



**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" - NÁHLÁ  
ZÁSTAVA OBĚHU**

MOBILE APPLICATION "PARAMEDIC DOGGY" - UNEXPECTED CARDIAC ARREST

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**NIKOL DUDOVÁ**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. TOMÁŠ CHLUBNA**

BRNO 2021

## Zadání bakalářské práce



Studentka: **Dudová Nikol**  
Program: Informační technologie  
Název: **Mobilní aplikace "Pejsek záchranář" - Náhlá zástava oběhu**  
**Mobile Application "Paramedic doggy" - Unexpected Cardiac Arrest**  
Kategorie: Uživatelská rozhraní

### Zadání:

1. Seznamte se s prostředím pro vývoj multimediálních mobilních aplikací (Unity)
2. Prostudujte problematiku první pomoci při náhlé zástavě oběhu
3. Navrhněte rozhraní a strukturu vzdělávací aplikace pro děti na výše uvedené téma
4. Aplikaci implementujte
5. Proveďte měření a uživatelskou studii hodnotící implementované výsledky
6. Vytvořte video reprezentující výsledky vaší práce

### Literatura:

- Berkman, E., Hooper, S.: Designing Mobile Interfaces, O'Reilly Media, Inc. 2011

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3 zadání

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

Vedoucí práce: **Chlubna Tomáš, Ing.**  
Konzultant: Zemčík Pavel, prof. Dr. 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

Tato bakalářská práce se zabývá návrhem a vývojem mobilní vzdělávací aplikace pro děti od 8 do 11 let. Cílem hry je naučit děti poskytnout první pomoc při náhlé zástavě oběhu u dospělého člověka. Cílem práce je položit teoretické základy pro vývoj mobilních her a teorii první pomoc. Následně jsou vybrané technologie použity prakticky při návrhu a vývoji hry.

## Abstract

This bachelor thesis contains the design and development of a mobile educational application for children from 8 to 11 years. The aim of the game is to teach children to provide first aid in the event of unexpected cardiac arrest of an adult. The aim of this work is to lay the theoretical foundations for the development of mobile games and providing first aid. Subsequently, selected technologies are used in practice when designing and developing the game.

## Klíčová slova

mobilní aplikace, první pomoc, náhlá zástava oběhu, C#, Unity, ZZS JMK

## Keywords

mobile app, first aid, unexpected cardiac arrest, C#, Unity, ZZS JMK

## Citace

DUDOVÁ, Nikol. *Mobilní aplikace "Pejsek záchranář" - Náhlá zástava oběhu*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Tomáš Chlubna

# Mobilní aplikace "Pejsek záchranář" - Náhlá zástava oběhu

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením pana Ing. Tomáše Chlubny. Další informace mi poskytli zástupci Zdravotnické záchranné služby Jiho-moravského kraje. Uvedla jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpala.

.....

Nikol Dudová  
16. května 2021

## Poděkování

Děkuji panu Ing. Tomáši Chlubnovi za podnětné připomínky, motivaci a inspiraci při psaní práce. Děkuji také ZZS JMK za možnost se na tomto projektu podílet. Dále děkuji panu Ing. Františku Svobodovi za korekturu a připomínky k práci. Díky patří také všem dětem, které pomohly hru otestovat. V neposlední řadě děkuji za lásku a podporu své rodině.

# Obsah

<b>1 Úvod</b>	<b>2</b>
<b>2 Teorie pro vývoj mobilních her a výuku první pomoci</b>	<b>3</b>
2.1 Vývoj her . . . . .	3
2.2 Animace . . . . .	5
2.3 C# . . . . .	6
2.4 Multiplatformní mobilní aplikace . . . . .	8
2.5 Canva . . . . .	9
2.6 Existující aplikace pro výuku první pomoci . . . . .	10
2.7 První pomoc . . . . .	11
<b>3 Návrh uživatelského rozhraní</b>	<b>13</b>
3.1 Úvodní obrazovka . . . . .	13
3.2 O aplikaci . . . . .	13
3.3 Úvodní video . . . . .	13
3.4 Kvízové otázky . . . . .	14
3.5 Mini hry . . . . .	14
3.6 Závěrečné video . . . . .	17
3.7 User flow . . . . .	17
<b>4 Implementace</b>	<b>18</b>
4.1 Použité nástroje a technologie . . . . .	18
4.2 Struktura . . . . .	18
4.3 Vyhodnocení úkolu . . . . .	19
4.4 Audio . . . . .	20
4.5 Mini hry . . . . .	22
4.6 Animace . . . . .	22
4.7 Grafika a fonty . . . . .	23
4.8 Výsledná aplikace . . . . .	26
<b>5 Testování aplikace</b>	<b>28</b>
5.1 Unity Remote . . . . .	28
5.2 Debugging pomocí Visual Studio . . . . .	28
5.3 Uživatelské testování . . . . .	29
<b>6 Závěr</b>	<b>31</b>
<b>Literatura</b>	<b>32</b>

# Kapitola 1

## Úvod

Aplikace, která má být výsledkem této práce, by měla přispět k šíření znalostí první pomoci mezi dětmi. Výuka formou hry má zpříjemnit proces učení a sloužit k tomu, aby nabyté znalosti nebyly hned zapomenuty. Její vývoj a faktická správnost je kontrolována Zdravotnickou záchrannou službou Jihomoravského kraje (dále ZZS JMK). Ústředním motivem hry je maskot ZZS JMK, pejsek Defík<sup>1</sup>. Pejška v plyšové podobě dostávají děti, když jsou ošetřované záchranáři. Je to terapeutický prvek, který má zmírnit případné trauma.

Pro aplikaci byla zvolena modelová událost náhlá zástava oběhu u ženy středního věku. Takový člověk může pocítit píchnutí na prsou a následně se sesune na zem. Dojde k přerušení krevního oběhu a často i zástavě dýchání. Je nezbytná pohotová reakce a záchrana se neobejde bez resuscitace zraněného.

Vývoj hry se skládá z několika základních částí. Nezbytné jsou grafické materiály, které tvoří uživatelské rozhraní. Logika uživatelského rozhraní se ovládána pomocí skriptů. Důležitým prvkem jsou také animace, které hru dělají dynamičtější a více zábavnou. Všemi těmito oblastmi se v práci budu zabývat.

Práce má několik důležitých kapitol. V kapitole číslo 2 je uveden přehled dostupných technologií pro vývoj mobilních her a to jak po grafické stránce, tak po programové. Také osvětluje problematiku první pomoci při náhlé zástavě oběhu. Krátce jsou rozebrány i multiplatformní aplikace a dva aktuální mobilní operační systémy, tedy Android a iOS. Dále jsou zmíněny již existující výukové aplikace pro děti a také alternativní technologie k Unity. Následuje popis návrhu hry doplněný o vizualizace, který lze najít v kapitole 3. Veškerý vývoj je popsán v kapitole 4 implementace. Jsou zde zmíněné použité technologie, práce s nimi a specifické implementační detaily, jako například tvorba animací. V poslední, 5. kapitole, je zmíněno testování a to jak pomocí nástrojů pro vývoj, tak testování s cílovými uživateli.

---

<sup>1</sup><https://www.zzsrmk.cz/defik-pejsek-zachranar>

## Kapitola 2

# Teorie pro vývoj mobilních her a výuku první pomoci

V kapitole zabývající se teorií je zdokumentován průzkum technologií, které budou použité k vývoji aplikace. Poslední část kapitoly se zabývá taktéž teorií první pomoci při náhlé zástavě oběhu, jelikož je tato znalost pro práci stěžejní.

### 2.1 Vývoj her

Ústředním pojmem této sekce bude herní engine. Jako hlavní zástupce herních enginů bude zmíněn program Unity. Dále budou uvedeny příklady dalších existujících herních enginů a jejich vlastnosti. Tyto programy pak budou s Unity také porovnány.

#### 2.1.1 Herní engine

Herní engine je software, který slouží k vytváření her. Je postavený na schopnosti oddělit základní softwarové jádro obsahující vykreslování grafických prvků, detekci kolizí a práci s audio prvky, od logické části, která se skládá z pravidel hry, celých hracích světů a ostatních souborů vlastněných vývojářem. Toto oddělení umožňuje znovupoužitelnost zdrojových kódů a dalších komponent hry. Čím pokročilejší funkce engine nabízí, tím rychlejší je vývoj her v něm. Významný aspekt je také cena hry, která zrychlením vývoje klesá. [21] Enginů existuje mnoho a rozsah nabízených funkcí je široký, pro příklad uvedu Quake, Unreal Engine, Unity nebo Godot. Unity se řadí mezi ty pokročilejší, jelikož nabízí engine pro vykreslování, fyziku a zvuky, navíc podporuje skriptování a vícevláknové aplikace. [12]

#### 2.1.2 Unity

Unity je engine pro vytváření her a multimediálních aplikací pro počítače, konzole i mobilní telefony. Tyto hry mohou být jak ve 2D tak ve 3D. Jedná se o velmi intuitivní nástroj lehký pro ovládání. Dostupný je také rozsáhlý manuál s různými tutoriály a dokumentace jeho rozhraní.

Vývoj her je multiplatformní, lze tedy vytvářet hry jak pro Windows a Linux, tak i pro Android a iOS. Při sestavení projektu se jen změní platforma, pro kterou má být hra použita, není třeba mít dva projekty pro každou platformu zvlášť. Další zajímavou věcí je Unity Asset Store, kde jsou ke stažení zdarma nebo za poplatek různé grafické materiály, v Unity nazývané Assety.

Pro účely vývoje pro mobilní telefony Unity nabízí vzdálené debuggování přímo na mobilním telefonu. Stačí si do telefonu stáhnout aplikaci UnityRemote5, v projektu se nastaví typ zařízení pro vzdálené připojení a pokud je aplikace otevřená a spustí se hra, vidíme ji jak v telefonu tak na počítači.

Při vytváření projektu se začíná vytvořením scény, což je jedna obrazovka hry. Uvnitř scény jsou objekty typu *Game Object*, což je základní třída všech entit. Každý objekt má v sobě nadále komponenty, které určují jeho vzhled a vlastnosti, ty mají základní třídu *Component*. *Rect Transform* je komponenta používaná pro prvky grafického uživatelského rozhraní, která obsahuje polohu, rozměry, míru rotace a přiblížení. Pokud ovládá objekt nějaký skript, je také přidán jako komponenta objektu. [26]

Co se týče tvorby skriptů, od verze 2018.2 lze používat výhradně jazyk C#, který bude v práci rozebrán později. Do verze 2018.2 byl podporovaný i takzvaný UnityScript, jehož syntaxe je podobná JavaScriptu. Používan byl zejména pro vývoj webových aplikací. [17]

### 2.1.3 Alternativní technologie

Unity samozřejmě není jediný herní engine, který existuje. V této kapitole bych ráda zmínila možné alternativy, které lze také využít pro vývoj her a vytváření animací. [4]

#### Godot

Godot je multiplatformní herní engine, který vznikl pod záštitou MIT, je vytvořen komunitou a tedy open-source. Lze v něm vyvíjet 2D a 3D hry pro PC, konzole, mobilní telefony i web. Stejně jako Unity, podporuje pro skriptování programovací jazyk C#. Navíc lze skripty psát i ve skriptovacím jazyce GDScript, jehož syntaxe se podobá jazyku Python. Godot podporuje i projekty, které v sobě obsahují soubory napsané v obou zmíněných jazycích. Pro vykreslování je používáno OpenGL E.S. 2.0 a pro psaní shaderů zjednodušený jazyk podobný jazyku GLSL. [2]

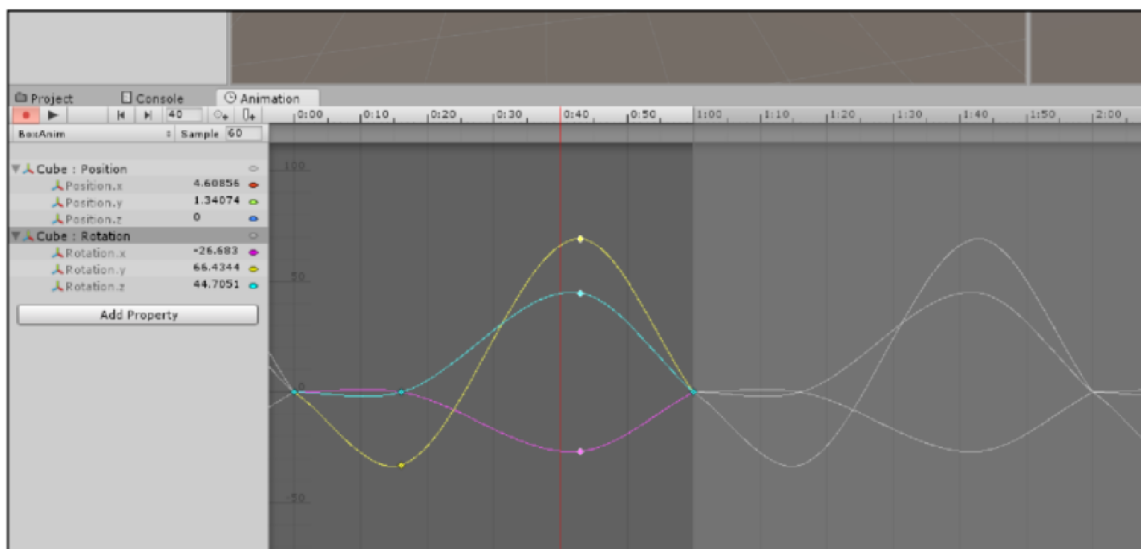
#### Unreal Engine

Unreal engine je další herní engine konkurující Unity. Vyvíjí ho softwarová společnost Epic Games, která se v poslední době proslavila především hrou Fortnite, právě ta byla vyvíjena v Unreal engine. Jedná se ale opět o open-source program, je tedy dostupný zcela zdarma. Na rozdíl od Unity není vhodným nástrojem pro vývoj her na mobilní telefony, i když umožňuje vývoj pro Android a iOS. Unreal engine je uzpůsoben spíše pro vývoj her na PC a konzole. Základní kód engine je napsán v C++, který podporuje i pro skriptování, a zbytek tvoří kód napsaný pomocí UnrealScript. Stejně jako Unity jej lze propojit s programem Visual Studio. [13]

#### Blender

Velmi rozšířeným programem zejména pro vytváření animací je otevřený program Blender, kromě toho slouží pro modelování a vykreslování 3D a 2D grafiky. Součástí je i Blender Game Engine, ve kterém lze vyvíjet hry. Využívá aplikační rozhraní OpenGL a je vyvíjen pod licenci GNU GPL. Zdarma je jak pro osobní tak komerční účely. [18]





Obrázek 2.1: Ukázka klíčových oken při animaci tuhého tělesa, obrázek převzatý z knihy [8]

## 2.2 Animace

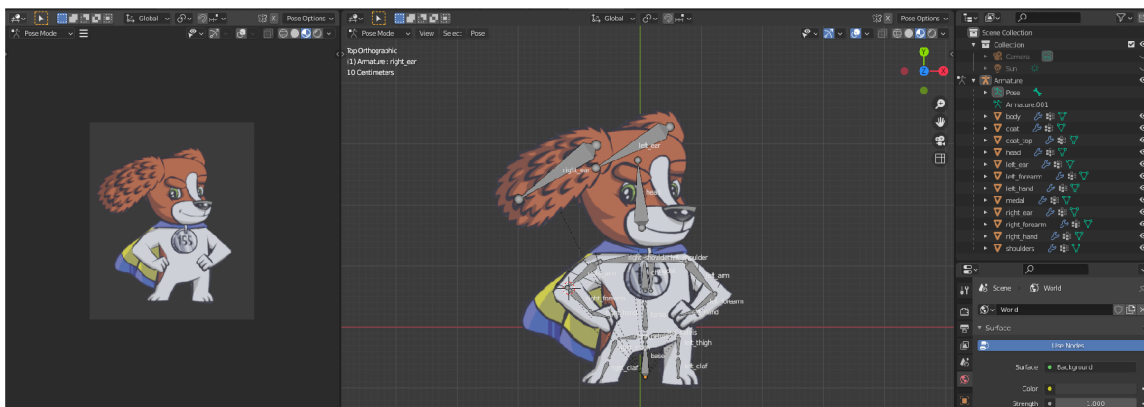
Animace je proměna objektu v čase, čas je rozdělen do časových oken (frames), se kterými se následně pracuje. Změny objektů při animacích probíhají v rámci klíčových časových oken (key frames), přechody mezi těmito okny jsou v Unity generovány automaticky. Doplnkovým oknům se říká tweens a matematicky se nazývá proces jejich generování interpolace. Z technického hlediska lze definovat několik typů animací, v Unity se s každým zmíněným typem pracuje pomocí jiných nástrojů, proto je nutné si animaci dobře promyslet a zvolit vhodné nástroje pro její vytváření.

První typ je tuhá animace tělesa (rigid body), při tomto typu animace se změny vlastností objektu vztahují k objektu jako celku, nijak jej nedělíme a neměníme jeho strukturu, změny nastávají pouze u proměnných na nejvyšší vrstvě objektu, například animace otevření a zavření dveří. Jak vypadá graf animace klíčových oken je vidět na obrázku 2.1.

Dalším typem je animace pomocí kostí (bone-based), objekt je rozdělen podle vnitřní geometrie na části, které se animují každá zvlášť. Je třeba nastavit jim váhu, aby animace vypadala věrohodně. Tento typ používáme typicky pro animování postav ve hrách, lepší podporu pro bone-based animace nabízí Blender, zejména pokud se jedná o 3D animace pomocí kostí. Jak vypadá obrázek pejska Defika při animaci pomocí kostí je ukázáno na obrázku 2.2.

Sprite animace jsou animace pomocí sekvence obrázků, pohyb je tedy rozkreslen do několika obrázků, které mají jasně dané pořadí. Lze je využívat u 2D animací postav nebo třeba grafického uživatelského rozhraní. Na obrázku 2.3 je sekvence pohybů pro tento typ animace.

Animace založené na fyzice jsou využívány pro nepředvídatelné situace, objekt má zkrátka takové vlastnosti, aby se choval v reálném čase co nejvíce realisticky, například pokud jde postava po nějaké římsě, která najednou skončí, spadne vlivem gravitace dolů na zem. Aby se tak stalo, musí se objektu nastavit vhodná váha, jinak by padal moc rychle nebo naopak moc pomalu.



Obrázek 2.2: Ukázka obrázku připraveného pro bone-based animaci v programu Blender



Obrázek 2.3: Sekvence obrázků pro sprite animaci

Animace proměny vzhledu objektu jsou pro výpočet velmi náročné, avšak výsledkem jsou jedny z nejvíce realistických animací. Objekt totiž obalíme geometrickou sítí podle jeho povrchu (typicky různě velké trojúhelníky) a při proměně vypočítáváme nový vztah dle hodnot vrcholů. Využívají se zde opět automaticky počítané doplňky časových oken, tedy přechody mezi jednou hodnotou klíčového časového okna na druhou hodnotu. Příkladem takové animace je třeba proměna člověka ve vlkodlaka ve fantasy hře. [8]

## 2.3 C#

C# patří mezi multiparadigmatické vysokoúrovňové objektově orientované programovací jazyky. Jeho autorem je společnost Microsoft a syntakticky vychází z jazyků C++ a Java. Patří tedy mezi jazyky vycházející z programovacího jazyka C. Nejnovější verze je C# 9.0, ke které patří cílový framework .NET 5. [23]

### 2.3.1 Objektově orientovaný jazyk

Vzhledem k tomu, že se jedná o objektově orientovaný jazyk, v implementaci se vyskytují tři základní pilíře tohoto paradigmatu, to jsou zapouzdření, dědičnost a polymorfismus. Na rozdíl od C++ je zde podporována pouze jednoduchá dědičnost, tedy že synovská třída nemůže mít více než jednoho rodiče.

Rozlišuje se několik typů modifikátorů přístupu - public, protected, internal a private. A na základě nich pak lze nastavit šest úrovní dostupnosti. Výchozím modifikátorem je private.

Nejedná se ale o čistě objektově orientovaný jazyk, jelikož v něm najdeme kromě typu třída i další - například rozhraní. Rozdíl mezi třídou a rozhraním je v tom, že rozhraní pouze popisuje třídní členy, implementaci těchto členů ale najdeme až ve třídě, která dané rozhraní implementuje. Tímto způsobem se dá v C# částečně obejít podmínka jednoduché dědičnosti. Další věc, kterou se liší od čistě objektově orientovaných jazyků, jsou třídní členy, mezi které se řadí nejčastěji metody, vlastnosti a události, ale existují i další méně používané.

Dále v C# neexistují globální proměnné ani metody. Lze ale používat proměnné veřejných tříd a statické metody. Také zde nenarazíme na dopřednou deklaraci, to znamená, že nezáleží na pořadí deklarace metod. V neposlední řadě se jedná o jazyk, který je tzv. case sensitive, tedy rozlišuje mezi velkými a malými písmeny ve všech identifikátorech. [19]

### 2.3.2 Common Type System

Proměnné jsou podle dat, které v sobě uchovávají, rozdělené na mnoho typů, ty mají dvě základní skupiny, hodnotové a referenční. Hodnotové datové typy jsou alokované na zásobníku, zatímco referenční jsou alokované na haldě. Unifikovaný systém typů (Common Type System) ale zajišťuje, že všechny tyto typy sdílí базový typ *Object*. Každý typ například podporuje zavolání metody *ToString* a tedy zkonvertování obsahu na řetězec. Toto ve výsledku ulehčuje práci s datovými typy a jejich konverze. [9]

### 2.3.3 Events & EventHandlers

Event je událost, která nastává na nějakém objektu. .NET framework definoval standardní model pro psaní takových událostí, jehož účelem je zachovat konzistenci mezi frameworkem a uživatelským kódem. Jádrem tohoto modelu je базová třída *System.EventArgs*, kterou události buď přímo využívají nebo si lze definovat vlastní třídu argumentů události, která z této базové musí dědit.

EventHandler je delegát reprezentující danou událost. Framework opět definuje generický delegát *System.EventHandler<T>*. Každý delegát události musí mít návratovou hodnotu *void* a přijímá zpravidla dva argumenty, první je typ objektu a druhým je třída odvozená právě od zmíněné třídy *System.EventArgs*. Typ objektu udává, na jakém objektu událost vzniká, který objekt ji bude posílat. Pomocí druhého parametru jsou předána data vztahující se k události, může se jednat například o polohu kurzoru v době kliknutí myši nebo text napsaný do textového pole.

Událost může být vyvolána z metody, která má vlastnosti *protected virtual* a její jméno začíná "On" (např. *OnClick*). Metoda také musí přijímat argument *EventArgs*. [9] Celý předpis metody včetně těla pak může vypadat třeba takto:

```
protected virtual void OnClick(MouseEventArgs e)
{
    Clicked?.Invoke(this, e);
}
```

### 2.3.4 Instance a statické objekty

Výchozím stavem v C# jsou objekty operující pomocí instancí daného typu. Statické objekty jsou takové objekty, které operují s typem přímo a nevytváří se jejich instance. Příkladem statické třídy je *Console*, v tomto případě nedává smysl vytvářet od objektu instance,

konzole je vždy jen jedna. Třída ale nemusí být statická a přesto může obsahovat nějaký statický člen.

Instance se používají v situaci, kdy existuje více objektů stejného typu, které ale mají různé vlastnosti. Statické třídy nebo členy jsou sdílené v celé aplikaci a obsahují tedy v každém případě stejné hodnoty, i když jsou volané z jiných míst aplikace. [9]

### 2.3.5 LINQ a Lambda výrazy

Další zajímavé funkce v tomto jazyce jsou například LINQ (Language Integrated Query) nebo lambda výrazy. LINQ je integrovaný dotazovací jazyk, který je součástí C# od verze 3.0. Ulehčuje programátorům práci s kolekcemi, databází a XML soubory. Pomocí něj lze třídit, vyhledávat a dotazovat se přímo v kódu.

Lambda výraz je forma zápisu, která může sloužit jako náhrada za anonymní funkce, obvykle zjednodušuje zapsání operace o jednom výrazu. Vytváří se podle daných předpisů a každý tento výraz lze převést na typ delegáta nebo strom výrazů. [7]

## Visual Studio

Tradiční prostředí pro vývoj v jazyce C# se nazývá Visual Studio, jedná se o velmi pokročilý nástroj, který lze po nainstalování příslušných rozšíření propojit s Unity. Poté když se v Unity přidá nějaký skript, je automaticky otevřen ve Visual Studiu. Takto lze přidávat breakpointy a díky nim odhalovat chyby ve skriptech při současném spuštění aplikace v Unity. [11]

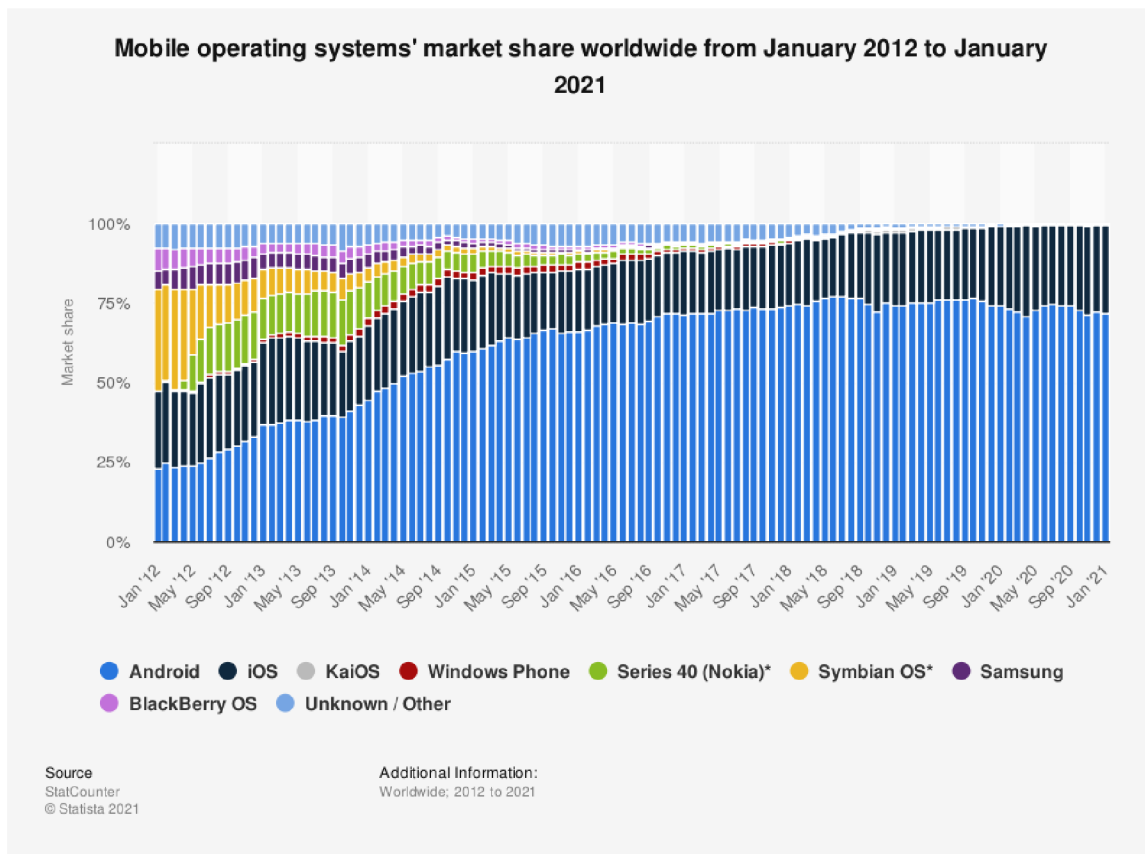
## 2.4 Multiplatformní mobilní aplikace

Multiplatformní mobilní aplikace jsou takové aplikace, které běží na více než jedné platformě. Unity má dobrou podporu pro tvorbu takových aplikací, při vytváření programu je sice potřeba vybrat, jestli chceme tu nebo onu platformu, ale toto je především kvůli debugování aplikace. Kód se napíše jednou a vygenerování výsledné hry se poté provede pro každou platformu zvlášť.

### Android

Operační systém Android byl poprvé vydán v roce 2008, je vyvíjen firmou Google jako otevřený software. Výrobci mobilních telefonů po celém světě jej přebírají a doplňují o jakousi nadstavbu, která je odlišuje od jiných výrobců. Zde ale vzniká jedna z nevýhod systému a tím jsou pozdní nebo žádné aktualizace. Čistý Android je aktualizován pravidelně, avšak až po vydání aktualizace mohou firmy synchronizovat jejich privátní změny v systému a poté teprve aktualizaci doručit svým uživatelům. V důsledku své otevřenosti má Android další problém, kterým je častější výskyt malware. [15] Jsou také známé důkazy o chybách v platformě, které narušují soukromí uživatelů. I přesto je ale Android nejrozšířenější z operačních systémů. [22]

Aplikace pro Android jsou dostupné v Google Play obchodě. Dostat aplikaci do Google Play je vcelku jednoduché, ne všechny aplikace jsou ověřené a může se stát, že stažená aplikace bude škodlivá. Obchod obsahuje velké množství aplikací zdarma, a samozřejmě i aplikace placené. Dále jsou tam také knihy a filmy. Uživatelské prostředí obchodu je hodnoceno velmi pozitivně a lidé si chválí i široký záběr při vyhledávání. [3]



Obrázek 2.4: Rozšíření mobilních operačních systémů od roku 2012 do roku 2021 [25]

## iOS

Firma Apple vydala operační systém pro jejich telefony poprvé v roce 2007, v té době se jmenoval iPhone OS, později byl název zkrácen na iOS. Na rozdíl od Androidu jej používá výhradně ve svých zařízeních, pravidelně vydávané aktualizace jdou rovnou k uživatelům a software telefonu by tak měl být bezpečnější a líp odladěný.

Obchod s aplikacemi pro iOS se nazývá App Store. Nachází se tam ale pouze hry a aplikace, žádné knihy ani filmy, pro ty má Apple jiné obchody. Aplikací v App Store je méně než v Google Play, jsou poctivě vybírány a testovány. Mnoho z nich je placených, ale samozřejmě najdeme i hry a aplikace, které jsou zcela zdarma.

Je zjevné, že cílové skupiny pro Android a iOS se liší. V dnešní době tvoří uživatelé Android a iOS telefonů 99% všech uživatelů mobilních telefonů, viz. obrázek 2.4. Konkrétně Android používá přes 72% z nich a ve více jak 26 procentech případů se jedná o zařízení s iOS.

## 2.5 Canva

Canva<sup>1</sup> je webový grafický editor pro vytváření grafiky různých rozměrů a účelů. Lze v ní vytvořit vše od vizitky přes prezentaci až po video. Pro tvoření jsou k dispozici různorodé fonty, pozadí ve formě fotek či ilustrací, spousta animovaných prvků. Umožňuje také na-

<sup>1</sup><https://www.canva.com/>

hrát vlastní obrázky. Velmi dobrou funkcí je odstranění pozadí fotky, následně lze využít možnost stáhnout návrh s průhledným pozadím, a tak mít čistý obrázek bez jakéhokoliv šumu v pozadí, který by mohl pro další práci vadit. [16]

Další zajímavou funkcí jsou šablony, které Canva nabízí. Stačí jen zadat téma, kterého se grafický návrh má týkat a zobrazí se nabídka mnoha zpracovaných grafických materiálů, které lze upravovat dle libosti. Je možné změnit barevné schéma, upravit text, smazat nebo naopak přidat prvky.

Pokud je člověk ve vytváření grafických návrhů nováček, v sekci Learn nabízí také několik tutoriálů. Ty se týkají fontů, pozadí, základů designu, obrázků, barev, ale i komplexnějších témat jako je branding nebo sociální síť. [10]

Alternativou k online grafickému editoru Canva je Crello, jejich funkce jsou téměř totožné. Canva má ale mnohem více funkcí dostupných již pro uživatele, kteří neplatí žádný poplatek. Co se týče prémiové verze, opět je rozsahem funkcí lepší Canva, také má větší databázi obrázků, šablon a dalších grafických prvků. Pro animování grafického obsahu má ale rozsáhlejší podporu Crello. [1]

## 2.6 Existující aplikace pro výuku první pomoci

V této kapitole bude zmíněno několik existujících aplikací, které jsou pro děti a mají sloužit k výuce první pomoci. V obchodě pro iOS App Store jsou dostupné dvě aplikace určené dětem, ty stejné jsou i v obchodu pro Android Google Play. První z nich je od Českého červeného kříže a jmenuje se První pomoc. Autorem druhé je asociace Záchranný kruh a jmenuje se První pomoc FREE. Existují od nich totiž ještě další aplikace k výuce první pomoci pro děti, ale ty už jsou placené.

### Aplikace První pomoc od Českého červeného kříže

První pomoc od ČČK je komplexní příručka první pomoci do telefonu. Lze zde najít postupy pro správné poskytnutí první pomoci při různých zdravotních komplikacích jako je alergická reakce nebo zlomenina. Další záložkou jsou situace a jak se na ně připravit, najdeme zde sekci bezpečnost u vody nebo dopravní nehoda. V záložce v nouzi najdeme případy urgentní pomoci, například je zde krvácení nebo dušení. V neposlední řadě si člověk může svoje znalosti otestovat, testů zde najdeme celkem 9 a podle nich můžeme také zhodnotit, které znalosti si máme rozšířit.

Celkově je aplikace vytvořena sympaticky, avšak není zaměřená na děti, nýbrž na dospělé. Může tedy sloužit jako studijní materiál pro lidi, kteří budou děti vzdělávat, ale nikoliv pro děti samotné.

### První pomoc od asociace Záchranný kruh

První pomoc FREE je aplikace zaměřená na děti, je plná obrázků a nejedná se o příručku první pomoci. Lekce jsou spíše pohádky s kreslenými animacemi, ve webové aplikaci jsou doprovázeny i textem. Většina informací v mobilní aplikaci je předávána zvukem a ne textem, proto je i pro děti, které neumí číst a psát.

Plná verze této aplikace se nachází na webu alik.cz. Vytvořila ji, stejně jako mobilní aplikaci, asociace Záchranný kruh pod záštitou generálního ředitele Hasičského záchranného sboru a Ministerstva školství, mládeže a tělovýchovy. Na začátku je možné si vybrat, které téma chceme slyšet a poté se již spustí vybraná lekce. Aplikací provází bernardýn

Benny, který vysvětluje situaci odehrávající se v pozadí. Na závěr některých lekcí je test nebo možnost si situaci zopakovat při jiné události. Například po shlédnutí lekce co dělat u autonehody je nabídnuta možnost zopakování první pomoci na příkladu úrazu na sjezdovce. [6]

## 2.7 První pomoc

Tato kapitola se zabývá situací náhlé zástavy oběhu a poskytnutím první pomoci člověku v tomto stavu. Správná první pomoc v této situaci se skládá z tří hlavních částí - zjištění stavu pacienta, volání na nouzovou linku a resuscitace.

### Náhlá zástava oběhu

Náhlá zástava oběhu je stav, kdy člověku z nějakého důvodu přestane fungovat krevní oběh v těle. Z 80 % jsou příčinou srdeční onemocnění - infarkt myokardu, srdeční selhání, plicní embolie, vada srdečních chlopní. Jedná se o jednu z nejvýznamnějších příčin úmrtí. [24] Odhaduje se, že na ni zemře 60 až 100 případů na 100 000 obyvatel každý rok. Prognóza pacientů není nijak povzbudivá, z pacientů, kteří jsou oživováni záchrannou službou, přežívá asi 5-15 %. [14] I pomoc laika ale výrazně zvyšuje pacientovu šanci na přežití. Klíčová je právě zmíněná resuscitace, která musí začít co nejdříve a způsob jejího provedení je také zásadní.

### Postup poskytnutí první pomoci

Pokud je osoba nalezena v bezvědomí, je třeba ji nejprve oslovit, pokud nereaguje, přistoupit blíž a zkusit bolestivý podnět, například štípnutí do ucha. Jestli člověk opět nezareaguje, následuje zjištění životních funkcí, tedy jestli dýchá a má tep. Dýchání lze zjistit pomocí zrcátka nebo bočním pohledem na hrudník. S tepem je to náročnější, lze zjistit přiložením dvou prstů na krkavici, ale může se stát, že záchránce ucítí svůj tep. Další krok je zavolat na číslo 155 a říct všechny podstatné informace. Volání na číslo 155 je důležité z toho důvodu, že při vytočení čísla 112 se v Česku člověk dovolá na dispečink hasičů, dispečer sice záchránce pomůže s péčí o zraněného, ale přepojování a předávání informací dispečerce pro záchranku, která až následně zalarmuje posádku sanitky, je zdoluhavé a v situaci, kdy člověk bojuje o život, se počítá každá sekunda. Správný postup při komunikaci s dispečerem by měl být následující:

1. Představit se
2. Co se stalo
3. Kde se to stalo a jak se tam pomoc může dostat
4. Jaké komplikace má pacient
5. Poté čekat na otázky operátora a nikdy první nezavěšovat telefon

Ve většině případů zůstává operátor na příjmu a navrhuje záchránce jak dále postupovat. Cílem aplikace ale je, aby se děti naučily být co nejvíce samostatné. Dalším krokem je zpravidla resuscitace. Její pravidla jsou:

1. Stlačovat uprostřed hrudní kosti mezi bradavkami

2. Hloubka stlačení by měla být 4-5 cm
3. Poměr stlačení a vdechů je 30:2
4. Frekvence stlačování by měla být 100-120/minutu

Resuscitace probíhá do úplného vyčerpání zachránce nebo do příjezdu záchranářů. [20]



## Kapitola 3

# Návrh uživatelského rozhraní

Tato kapitola se bude týkat návrhu výsledné aplikace. Scénář aplikace dodali lidé ze ZZS JMK, stejně jako seznam kvízových otázek. Dle těchto informací jsem připravila návrh aplikace, která se bude dělit na 5 hlavních částí. V každé části bude jak slovní popis obrazovky, tak její grafický návrh. Při návrhu byly použity pravidla UX a pravidla pro správný návrh grafického uživatelského rozhraní pro mobilní aplikace. [5]

### 3.1 Úvodní obrazovka

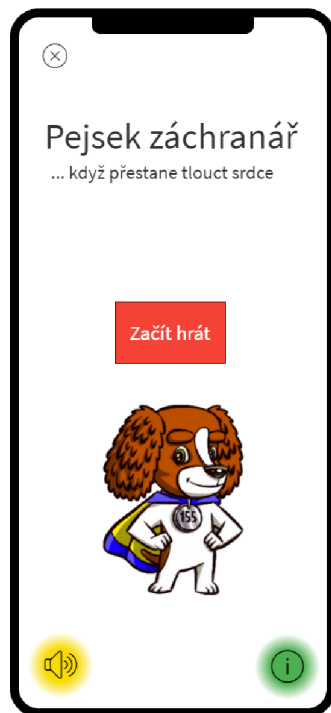
Aplikace začíná úvodní obrazovkou, na pozadí je ústřední motiv, který se vyskytuje ve všech scénách aplikace, jedná se o obrázek parku se dvěma lampami a lavičkou, v pozadí jsou stromy a vysoké budovy města. V levém horním rohu je křížek pro rychlé opuštění a ukončení aplikace, v levém dolním rohu se nachází tlačítko pro spuštění zvuku, v pravém dolním rohu je pak tlačítko, které vede na obrazovku s informacemi o aplikaci. Uprostřed scény nahoře je název aplikace, v samém středu se nachází velké červené tlačítko s nápisem start, pod kterým se nachází postava pejska, které vlaje ve větru hrdinský plášť. Název aplikace je Pejsek záchranář, podnadpis zní "...když přestane tlouct srdce", což je jen trochu poetičtěji nazvaná událost náhlé zástavy oběhu. Vizualizace scény na obrázku 3.1a obsahuje všechny prvky kromě pozadí.

### 3.2 O aplikaci

Z úvodní scény se kliknutím na ikonu s písmenem "i" dostaneme na obrazovku s informacemi o aplikaci. Zde bude uveden informační text. Nebude chybět ani verze aplikace. Důležitou součástí této obrazovky bude také seznam licencí pro použitou grafiku. Obsah doplní také logo Fakulty informačních technologií a logo ZZS JMK. Křížkem se uživatel dostane zpět na úvod aplikace. Náhled obrazovky je na obrázku 3.1b.

### 3.3 Úvodní video

Jelikož bude video pro uvedení do situace na šířku, je potřeba telefon otočit. Nejdříve se tedy zobrazí výzva pro otočení - na obrázku 3.2, buď na ni člověk zareaguje a telefon otočí, nebo se video začne přehrávat až po uplynutí odpočítávání v dolní části, které bude nastaveno na 10 sekund. Video by se mělo odehrávat v parku, kde se udělá špatně paní středního věku



(a) Návrh rozložení úvodní obrazovky



(b) Návrh scény s informacemi o aplikaci

Obrázek 3.1: Návrh úvodní scény a obrazovky informace o aplikaci

a upadne na zem. Paní nereaguje a nedýchá. Na pomoc příběhne pejsek záchranář, potom se hra přepne na další scénu.

### 3.4 Kvízové otázky

Jedna z částí výuky budou kvízové otázky. Celkem budou otázky tři, znění otázky bude nahoře a následovat budou možnosti odpovědi, vždy bude na výběr ze tří možností A, B a C. Původní verze byla, aby po kliknutí na odpověď byla otázka rovnou vyhodnocená, ale nakonec byla zvolena verze s potvrzujícím tlačítkem, aby měl uživatel možnost odpověď ještě změnit než mu bude prozrazeno, že odpověděl dobře nebo špatně. Po vyhodnocení se zobrazí animace pejska, v případě správné odpovědi bude vrtět ocáskem a ukazovat palec nahoru, následně se ikona vyhodnocovacího tlačítka změní na šipku vpravo pro přechod na další scénu. Pokud odpověď bude špatná, vyskočí animace pejska kroutícího rukou se vztyčeným ukazováčkem, tlačítko pro kontrolu výsledků se nyní změní na šipku ve směru kolečka, která znamená opakovat pokus. Náhled rozložení obrazovky při scéně kvízové otázky je na obrázku 3.3

### 3.5 Mini hry

Aby aplikace nebyla jen odpovídáním na otázky a byla pro děti zábavnější, vložila jsem do ní dvě hry. V těchto hrách si dítě nacvičí čísla složek záchranného systému a určí důležité informace, které by měly říct při volání na záchranný systém.



Obrázek 3.2: Návrh obrazovky s výzvou k otočení telefonu

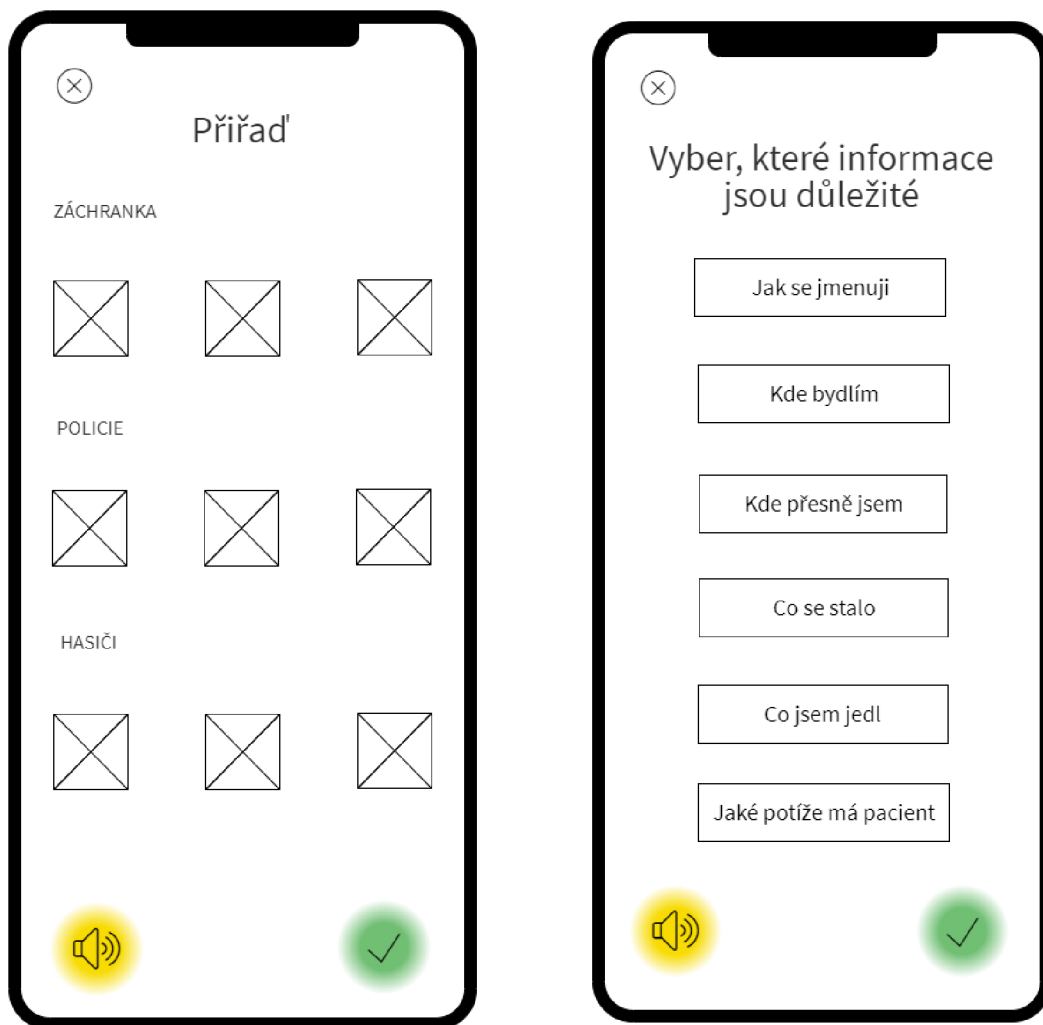


(a) Návrh vzhledu první kvízové otázky

(b) Návrh vzhledu druhé kvízové otázky

(c) Návrh vzhledu třetí kvízové otázky

Obrázek 3.3: Návrh kvízových otázek



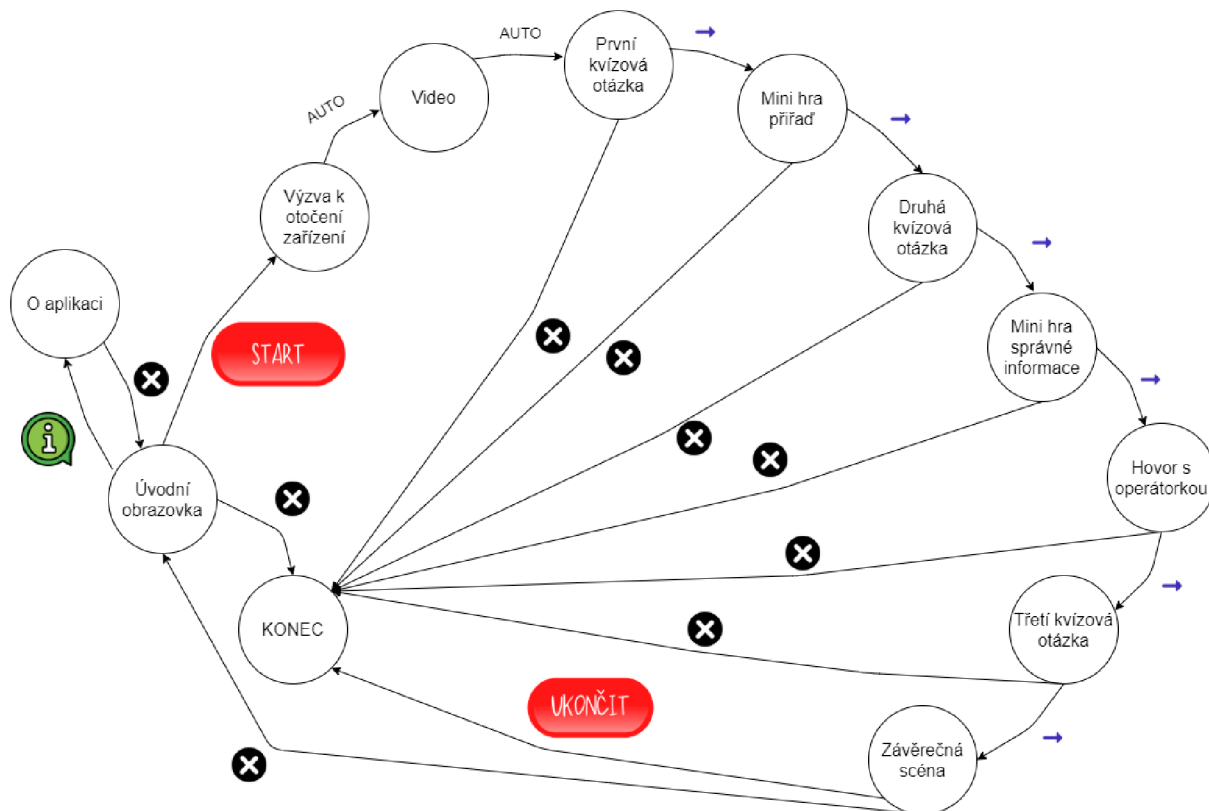
(a) Rozložení mini hry přiřad správně

(b) Rozložení mini hry vyber důležité informace

Obrázek 3.4: Návrh mini her

První hrou bude spojování, dítě má přiřadit ke každé ze složek záchranného systému pomůcku pro zapamatování čísla, telefonní číslo a auto, kterým dojedou na pomoc. V této hře se má uživatel naučit jaké číslo má která složka. Pro některé děti je zapamatování čísla těžké, proto je zde i nápověda, která by jim v přiřazení měla pomoci. Aby si vytvořili úplnou asociaci, přidala jsem ještě obrázky aut, které by měly být nejjednodušší přiřadit.

Druhá hra navazuje na vytočení čísla. Dítě musí v seznamu informací vybrat, které jsou ty správné a mělo by je sdělit operátorce při volání na nouzové telefonní číslo. Pokud dítě ví, že jsou tyto informace třeba sdělit, může si na ně předem připravit odpovědi a hovor tak bude rychlejší. Nestačí jen volat, že se tady něco stalo. Je nutné vědět přesné místo, jak se k tomu místu záchranáři dostanou. Musí mít v době volání přehled o počtu zraněných a jejich komplikacích. Obrázek 3.4 ukazuje rozložení prvků při návrhu mini her.



Obrázek 3.5: Graf možností průchodu uživatele aplikací

### 3.6 Závěrečné video

V závěrečné scéně by měl pes poděkovat uživateli za skvělou pomoc při záchraně života. Na obrazovce by se také měla objevit sanitka. Následně by měl vyskočit dialog, jestli se chce uživatel vrátit na domovskou obrazovku nebo ukončit aplikaci.

### 3.7 User flow

Průchod uživatele aplikací je navrhnut tak, že aplikace jde ukončit z úvodní nebo závěrečné scény. Z téměř všech ostatních scén ve hře se po stisknutí křížku uživatel nejprve dostane na úvodní obrazovku a až pak může aplikaci ukončit. Pokud uživatel neukončí hru křížkem, je nastavena pevná hierarchie po sobě jdoucích scén, která začíná stisknutím tlačítka start na úvodní obrazovce. Přejchod, který nevyžaduje uživatelskou akci je v diagramu 3.5 označen jako "AUTO". Ostatní přechody mají u hrany vždy tlačítka, pomocí kterých je přechod proveden.

## Kapitola 4

# Implementace

V kapitole implementace bude důkladně rozebrán vývoj aplikace, použité technologie a zmíněny zajímavé implementační detaily. Licence pro Visual Studio, Canvu a Icons8 jsem měla dostupné díky Github Student Developer Pack, o který je možné si zažádat po zadání čísla ISIC karty. Pro každý z dostupných software jsou pravidla trochu jiná, pro některý je profesionální licence na půl roku a pro jiný na rok.

### 4.1 Použité nástroje a technologie

Při implementaci jsem použila pro vytvoření hry nástroj Unity a jeho přidružený Unity Hub, grafický editor Canva pro vytvoření grafických podkladů - tlačítka, obrázky, texty. Skripty jsou napsané v jazyce C# pomocí vývojového prostředí Visual Studio 2019. Wireframy aplikace jsem kreslila pomocí nástroje Mockflow<sup>1</sup>. Zvukové nahrávky od ZZS JMK jsem rozstříhala pomocí online nástroje MP3cut<sup>2</sup>.

### 4.2 Struktura

V této části bude objasněna struktura projektu se zaměřením na adresář Assets a dále se definuje obecná struktura scény, ve které jsou pro vytváření logických skupin objektů používány prázdné objekty.

#### 4.2.1 Struktura projektu

Projekt v Unity má základní složku Assets, ve které se nachází všechna data vytvořená vývojářem. Tato složka má pět mnou vytvořených podadresářů podle typu dat, které obsahují. Každá naimportovaná položka má k sobě partnera v podobě meta datového souboru. V těchto meta souborech jsou uložena všechna data o daném objektu v Unity.

Složka Animations obsahuje animace včetně animátorů objektů. To jsou kontrolery sloužící pro správu animací daného objektu. Pomocí animátoru se například nastavuje pořadí průběhu animací a jejich spouštěče.

Všechny zvukové nahrávky, které se v aplikaci objevují, jsou uloženy v adresáři Audio.

Asi nejvíce obsáhlá je složka Graphics, která obsahuje úplně všechny obrázky, které se v aplikaci objevují. A nejen to. Jelikož jsem si vybrala font Jenthill Caps, který byl dostupný

---

<sup>1</sup><https://wireframepro.mockflow.com/>

<sup>2</sup><https://mp3cut.net/>

jen v Canvě, všechny texty jsou zde též a do aplikace jsou zpravidla naimportovány jako obrázky.

Ve složce Scenes jsou uloženy všechny scény hry, kterých je celkem 11. Správu scén má na starost balíček *UnityEngine.SceneManagement*. Ten je potřeba použít ve skriptu, pokud se v něm pracuje se scénou, například se scénou načítá nebo se v ní hledá nějaký Game Object.

Poslední mnou vytvořenou složkou je Scripts. Zde jsou všechny skripty použité v projektu. Pokud jsou ve scéně použité společné objekty jako animace nebo tlačítka, i skript k nim je použit ten stejný.

### 4.2.2 Struktura scény

V každé scéně jsem se snažila dodržovat podobnou strukturu, aby byla orientace rychlejší a přehlednější. Kořenovými prvky každé scény jsou hlavní kamera a systém událostí. Kamera v sobě obsahuje vždy canvas. Ten slouží pro zobrazení prvků uživatelského rozhraní, jako výplň canvasu používám panel, který má přiřazeno pozadí. Pozadí je stejné ve všech scénách kromě té s výzvou k otočení telefonu. Prvky canvasu pak rozdělují na pozadí a popředí pomocí prázdných game objects. Součástí těch už jsou texty, obrázky, tlačítka atd.

Tato struktura nemusí být všude dodržena úplně přesně, ale takto jsem si ji na začátku nastavila. Například pokud v popředí je větší skupina stejných objektů, i ty jsou ještě zařazeny ve vlastní složce.

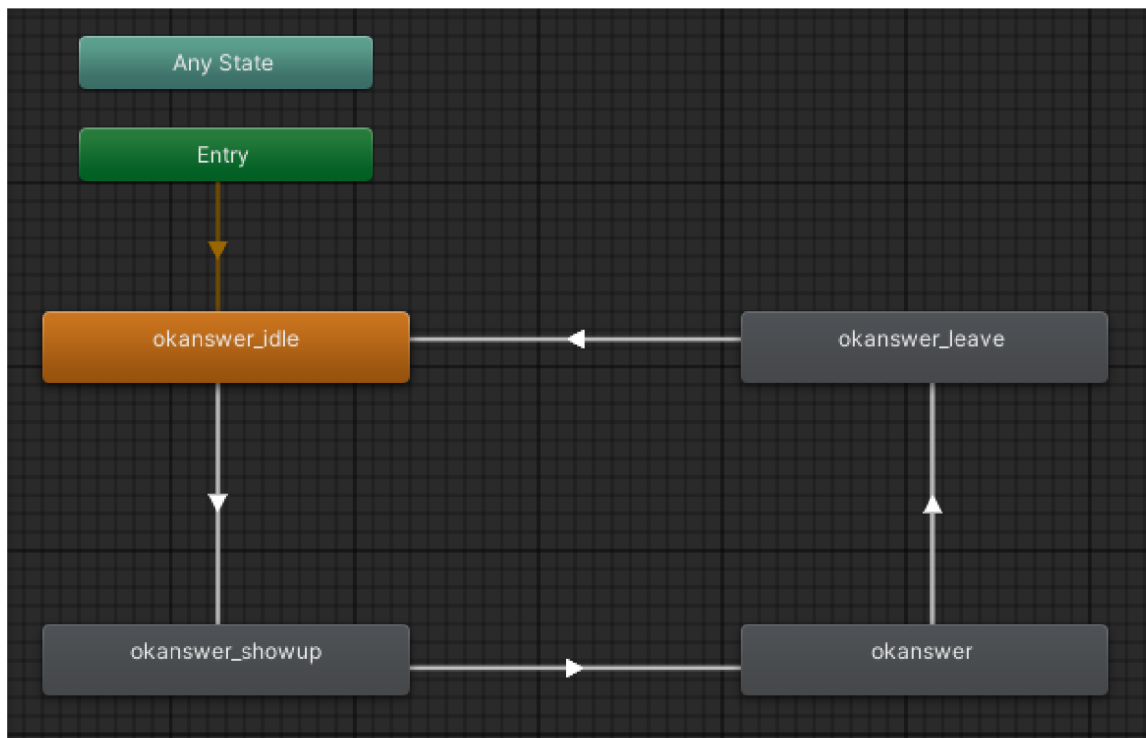
## 4.3 Vyhodnocení úkolu

Ať už se jedná o kvízovou otázku nebo mini hru, dítě vždy dostane zpětnou vazbu na svoji odpověď. Vyhodnocení probíhá pomocí animace pejska. Ten buď ukáže gesto, že je odpověď správná a u toho řekne "Výborně!", nebo je odpověď špatná, pejsek ukáže gesto nesouhlasu a řekne buď "To není dobře", nebo "Zkus to znovu".

Tento pop up je složen z několika částí. V první řadě musí být součástí scény obrázek pejska pro oba dva případy. Objekt pejska má v sobě navíc komponentu animátoru a skript, který jej řídí. Animace obrázku jsou celkem čtyři. První je stav, kdy je obrázek mimo scénu, schovaný. Následuje animace objevení obrázku a jeho postupného přiblížení. Přejít mezi prvním a druhým stavem je spuštěn triggerem *OkAnswerTrigger*. Pak následuje hlavní animace, tedy gesto pejska. Buď ukazuje palec nahoru a vrtí ocáskem a nebo má ruku vpřed, ukazuje prstem a kroutí zápěstím. Tato animace jde ve smyčce dokud uživatel nezmáčkne tlačítko. Tím je změněn trigger *OkAnswerLeaveTrigger*, který vyvolá poslední animaci. Obrázek pomalu zmizí a poté plynule přejde do stavu, kdy není vidět. Stav animátoru a přechody mezi nimi jsou přehledně ukázány na obrázku 4.1.

Ve skriptu *OkAnswerDoggie.cs* jsou dvě funkce. *PopUp* pro zobrazení pejska, nastavení triggeru a spuštění zvuku. Funkce *Leave* nastavuje pouze trigger pro spuštění animace zmizení. Obě tyto funkce jsou veřejné a jsou volány z jiných skriptů aplikace. Součástí skriptu jsou také tři veřejné třídní proměnné, ve kterých jsou uchovány objekty potřebné ve funkcích. Hodnoty těchto proměnných je možno nastavit přímo v Unity pouhým přetažením objektu do daného políčka.

Samotné vyhodnocení otázky je u kvízu řízeno pomocí skriptu patřícího k tlačítku pro kontrolu odpovědi - *CheckButton*. V *Update* funkci průběžně kontroluji, zda uživatel zaklikl nějakou odpověď a pokud ano, kontroluji její správnost a pravdivostní hodnotu ukládám do proměnné *valid*. Pokud je pak stisknuto tlačítko pro zkontrolování, jednoduše se jen



Obrázek 4.1: Stavy animátoru při animaci správného odpovězení na otázku

zkontroluje, zda poslední stisknutá možnost byla validní. Vyskočí příslušná animace pejska, schová se *CheckBox* a místo něj se ukáže buď *AgainButton* pro opakování odpovědi, nebo *NextButton* pro přechod na další obrazovku. Zobrazování a schovávání tlačítek má na starost skript *QuizScene*, který je komponentou canvasu.

## 4.4 Audio

Při načtení scény se zvuk spouští automaticky. Poté, co se nahrávky dostanou na konec, objeví se tlačítko pro opakování zvuku. Pokud je součástí scény zvuk, je přidán skript *SceneWithVoice* mezi komponenty canvasu. Skript obsahuje funkci *Start*, ve které najde a uloží objekt tlačítka do pole třídy. Další dvě metody slouží k zobrazení a schování tohoto tlačítka.

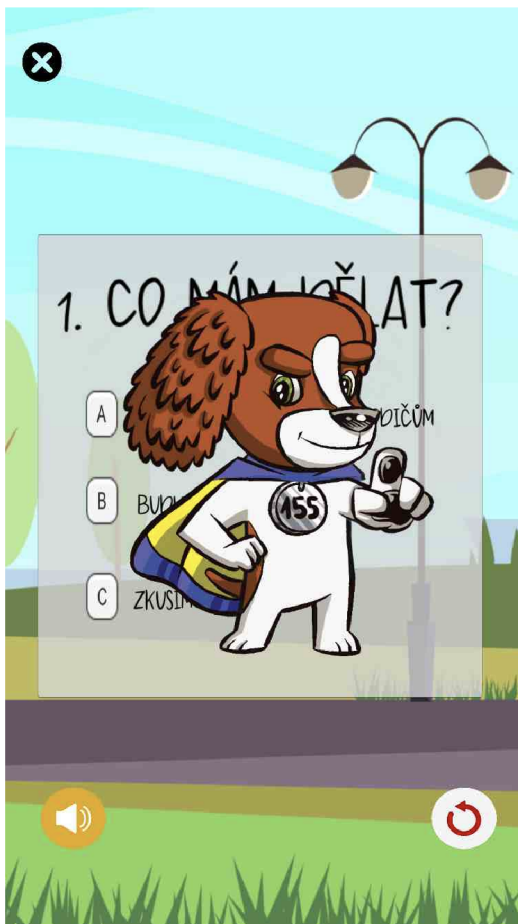
Pro pouštění audia u scén s více nahrávkami slouží skript *MultipleAudio*, který má vystavené pole pro uložení nahrávek. Toto pole je viditelné v Unity, pro každé použití lze poli nastavit různou délku a přetáhnout příslušné nahrávky. Hlavní funkce tohoto skriptu je coroutine *PlayEngineSound*, která je volána přímo z funkce *Start*. Coroutine je funkce, která vrací data pomocí *yield*. Na rozdíl od klasických funkcí, tato je vyhodnocována po rámcích, může uvnitř ní být čekání a nezastaví se ostatní chování aplikace. Jedná se o jakousi asynchronní formu volání. Zavolání této funkce probíhá příkazem

```
StartCoroutine(PlayEngineSound());
```

a pokud je třeba přerušit její vykonávání, zavolá se

```
StopCoroutine(PlayEngineSound());
```





(a) Animace špatné odpovědi spolu s tlačítkem pro opakování



(b) Animace správné odpovědi spolu s tlačítkem pro přepnutí scény

Obrázek 4.2: Animace vyhodnocení otázky ve výsledné aplikaci

Já zastavení používám v případě, že dítě odpoví dříve, než se stihne pustit audio ke všemu textu na obrazovce.

## 4.5 Mini hry

Sekce mini hry obsahuje popis výsledné podoby implementace logické části i grafického uživatelského rozhraní obou her obsažených v aplikaci.

### 4.5.1 Mini hra přiřazování

Zadání hry Přiřad je následující: *V této hře máš za úkol přiřadit ke každé složce záchranného systému správné telefonní číslo, pomůcku na jeho zapamatování a jejich auto. Složky záchranného systému jsou záchranka, policie a hasiči.* Na scéně jsou 3 řady po 3 rámečcích na uložení obrázku. Každý z nich na začátku hry již obsahuje nějaký obrázek. Hra je nastavená tak, že pokud uživatel přetáhne obrázek do jiného políčka, jeho původní obsah se objeví na původní pozici přetahované položky. Výsledný vzhled je na obrázku 4.3.

V kódu to funguje následovně. Hra je řízena skriptem *DragAndDrop*, který je komponentou všech položek, jež je možno přetahovat. Při inicializaci scény se do soukromého pole uloží komponenty *RectTransform* a *CanvasGroup*. V metodě *Start* se uloží pozice položky jako první hodnota do soukromého pole. Pokud uživatel začne táhnout, aktualizuje se poloha položky, obrázku se zvýší průhlednost a vypne se blokování kolizí. Při táhnutí se průběžně aktualizuje poloha tažené položky, k aktuálním hodnotám se vždy přičte delta, tedy hodnota, o kterou se od posledního vyhodnocení obrázek posunul. Jakmile uživatel táhnout přestane, nastává nejdůležitější moment. Pokud obrázek pustil někde mimo slot pro uložení, vrátí se tažená položka na své původní místo. V opačném případě se pozice položky nastaví na pozici obrázku ve slotu (popřípadě pozici slotu). Původní obsah slotu je přesunut na výchozí pozici tažené položky. Algoritmus pro tuto hru jsem převzala z tutoriálu na YouTube<sup>3</sup>.

### 4.5.2 Mini hra výběr důležitých informací

V druhé mini hře má uživatel za úkol vybrat z uvedených informací jen ty důležité. Scéna se od návrhu liší přítomností toggle tlačítek. Unity totiž neumožňuje současný výběr více tlačítek. Proto jsem se rozhodla vyrobit dvojice tlačítka a zaškrťovacího tlačítka, které jsou spolu propojené. Uživatel může klikat na obě dvě z nich.

Algoritmus výběru funguje tak, že ať uživatel klikne na tlačítko s textem nebo zaškrťovací tlačítko, rozhoduje se podle stavu zaškrťovacího tlačítka. V druhém případě se rozhoduje podle aktuálního stavu, tedy pokud je tlačítko zaškrtnuté, nastaví se barvy tlačítka s textem na žlutou. Pokud tlačítko zaškrtnuté není, barvy jsou nastavené na bílou. V případě prvním, tedy kliknutí na tlačítko s textem, je scénář opačný a stav zaškrťovacího tlačítka se nastavuje zpětně bez notifikace. Jak hra vypadá je možné vidět na obrázcích 4.4a a 4.4b.

## 4.6 Animace

V této sekci bude rozebrána implementace několika hlavních animací, které jsou součástí výsledné aplikaci.

<sup>3</sup>[https://www.youtube.com/watch?v=BGr-7GZJNXg&ab\\_channel=CodeMonkey](https://www.youtube.com/watch?v=BGr-7GZJNXg&ab_channel=CodeMonkey)



Obrázek 4.3: Vzhled mini hry přiřad

#### 4.6.1 Animace pejska

Při animacích pejska Defíka jsem používala vždy sekvenci obrázků. Animovaný objekt má obvykle 4 základní stavy. První, ve kterém není grafika vidět, poté další, při kterém se objekt pomalu objeví, následuje hlavní animace (například gesto pejska) a jako poslední je zmizení objektu, toto probíhá na základě nastavení spouštěče, stejně jako objevení. Po zmizení se objekt vrací do stavu, kdy není vidět. Nastavení přechodů mezi stavy má na starost animátor objektu.

#### 4.6.2 Úvodní a závěrečná animace

Scéna pro úvod do situace, která měla být původně video, je nakonec realizovaná pomocí animace. Začíná jednoduchou animací panelu s pozadím parku, ten se přejede zleva doprava. Obraz zůstane stát u scény s lavičkou, kde leží paní v bezvědomí a u ní klečí pán, který se drží za hlavu a je očividné, že neví, co má dělat. Na tuto scénu se obraz přiblíží, aby ji dítě vidělo zblízka. Poté se pohled začne oddalovat a při opuštění stavu *zoom in* se spustí animace běžícího pejska. Jakmile Defík doběhne na místo, scéna se automaticky přepíná na první otázku. Na obrázku 4.5 je graf stavů animátoru a jejich přechodů.

Závěrečná animace se skládá z pozadí parku, ve kterém stojí auto záchranky, a popředí, což je animace pejska Defíka, který ukazuje palec nahoru. U této scény je také zvuk, kdy pejsek poděkuje za výborně odvedenou práci. Nakonec vyskočí dialog, zda si chce hráč zopakovat lekci nebo chce aplikaci opustit.

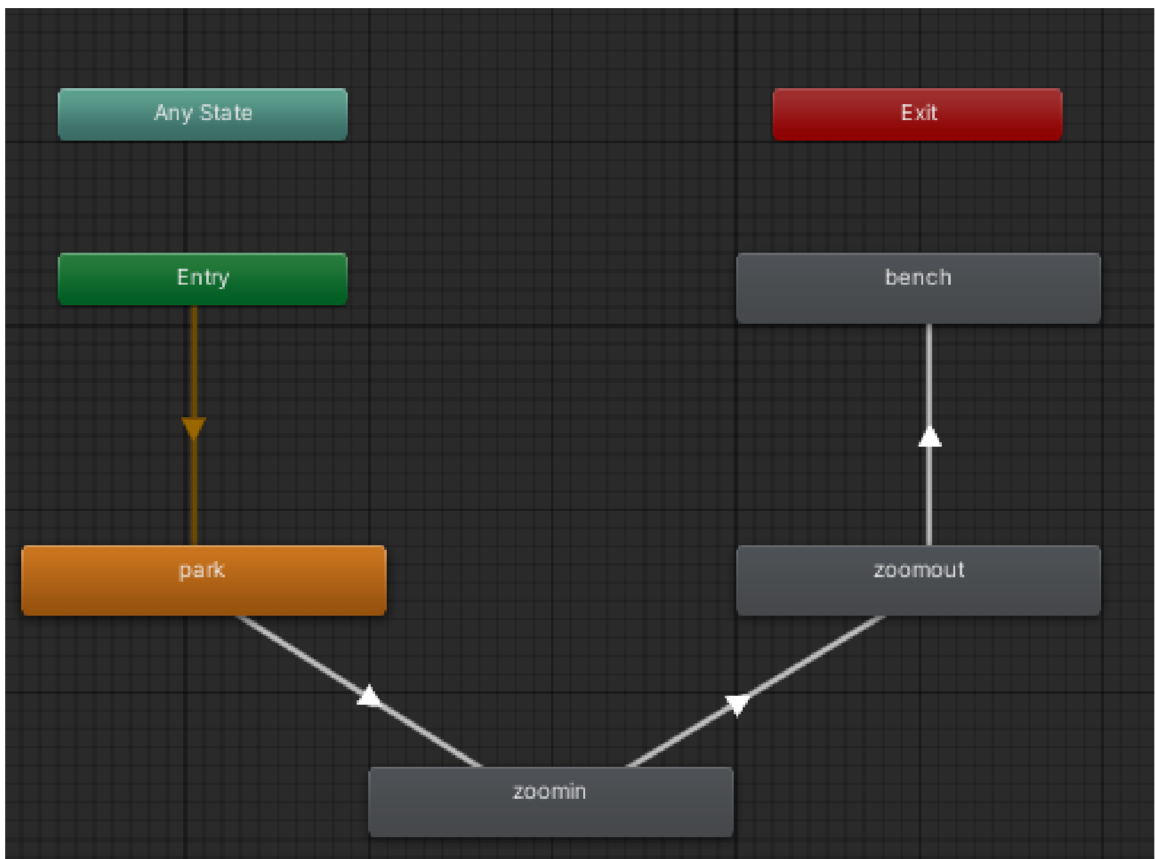
### 4.7 Grafika a fonty

Grafiku jsem vytvářela především v nástroji Canva. Pro aplikaci jsem zvolila font Jenthill Caps, jedná se o tiskací písmo, je celkem dobře čitelné a stylově se hodí ke kreslené aplikaci

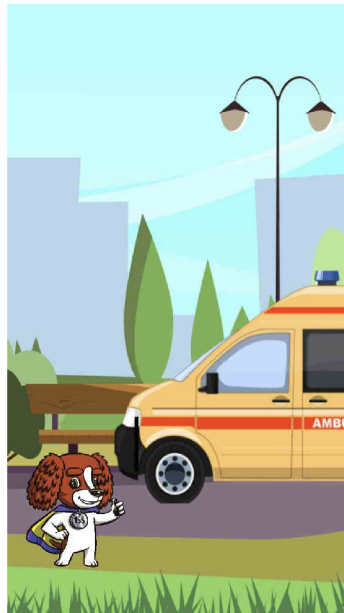


(a) Hra vyber správné informace - výchozí stav obrazovky (b) Hra vyber správné informace - vybrané správné položky

Obrázek 4.4: Výsledná podoba hry vyber správné informace



Obrázek 4.5: Stavy animátoru při úvodní animaci parku



Obrázek 4.6: Výsledná podoba závěrečné scény

2. JAKÉ MÁM ZAVOLAT  
ČÍSLO?



- (a) Text zadání druhé kvízové otázky  
(b) Pomůcka pro zapamatování čísla 150  
(c) Ikona informace použitá na úvodní obrazovce

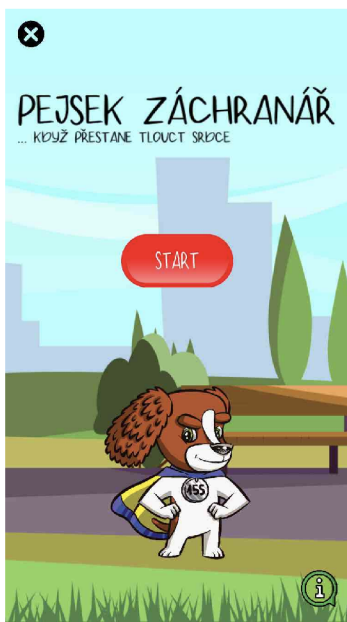
Obrázek 4.7: Grafika použitá z Canvy

pro děti. Z dostupné knihovny prvků jsem použila hlavní červené tlačítko, ostatní tlačítka jsou kulatá a doplněná ikonou buď z Icons8 nebo z knihovny přímo v Canvě. Funkce smazání pozadí, kterou Canva nabízí, jsem využívala velmi často. Většina obrázků, které jsou v aplikaci, byla totiž s pozadím a to bylo pro použití ve hře nevhodné. Na obrázku 4.7 jsou ukázky grafiky z databáze Canva.

Pokud v databázi Canvy nebyly takové materiály, které by se do aplikace hodily, hledala jsem ještě na Icons8. Na této stránce jsou nejen ikony, jak napovídá název, ale také fotky, ilustrace a hudba.

## 4.8 Výsledná aplikace

Výsledná aplikace se od návrhu liší jen minimálně. Původní návrh měl obsahovat místo úvodní a závěrečné animace videa. Kromě toho je koncept zachován, jak jde vidět na obrázcích 4.8.



(a) Výsledná podoba úvodní obrazovky



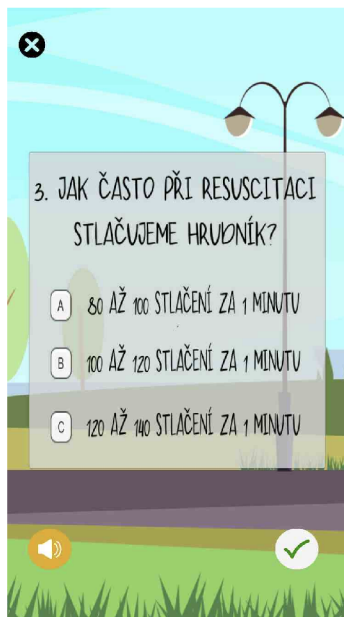
(b) Výzva k otočení telefonu



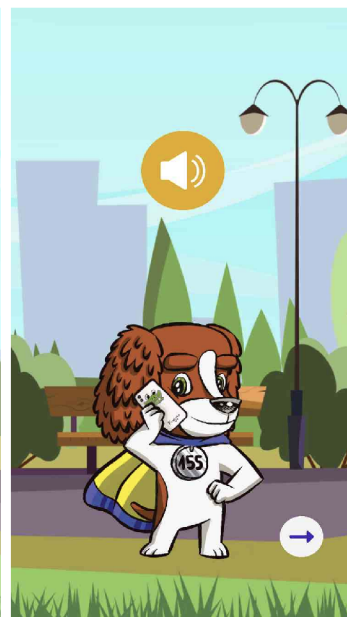
(c) Výsledná podoba otázky číslo 1



(d) Výsledná podoba otázky číslo 2



(e) Výsledná podoba otázky číslo 3



(f) Výsledná podoba scény rozhovoru s operátorkou

Obrázek 4.8: Fotky výsledné podoby aplikace

## Kapitola 5

# Testování aplikace

V kapitole testování bude rozebráno jak testování funkčnosti aplikace pomocí Unity Remote, tak i testování s cílovou skupinou uživatelů. C# umožňuje vytváření unit testů, avšak pro vytváření her jsou použité tak specifické skripty bez jakéhokoliv rozhraní či návrhového vzoru, že jejich použití zde mi nedává smysl.

### 5.1 Unity Remote

Aplikace Unity Remote je volně dostupná ke stažení jak v App Store tak Google Play. Slouží k odladění mobilních aplikací vyvíjených v Unity. Aplikace totiž může reagovat jinak, když je ovládaná přes počítač myší a nebo přes mobilní telefon prstem. Jednu z takových chyb jsem odhalila v samém začátku vývoje. U mini hry s přiřazováním ke složkám záchranného systému pracuji s událostmi vyvolanými táhnutím a puštěním obrázku. Tyto události jsou ale jiné pro počítač a jiné pro mobilní telefon. Celou hru jsem měla hotovou, na počítači vše fungovalo v pořádku. Poté jsem si hru zkusila pustit přes Unity Remote a našla několik chyb v chování.

Nakonec jsou odchyťovány události pomocí 4 handlerů a následně volány příslušné metody. Handler *IPointerDownHandler* je používán pro detekci stlačení myši. Pokud je kliknuto a uživatel začne táhnout, tuto událost zachytí *IBeginDragHandler*. Ukončení táhnutí zachycuje *IEndDragHandler*. Pomocí *IDragHandler* odchyťávám události v průběhu táhnutí.

Další chybou, na kterou jsem díky testování přímo v zařízení přišla, bylo otáčení obrazovky. Pokud byl telefon otočený na bok, celá aplikace se otočila také. Na to ale grafika nebyla připravená a vypadalo to velmi ošklivě. Dohledala jsem, že lze pro aplikaci nastavit její orientace a tím zablockovat její otočení, i kdyby došlo k otočení zařízení. Důležité ale bylo, aby nadále fungovala detekce otočení pro obrazovku před spuštěním videa. Pokud se odchyťí událost otočení zařízení, začne se video přehrávat okamžitě a nečeká se na odpočet.

### 5.2 Debugging pomocí Visual Studio

Skripty psané v jazyce C# jsem odladřovala pomocí debug módu ve Visual Studiu. Jednoduše jsem si nastavila breakpoint na sledované místo, spustila debug mód a poté se přepnula do Unity, kde jsem zapnula hru. Pokud se na požadovanou část skriptu při hře narazilo, byla pozastavena a automaticky se přepnul kontext do Visual Studia. Mohla jsem zkontrolovat, jaké hodnoty mají proměnné a kudy kód prochází. Při odchyťování událostí jsem



tuto funkci využívala často pro zjištění, zda je při dané události opravdu zavolána příslušná metoda.

Další způsob testování, který jsem využívala, bylo vypisování do konzole. Toto je umožněno vložením příkazu

```
Debug.Log("Text");
```

na libovolné místo ve skriptu. Výpisy jsou pak ukazovány jak v konzoli ve Visual Studiu, tak i v konzoli v Unity. Při řešení hry s výměnou obrázků jsem tento způsob využívala pro rozpoznávání odchycených událostí.

### 5.3 Uživatelské testování

Pro uživatelské testování bylo vybráno 8 žáků z první a druhé třídy. Každé dítě dostalo telefon s nainstalovanou aplikací, následně se hra spustila. V první fázi jsem dítěti pouze řekla, aby si zkusilo zahrát hru, že jej budu jen pozorovat a potom to společně rozebereme.

Tři děti z první třídy využívaly tlačítko pro opětovné spuštění zvuku, jelikož ještě neumí dobře číst. Průchod aplikací jim zabral o něco více času než druhákům, jelikož jsem dětem čas nestopovala, aby nebyly ve stresu, nevím přesné číslo. Odhaduji, že mladším dětem trval průchod aplikací o 30 sekund až minutu déle. Celkem čtyři děti měly problém s mini hrou přiřad, kde je třeba přetahovat položky mezi rámečky. Většina z nich nerozuměla dobře zadání a u jednoho byl problém s přiřazením hadice, jelikož se učil pomůcku rybníček.

Měla jsem připravený seznam otázek, které jsem s dětmi chtěla po konci hry projít. Některé odpovědi vyplynuly přímo z pozorování chování dítěte při hraní hry. Zde je uveden seznam připravených otázek, na které jsem se dětí ptala:

1. Je písmo dost velké a čitelné?
2. Jsou úkoly popsané dostatečně srozumitelné? Rozumíš tomu, co se po tobě chce?
3. Dokážeš krátce shrnout postup první pomoci při náhlé zástavě oběhu?
4. Co se ti na aplikaci nejvíce líbilo?
5. Co tě nejméně bavilo?

Na první otázku byly reakce vesměs kladné. Kromě písmena D, které čtyři děti zmínily jako špatně čitelné, neměly s fontem a ani jeho velikostí problém.

Druhá otázka většinou vyplynula z průběhu hry a toho, jestli se dítě snažilo požádat mě o pomoc. S kvízovými otázkami neměl problém nikdo. Mini hra s výběrem důležitých informací byla také vcelku pochopitelná, opravila bych čtení možností, které bylo pro 2 děti moc rychlé. Ne všechny děti si ale napoprvé dokázaly poradit s hrou přiřazování ke složkám záchranného systému, 2 děti mě požádaly o pomoc, 1 z nich si nakonec poradil a dalšímu stačilo říct, že pomůcka pro hasiče může být rybníček i hadice. Dvě děti, které si s hrou nevěděly rady, mají telefon s tlačítky a hry hrají velmi málo nebo spíš vůbec.

Otázku shrnutí toho, co se měly děti naučit, hodnotím zpětně jako velmi sebevědomou. Mnoho dětí se stydělo a těch, které dokázaly odpovědět opravdu správně, bylo jen pár. Konkrétně 2 děti, obě z druhé třídy a řekly mi, že chodí do skauta, kde se to učily. Ostatních šesti dětí jsem se dále doptávala alespoň jestli si pamatují číslo na záchranku a na to už polovina dokázala říct správnou odpověď.

Dětem se nejvíce líbila postava pejska Defíka, někdo také řekl, že jej bavila hra s přiřazováním. Dvě děti bavily kvízové otázky.

Šest dětí, což je velká část z testovaných, se shodlo, že je příliš nebavila část s rozhovorem s operátorkou, kde se jen poslouchá návod na resuscitaci a někteří u toho nedávali pozor. Kdybych měla tuto část vylepšit, udělám ji formou animace, tedy vše co paní operátorka říká, by bylo předváděno ve hře.

## Kapitola 6

# Závěr

Aplikace Pejsek záchranář - náhlá zástava oběhu se podařila vytvořit dle zadaného scénáře. Nad rámec tohoto scénáře byla obohacena o dvě mini hry. Na základě uživatelské studie se dá říct, že aplikace svůj účel splnila a má potenciál sloužit jako praktická pomůcka výuky první pomoci.

Pokud bych měla navrhnout rozšíření hry, tak bych vytvořila názornou scénu s návodem na resuscitaci, která děti v aktuální podobě dle testování nebaví. Popřípadě by se dala aplikace rozšířit o část, jak se zachovat, když najdeme takto bezvládného člověka. Nejdříve je potřeba jej oslovit, pokud nereaguje, zkusit bolestivý podnět. Na to aktuálně v aplikaci není kladen důraz.

V porovnání s lekcí o zástavě dechu a oběhu z aplikace od Záchranného kruhu je moje aplikace více hravá a zapojuje uživatele. Lekce je sice názorná a ukazuje prováděné úkony na zraněném, ale je dlouhá a plná informací, které z mého osobního pohledu nejsou zas tolik důležité. Netrénuje se zde volání správného čísla ani tempo resuscitace.

Přínos hry vidím zejména v šíření základních znalostí první pomoci mezi děti a to pomocí hry, která je zcela zadarmo. Co se týče práce, věřím, že může být dobrým odrazovým můstkem pro případné následovníky, kteří budou portfolio her pro děti od ZZS JMK rozšiřovat, ale i pro jiné vývojáře zajímající se o vývoj her v Unity.

Jsem ráda, že jsem dostala možnost zpracovat bakalářskou práci s takovým přesahem. Doufám, že bude aplikace dobře sloužit všem, kteří se rozhodnou ji sobě nebo svým dětem stáhnout do chytrého telefonu.

# Literatura

- [1] *Crello vs Canva – What’s the Best Choice For 2021?* [online]. [cit. 2021-04-10]. Dostupné z: <https://www.ecomsay.com/crello-vs-canva/>.
- [2] *Godot* [online]. [cit. 2020-12-27]. Dostupné z: <https://docs.godotengine.org/en/stable/>.
- [3] *Google Play vs App Store* [online]. [cit. 2021-01-27]. Dostupné z: <https://smartmania.cz/google-play-vs-app-store-vyhody-aplikace/>.
- [4] *What is the best alternative to Unity3D?* [online]. [cit. 2020-12-27]. Dostupné z: <https://www.slant.co/options/1047/alternatives/~unity3d-alternatives>.
- [5] *Designing Mobile Interfaces*. 1. vyd. O’Reilly Media, Inc., 2011. ISBN 9781449394639.
- [6] *První pomoc* [online]. asociace Záchraný kruh, 2011 [cit. 2021-04-10]. Dostupné z: <https://www.alik.cz/h/prvni-pomoc>.
- [7] *Lambda expressions (C# reference)* [online]. Microsoft, 2020 [cit. 2021-04-10]. Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/csharp/language-reference/operators/lambda-expressions>.
- [8] ALAN, T. *Unity Animation Essentials*. 1. vyd. Packt Publishing Ltd, 2015. ISBN 978-1-78217-481-3.
- [9] ALBAHARI JOSEPH, A. B. *C# 7.0 in a nutshell*. 1. vyd. O’Reilly Media, Inc, 2018. ISBN 978-1-491-98765-0.
- [10] BAND, B. *Canva* [online]. [cit. 2021-04-10]. Dostupné z: <https://search.proquest.com/openview/ea3941b8bc20c507fcae79a412820f80/1?pq-origsite=gscholar&cbl=296199>.
- [11] CHOWDHURY, K. *Mastering Visual Studio 2017*. 1. vyd. Packt Publishing Ltd, 2017. ISBN 9781787280656.
- [12] CVETKOVIĆ, D. *Simulation and Gaming*. 1. vyd. InTech, 2018. ISBN 978-953-51-3803-7.
- [13] EPIC GAMES, I. *Unreal Engine 4 Documentation* [online]. [cit. 2021-04-10]. Dostupné z: <https://docs.unrealengine.com/en-US/index.html>.
- [14] FRANĚK, M. O. *Mimonemocniční náhlá zástava oběhu a neodkladná resuscitace dospělých v terénu* [online]. -: -, duben 2011 [cit. 2019-10-02]. Dostupné z: [https://www.zachrannasluzba.cz/zajimavosti/2010\\_resuscitace.pdf](https://www.zachrannasluzba.cz/zajimavosti/2010_resuscitace.pdf).

- [15] GARGENTA, M. *Learning Android*. 1. vyd. O'Reilly Media, Inc., 2011. ISBN 9781449307240.
- [16] GEHRED, A. P. *Canva* [online]. [cit. 2021-04-10]. Dostupné z: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7069818/>.
- [17] GIBSON, J. *Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game - with Unity and C#*. 2. vyd. Addison-Wesley, 2015. ISBN 978-0-321-93316-4.
- [18] GUMSTER, J. van. *Blender for dummies*. 4. vyd. Wiley and sons, 2020. ISBN 978-1-119-61696-2.
- [19] HEJLSBERG, A., TORGERSEN, M., WILTAMUTH, S. a GOLDE, P. *The C# Programming Language*. Pearson Education, 2008. Microsoft Windows Development Series. ISBN 9780321592255. Dostupné z: <https://books.google.cz/books?id=ICe7ea4RscUC>.
- [20] JARMILA, K. *První pomoc I: pro studenty zdravotnických oborů. 2., přeprac. a dopl. vyd.* 2. vyd. GRADA Publishing, 2012. ISBN 978-80-247-4199-4.
- [21] JASON, G. *Game engine architecture*. 3. vyd. Boca Raton: Taylor & Francis, CRC Press, 2018. ISBN 978-1-13803-545-4.
- [22] KARTHICK S, D. S. B. Android Security Issues and Solutions. *ICIMIA*. 2017, s. 686–689.
- [23] MICROSOFT. *What's new in C# 9.0* [online]. [cit. 2021-04-10]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-9>.
- [24] MUDR. MGR. JAN BYDŽOVSKÝ, D. *Předlékařská první pomoc*. 1. vyd. Grada Publishing, a.s., 2011. ISBN 978-80-247-2334-1.
- [25] STATISTA. *Mobile operating systems' market share worldwide from January 2012 to January 2021* [online]. Statista, 2021 [cit. 2021-04-10]. Dostupné z: <https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/>.
- [26] TECHNOLOGIES, U. *Unity User Manual (2019.4 LTS)* [online]. [cit. 2021-04-10]. Dostupné z: <https://docs.unity3d.com/2019.4/Documentation/Manual/UnityManual.html>.