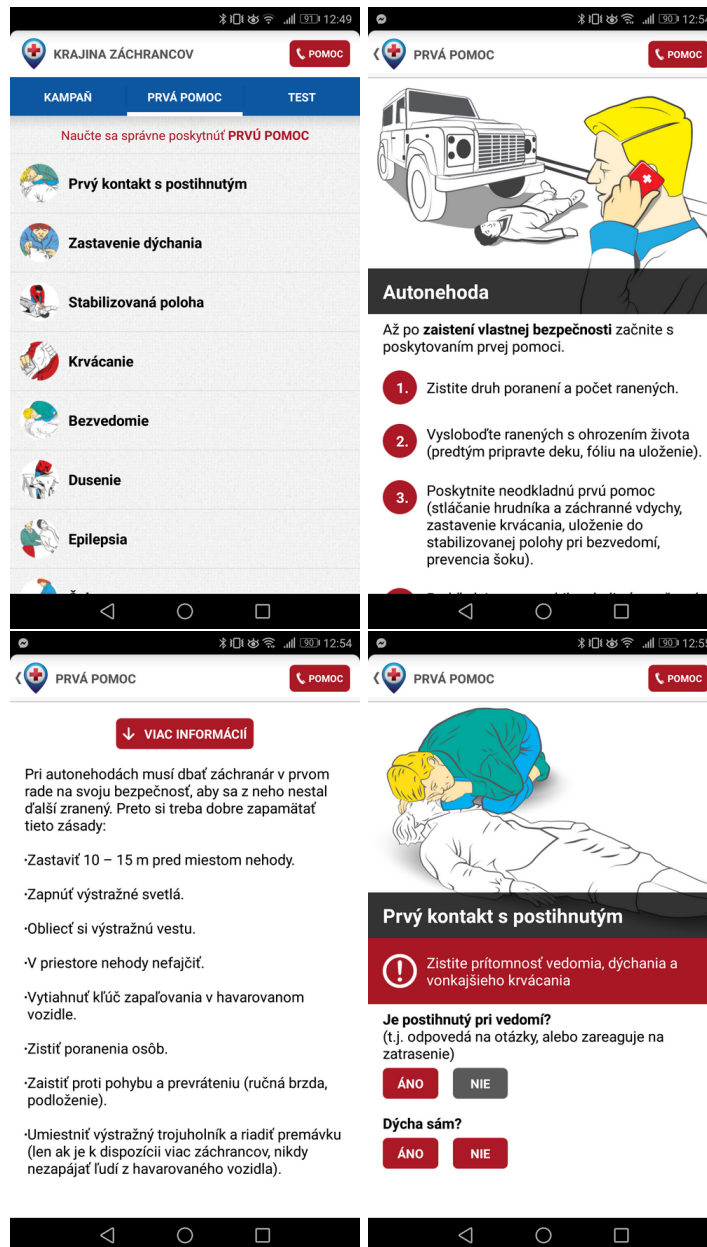
Obrázek 3.5: Vzhled aplikace První pomoc¹⁶

Aplikace Prvá pomoc

Jedná se o slovenský ekvivalent předchozí aplikace s hodnocením 4.5 hvězdiček z 5¹⁷. Video jsou doplněna o příběh, což dokáže děti více nadchnout. Navíc obsahuje počáteční dotazník (viz obrázek 3.6), který uživatele nasměruje na patřičnou první pomoc v dané situaci – uživatel odpovídá na otázky typu „Je postihnutý pri vedomí?“, „Dýchá sám?“, a podobné.

¹⁶obrázky jsou snímky obrazovky pořízené na mém mobilním telefonu; aplikace byla stažena z Google Play Store <https://play.google.com/store/apps/details?id=com.cube.gdpc.cze>

¹⁷<https://play.google.com/store/apps/details?id=com.cube.gdpc.cze>



Obrázek 3.6: Vzhled aplikace Prva pomoc¹⁸

¹⁸obrázky jsou snímky obrazovky pořízené na mém mobilním telefonu, aplikace byla stažena z <https://play.google.com/store/apps/details?id=com.cube.gdpc.cze>

Aplikace Animovaná první pomoc

Tato hra od Záchraného kruhu je bohužel placená – na Google Play Store stojí 85 Kč¹⁹, proto nebylo možné ji osobně vyzkoušet. Jakékoli hodnocení také chybí, i když má aplikace na Google Play přes 100 stažení.

Na internetu je však k dispozici několik videí přímo od Záchraného kruhu, která mají stejnou grafiku jako tato aplikace (viz obrázek 3.7). Po jejich zhlédnutí lze předpokládat, že hra bude koncipovaná podobně. Jedná se o jednoduchá instruktážní videa, která pomocí jednoduchého jazyka popisují postup při provádění první pomoci. Instrukce jsou udávány hlasovými pokyny, a současně se řečený text objevuje i na obrazovce. Kromě samotných rad, jak postupovat při daném problému, jsou součástí videa i doprovodné informace, které dítěti poskytnou základní vědomosti o lidském těle.

Stejně tak existuje hra se stejnou grafikou, také od Záchraného kruhu, a to na stránkách Alík. Děti hrou provází bernardýn, který dětem říká, co mají dělat a proč. Nejedná se o hru v pravém slova smyslu, jelikož dítě samo nic nedělá, jen získává informace od pejska.



Obrázek 3.7: Vzhled aplikace Animovaná první pomoc²⁰

Aplikace Baby Panda's First Aid Tips

Jedná se o hru, která má podobný koncept jako Pejsek záchranář. Aplikace má hodnocení 4 hvězdičky z 5 a přes 5 milionů stažení²¹. Je zaměřena na předškolní děti a žáky 1. stupně základní školy. Uživatele provází hrou robot, který pomáhá ošetřovat zraněné děti. Po každém ošetření dítěti následuje sada otázek, u kterých si uživatel připomene získané zkušenosti. Každé ošetření obsahuje řadu úkonů – například při ošetřování holčičky, která se opařila vařící vodou, dítě nejdříve opláchně ruku holčičky studenou vodou, poté

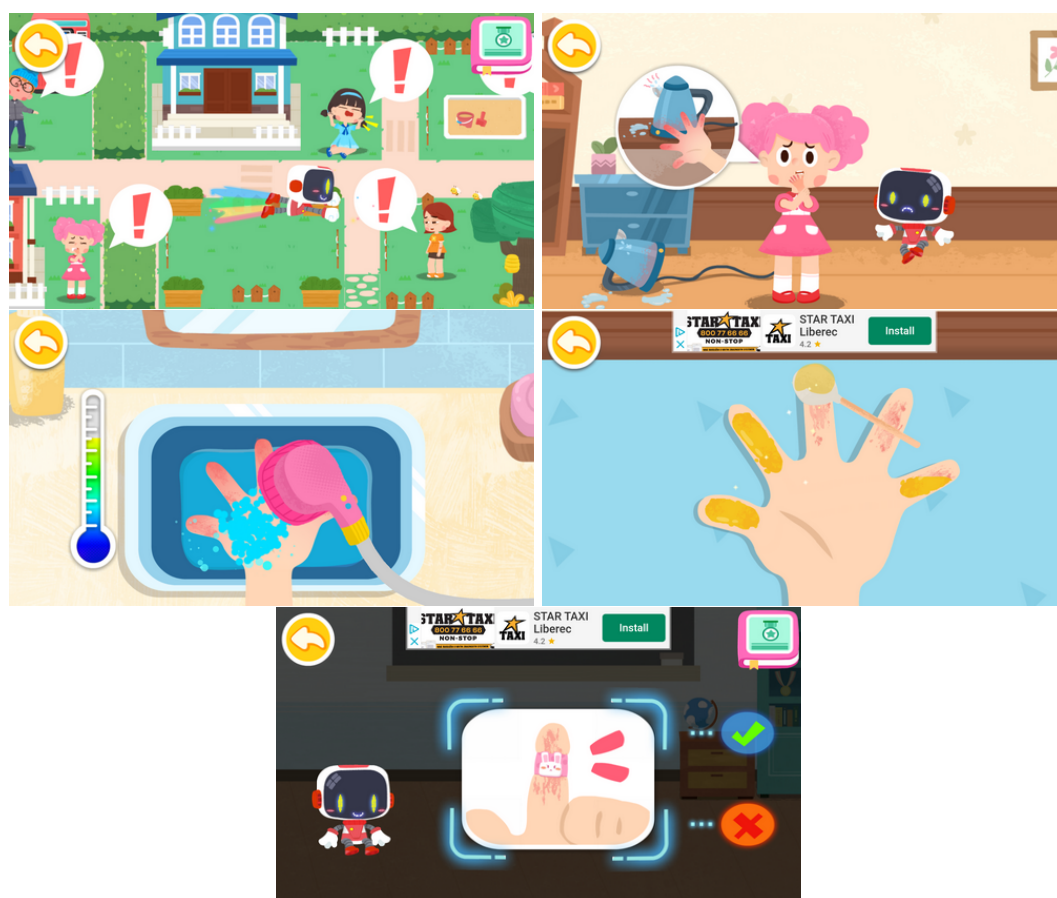
¹⁹<https://play.google.com/store/apps/details?id=air.cz.ucenibezmuceni.animovanaprvnipomoc>

¹⁹<https://www.alik.cz/h/prvni-pomoc>

²⁰obrázky byly převzaty z oficiální stránky aplikace na Google Play Store <https://play.google.com/store/apps/details?id=air.cz.ucenibezmuceni.animovanaprvnipomoc>

²¹<https://play.google.com/store/apps/details?id=com.sinyee.babybus.ambulance>

namaže zraněná místa mastí a následně ruku zakryjí obvazem. Mezitím robot říká, proč se jednotlivé úkony provádějí, a také co naopak nedělat. Příklady úkolů, které má uživatel splnit, jsou uvedeny na obrázcích 3.8.



Obrázek 3.8: Vzhled aplikace Baby Panda's First Aid Tips²²

Hra je zajímavá, poutavá a přitom jednoduchá a dítě se opravdu něco naučí. Je ovšem v angličtině, kterou mladší děti neovládají. Bylo by jistě přínosné vytvořit podobnou hru pro děti i v češtině.

²²obrázky jsou snímky obrazovky pořízené na mém mobilním telefonu; aplikace byla stažena z Google Play Store <https://play.google.com/store/apps/details?id=com.sinyee.babybus.ambulance>

Kapitola 4

Návrh struktury a vzhledu aplikace

Tato kapitola obsahuje návrh jednotlivých prvků hry tak, aby byla co nejvíce přizpůsobena jejímu účelu, tj. vzdělávání dětí. Obsahuje tedy aplikování teorie o vzdělávání dětí a gamifikace do praxe. Dále obsahuje první návrh vzhledu aplikace a rozšíření, prvotní návrh potřebných scén ve hře a způsob práce s daty, tj. otázkami a odpověďmi, které budou ve hře potřeba.

4.1 Analýza současného stavu

Z průzkumu existujících aplikací o první pomoci (viz předchozí kapitola), lze vyvodit výsledky uvedené v tabulce 4.1:

Název aplikace	česky	pro děti	zábavné	hra	zdarma
První pomoc	Ano	Ne	Ne	Ne	Ano
Prvá pomoc	Ne	Částečně	Částečně	Ne	Ano
Animovaná první pomoc	Ano	Ano	Ano	Ne	Ne
Baby Panda's First Aid	Ne	Ano	Ano	Ano	Ano

Tabulka 4.1: Vlastnosti některých existujících aplikací, které vzdělávají uživatele v první pomoci

Aplikace První pomoc je pro děti nevhodná. Postupy při první pomoci jsou sice popsány dobře, ale mladším dětem dělá delší text zpravidla problém, a některým slovům by děti nižšího věku nemusely rozumět. Navíc se jedná zejména o vzdělávací aplikaci než hru.

U aplikace Prvá pomoc, která je k dětem více vstřícná, je bohužel ve slovenštině. Dnešní české děti slovenštině nerozumí a aplikace je pro ně tedy prakticky nepoužitelná. Navíc se stále nejedná o hru, i když by tato aplikace byla pro děti o něco více zábavnější než předchozí.

Animovaná první pomoc by děti mohla bavit, ale stále se nejedná o hru v pravém smyslu slova, jelikož se jedná spíše o videa. Na mobilním telefonu ji navíc nelze stáhnout zdarma, přičemž děti mají častěji přístup k mobilnímu telefonu, než k počítači.

Aplikace Baby Panda's First Aid je výbornou hrou pro děti, je ale bohužel v angličtině, kterou mladší děti tolik neovládají.

Z uvedených informací je patrné, že současné době není na trhu žádná podobná aplikace v češtině, která by cílila na stejnou věkovou skupinu – předškoláci a žáci prvního stupně

základní školy. Jediná česká aplikace, která se zaměřuje na vzdělávání dětí, je placená. Navíc se nejedná o hru, ale o edukační videa.

Výbornou vzdělávací hrou pro děti je Baby Panda's First Aid, která je bohužel v angličtině. Jedná se ale o velmi dobře koncipovanou hru, která by mohla být inspirací pro Pejska Záchranáře.

Na základě analýzy současného stavu a domluvy se Zdravotnickou záchrannou službou Jihomoravského kraje jsem si jako téma hry zvolila cévní mozkovou příhodu. Mrtvice je druhou nejčastější příčinou úmrtí v České republice a přitom její příznaky nepozná až 70 % obyvatel, a proto je důležité šířit povědomí o tomto problému¹.

Vzhledem k informacím o mobilních operačních systémech v ČR (kapitola 2.2) je aplikace Pejska Záchranář vyvíjena pro operační systém Android. Vzhledem k poměru mobilních operačních systémů není nutné vyvíjet aplikaci i pro iOS, protože jej používá jen malá část uživatelů – proto je zvolen nativní vývoj aplikace. Pro vývoj aplikace je použito vývojové prostředí Unity 2.4.

4.2 Návrh hry pro děti

Jelikož se jedná o aplikaci pro děti, je potřeba ji i její vývoj přizpůsobit dětským potřebám.

Jako člověk, kterého je potřeba zachránit, byl vybrán dědeček a příběh se odehrává doma. Díky tomu se dítě může vžít do představy, že se jedná o jeho vlastního dědečka, a díky tomu vzniká pozitivní motivace spolu se sociálním motivem 3.1 – dítě cítí odpovědnost za 'svého' prarodiče, a zároveň mu samo chce pomoci, protože jej má rádo. Ačkoli jsou si dospělí plně vědomi toho, že se jedná o smyšlený příběh, děti často příběhy přenášejí do skutečného života. Je zřejmé, že dítě může vycítit i jistou negativní motivaci – pokud dítě nic neudělá, dědeček by mohl zemřít, a případně by mu to rodiče mohli klást za vinu. Děti v tomto věku ale spíše pocítí pozitivní motivaci, zvláště má-li k dispozici pejska záchranáře, který mu rád pomůže.

Na poznacích z kapitoly 3.1 je postaveno také rozhodnutí, že dítě bude hned po zvolení odpovědi vědět, jestli byla správná nebo špatná, a ne aby příběh pokračoval dál, a na konci se např. nepovedlo dědečka zachránit. Hráč by nevěděl, v jaké otázce udělal chybu, a navíc by to mohlo působit demotivačně, jelikož dítě selhalo a nedokázalo dědečka zachránit. Současně by dítě mělo dostat vysvětlení, proč je odpověď špatná (pokud je to možné). Díky tomu se více naučí a je možné, že ho to posune dále v jeho myšlení. Současně by možnost prohrát a tedy nezachránit dědečka mohla být u takového typu hry silně demotivující, a aplikace by pak neměla kýžený efekt. Je-li cílem aplikace dítě naučit jak správně postupovat, nemělo by být trestem za špatnou odpověď něco tak smutného, jako je smrt prarodiče. Tím spíše, že děti jsou schopny se do příběhu zcela ponořit.

Kromě samotné reakce pejska, zda byla odpověď špatná nebo dobrá, je potřeba zvolit vhodný výraz a gesto pejska, aby dítě vidělo reakci co nejdříve. Mezi nejlepší návrhy patří mírně zamračený pejska kroučící prstem jako symbol špatné odpovědi a usmívající pejska vrtící ocáskem a ukazující zdvižený palec.

Jelikož dítě má potřebu být oceňováno za dobře provedenou práci, pejska by měl dítě také občas pochválit – oblíbené pochvaly dětí jsou „Dobře!“, „Výborně!“, „Skvěle!“, „Dobrá práce!“ a podobně. Na konci, po úspěšném zachránění dědečka, by dítě mělo slyšet, že odvedlo dobrou práci, že díky němu je dědeček zdravý, nebo něco podobného.

¹<https://www.casjemozek.cz/>

Při průzkumu dětí daného věku vyšlo také najevo, že u her, kde právě takto zodpovídají otázky, hrozí, že dítě bude mít pocit, že se neblíží ke zdárnému konci, a hru těsně před úspěšným dokončením vzdá. Proto by ve hře mělo být kupříkladu počet zbývajících otázek nebo nějaký ukazatel, který by hráči řeklo, o kolik už postoupilo.

4.3 Využití gamifikace

V této práci je využita technika příběhu. Uživatel se pomocí odpovědí pokouší pomoci člověku, který je v ohrožení života – v tomto případě u něj nastala cévní mozková příhoda. Jen správné zodpovězení otázky může dítě motivovat k zodpovězení další – při správné odpovědi, respektive při pochvalném gestu pejska, se uvolňuje dopamin, který je tzv. drogou štěstí. Po uvolnění dopaminu mozek touží po dalším dopaminu, a tak je motivován pokračovat ve hře. Tím, že se dítěti na konci příběhu podaří dědečkovi zachránit život, má radost z úspěchu, má ze sebe dobrý pocit, a díky tomu si pravděpodobně zapamatuje více informací. Navíc jej pozitivní zkušenost může motivovat k tomu, aby si stáhlo další hry z této série nebo aby si zjistilo další informace o první pomoci z jiných zdrojů.

Nemožnost prohrát hru a také žádná penalizace za špatnou odpověď je také jedním z prvků gamifikace a děti se tak necítí pod tlakem. Pokud pejsek hráče jen opraví, tak se hráč pravděpodobně pokusí odpovědět znovu (průzkum proveden Markem Roberem a výsledky zveřejněny v rámci přednášky na TeddXPenn: The Super Mario Effect – Tricking Your Brain into Learning More²).

Dalším prvkem, kterého by práce mohla využít, je tzv. progress bar (ukazatel průběhu), který by uživateli ukazoval, jak v záchraně dědečka postoupil a kolik otázek mu zhruba zbývá (vzhled progress baru viz obrázek 4.1). Pejsek by měl po každé správné odpovědi dítě pochválit, aby obdrželo okamžitou zpětnou vazbu.



Obrázek 4.1: Progress bar (ukazatel průběhu) použitý v aplikaci

Odměnou by pro děti mohlo být i to, že na závěru hry se pejsek vrátí za svou rodinou, nebo mu mohou dát granule. Případně lze využít techniky odměn a udělit dítěti po úspěšném ukončení hry odznak nebo diplom.

4.4 Návrh vzhledu aplikace

Z důvodu, že je zadání bakalářské práce součástí série her s Pejskem Záchranářem, byl vzhled pejska navrhnout grafikem záchranné zdravotnické služby Jihomoravského kraje, panem Daliborem Krchem. Tento návrh byl použit pro vytvoření postavy pejska a animací, které jsou ve hře použity. Ostatní grafiku bylo potřeba vytvořit, nebo použít grafiku, která je dána volně k použití (tzv. stock grafika). Vzhledem k předem danému stylu pejska bylo vhodnější vytvořit grafiku vlastní.

Hlavní příběh se má odehrávat v domě, kde dítě bydlí. Dítě přichází do pokoje, kde dědeček sedí v křesle. Dědeček ale vypadá zvláštně – má pověšený koutek úst a koutek

²https://www.youtube.com/watch?v=9vJRopau0g0&ab_channel=TEDxTalks

oka, a má ochablou celou polovinu těla. V tom se v místnosti objeví pejsek Záchranář, aby dědečkovi pomohl. Návrh je představen obrázkem 4.2.



Obrázek 4.2: Návrh obrazovky hlavního příběhu – pejsek, dědeček, komiksová bublina a tlačítka s odpověďmi

Po konzultaci s Mgr. Miladou Smutnou jsem dospěla k názoru, že pouhý hovor s paní ze střediska a záchrana životů by děti nemuselo příliš bavit. Bylo by vhodné do hry zakomponovat pár minihry. Tyto minihry by na sebe měly navazovat, a stejně tak i na samotný příběh. Jejich návrhy jsou uvedeny na obrázcích 4.3, 4.4 a 4.5.

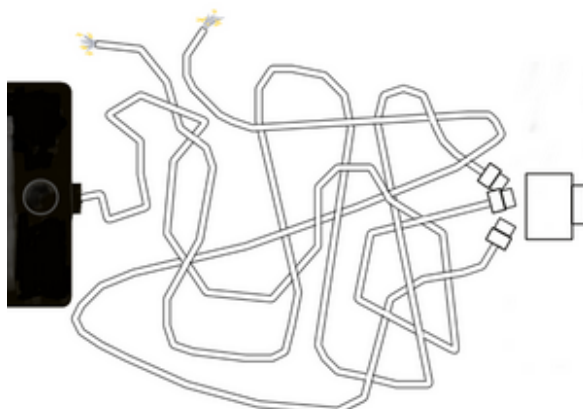
První minihra by mohla být inspirována hrami Hidden Objects, kdy děti hledají určité předměty, které jsou ukryty na obrázku. Zasazení do příběhu by bylo následující: Pejsek chce dědečkovi pomoci a zavolat mu záchranku. Nemá u sebe ale mobilní telefon, a musí tak použít dědečkův telefon. Dědeček schovává telefon do šuplíku, ve kterém je ale velký nepořádek, a tak mu děti musí s hledáním pomoci.



Obrázek 4.3: Návrh minihry 1 – hledání mobilního telefonu v šuplíku

Po úspěšném nalezení telefonu pejsek by mohl zjistit, že je telefon zcela vybitý, a je potřeba jej nejdříve nabít. Druhou minihrou by tedy mohl být úkol zvolit, který kabel

použít, aby se telefon mohl nabít. V šuplíku se totiž do sebe zamotalo několik drátů a tvoří nepřehledné klubko.



Obrázek 4.4: Návrh minihry 2 – nalezení funkčního kabelu pro nabití telefonu

Třetí minihrou by mohl být úkol zvolit, do jaké zásuvky zapojit nabíječku. Ze čtyř zásuvek je totiž funkční jen jedna. U každé zásuvky je nakreslen obrázek, a nad zásuvkami je napsaná hádanka. Pokud dítě úspěšně hádanku vyluští a zapojí nabíječku do správné zásuvky.



Obrázek 4.5: Návrh minihry 3 – nalezení fungující elektrické zásuvky na základě hádanky

Po nabití telefonu by následoval hlavní příběh hry – dítě pomůže pejskovi vytočit číslo záchranné služby, a poté už dítě odpovídá na dotazy paní ze střediska a pokouší se dědečkovi zachránit život.

4.5 Návrh jednotlivých scén aplikace

Dle přání Zdravotnické záchranné služby by postup hry měl být následující:

Hra začíná videem, kde se dědeček dívá na televizi. Najednou se zhroutlí ke straně, hlava mu klesne na rameno, svěsí se pravý koutek oka a úst a levé končetiny se uvolní – dědeček

je zasažen cévní mozkovou příhodou. V ten moment příběhne pejsek, a pomáhá hráči se záchranou dědečka. Poté následuje hlavní příběh hry proložený minihrami. Po zodpovězení poslední otázky je vidět přijíždět sanitku, která dědečka odváží do nemocnice. Jakmile je dědeček v nemocnici, je zachráněn a pejsek skáče radostí.

Pro samotnou hru budou tedy potřeba scéna pro hlavní příběh, scény pro minihry, scéna se sanitkou a scéna v nemocnici – tedy minimálně 6 scén.

Kromě hlavního příběhu je potřeba vytvořit úvodní scénu, tedy menu, kde se bude dát hra pustit, vypnout, bude možné zobrazit informace o hře, a také bude obsahovat odkaz na oficiální internetovou stránku pejska záchranáře. Jelikož jsme od ZSSJMK dostali instrukci, ať se informace o hře zobrazí i po úspěšném dokončení hry, bylo by praktické vytvořit zvlášť scénu, která bude obsahovat právě tento text – při změně textu proběhne úprava jen na jednom místě, informace lze teoreticky spustit odkudkoli...

4.6 Potřebná data a jejich uložení

Hráč by se v příběhu měl posouvat zodpovídáním série otázek. Každá otázka má mít určitý počet odpovědí, z nichž je správná jen jedna. Po správné odpovědi je hráči položena následující otázka. Jako vhodné řešení se jeví vytvoření dvou tříd – otázky, a odpovědi.

Otázka by měla obsahovat:

- text znění otázky
- text, který se zobrazí nad polem odpovědí – aby hráč stále věděl znění otázky (i když kratší), když zvolí špatnou odpověď a otázka v komiksové bublině se přepíše
- zda pejsek volá, tj. má telefon u ucha
- zda mluví paní ze střediska – na zobrazení ikony paní
- přístupová cesta k souboru se zvukovou stopou – pejsek říká nahlas otázku

U odpovědi je třeba znát následující informace:

- text odpovědi
- obrázek, který se má zobrazit na tlačítku odpovědi (může být null – ne všechny odpovědi mají obrázek)
- zda je odpověď správná
- reakce pejska na tuto odpověď – co má pejsek říci
- přístupová cesta k souboru se zvukovou stopou s reakcí pejska

Návrh práce s otázkami je následující: po spuštění hry se všechny otázky vloží do zásobníku (LIFO princip – last in, first out), ze kterého se postupně odebírají.

Otázky by bylo vhodné načítat ze strukturovaného souboru. Nejlepším řešením v případě použití Unity je JSON soubor – pro programovací jazyk C# jsou dostupné knihovny pro zpracování JSON souborů a formát je poměrně lehce čitelný člověkem a je snadné jej upravovat.

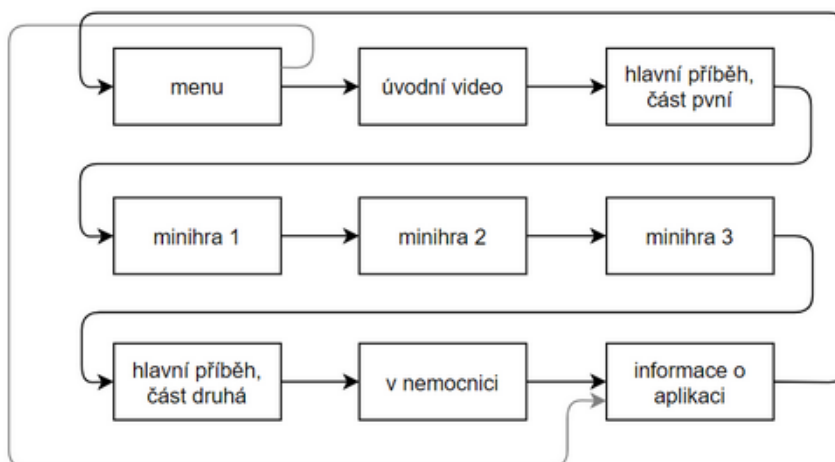
Kapitola 5

Implementace aplikace Pejsek Záchranář

Tato kapitola popisuje implementaci aplikace – práci s daty, základní principy implementace a řízení animací. Na konci kapitoly je popsáno testování aplikace. Aplikace je vyvíjena pro operační systém Android pomocí vývojového prostředí Unity.

5.1 Rozdělení aplikace na scény

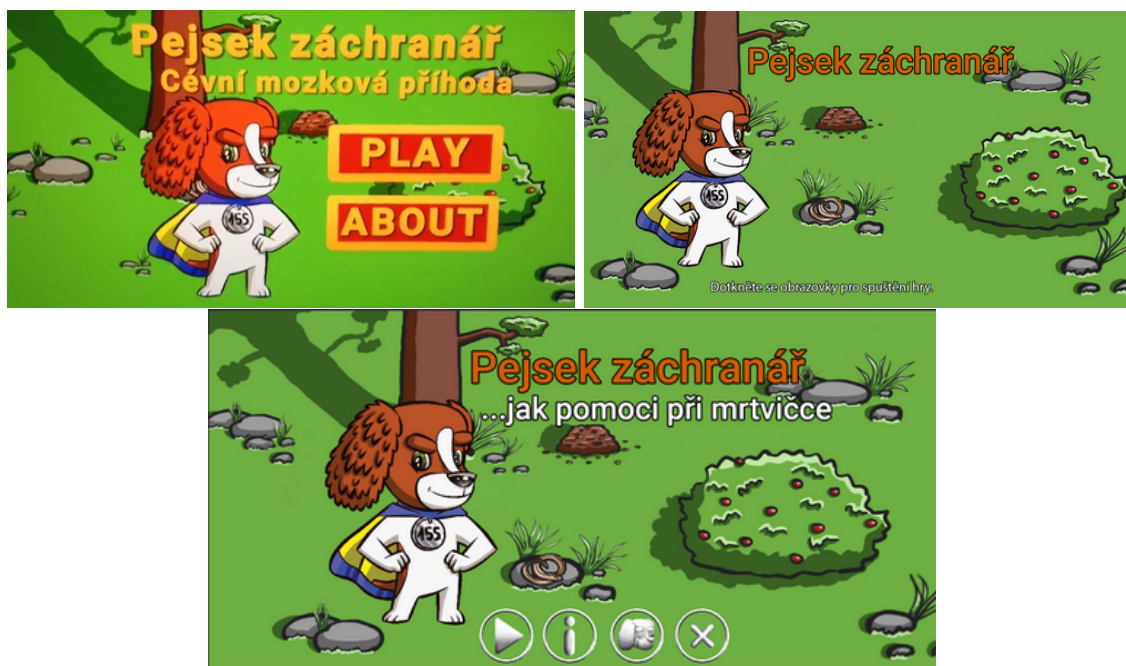
Aplikace obsahuje celkem 11 scén – menu se zvukem, menu bez zvuku, počáteční video, první část hlavního příběhu, tři minihry, druhou část hlavního příběhu, příjezd sanitky, scéna v nemocnici a scéna s informacemi o aplikaci. Jejich posloupnost prezentuje graf 5.1. Aby byla zaručena univerzálnost skriptů při načítání další scény, bylo potřeba vytvořit někdy scénu dvakrát (například scénu hlavního příběhu). Díky tomu lze načíst vždy scénu s build indexem (číslem sestavení) o jednu větší než je hodnota build indexu aktuální scény.



Obrázek 5.1: Jak po sobě následují jednotlivé scény hry

5.2 Uživatelské rozhraní a vzhled aplikace

Menu aplikace prošlo několika etapami vývoje. Původně jsem se držela přání, aby ovládací tlačítka byla červená, hra však působila starým dojmem snad v jakékoli variantě. Poté bylo navrženo jednoduché menu, přičemž by se hra spustila tím, že by se hráč pouze dotkl obrazovky. Nakonec zvítězila varianta se stříbrnými tlačítky na dolní části obrazovky. Vzhledy jednotlivých verzí jsou uvedeny na obrázcích 5.2.



Obrázek 5.2: Postupný vývoj hlavní obrazovky hry (menu) – vybrána byla poslední varianta

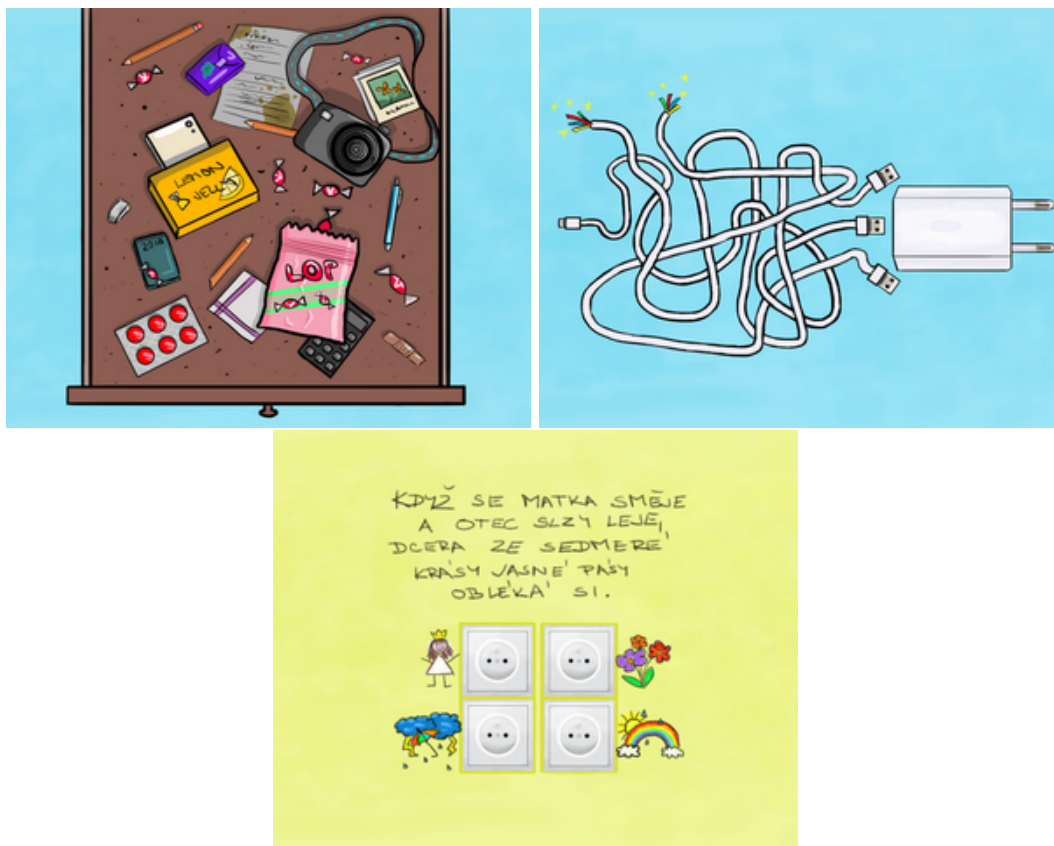
V rámci vývoje vzniklo několik návrhů obrazovky hlavního příběhu i dalších obrazovek. Výsledná grafika byla zvolena dle průzkumu, při kterém se návrhy rozeslaly několika lidem, a byly vybrány nejčastěji zvolené návrhy.

Jako hlavní obrazovka byla zvolena varianta se světle žlutou zdí, světlou skříní a modrým kobercem - jak lze vidět na obrázku 5.3.



Obrázek 5.3: Obrazovka hlavního příběhu (výsledná grafika)

Obrazovky miniher byly postupně upravovány dle doporučení lidí, kteří aplikaci testovali. Finální verze jsou uvedeny na obrázcích 5.4.



Obrázek 5.4: Obrazovky miniher – hledání telefonu v šuplíku, nalezení správného kabelu a hádanka

Pro tlačítka s odpověďmi bylo vytvořeno šest návrhů, z nichž byla nejoblíbenější šestá (poslední) varianta, která je ve hře použita (viz obrázek 5.5).



Obrázek 5.5: Návrh tlačítek s odpověďmi – vybrána byla poslední varianta

Animace

Unity umožňuje velmi jednoduché vytváření animací pomocí vložení obrázku do časové osy. Přepínání mezi jednotlivými animacemi daného objektu je možné pomocí tzv. triggerů, případně podle hodnot určité proměnné (menší, větší, rovno či nerovno nějaké hodnotě). Pro vytvoření animací je potřeba zvolit jaké animace budou použity, a poté jednotlivé obrázky nakreslit.

Pro pejska byly vybrány následující animace a jsou prezentovány sadou obrázků 5.6:

- stojící pejsek vrtící ocasem (jehož základ je použit u dalších animací)
- reakce na správnou odpověď
- reakce na špatnou odpověď
- mluvící pejsek
- telefonující pejsek
- běžící pejsek
- skákající pejsek
- pejsek s vlajícím pláštěm



Obrázek 5.6: Jednotlivé animace pejska záchranáře – stojící, mluvící a telefonující pejsek, reakce na odpovědi, běžící a skákající pejsek

Pro dědečka bylo potřeba vytvořit animace příchod mrtvice, zvednutí ruky, a mluvení, a pro úvodní video a scénu v nemocnici bylo potřeba vytvořit usmívajícího se dědečka - jak je ilustrováno na obrázku 5.7.



Obrázek 5.7: Animace dědečka – zvednutí ruky a mluvení

Jelikož jedním požadavkem bylo postupné objevení nápisu FAST, byly navrženy i animace postupného objevení tohoto textu – text se vynoří odnikud jako malý nápis, poté se postupně zvětšuje směrem od středu, po dosažení dané velikosti se zvýrazní první písmeno, které je písmenem ve zkratce FAST, a putuje směrem vzhůru, kde se nápis postupně zformuje. Zbytek nápisu mezitím pozvolna zmizí - jeden takový krok je uveden na obrázku 5.8.



Obrázek 5.8: Příklad kroku animace nápisu FAST (metoda pro první pomoc při mrtvici)

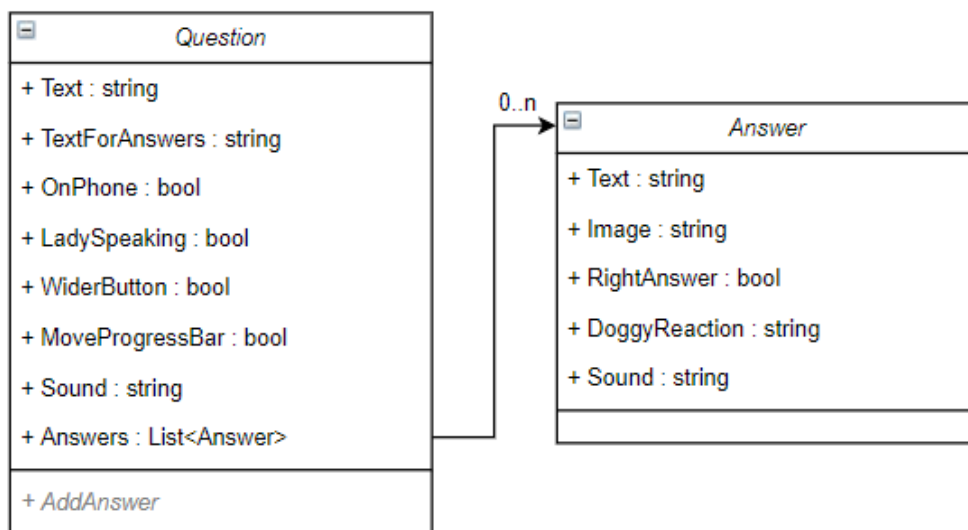
Poslední potřebnou animací byl průjezd sanitky a bylo potřeba zvolit vhodnou reprezentaci pohybu, neboť samotné posouvání auta do stran působí fádně. Nejlépe vypadá

animace, kdy se auto mírně pohybuje vzhůru a zpět dolů. Dále bylo potřeba navrhnout objevení a zmizení komiksově bubliny.

5.3 Zadávání otázek a práce s nimi

Principem hry je, že pejsek Defík klade otázky, na které hráč odpovídá, a tím se posouvá příběh hry. Tyto otázky je potřeba nějak uložit, načíst a pracovat s nimi. Současně jsou pro každou otázku uloženy možné odpovědi, mezi kterými uživatel vybírá (odpovědi, kde by hráč musel sám psát odpověď, by pro děti mohly být příliš obtížné).

Otázky jsou v aplikaci uloženy ve třídě `Question`, a odpovědi na ně ve třídě `Answer`. `Question` obsahuje seznam odpovědí. Jejich třídní diagram je uveden na obrázku 5.9.



Obrázek 5.9: Třídy `Question` a `Answer` použité pro uložení otázek a odpovědí

Atributy třídy `Question`:

- `Text` – text, který říká pejsek, resp. který se objeví v komiksově bublině
- `TextForAnswers` – zkrácený text, který se má zobrazit nad možnými odpověďmi (aby hráč věděl, jaké bylo zhruba znění otázky, když například zvolí špatnou odpověď a text v bublině se přepíše)
- `OnPhone` – zda pejsek volá, tedy drží v pacce telefon (pro zvolení správné animace)
- `LadySpeaking` – zda má větu říkat paní ze střediska (pro zobrazení její ikony)
- `WiderButton` – zda mají být tlačítka odpovědí širší (je-li odpověď moc dlouhá)
- `MoveProgressBar` – zda se po zodpovězení má posunout progress bar (tj. zda se otázka počítá jako důležitá pro posun ve hře)
- `Sound` – cesta k souboru se zvukovou stopou (znění otázky)

- **Answers** – seznam možných odpovědí na otázku
- **AddAnswer** – funkce pro vložení odpovědi do seznamu otázek; jelikož jsou otázky načítány ze souboru, je funkce ve zdrojovém souboru zakomentována, ale nebyla smazána pro případné budoucí použití

Atributy třídy **Answer**:

- **Text** – text odpovědi
- **Image** – obrázek odpovědi (např. tvář dědečka); může být **null**
- **Right Answer** – zda je tato odpověď správná
- **DoggyReaction** – jak pejsek na danou odpověď reaguje (text odpovědi)
- **Sound** – zvuková stopa pejskovy reakce, respektive cesta k souboru

Aby bylo co nejjednodušší změnit znění i množství otázek, které má hráč zodpovědět, jsou otázky načítány ze souboru typu JSON. Díky tomu je možné pro každou scénu použít jiný seznam otázek, aniž by se muselo zasahovat do skriptu – stačí načíst konkrétní soubor. Soubory také může vytvářet či upravovat kdokoli jen s použitím vzoru, a to aniž by daný člověk měl přístup do zdrojových souborů aplikace. Programovací jazyk **C#**, ve kterém se aplikace pro Unity píše, umožňuje použití knihovny **Newtonsoft.Json**, která usnadňuje práci s JSON soubory. Pro jejich načtení stačí jeden příkaz (je-li soubor správně naformátován).

Formát JSON souboru vychází z třídy **Questions** a vypadá následovně:

```
{
  "Text":"text, který pejsek říká",
  "TextForAnswers":"text nad možnými odpověďmi",
  "OnPhone":zda pejsek drží v pacce telefon,
  "LadySpeaking":zda mluví paní ze střediska,
  "WiderButton":zda je potřeba, aby byly tlačítka odpovědí širší,
  "MoveProgressBar":zda otázka ovlivňuje progress bar (tj. jestli se má progress
bar posunout),
  "Sound":"cesta ke zvukovému souboru, který se má spustit",
  "Answers":[
  {
    "Text":"text této odpovědi",
    "Image":"cesta k obrázku dané odpovědi",
    "RightAnswer":zda je odpověď správná,
    "DoggyReaction":"jak pejsek na tuto odpověď reaguje",
    "Sound":"cesta ke zvukovému souboru odpovídající reakci pejska"
  },
  další možné odpovědi
  ]
}
```

Načítání otázek ze souborů má na starosti třída **FillQuestions**, od které si může otázky převzít libovolný skript (zpravidla je však načítá **BubbleScript**).

Který soubor načíst je určeno pomocí build indexu (česky čísla sestavení) aktivní scény. Otázky se načtou do seznamu, následně se seznam otočí, protože v souboru jsou otázky napsány v chronologickém pořadí, ale otázky se pro jednodušší práci s nimi vkládají do zásobníku (neboli LIFO – last in first out). Po otočení seznamu jsou otázky postupně vloženy do zásobníku pomocí příkazu `Push()`.

Zpracování otázky

Otázky se postupně načítají z jednoho zásobníku – první otázka se načte ihned při načtení dané scény, ostatní otázky se načítají v momentě, kdy je správně zodpovězena předchozí otázka a hráč se dotkne displeje. Aktuální otázka se uloží do proměnné, aby s ní bylo možné pracovat. Po vyjmutí otázky ze zásobníku je zavolána funkce třídy `DoggyScript`, která vybere správnou animaci pejska, a případně i funkci pro výběr animace dědečka z třídy `GrandpaScript`. Poté je spuštěn koprogram (anglicky *coroutine*) pro vypsání textu otázky, a koprogram pro připravení tlačítek (ten se nachází ve třídě `Buttons`).¹

Koprogram pro vypsání textu zavolá funkci, která načte a spustí zvukovou stopu (cesta k souboru je předána v argumentech). Následně je zkontrolováno, zda větu neříká paní ze střediska, a případně je zobrazena bublina s ikonou volající paní. Poté probíhá postupné vypisování písmen z textu. Rychlost generování je náhodná, v řádech milisekund. Pokud se hráč během vypisování textu dotkne obrazovky, tak se ihned zobrazí celý text.

Jak probíhá příprava tlačítek v koprogramu je popsáno v následující podkapitole. Stisknutí tlačítek je zachycováno pomocí funkcí z té samé třídy, které jsou na dané tlačítko přímo navázány. Při stisknutí nějakého tlačítka je zavolána funkce, která vyhodnotí, zda byla odpověď správná. Pokud odpověď nebyla správná, pejsek patřičně zareaguje, a tlačítko je deaktivováno. Jinak jsou deaktivovány všechny tlačítka, pejsek dítě pochválí, a pokud je to potřeba, je zavolána funkce pro animaci nápisu FAST ze třídy `FastScript`. Poté je zkontrolováno, zda se má posunout progress bar, a případně je zavolána funkce, která jej má na starosti. Správnou odpovědí je také nastavena proměnná, která označuje, že je otázka úspěšně zodpovězena, na hodnotu `true`. Díky tomu je možné načíst další otázku pomocí pouhého dotyku obrazovky.

5.4 Třídy a s nimi spojené skripty

V Unity má každý prvek scény, jehož chování či vzhled chceme v průběhu hry měnit, vlastní skript, a tím pádem i třídu, která definuje jeho chování. Jedinou výjimkou jsou prvky, které mají nastaveny jedinou animaci – například pejsek v úvodní obrazovce hry.

Skript může obsahovat dvě speciální funkce – `Start` a `Update`. Funkce `Start` je zavolána hned po načtení scény, funkce `Update` je volána s každým snímkem (anglicky *frame*) a využívá se tedy pro zachytávání některých vstupů od uživatele, např. dotyku obrazovky (toho je využito ve skriptu, který ovládá komiksovou bublinu pejska a celý dialog). Pokud daný skript nemá ani funkci `Start`, ani `Update`, je objekt zcela závislý na jiném, který zavolá některou z jeho funkcí.

Jelikož je pro každý objekt potřeba zvláštní třída, nebylo by přínosné kreslit diagram všech tříd, neboť na sobě nejsou nijak závislé. Jedinou výjimkou jsou třídy, jejichž diagram je uveden níže. [5.10](#)

¹koprogram funguje na podobném principu jako např. funkce – je zavolán z hlavního programu, ale na rozdíl od funkcí je možné se z něj do hlavního programu vrátit víckrát, a to pomocí příkazu `yield`; toho se v tomto projektu využívá pro čekání pomocí příkazu `Wait`

Třídy a skripty lze rozdělit do několika kategorií: skripty pro samotnou hru a minihry, skripty pro úvodní video a video projíždějící sanitky, skript pro závěrečnou scénu a skript pro menu.

Funkčnost menu je pokryta jediným skriptem, resp. třídou, která obsahuje funkce pro registraci kliknutí na tlačítka (spuštění hry, informace, načtení internetové stránky pejska záchranáře a ukončení hry). Veškeré animace jsou řízeny z Unity, neboť jsou jednoduché a nenachází se tam žádné přechody.

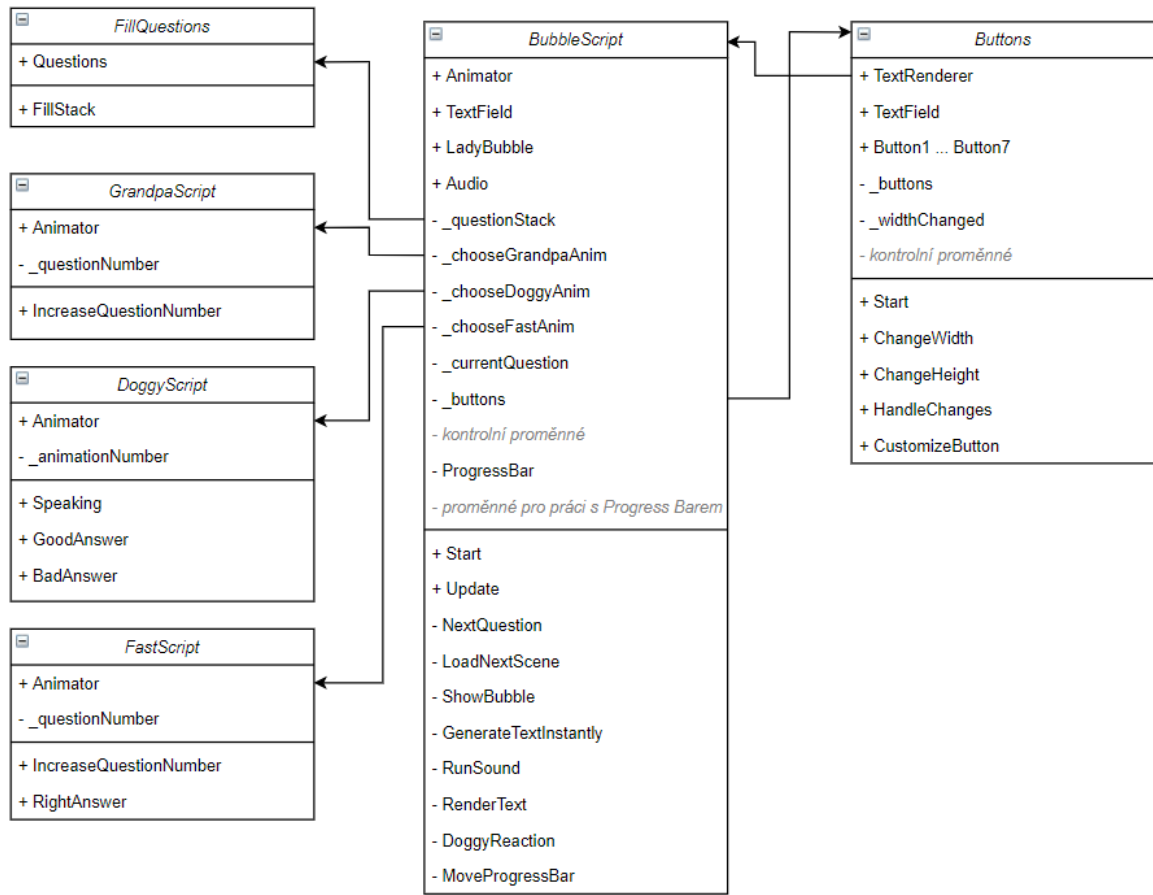
V úvodním videu jsou potřeba tři skripty, z nichž všechny ovládají animace objektů ve scéně: pejska, dědečka a světlo z televize. Aby byla případná úprava co nejjednodušší, jsou všechny animace řízeny z jednoho skriptu pomocí funkcí. Při načtení scény se v `Start` funkci této třídy zavolá funkce, která postupně volá funkce jiných tříd, které zajistí změnu animace požadovaného objektu. Jejich načasování probíhá pomocí příkazu `Wait`.

Hlavní příběh je řízen třídou `BubbleScript` a `Buttons` případně jejich modifikacemi pro minihry. Dále je použita jedna třída pro načítání obrázků a tři třídy pro správu animací dědečka, pejska a nápisu FAST. Třída `BubbleScript` je navázána na komiksovou bublinu ve scéně. Vyskytuje se ve dvou modifikacích – `BubbleScript` a zjednodušený skript pro minihry, který neobsahuje některé funkce a kontroly, které nejsou v minihrách potřeba. Třída pomocí funkce `Update` zachytává vstupy uživatele, a dle konkrétní situace ve hře volá patřičné funkce – načtení další otázky, urychlení výpisu textu dialogu nebo načtení další scény. Kromě dialogu včetně zvukových stop zajišťuje také posouvání hodnoty progress baru (ukazatele průběhu) a pomocí funkcí jiných tříd aktivuje animace pejska `Defíka`, dědečka i nápisu FAST.

Pro video jedoucí sanitky je potřeba jediná třída, která spustí animaci auta a postupně posouvá auto kupředu. Jakmile se auto dostane na určitou pozici, je otočeno a vrací se zpět.

V závěrečném videu je potřeba jen třída (skript), která ovládá tlačítko na ukončení hry. Zde je funkce navázána na dané tlačítko a jeho stisk zachytává Unity. Animace pejska je jednoduchá a je tedy zajištěna Unity.

Jelikož jsou třídy použité v hlavním příběhu hry mírně složitější, je vhodné uvést jejich diagram - viz obrázek 5.10.



Obrázek 5.10: Diagram některých tříd – BubbleScript je důležitý pro běh hry, ostatní třídy znázorňují jejich vztahy

Nejdůležitější třídou je **BubbleScript**, která obsahuje veškerou logiku pro ovládání hry – sama zajišťuje jen práci s otázkami, které jsou uloženy v zásobníku, a voláním funkcí z ostatních tříd zajišťuje spuštění animací a práci s tlačítky.

Ve funkci **Start**, která je zavolána v momentě načtení herní scény, jsou načteny všechny potřebné skripty (skript na ovládání tlačítek, skripty zajišťující animace objektů, ...), potřebné obrázky, je uložen build index scény, je nastaven progress bar na správnou hodnotu, a následně je zavolána funkce na načtení první otázky.

Funkce **Update** je volána při každém snímku a zachytává vstupy od uživatele – pokud se hráč dotkne obrazovky a předchozí otázka je zodpovězena, veškeré aktivity jsou dokončeny a v zásobníku je ještě nějaká otázka, načte se další otázka. Pokud je již zásobník prázdný, je načtena další scéna (protože současná je již dokončená), Pokud ovšem nejsou aktivity jako vypisování dokončeny, tak se stiskem obrazovky zrychlí výpis textu (nastavením kontrolní proměnné na správnou hodnotu).

Funkce, která má na starosti načítání otázek, funguje následovně: funkce vyjme z vrcholu zásobníku otázku, nastaví příznak nezodpovězené otázky a zavolá funkce z jiných tříd, které zajišťují zvolení správných animací. Pomocí koprogramu třídy **Buttons** nachystá odpovědní tlačítka a zavolá koprogram, který zajistí vytištění textu.

Vytištění textu probíhá tak, že nejdříve je textové pole nastaveno na prázdný řetězec, poté je s jistým zpožděním zavolána funkce pro spuštění zvukové stopy a pokud daný text

říká paní ze střediska, tak je zobrazena její ikona. Poté probíhá postupné tištění textu – před vytištěním každého znaku se pomocí funkce `WaitForSeconds` čeká náhodný čas v rádech milisekund. Pokud se kdykoli v průběhu generování textu hráč dotkne obrazovky, je postupné generování přerušeno a text je vytištěn ihned. Po dokončení tisku je nastaven příznak dokončené aktivity.

Reakci na zvolenou odpověď také zajišťuje tato třída – nejdříve je schována ikona paní ze střediska (pokud je aktivní), poté se pomocí indexu, který je předán v argumentu funkce, načte zvolená odpověď, a pomocí jejího atributu `RightAnswer` se zjistí, zda je odpověď správná nebo špatná. Pokud je odpověď špatná, je zvolena odpovídající animace a dané tlačítko je deaktivováno. Pokud je odpověď správná, jsou deaktivována všechna tlačítka, je vybrána odpovídající animace a případně i animace FAST nápisu a je zavolána funkce pro posunutí progress baru. Na konci funkce je v každém případě zavolána funkce pro vytištění textu (a tím se zajistí i spuštění zvukové stopy).

Posouvání progress baru funguje na tom principu, že je k současné šířce výplně progress baru přičtena šířka jednoho kroku, a poté je daný objekt nastaven na odpovídající šířku. Délka jednoho kroku se vypočítá podle množství otázek, které posouvají progress bar, a celkové šířce progress baru.

Jak tyto funkce vypadají ve hře je reprezentováno obrázky [5.11](#).



Obrázek 5.11: Ukázky - upravená tlačítka při dlouhém textu, chování aplikace když je zvolena špatná odpověď a když je zvolena správná odpověď (změna zobrazení tlačítek, animace, posunutí progress baru)

Zpracování zvolené odpovědi

Každá otázka může mít až 7 možných odpovědí, z nichž je správná jen jedna (pro každou otázku byly vybrány možné odpovědi, které se hráči zobrazí, a největší počet možných odpovědí bylo 7 – více odpovědí na dané otázky není potřeba – příklad odpovědí na otázku viz obrázek 5.13). Každé odpovědi odpovídá jedno tlačítko (*anglicky button*), na kterém je zobrazen text nebo obrázek odpovědi - jak je vyobrazeno na obrázku 5.12.



Obrázek 5.12: Příklad odpovědního tlačítka s obrázkem a s textem



Obrázek 5.13: Ukázka – možné odpovědi u jedné z otázek

Kliknutí na tlačítko je zaregistrováno pomocí funkce `ButtonXClicked`, kde `X` reprezentuje číslo tlačítka. Tato funkce je navázána přímo na konkrétní tlačítko, které ji při stisknutí zavolá (samotné zachycení dotyku obrazovky na místě tlačítka je zajištěno knihovnamí Unity). Ovládání tlačítek, včetně zmíněné funkce `ButtonXClicked`, je zajištěno třídou `Buttons`.

Funkce `ButtonXClicked` zavolá funkci z třídy `BubbleScript` a poskytne jí číslo dané odpovědi, aby mohla odpověď řádně zpracovat – veškerá logika zpracování otázek a odpovědí se tedy nachází v `BubbleScriptu`. `Buttons` kromě reakce při stisknutí tlačítka zajišťuje úpravu tlačítek – zobrazení potřebného počtu tlačítek, změna textu a obrázku tlačítka, úprava jejich šířky a výšky dle délky textu. Aby změny neprobíhaly zbytečně, neboť některé z nich jsou náročné, využívá skript několika booleovských proměnných jako příznaků o změnách (porovnání upraveného a implicitního vzhledu tlačítek viz obrázky 5.11 a 5.13).

Obdobně fungují i modifikace skriptu `Buttons` pro minihry. Liší se počtem tlačítek a neobsahují funkce pro jejich modifikace.

Zvuky aplikace

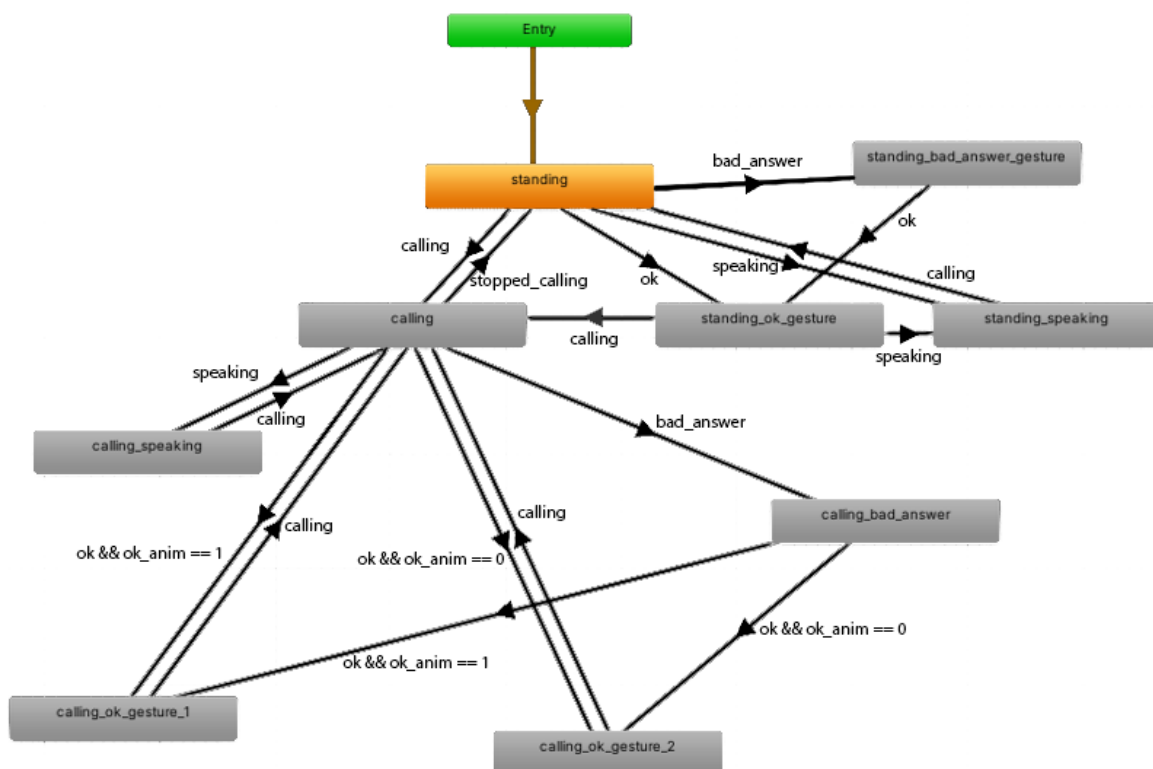
Všechny zvuky v aplikaci jsou přehrávány pomocí Unity objektu **Audio**. Tomuto objektu je předána cesta ke zvukovému souboru, objekt soubor načte a přehraje. Každá věta je tedy uložena ve zvukovém souboru (ve formátu **mp3**).

Pejsek byl namluven hercem Michalem Isteníkem, paní ze střediska byla namluvena tiskovou mluvčí Zdravotnické záchranné služby Jihomoravského kraje.

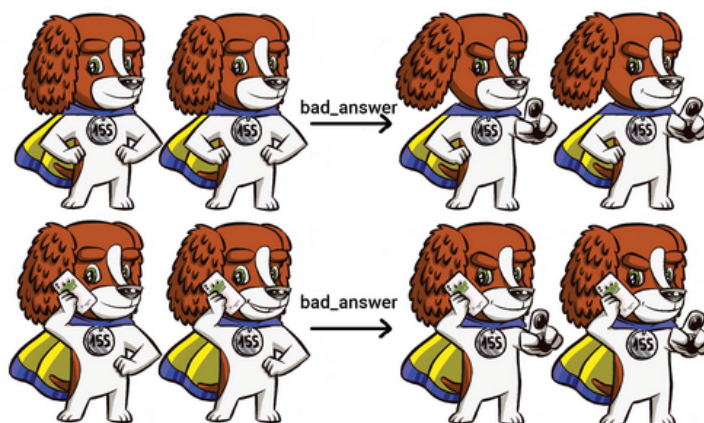
5.5 Přepínání animací objektů

Animace je vždy navázána na jeden konkrétní objekt. Mezi všemi animacemi daného objektu jsou vazby, které určují, do jakého stavu se může objekt dostat ze stavu aktuálního, a za jakých podmínek.

Animace veškerých objektů se ve hře mění pomocí tzv. triggerů (česky spouštěčů) nebo pomocí porovnání proměnné s určitou hodnotou. Díky systému animací v Unity je možné použít jeden trigger na spuštění více animací, které se liší pouze výchozí pozicí. Toho je využito například u animací pejska – reakce na správně či špatně zodpovězenou otázku – znázorněno obrázky 5.14 a 5.15. Trigger **bad_answer** (viz obrázek 5.14), který je spuštěn při špatném zodpovězení libovolné otázky, spustí animaci **standing_bad_answer_gesture** pokud se v danou chvíli pejsek nachází ve stavu **standing**. Pokud se však nachází ve stavu **calling**, je spuštěna animace **calling_bad_answer**.



Obrázek 5.14: Graf znázorňující přechody mezi animacemi pejska – obdélníky reprezentují stavy objektu pejsek, šipky znázorňují přechody a popis šipek určuje podmínku pro uskutečnění přechodu



Obrázek 5.15: Grafická ukázka přepínání animací na základě triggeru a aktuálního stavu

U animace dědečka a nápisu FAST se animace spouští pomocí porovnávání hodnoty proměnné. Hodnota, se kterou je proměnná porovnávána, je ručně zadána v Unity editoru, a rovná se číslu otázky, při kterém má být animace spuštěna. Animace pejska jsou spouštěny ze skriptu dle vlastností otázky či odpovědi – příznaky indikující, že pejsek drží v pacce telefon, že mluví paní ze střediska, a booleovská hodnota `RightAnswer` určující zda je konkrétní odpověď správná.

5.6 Testování a vydání výsledné aplikace

Aplikaci je potřeba během vývoje otestovat, aby byla ověřena její funkčnost a bylo odhaleno co nejvíce chyb. Po jejím vytvoření je také potřeba se ujistit, že splňuje cíle této aplikace. Klíčovými cíli této aplikace je naučit uživatele rozpoznávat příznaky cévní mozkové příhody a naučit ho jak reagovat při situaci, kdy je ohroženo něčí zdraví či život. Testy jsou tedy rozděleny na 2 části: testy, které ověřují funkčnost, a uživatelské testy, které ověřují splnění cílů aplikace.

Testování částí aplikace bylo nutné provádět při běhu aplikace ve vývojovém prostředí. Testování jednotlivých částí (Unit testy) nebylo možné, protože jednotlivé skripty jsou vázány na objekty ve hře. Bylo možné ale využít debugger² a sledovat tak hodnoty proměnných či předávaných argumentů v programu. Důležité části, které bylo nutné otestovat, patří:

- načítání souboru s otázkami (soubor ve formátu JSON) – otestováno pomocí různých validních i nevalidních souborů
- postupné načítání otázek ze zásobníku – kontrola pořadí otázek z různých souborů
- správné zobrazení tlačítek s odpovědmi a jejich přizpůsobení délce textu – zadávání různých délek odpovědí
- správné načítání obrázků k odpovědím – vkládání cest k různým (i neexistujícím) obrázkům
- správné spouštění animací – testování různých kombinací validních přechodů

²softwarový nástroj, který se používá pro nalezení chyby v programu

- správná funkčnost komiksové bubliny – testování schovávání a objevování bubliny a ikony paní ze střediska
- reakce stisknutých tlačítek či dotyku obrazovky

Aplikace byla testována na řadě zařízení s operačním systémem Android s různými verzemi Andriodu a s různě velkým displejem. Konkrétně se jednalo o mobilní telefony Huawei P9 Lite, Samsung Galaxy A7, OnePlus 7 Pro, Xiaomi Redmi Note 10 Pro a o tablet Samsung Galaxy Tab A 9.7. Díky starším zařízením bylo ověřeno, že aplikace funguje správně i na starších verzích operačního systému, a díky menším obrazovkám bylo otestováno správné rozložení tlačítek, aby se bez problému vešla i na displeje s poměrem stran 4:3.

Aby bylo otestováno splnění klíčových cílů aplikace, testovala jsem ji na svém bratranci a sestřenci a jejich kamarádech, kteří jsou v cílové skupině, tj. děti ve věku 7 – 10 let. Při takovém testování je potřeba uživatele v tichosti sledovat a všimnout si jejich kroků (nikoli do hry vstupovat a radit). Cílem bylo zjistit, zda:

- je aplikace intuitivní
- dětem nedělá problém ji ovládat
- není moc těžká
- není až příliš jednoduchá

Po absolvování hry bylo potřeba ověřit, zda děti nyní rozpoznají příznaky mrtvice a vědí, co mají v takové situaci (či podobných situacích) dělat. Proto byly dětem položeny následující otázky:

- Víš o jakou nemoc/problém se jednalo?
- Víš jaké jsou příznaky cévní mozkové příhody neboli mrtvice?
- Pamatuješ si metodu zmíněnou ve hře, která se pojí právě s příznaky?
- Co máš dělat, pokud uvidíš někoho s takovými příznaky?
- Co budeš dělat, pokud uvidíš někoho, kdo je nějak zraněný nebo mu je opravdu špatně?
- Naučil jsi se tyhle věci ze hry, nebo jsi už nějaké věděl předtím?

Pomocí jejich pozorování při hraní hry jsem získala informace o tom, co a jak upravit či předělat, aby pro ně byla hra srozumitelná. Většina dětí na poslední otázku odpověděla, že znala telefonní číslo zdravotnické záchranné služby (155), ale ostatní informace pro ně byly nové. Dle průzkumu aplikace naplňuje své cíle a děti se díky ní dozví nové cenné informace. Po odehrání jsem se jich také ptala na jejich pocity ohledně hry, co by upravily, jejich názor na vzhled a ovládání. Dle jejich připomínek byla aplikace také upravena – problémy byly například v příliš pomalé odezvě tlačítek, ne příliš intuitivní přepínání dialogu, malé obrázky u odpovědí.

Dále nám bylo nabídnuto otestování hry v jedné soukromé škole, ale výsledky tohoto testování nedorazily před vytištěním textu bakalářské práce. Aplikace podle zpětné vazby bude samozřejmě upravena, nelze zde ale její výsledky prezentovat.

Vydání aplikace je plánováno buď na 15. 5. (den linky 155) nebo na 1. 6. 2021 (narození pejska Defíka, maskota záchranné zdravotnické služby Jihomoravského kraje). Tato data bylo vybráno lidmi ze ZZSJK – záchranné zdravotnické služby Jihomoravského kraje.

Kapitola 6

Závěr

Cílem této práce bylo vytvořit vzdělávací aplikaci, která by uživatele naučila první pomoc při cévní mozkové příhodě neboli mrtvici. Tento cíl byl splněn.

Před samotným vývojem aplikace jsem nastudovala příznaky mrtvice a první pomoc v takové situaci. Jelikož jsem takovou aplikaci vyvíjela poprvé, bylo nutné se seznámit s možnostmi vývoje her a s operačními systémy pro mobilní telefony. Poté jsem prozkoumala různá vývojová prostředí a seznámila jsem se s vybraným vývojovým prostředím Unity. Jelikož je aplikace zaměřena na děti ve věku 7–10 let (první stupeň základní školy), nastudovala jsem psychologii a vzdělávání dětí, aby aplikace měla kýžený vzdělávací efekt. Veškeré poznatky získané při tomto studiu jsou shrnuty v kapitolách 2 a 3. Následně jsem navrhla strukturu i vzhled aplikace – návrh je popsán v kapitole 4 a výsledný vzhled aplikace je popsán na konci 5. kapitoly.

Implementace aplikace, její testování a ohodnocení uživateli jsou také popsány v kapitole 5. Po veškerých úpravách, které byly provedeny na základě výsledků testování, bylo vytvořeno ukázkové video (to lze nalézt na paměťovém médiu, které je přiloženo k této práci, a odkaz na něj se nachází také v dokumentaci). Pro samotnou implementaci bylo důležité správné uložení dat (otázek a odpovědí) a následná práce s nimi. Díky tomu byla implementace ostatních věcí jednodušší, univerzálnější a lépe čitelná. Také se v aplikaci vyskytovalo méně chyb a případné chyby bylo lehké odhalit a opravit.

Děti, které výslednou aplikaci vyzkoušely, následně dokázaly popsat příznaky mrtvice a věděly, co mají v takovou chvíli dělat – hra tedy splnila svůj účel.

Jelikož se jednalo o aplikaci na zakázku, bylo pevně určeno téma a zaměření aplikace, ale rozhodla jsem se zadání rozšířit o tři minihry, které do samotné hry přidají ozvláštňující prvek a udělají hru více poutavou. Zároveň jsem vytvořila vlastní grafiku, neboť grafik, který ji měl původně vytvářet, objednávku odmítl.

V budoucnu by bylo vhodné aplikaci upravit dle výsledků širšího testování, které ale nyní nejsou známy (testování na menším počtu uživatelů již proběhlo). Další možností je rozšíření aplikace – přidání dalších miniher či různých aktivit zcela mimo hlavní příběh, například ošetřování člověka s mrtvicí před příjezdem sanitky nebo kvíz na otestování právě získaných vědomostí. Dále je možné vytvořit další aplikace podobného charakteru, jen pro jiné zdravotní problémy, či zcela jinou hru s pejskem záchranářem, díky čemuž by se děti více seznámili s tímto maskotem zdravotnické záchranné služby Jihomoravského kraje – tuto možnost jsem osobně konzultovala se zadavatelem aplikace a vytvoření takové hry by uvítali.

Literatura

- [1] BEN-KIKI, O., EVANS, C. a DÖT NET, I. *YAML Ain't Markup Language (YAML™) Version 1.2* [online]. 3. vyd. YAML™, říjen 2009 [cit. 2021-02-27]. Dostupné z: <https://yaml.org/spec/1.2/spec.html>.
- [2] BLAHOVCOVÁ, M. *Emocionalita dětí mladšího školního věku*. Plzeň, CZ, 2015. [cit. 2021-02-27]. Diplomová práce. Západočeská univerzita v Plzni, fakulta pedagogická, katedra psychologie. Dostupné z: <http://hdl.handle.net/11025/19901>.
- [3] BRAY, T. *The JavaScript Object Notation (JSON) Data Interchange Format* [Internet Requests for Comments]. RFC 7159. Reston, Virginia, U.S.: Internet Society, 2014 [cit. 2021-03-12]. Dostupné z: <http://www.rfc-editor.org/rfc/rfc7159.txt>.
- [4] CHOU, Y. *Actionable Gamification: Beyond Points, Badges, and Leaderboards*. 1. vyd. Birmingham, United Kingdom: Packt Publishing Ltd, 2019 [cit. 2021-02-19]. ISBN 9781839210778.
- [5] DUSTIN, K. *The Power of Video Game Engines: Every Game Developer's (Not-So-Secret) Weapon* [online]. St. Petersburg, Russia: Game Designing, 2021 [cit. 2021-03-27]. Dostupné z: <https://www.gamedesigning.org/career/video-game-engines/>.
- [6] EL KASSAS, W. S., ABDULLAH, B. A., YOUSEF, A. H. a WAHBA, A. M. Taxonomy of Cross-Platform Mobile Applications Development Approaches. *Ain Shams Engineering Journal*. 1. vyd. Egypt: [b.n.]. 2017, sv. 8, č. 2, s. 163–190, [cit. 2021-02-16]. DOI: 10.1016/j.asej.2015.08.004. ISSN 2090-4479. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S2090447915001276>.
- [7] FEDEWA, J. *What Is the Google Play Store?* [online]. How to Geek, 2021 [cit. 2021-04-03]. Dostupné z: <https://www.howtogeek.com/709926/what-is-the-google-play-store/>.
- [8] FINGAS, J. *Google Play protects your Android phone against rogue apps* [online]. New York, U.S.: Engadget, 2017 [cit. 2021-04-07]. Dostupné z: https://www.engadget.com/2017-05-17-google-play-protect.html?guce_referrer=aHR0cHM6Ly91bi53aWtpcGVkaWEub3JnLw&guce_referrer_sig=AQAAAMM-k1pXNrcrpbEoHwfXo7CMjZ7fYzhiXLV-1XmgdEltWB3EVvwUtUW9m5_ys6dEmVCwFyCY2TdOfTpKEVuliTE8ARzKpc9HomwM11R3oEif6S6whQ85i5gWc0w-XWnSfkTCeDOEWhNM0G3oeABbEuHQyJAyx5UULc2o-ZuC8heZ.
- [9] FISHER, T. *What Is an APK File?* [online]. New York City, U.S.: Lifewire, 2021 [cit. 2021-04-07]. Dostupné z: <https://www.lifewire.com/apk-file-4152929#:~:text=A%20file%20with%20the%20APK,be%20found%20on%20other%20websites>.

- [10] *Configure your build* [online]. California, United States: Android Developers, 2021 [cit. 2021-04-08]. Dostupné z: <https://developer.android.com/studio/build>.
- [11] GREGORY, J. *Game Engine Architecture*. 1. vyd. Wellesley, Massachusetts: Taylor and Francis Group, 2009 [cit. 2021-03-27]. ISBN 978-1-4398-6526-2.
- [12] LAŠEK, J. *Psychologie vývojová* [studijní opora]. Hradec Králové, Cz: Univerzita Hradec Králové, Pedagogická fakulta, 2015 [cit. 2021-02-27]. Dostupné z: <https://www.uhk.cz/file/edee/pedagogicka-fakulta/pdf/pracoviste-fakulty/katedra-socialni-patologie-a-sociologie/dokumenty/studijni-opory/socialni-komunikace-v-neziskovem-sektoru/psychologie-vyvojova.pdf>.
- [13] LINHART, J. Teorie učení a zvyšování účinnosti vyučování počátečnímu čtení. *Pedagogika: Časopis pro vědy o vzdělávání a výchově* [online]. 1. vyd. Praha, CZ: Univerzita Karlova v Praze – Pedagogická fakulta. 1986, č. 5, [cit. 2021-02-27]. ISSN 2336-2189. Dostupné z: <https://pages.pedf.cuni.cz/pedagogika/?p=4542>.
- [14] NIEMEYER, P. a LEUCK, D. *Learning Java*. 4. vyd. Massachusetts, United States: O'Reilly Media, 2013 [cit. 2021-04-15]. ISBN 978-1-449-31924-3.
- [15] ROSENBERG, J. *Extensible Markup Language (XML) Formats for Representing Resource Lists* [Internet Requests for Comments]. RFC 4826. Reston, Virginia, U.S.: Internet Society, 2007 [cit. 2021-03-12]. Dostupné z: <https://tools.ietf.org/rfc/rfc4826.txt>.
- [16] RŮŽIČKA, E. a KOLEKTIV. *Neurologie*. 1. vyd. Praha, CZ: Triton, 2019 [cit. 2021-02-18]. ISBN 978-80-7553-681-5.
- [17] ŠTÍPEK, P. *Co děti potřebují* [online]. Praha, Cz: Psychologie.cz, 2016 [cit. 2021-02-27]. Dostupné z: <https://psychologie.cz/co-deti-potrebuji/>.
- [18] SURIN, T. *Inspecting APK Files* [online]. Vídeň, Rakousko: PSPDFKIT, 2018 [cit. 2021-04-07]. Dostupné z: <https://pspdfkit.com/blog/2019/inspecting-apk-files/>.
- [19] WOOD, L. a REINERS, T. Gamification. In: KHOSROW-POUR, M., ed. *Encyclopedia of Information Science and Technology*. 3. vyd. Pennsylvania, United States: Information Science Reference, 2015, s. 3039–3047 [cit. 2021-02-19]. DOI: 10.4018/978-1-4666-5888-2.ch297. ISBN 1466658886.