

# Vývoj webové hry pro výuku základů počítačových sítí

Bc. Zdeněk Karlík

---

Diplomová práce  
2024



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
Ústav informatiky a umělé inteligence

Akademický rok: 2023/2024

# ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Bc. Zdeněk Karlík  
Osobní číslo: A23541  
Studijní program: N0613A140022 Informační technologie  
Specializace: Softwarové inženýrství  
Forma studia: Kombinovaná  
Téma práce: Vývoj webových hry pro výuku základů počítačových sítí  
Téma práce anglicky: Development of a Web Game for Teaching the Basics of Computer Networks

## Zásady pro vypracování

- Vypracujte stručnou literární rešerši na téma vývoj počítačových her a použitých technologií.
- Vhodně navrhnete úlohy s přihlédnutím ke kurikulu SŠ.
- Navrhnete webovou aplikaci pro výuku základů počítačových sítí.
- Naprogramujte navrženou webovou aplikaci.
- Vytvořte adekvátní popis výsledné aplikace.
- Sestavte a vyhodnotte anketu na vybraném vzorku studentů.

Forma zpracování diplomové práce: **tištěná/elektronická**

**Seznam doporučené literatury:**

1. STUART, Graeme. *Introducing JavaScript Game Development: Build a 2D Game from the Ground Up*. Imprint: Apress, 2017. ISBN 978-1484232514.
2. SHANKAR, Aditya Ravi. *Pro HTML5 Games: Learn to Build your Own Games using HTML5 and JavaScript*. 2nd ed. New York, NY: Springer Science+Business Media, 2017. ISBN 978-1484229095.
3. DUCKETT, Jon. *PHP & MySQL: server-side web development*. Hoboken, New Jersey: Wiley, 2022. ISBN 978-1119149224.
4. LOWE, Doug. *Networking All-in-One For Dummies*. 8th ed. Indianapolis: John Wiley, 2021. ISBN 978-1119689010.
5. FULLERTON, Tracy. *Game Design Workshop: A Playcentric Approach to Creating Innovative Games, Fourth Edition*. 4th ed. A K Peters/CRC Press, 2018. ISBN 978-1138098770.

Vedoucí diplomové práce: **Ing. Bc. Pavel Vařacha, Ph.D.**  
Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: **5. listopadu 2023**

Termín odevzdání diplomové práce: **13. května 2024**

**doc. Ing. Jiří Vojtěšek, Ph.D. v.r.**  
děkan



**prof. Mgr. Roman Jašek, Ph.D., DBA v.r.**  
ředitel ústavu

Ve Zlíně dne 5. ledna 2024

**Prohlašuji, že**

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

**Prohlašuji,**

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 7.5.2024

Zdeněk Karlík, v.r.  
podpis studenta

## **ABSTRAKT**

Diplomová práce se zaměřuje na vývoj webové hry pro výuku počítačových sítí. Hlavním cílem diplomové práce bylo vytvoření interaktivní webové hry, která bude sloužit pro podporu výuky základů počítačových sítí. Do této aplikace bylo implementováno celkově 65 úloh, vytvořených na základě školského vzdělávacího plánu používaného na střední škole. Úlohy jsou poté plněny s pomocí pěti miniher. Uživatel má možnost hrát hru prostřednictvím herního módu kampaně zaměřeného na výuku základů počítačových sítí nebo si vyzkoušet libovolnou úroveň. Diplomová práce dochází k závěru, že výsledná webová hra by mohla studentům a začátečnickům pomoci k lepšímu pochopení problematiky počítačových sítí.

Klíčová slova: Webová aplikace, webová hra, hra, počítačové sítě, výuka

## **ABSTRACT**

The thesis focuses on the development of a web-based game for teaching computer networks. The main aim of master thesis was to create an interactive web game designed to support the teaching of basic computer network principles. A total of 65 tasks have been implemented into this application, created based on the educational curriculum used in secondary schools. These tasks are then completed with the help of five mini-games. The user has the option to play the game through the campaign mode which focuses on teaching the basics of computer networks, or to try out any level independently. In brief, the master thesis concludes that the resulting web game could assist students and beginners in achieving a better understanding of computer network concepts.

Keywords: Web application, web game, game, computer networks, teaching

Tímto bych chtěl poděkovat panu Ing. Bc. Pavlovi Vařachovi, Ph.D. za poskytování cenných rad, ochotu a dohled při vedení mé diplomové práce.

Prohlašuji, že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

# OBSAH

<b>ÚVOD</b> .....	<b>10</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>11</b>
<b>1 MOTIVACE K VÝVOJI A REŠERŠE EXISTUJÍCÍCH ŘEŠENÍ</b> .....	<b>12</b>
1.1 SIMULÁTORY SÍTÍ .....	12
1.2 ONLINE TESTY .....	13
1.3 MOBILNÍ APLIKACE COMPUTER NETWORK TUTORIALS .....	14
<b>2 VÝVOJ POČÍTAČOVÝCH HER</b> .....	<b>15</b>
2.1 HERNÍ ENGINY .....	15
2.2 PROGRAMOVACÍ JAZYKY .....	17
2.3 WEBOVÉ HRY .....	18
2.4 VÝVOJ HER A FÁZE VÝVOJE .....	19
2.4.1 Před-produkční fáze .....	19
2.4.2 Produkční fáze.....	19
2.4.3 Po-produkční fáze .....	21
2.5 TRENDY VE VÝVOJI HER .....	21
2.6 HERNÍ DESIGN .....	22
2.6.1 Způsoby inovací ve hrách .....	22
2.7 VÝVOJ HER TYPU GOAL-ORIENTED A GAME-BASED LEARNING.....	22
<b>3 TECHNOLOGIE VYUŽÍVANÉ PŘI VÝVOJI HRY</b> .....	<b>24</b>
3.1 WEBOVÉ TECHNOLOGIE .....	24
3.1.1 HTML .....	24
3.1.2 CSS.....	25
3.1.3 Javascript.....	25
3.1.4 OOP v Javascriptu.....	26
3.2 NÁVRH APLIKACE.....	27
3.2.1 Účely využití UML .....	27
3.2.2 Analýza požadavků .....	28
3.2.3 UC model .....	29
3.2.5 Scénáře .....	30
3.2.6 Návrh UX/UI designu .....	31
3.3 SYSTÉMY PRO SPRÁVU VERZÍ .....	33
3.4 VÝVOJOVÉ PROSTŘEDÍ .....	33
3.5 UMĚLÁ INTELIGENCE .....	34
<b>II PRAKTICKÁ ČÁST</b> .....	<b>35</b>
<b>4 NAVRŽENÍ ÚLOH NA ZÁKLADĚ KURIKULA SŠ</b> .....	<b>36</b>
4.1 PRŮZKUM KURIKULA SŠ .....	36

4.3	SHRnutí ÚLOH NA ZÁKLADĚ KURIKULA.....	38
4.5	GUESS THE WORD .....	42
4.6	PICTURE MATCH .....	42
4.7	GUESS THE PICTURE.....	43
4.8	PUT THE WORDS .....	43
4.9	COUNT THE NUMBER.....	43
<b>5</b>	<b>NAVRŽENÍ WEBOVÉ APLIKACE.....</b>	<b>45</b>
5.1	SBĚR A ANALÝZA POŽADAVKŮ.....	45
5.1.2	Funkční požadavky všech her .....	45
5.1.3	Nefunkční / systémové požadavky.....	46
5.2	MODEL PŘÍPADŮ UŽITÍ .....	46
5.3	SCÉNÁŘE .....	50
5.4	NAVRŽENÍ APLIKACE POMOCÍ WIREFRAMŮ .....	67
<b>6</b>	<b>NAPROGRAMOVÁNÍ WEBOVÉ APLIKACE.....</b>	<b>75</b>
6.1	SKRIPT MENU.JS .....	75
6.2	TŘÍDY JEDNOTLIVÝCH HER.....	78
6.2.1	Třída Game.....	78
6.2.2	Třída DragableGame .....	81
6.2.3	Třída GuessTheWord .....	85
6.2.4	Třída GuessThePicture.....	86
6.2.5	Třída PictureMatch.....	88
6.2.6	Třída PutTheWords a CountTheNumber .....	89
6.3	TŘÍDA TIMER.....	89
6.4	POPUP.....	90
<b>7</b>	<b>POPIS APLIKACE A JEJÍHO GUI.....</b>	<b>91</b>
7.1	MÓD KAMPAŇ .....	92
7.2	MÓD LEVELY.....	92
7.3	GUI MINIHRY .....	93
7.4	TABULKA A OBECNÉ OVLÁDÁNÍ .....	100
<b>8</b>	<b>OPTIMALIZACE APLIKACE .....</b>	<b>102</b>
<b>9</b>	<b>SESTAVENÍ A VYHODNOCENÍ ANKETY .....</b>	<b>103</b>
9.1	NADEFINOVÁNÍ CÍLŮ ANKETY .....	103
9.2	OSOBNÍ DOTAZY .....	103
9.3	DOTAZY Z OBLASTI POČÍTAČOVÝCH SÍTÍ.....	106
9.4	DOTAZY TÝKAJÍCÍ SE HRANÍ HER.....	108
9.5	VYHODNOCENÍ ANKETY A NAPLNĚNÍ CÍLŮ .....	118
<b>10</b>	<b>VYPRACOVÁNÍ ZÁSAD .....</b>	<b>119</b>



<b>11 BUDOUCNOST APLIKACE.....</b>	<b>120</b>
<b>ZÁVĚR .....</b>	<b>121</b>
<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>	<b>126</b>
<b>SEZNAM OBRÁZKŮ .....</b>	<b>129</b>
<b>SEZNAM TABULEK.....</b>	<b>131</b>
<b>SEZNAM GRAFŮ .....</b>	<b>132</b>
<b>SEZNAM PŘÍLOH.....</b>	<b>133</b>

## ÚVOD

V dnešní digitální době se stává vzdělání v oblasti počítačových sítí nezbytností. S nárůstem významu počítačových sítí ve všech oblastech společnosti je stále důležitější, aby studenti a začátečníci měli přístup k efektivním vzdělávacím prostředkům, které jim pomohou lépe porozumět principům, protokolům a technologiím spojeným s počítačovými sítěmi. Ačkoliv existuje mnoho zdrojů pro studium tohoto oboru, mnoho začátečníků a studentů má stále problém pochopit základní principy počítačových sítí.

Hlavním cílem této diplomové práce je vytvoření interaktivní webové hry, která bude sloužit jako vzdělávací nástroj pro výuku základů počítačových sítí. Hra má uživatelům poskytovat intuitivní a zábavný způsob výuky, což by mohlo vést ke zvýšení zájmu o počítačové sítě.

V diplomové práci je provedena analýza, která poslouží jako základ pro návrh webové hry určené k výuce počítačových sítí. Tato analýza zahrnuje studium existujících aplikací, rozbor výhod a nevýhod technologií pro vývoj hry a volbu témat na základě školského vzdělávacího plánu. Dále je v práci využita anketa jako doplňující metoda sběru dat, která umožní získat hlubší pochopení potřeb a očekávání uživatelů v této oblasti. Zároveň anketa poskytuje odpovědi na stanovené cíle.

Diplomová práce je rozdělena na teoretickou a praktickou část. Součástí teoretické části je popis motivace k vývoji webové hry, průzkum aplikací a her zaměřených na podporu výuky počítačových sítí a vypracování přehledu odborné literatury na téma vývoj počítačových her a použitých technologií. Praktická část se zaměřuje na implementaci navržených úloh na základě kurikula střední školy. Dále se tato část práce věnuje návrhu interaktivní webové hry zahrnující sběr a analýzu požadavků, vytvoření modelu případů užití s příslušnými scénáři a návržení jednotlivých částí hry pomocí wireframů. Poté následuje část zaměřená na naprogramování webové aplikace, kde bude popsán proces implementace, včetně důležitých technických a vizuálních detailů hry. Závěr praktické části bude zaměřen na sestavení ankety, která bude sloužit k získání uživatelské zpětné vazby. Anketa bude zahrnovat také nadefinování hlavních a vedlejších cílů. Data získaná z ankety budou poté současně se stanovenými cíli vyhodnocena.

## **I. TEORETICKÁ ČÁST**

## 1 MOTIVACE K VÝVOJI A REŠERŠE EXISTUJÍCÍCH ŘEŠENÍ

Hlavní motivací k vytvoření této práce bylo posílení schopností v oblasti webových aplikací. Kromě samotného záměru zlepšení schopností při tvorbě této diplomové práce byla motivací také nabídka střední školy na vytvoření aplikace pro podporu výuky počítačových sítí. Během procesu tvorby došlo také ke zlepšení znalostí v oblasti softwarového inženýrství, vzhledem k tomu, že postup zahrnoval sběr a analýzu požadavků, návrh aplikace včetně sepsání scénářů, implementaci samotné aplikace s využitím webových technologií, optimalizaci pro zajištění efektivity a rychlosti aplikace, a nakonec nasazení na produkci.

Během vytváření diplomové práce docházelo k pravidelnému zaznamenávání časových úseků věnovaných jednotlivým fázím procesu. Tyto intervaly byly evidovány a jsou uvedeny na Obrázku 1.

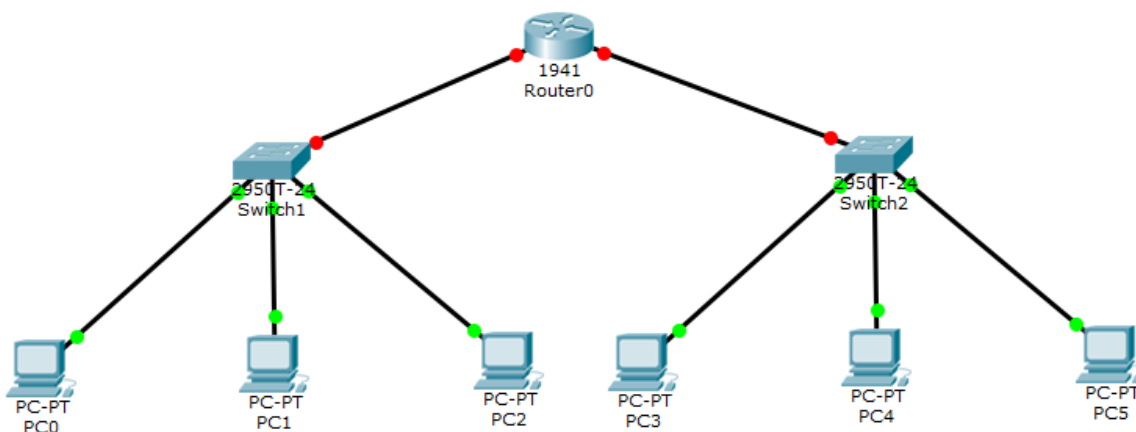
Měsíce	Studium problematiky	Návrh Softwaru	Programování	Práce s AI	Psaní	Celkem
Do Září	20	1	15	3		39
Říjen	5		9	2	2	18
Listopad		1	10			11
Prosinec		2	16	1	2	21
Leden	2		3			5
Únor	1	3	33		40	77
Březen	8	16	38	5	26	93
Duben	2		2		82	86
CELKEM	38	23	126	11	152	350

Obrázek 1 Časové úseky strávené nad diplomovou prací

V rámci diplomové práce byl proveden průzkum aplikací a her zaměřených na podporu výuky počítačových sítí. Tyto podpůrné aplikace byly poté rozděleny do tří základních skupin a to simulátory, online testy a mobilní aplikace.

### 1.1 Simulátory sítí

Tvorbou a simulací počítačových sítí se zabývají simulátory. Jedná se o typ aplikace, jejímž hlavním cílem je umožnit uživateli experimenty a praktická cvičení bez využití fyzických zařízení. Simulace nemusí být využity pouze pro výuku, ale také pro testování a ladění síťových konfigurací bez rizika poškození reálné sítě [1]. Mezi populární simulátory patří Cisco Packet Tracer, GNS3 a Netsim. Za nevýhodu lze považovat, že uživatel musí mít alespoň základní znalosti počítačových sítí, aby se byl schopný v simulátoru orientovat.

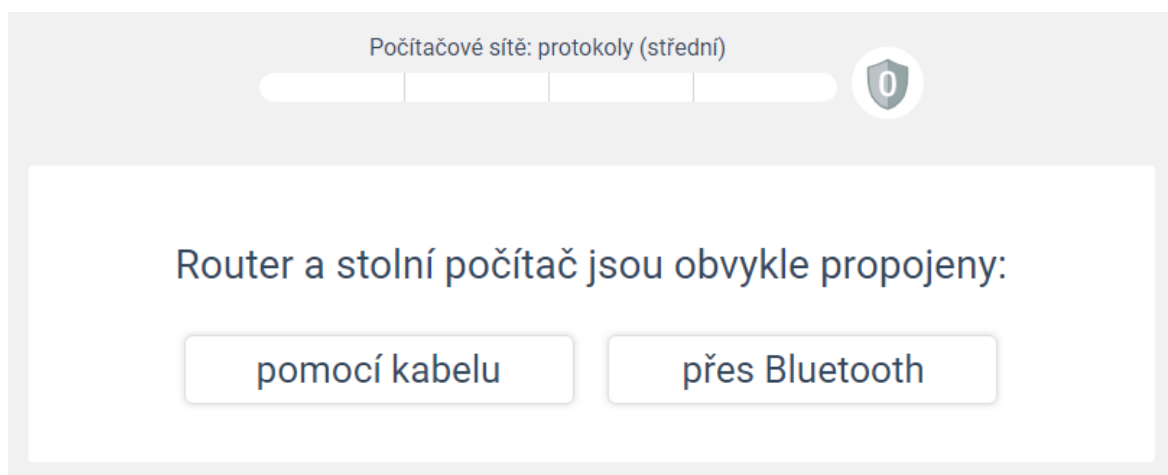


Obrázek 2 Síť ze simulátoru Cisco Packet Tracer

## 1.2 Online testy

Pro podporu výuky základů počítačových sítí lze využít také online testů. U aplikací tohoto typu má uživatel k dispozici více možností, mezi kterými je minimálně jedna správná odpověď. Testy lze tvořit také prostřednictvím různých kvízových aplikací jako je Quizizz.

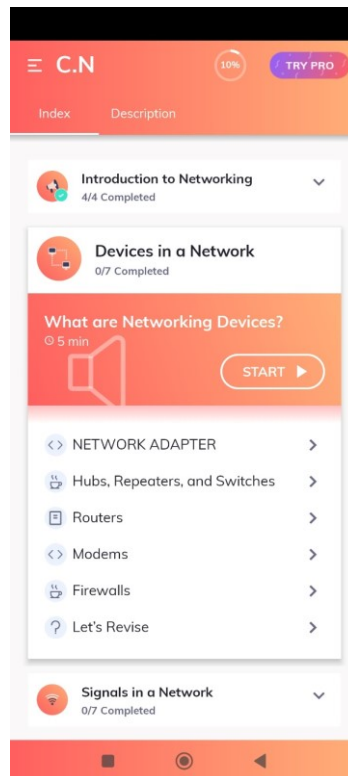
Jednou z webových stránek, která nabízí výuku počítačových sítí formou testů, je *umimeinformatiku.cz*. Tento web nabízí více testů z oblasti počítačových sítí, a navíc poskytuje také výukové materiály [2]. Výhodou může být rovněž možnost volby náročnosti testu mezi lehkou, střední a těžkou.



Obrázek 3 Snímek obrazovky testu na odkaze umimeinformatiku.cz

### 1.3 Mobilní aplikace Computer Network Tutorials

Mobilní aplikace Computer Network Tutorials je určena pro začátečníky v oblasti počítačových sítí. Součástí aplikace jsou výukové materiály, které se skládají z textových popisů, grafů, obrázků a videí. Při používání aplikace dochází k pravidelnému testování znalostí, bez kterého není možné pokračovat v rozvoji. Uživatelé, kteří dokončí proces výuky, dostanou navíc certifikát [3].



Obrázek 4 Snímek obrazovky aplikace Computer Network Tutorials

## 2 VÝVOJ POČÍTAČOVÝCH HER

V úvodu literární rešerše je nutné objasnit pojem počítačová hra. Jedná se o interaktivní elektronickou hru, jejímž hlavním cílem je pobavit hráče [4]. Tyto elektronické hry lze kategorizovat dle různých charakteristik jako jsou herní interakce, způsob hraní her a typ platformy. Vzhledem k velkému množství charakteristik a různorodosti hráčů je dnes možné pozorovat také široké spektrum herních žánrů, zahrnující akční, dobrodružné, bojové, závodní, RPG, střílečky, simulátory, sportovní a strategické hry [4].

V současnosti jsou počítačové hry a jejich hraní běžné téměř v každé domácnosti. Pokud se ovšem vrátíme k počátkům počítačových her, zjistíme, že skutečný začátek měly ve vědeckých laboratořích a byly součástí doktorských disertačních prací. Jako první hru považujeme z historického hlediska OXO, kterou v České republice známe pod názvem piškvorky. Byla vytvořena britským profesorem Alexandrem Shafto Douglasem, který tou dobou studoval na univerzitě v Cambridge. Mezi další významné hry, které byly vytvořeny, řadíme Tennis For Two a Spacewar [5].

Postupně docházelo k vývoji herního průmyslu a došlo také k vytvoření prvních herních konzolí, arkádových her, 3D her a postupem času se herní průmysl vyvinul až k hraní počítačových her pomocí webových stránek a sociálních médií. Dnes je již standardem hrát hry na většině zařízení včetně mobilních telefonů, tabletů a VR brýlí [5].

Pro vývoj her lze v dnešní době využít více možností. Mezi tradiční způsoby vývoje počítačových her patří využití herních engineů, programovacích jazyků a webových technologií.

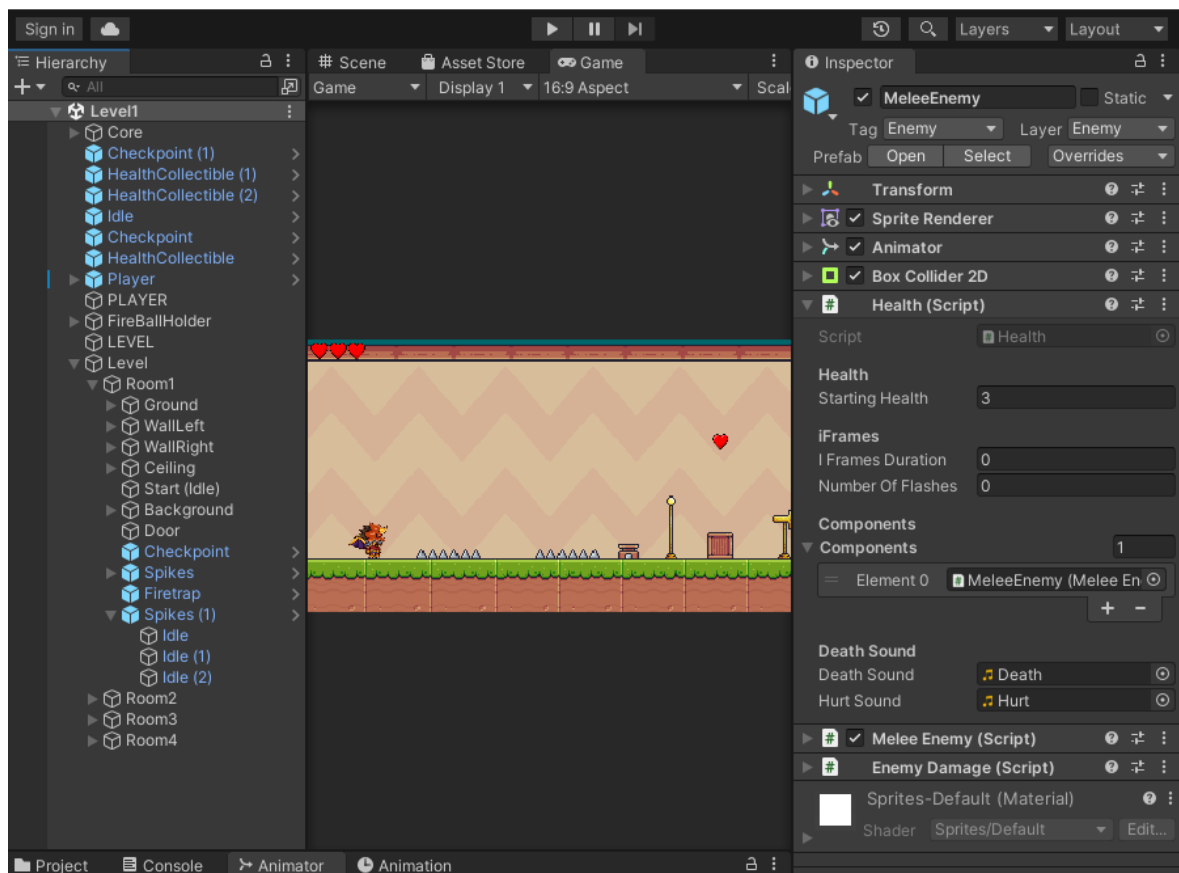
### 2.1 Herní enginey

Herní enginey začaly vznikat v polovině 90. let a poprvé se objevily u stříleček, které využívaly hru z pohledu první osoby. Jednou z populárních her postavených na herním engineu je Doom, který byl koncipován s cílem oddělit základní softwarové komponenty, jako je detekce kolizí, audio systém a 3D grafický vykreslovací systém, od grafických prvků, herních světů a pravidel hry [6]. Díky tomu mohli vývojáři opakovaně využívat stejné softwarové komponenty společně s různými grafickými prvky, odlišnými herními světy a rozdílnými pravidly. Termín herní engine by bylo možné definovat jako software, který lze snadno rozšířit a využít bez větších uprav jako základ pro více her [6]. Zároveň je nutné konstatovat, že žádný herní engine není univerzální pro všechny žánry her a pro všechny

typy hardwaru. Díky tomu tedy můžeme říct, že čím více je herní engine obecný, tím méně optimální bude pro vývoj konkrétní hry na dané platformě [6].

Nejrozšířenějším herním enginem je v současné době Unreal Engine, který byl vyvinutý společností Epic Games v roce 1998 [7]. Výhodou je, že jej lze využít pro vývoj her na různých platformách. Mezi další výhody spadá především jeho vysoký výkon v porovnání s ostatními enginy a také lepší technologie pro práci s grafikou. I díky tomu jsou vývojáři schopni s jeho pomocí sestavit téměř jakoukoliv 3D hru, která využívá pohled z první anebo třetí osoby [6]. Unreal Engine umožňuje vytvářet 2D i 3D hry [7].

Herní engine Unity je stejně jako Unreal Engine multiplatformní a umožňuje vývoj 2D i 3D her [7]. Pro vývojáře je velkou výhodou jeho intuitivní prostředí editoru (viz Obrázek 5), díky kterému lze jednoduše vytvářet a manipulovat s jednotlivými prvky hry, které tvoří herní světy. Součástí editoru je také možnost rychlého zobrazení hry. Další výhodou je sada nástrojů sloužících k analyzování a optimalizaci hry pro jednotlivé platformy. Standartními jazyky pro skriptování, které podporuje herní engine Unity, jsou C#, Javascript a Boo [6].



Obrázek 5 Editor herního enginu Unity



Dalším herním enginem, který lze využít pro vývoj počítačových her, je GameMaker Studio. V rámci tohoto editoru je možnost implementace jednotlivých prvků do hry za pomoci *drag and drop* funkce, díky které lze vytvořit hru prakticky bez znalosti programování [7]. Nevýhodou u této funkce je, že vývojáři jsou až příliš omezeni na předdefinované prvky editoru. Další možností pro implementaci jednotlivých částí hry je využití skriptovacího jazyka GLM, což je speciální programovací jazyk GameMakeru [8]. Engine umožňuje vývoj 2D i 3D her [8].

Amazon Lumberyard je herní engine, který je zaměřený pouze na 3D hry [7]. Jeho unikátní vlastností je, že se zaměřuje na hry pro více hráčů, což ale neznamená, že neumožňuje také vývoj her pro jednoho hráče. Druhou specifickou vlastností je zaměření na hry typu AAA, což je označení pro hry, které kladou důraz na nejvyšší kvalitu, ale také na velmi vysoký rozpočet [9]. Z tohoto důvodu si vývoj pomocí Amazon Lumberyard mohou dovolit pouze přední studia, která mají k dispozici dostatečný finanční kapitál.

Mezi další významné herní enginy by bylo možné zařadit CryEngine, Godot a Cocos2d [7]. Pro vývoj počítačových her lze využít širokou škálu herních enginů, kdy zásadními kritérii pro jejich volbu jsou požadavky na hardware, grafické schopnosti, herní žánr a volba platformy. Optimální herní engine by měl vývojářům především usnadnit práci a zkrátit dobu vývoje výsledného softwaru.

## 2.2 Programovací jazyky

Další možností vývoje počítačových her je využití výhradně programovacích jazyků bez herních enginů nebo jiných externích nástrojů. Obecně lze říci, že varianta programování počítačových her jen za pomoci programovacího jazyka je v dnešní době méně častá než s využitím herních enginů [10]. Toto tvrzení souvisí s tím, že naprogramování hry bez využití enginu je mnohem náročnější než s ním [11]. Na druhou stranu si vývojáři mohou vytvořit implementaci přímo na míru danému hernímu problému. Každý programovací jazyk má své výhody a nevýhody. Mezi nejoblíbenější jazyky, v oblasti vývoje počítačových her, řadíme C++, C#, Javu a Python [12].

C++ je staticky typovaný a kompilovaný jazyk, díky čemuž pracuje velmi efektivně s pamětí [12]. Další jeho výhodou je kontrola nad hardwarem a grafickými procesy, což umožňuje při správné implementaci zdrojového kódu vyšší rychlost oproti ostatním jazykům [12]. Z tohoto důvodu je ale také jazyk C++ náročnější na naučení než ostatní programovací jazyky, což je zároveň i jeho největší nevýhodou.

C# je programovací jazyk, který je výhodný především z důvodu přístupu k velké sadě knihoven a vývojových nástrojů. Pokud hledáme objektivě orientovaný jazyk, který je staticky typovaný a na rozdíl od C++ velmi dobře čitelný, snadno použitelný a jednoduchý na naučení, jednou z optimálních možností může být právě C# [12].

Java je velmi univerzální jazyk, který lze využít i pro programování počítačových her. Jeho hlavní výhodou je vestavěná přenositelnost pomocí JVM [12]. Díky tomu jsme schopni spustit kód Javy na různých platformách. Součástí implementace Javy je také programování socketů, čehož lze velmi dobře využít při tvorbě her pro více hráčů. Její největší nevýhodou je omezený přístup k hardwaru nízké úrovně a větší velikost souborů her než u ostatních programovacích jazyků [12].

V případě, že by bylo potřeba využít interpretovaný programovací jazyk vysoké úrovně, vhodnou volbou je Python, který může být také ideálním řešením, zejména pokud vývojář potřebuje vytvořit rychlý prototyp hry. Dalšími jeho přednostmi jsou jednoduchá syntaxe a velká standardní knihovna pro podporu grafiky, počítačových sítí a zvuku. Naopak jeho nevýhodou je, že nemá žádný přístup k hardwaru a správě paměti, což může negativně ovlivňovat rychlost výsledné hry [12].

### 2.3 Webové hry

Díky webovým technologiím mohou vývojáři implementovat hry pro počítač také prostřednictvím webu. V dnešní době jsou již webové hry srovnatelné s kvalitou her, které jsou vytvářeny přímo pro počítače a jejich nativní operační systémy [13]. Webové hry navíc poskytují celou řadu výhod pro vývojáře. První výhodou je, že vývojáři nemusí cílit pouze na hráče počítačových her. Díky responzivnímu designu a rozvoji chytrých zařízení lze webovou hru tvořit zároveň pro telefony, tablety, počítače, a dokonce i chytré televizory [13]. Dalším přínosem je, že nedochází k omezení propagace softwaru v cizím obchodě. Webovou hru lze propagovat svépomocí, a navíc veškerý příjem ze hry patří vývojáři, který nemusí poskytovat podíl z příjmu žádné společnosti zajišťující distribuci produktu [13]. Třetí výhodou je, že jsme kdykoliv schopni hru sami aktualizovat, předělat nebo pozměnit. Tohle může být užitečné především z důvodu, že jsme schopni rychleji reagovat na zjištěnou chybu a nemusíme čekat na schválení od externí společnosti, která nám zprostředkovává distribuci [13]. V rámci benefitů je zařazena také možnost vlastní analýzy [13]. Díky tomuto bodu jsme schopni sledovat informace, které jsou pro nás skutečně důležité a ne ty, které nám poskytne někdo jiný. Posledním významným benefitem,

který přináší webové hry, je, že hráči mohou hrát hru kdekoliv a kdykoliv. Této výhody lze využít u her, které potřebují pravidelnou kontrolu aktuálního stavu.

Kromě výhod lze ale také u webových her nalézt spoustu nevýhod. Obecně jako největší nevýhoda webových her je považována závislost na internetovém připojení, bez kterého hru nelze spustit. Dalším záporným bodem je, že z historického hlediska nikdy nedosáhly příjmy z webových her příjmům z her vydaných na konkrétních platformách i přesto, že hratelnost, grafika a zábava webových her byly na srovnatelné úrovni [14]. Velkou nevýhodou jsou technologická omezení. I přesto, že v dnešní době jsou moderní prohlížeče velmi výkonné, nemohou se měřit s nativními hrami, které mají přímý přístup k hardwaru a správě paměti [14].

## **2.4 Vývoj her a fáze vývoje**

Vývojový cyklus nové počítačové hry se u jednotlivých autorů a vývojářských studií liší, ale i přesto postup vývoje počítačové hry standardně probíhá ve třech hlavních fázích, mezi které řadíme před-produkční, produkční a po-produkční fázi [15].

### **2.4.1 Před-produkční fáze**

Jedná se o počáteční fázi vývoje. Zde by mělo dojít k představení hlavního nápadu na hru. Autor definuje, o čem finální hra bude a určí cílové publikum, na které se bude hra zaměřovat [15]. Dochází zde k průzkumu potenciálního trhu a především konkurence. Také je určeno několik zásadních aspektů, jako platforma, na kterou bude vývoj zaměřen, časový interval vývoje, rozpočet na dokončení hry a jakým způsobem bude výsledný produkt zpeněžen [16].

V rámci této fáze dochází také k vypracování dokumentu, který nese název GDD [15]. Tento dokument slouží k tomu, aby pomohl proniknout do vize projektu a je průběžně aktualizován současně s vývojem produktu. Dokument zachycuje průběh hry, příběh, žánr a mechanismy, které by měly být během hraní využívány [15].

### **2.4.2 Produkční fáze**

Produkční fáze je obecně považována za nejdélší fázi vývoje [15]. Dochází zde ke zdokonalení příběhové linie, nastavení pravidel hry, navržení světů a jednotlivých úrovní a vytváření nových prvků hry, mezi které řadíme např. postavy, rekvizity nebo prostředí.

Součástí této fáze je také získání určitých milníků, kterých by počítačová hra měla dosáhnout [15].

Prvním milníkem produkční fáze je vytvoření prototypu. Jedná se o počáteční test hry, ve kterém dochází k prvotnímu představení. Při reálném vývoji se často hra ani do této fáze nedostane [15]. Vytváření prototypu je jediná fáze, která vzniká už v před-produkci, ale prototyp je často dokončen a představen až v produkční fázi.

V další části, která se nazývá první hratelnou verzí, dochází k prezentaci základního designu a hratelnosti hry [15]. Ačkoliv si v této části vývoje mohou již členové vývojářského týmu hry vyzkoušet, jedná se o velmi omezenou verzi a většina herních prvků je v budoucnosti nahrazena.

Z hlediska zisku financí je velmi podstatným milníkem třetí fáze a tou je vertikální řez. Jedná se o etapu hry, která slouží k zaujmutí investorů [15]. Většinou se jedná o krátkou část hry, kde se snaží vývojáři svou hru co nejlépe představit.

Čtvrtou fází je pre-alfa, ve které by měla být již vyvinuta většina obsahu finální počítačové hry [15]. V této etapě také dochází k poslednímu klíčovému rozhodnutí ohledně obsahu výsledné hry, a tím je rozhodnutí, zda má být hra zjednodušena či naopak rozšířena o další nezbytné prvky [15].

Jakmile vývojářský tým dokončí etapu pre-alfa, přichází na řadu fáze alfa. Součástí tohoto milníku je již kompletně vytvořená počítačová hra, která by měla být plně hratelná od samého počátku až do konce a měly by být implementovány veškeré herní prvky [15]. Co se týká zvukových a grafických efektů, tak by měly být již součástí hry, ale může se jednat o zástupné prvky, které budou ještě v budoucnu nahrazeny. Hlavními členy, účastníky se této fáze, jsou testeři, kteří mají za úkol odhalit a nahlásit nalezené chyby vývojářskému týmu [15].

Předposledním milníkem, který se týká vývoje, je beta fáze. V tento moment by měl být součástí hry již kompletní obsah a veškeré grafické i zvukové prvky by měly být implementovány ve hře. Hra by také měla být plně kompatibilní pro specifikovaný hardware a software. Vývojářský tým se v této fázi soustředí spíše na optimalizaci a případné přidávání nových funkcí [15]. Pro tuto etapu je také velmi důležité, že ji netestují pouze testeři, kteří jsou součástí vývojářského týmu, ale také běžní lidé.

Závěrečná etapa se nazývá gold master. Jedná se o finální verzi počítačové hry, která bude představena veřejnosti a odeslána vydavatelům [15].

### 2.4.3 Po-produkční fáze

Ačkoliv je výroba počítačové hry dokončena, vývojářská práce tím nekončí. Obvyklý postup po-produkční fáze je takový, že část vývojářského týmu je přesunutá na jiný projekt, ale část týmu zůstává na dosavadním projektu a stará se o jeho údržbu [15]. Součástí této údržby může být opravování nalezených chyb, vytváření bonusových částí nebo vytváření nových rozšíření hry v podobě DLC. Pro tuto etapu je podstatná zpětná vazba od hráčů, na kterou by se vývojáři měli snažit co nejvíce reagovat kvůli dlouhodobé životnosti projektu [16].

## 2.5 Trendy ve vývoji her

Veškeré oblasti průmyslu mají vždy určité trendy, které jsou v populaci aktuálně oblíbené. Jinak to není ani v případě herního průmyslu, který se v průběhu 21. století velmi změnil.

Populárním trendem jsou nyní hry typu AAA. Je to z důvodu toho, že nabízejí herní kvalitu a zážitek nejvyšší úrovně. Mezi tyto hry řadíme třeba GTA 5, Witcher 3 nebo Far Cry 5. Tyto hry často poskytují až fotorealistickou grafiku (viz Obrázek 6), což ovšem vyžaduje rozsáhlý tým zkušených vývojářů a spoustu financí [17].



Obrázek 6 GTA V – AAA hra [15]

Druhým trendem je využití cloudových her. Jedná se o princip, kdy uživatel platí poplatek v podobě předplatného a může využívat hraní her na vzdálených serverech [17]. Tyto servery současně vrací stream uživateli, díky čemuž hráč nemusí vlastnit žádné výkonné zařízení, protože veškeré výpočty implementované ve hře probíhají na straně serveru.

Třetí bod, který je dnes velmi populární v herní komunitě, je hra typu As-An-Art. Jedná se o trend, který se snaží vnímat hraní her jako formu umění [17]. Prostřednictvím těchto her

dochází k osobnímu spojení mezi uživateli, a dokonce typ těchto her dokáže v hráčích vyvolávat silné emoce a ovlivnit jejich chování.

Mezi další populární trendy se řadí využití rozšířené reality, vývoj her pro mobilní platformy a využívání sítě typu 5G díky její vyšší rychlosti a nižší latenci oproti předchozím generacím [17].

## 2.6 Herní design

Herní design je velmi důležitá část herního vývoje. Jedná se o oblast, která velmi ovlivňuje výslednou hratelnost hry a je založena především na zkušenostech samotných hráčů. Z hlediska herního designu je zásadní především poskytnout zážitek ze hry cílové skupině hráčů. Na druhou stranu se designéři musí snažit za pomoci inovací vytvořit dílo, které výslednou hru odliší od všech ostatních, jenž jsou k dispozici na trhu [18].

### 2.6.1 Způsoby inovací ve hrách

Inovace her není snadnou záležitostí, vzhledem k tomu, že hráči jsou zvyklí na určitý standard v oblasti hraní počítačových her. Pokud by došlo k příliš velkým změnám, mohli by to hráči vnímat naopak spíše negativně a hra by nemusela dosáhnout požadovaných výsledků [18].

Mezi tradiční oblasti z hlediska inovace řadíme navržení unikátních herních mechanik, které se v ostatních hrách nevyskytují. Dalšími možnostmi je proniknutí do platforem, které nejsou tak hojně využívány, vytvoření her inspirovaných každodenním životem a poskytnutí nových modelů, které hráčům poskytují určitou výhodu, jako je hraní hry zdarma [18].

Z hlediska designu jsou pro hru klíčové čtyři hlavní problémy. Mezi tyto problémy řadíme vytvoření příběhové linie a její hratelnosti, prohloubení empatických schopností postav ve hrách, implementace prvků vzbuzujících v hráči emociální reakci a snaha o objevení vztahů mezi hrou a učením [18].

## 2.7 Vývoj her typu goal-oriented a game-based learning

Moderní hry se často vyznačují tím, že jsou typu goal-oriented. Tento typ her je založený na tom, že hráč je motivován k postupnému dosažení lepších výsledků. Motivací často bývá porážení oponenta, ostatních hráčů nebo přiblížení se k cíli hry. Tohoto přístupu lze využít u her typu game-based learning, které se ale navíc zaměřují na to, aby hráče vzdělávaly

v určité oblasti. Díky tomu lze dosáhnout úspěchů při vzdělávání za pomoci hraní počítačových her [19].

### 3 TECHNOLOGIE VYUŽÍVANÉ PŘI VÝVOJI HRY

Pro vývoj webové hry, která je součástí této diplomové práce, byly využity určité technologie. Mezi tyto technologie patří programovací jazyky pro vývoj webových aplikací, grafický jazyk pro modelování softwaru, UX/UI design, systém pro správu verzí, vhodné IDE a také systémy založené na umělé inteligenci.

#### 3.1 Webové technologie

S postupným rozvojem webových stránek a aplikací se objevily také určité techniky a pravidla. Pokud vývojář chce, aby web fungoval a vypadal dle nějakých předpokladů, musí znát alespoň základní webové technologie, mezi které řadíme programovací jazyky HTML, CSS a Javascript [20].

##### 3.1.1 HTML

Značkový jazyk HTML vznikl v 90. letech 20. století a byl postaven na vytvoření dokumentu, jehož součástí jsou elementy reprezentující jednotlivé části webové stránky, jako nadpis, textová oblast nebo odstavec. Tyto elementy jsou často označovány jako značky. Postupem času se jazyk vyvíjel až do současné verze HTML5. Při vytváření nových verzí jazyka HTML se klade důraz na zpětnou kompatibilitu se staršími verzemi webových prohlížečů. Díky této vlastnosti lze spustit webové stránky postavené na verzi HTML5 i na zastaralých prohlížečích [21].

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Titulek</title>
  </head>
  <body>
    Obsah stránky
  </body>
</html>
```

Obrázek 7 Základní struktura HTML kódu

Syntaxe jazyka HTML je založena na psaní tagů, což jsou elementy, které jsou obklopeny špičatými závorkami. Pro tagy není důležité, zda jsou napsány pomocí malých nebo velkých písmen, a dokonce písmo můžeme i kombinovat a neovlivní to jejich funkčnost [22]. Jako standard se ale využívá malé písmo. Tagy můžeme poté rozdělit na párové a nepárové [21].



Párové tagy se píšou ve dvojicích, kdy první tag označujeme jako otevírací a druhý jako uzavírací, jehož součástí je kromě elementu i lomítko. Typickým příkladem párového tagu může být odstavec, který musí někde začít a někde skončit. V případě nepárových tagů se využívá pouze jeden samostatný tag. Většinou se jedná o tagy, do kterých se standardně nekládá již další obsah. Za typický nepárový tag lze považovat obrázek.

Jednotlivé tagy mohou také nést informace o obsahu v podobě atributů. Tyto atributy jsou zapisovány za názvem elementu a jeden tag jich může obsahovat více. Zápis atributů je složený ze tří částí, kterými jsou název, přiřazovací znak rovnítko a hodnota atributu, která se standardně píše do uvozovek, ale není to podmínkou [21].

### 3.1.2 CSS

CSS je kaskádový jazyk, který vzniknul v roce 1996. Současná verze je CSS3, která je také zpětně kompatibilní s předchozími verzemi [21]. Jedná se o jazyk, který je stejně jako HTML5 standardizován ve všech webových prohlížečích a slouží k popisu toho, jak by měly jednotlivé elementy jazyka HTML vypadat [23].

V dnešní době se CSS využívá především ke stylování prvků vytvořených značkovacím jazykem. Díky stylování lze změnit barvu textu, barvu pozadí, velikost písma, odsazení a mnoho dalších estetických vlastností [24].

Mezi jeho klíčové vlastnosti kaskádového jazyka verze CSS3 spadá plná podpora všech moderních prohlížečů, vytváření pokročilých animací a responzivní design, který nám umožňuje tvorbu webových stránek pro různé typy zařízení [24]. Pro tvorbu responzivního designu využívá media queries, které umožňují jednodušší adaptaci na konkrétní zařízení a specifikaci stylizačních pravidel.

Dalšími zásadními vlastnostmi, které jsou jen součástí poslední verze CSS3, jsou seskupení kódu do modulů, podpora vícesloupcových textových bloků a využívání různých formátů pro definování barvy. Mezi tyto formáty spadají RGB, RGBA, HSL, HSLA, hexadecimální kód a také anglický název barvy [24].

### 3.1.3 Javascript

Programovací jazyk, který zprostředkovává dynamický obsah na webu, se nazývá Javascript. Mezi jeho základní schopnosti patří aktualizování obsahu, animování prvků na stránce a ovládání multimédií. V základu je Javascript využíván na straně klienta a umožňuje nám využívat běžné programovací funkce jako ukládání hodnot do proměnných, operace

s proměnnými a spouštění kódu v reakci na nějakou událost, která nastala na webové stránce [25].

Pro usnadnění práce vývojářů bylo do Javascriptu implementováno rozhraní API, jehož prostřednictvím jsou vývojáři schopni realizovat jinak velmi složitou logiku do výsledné aplikace. Součástí implementace webových prohlížečů je několik základních rozhraní API. Pro manipulaci s HTML a CSS prostřednictvím jazyka Javascript lze využít DOM API. Díky němu jsou vývojáři schopni za pomoci Javascriptu tvořit, editovat a odstraňovat obsah HTML nebo měnit stylování jednotlivých elementů [25]. Druhým rozhraním, které nám poskytuje možnost získání geografických informací, je Geolocation API. Tuto vlastnost lze využít pro nalezení polohy a souřadnic zařízení. Při vytváření animované 2D nebo 3D grafiky se využívá dalších dvou rozhraní API pro Canvas a WebGL, což může být prospěšné při vývoji webových her [25]. Posledním rozhraním, které je zásadní pro vývoj moderních webových stránek, je API pro audio a video. Zde stejně jako u grafiky lze volit ze dvou technologií, a to HTMLMediaElement a WebRTC [25].

Speciálním typem HTML elementu je *canvas*, který slouží pro vytvoření prázdného pozadí, na které lze vykreslit libovolný grafický prvek. Prostřednictvím elementu *canvas* lze poté přistoupit k API, které nám umožní kreslení tvarů a čar. Důležité je, že s využitím DOM API, lze přistoupit k prvku typu *canvas* a měnit jeho vlastnosti jako velikost, tvar, barvu a jiné [26].

### 3.1.4 OOP v Javascriptu

OOP je způsob programování založený na vytváření tříd a jejich instancí, které jsou nazývány objekty. Součástí většiny tříd jsou poté atributy a metody, které mohou jejich instance používat. Třída samotná není reálným objektem, ani ho nijak nezastupuje. Naopak její instance mohou reprezentovat objekty z reálného světa.

V programovacím jazyce Javascript se třídy vytvářejí pomocí klíčového slova *class*. Jedná se o slovo, které je vyhrazeno pouze pro tvorbu nových tříd a nelze jej využít pro jiný účel [27].

Využívání OOP je založeno na čtyřech principech a to abstrakci, zapouzdření, dědictví a polymorfismu. Co se týká abstrakce, jedná se o pojem, který zajišťuje skrytí určité aplikační logiky, která není z uživatelského hlediska důležitá. Druhým pojmem je zapouzdření, které nám umožňuje tvorbu interních metod a vlastností, ke kterým lze přistupovat pouze zevnitř třídy. Javascript využívá specifickou syntaxi pro vytvoření soukromé vlastnosti nebo

metody. Ve většině programovacích jazyků je využíváno klíčového slova `private`, ale v Javascriptu musíme využít znaku `#`, který se píše před název vlastnosti nebo metody [27]. Dalším důležitým termínem, z oblasti OOP, je dědičnost. Princip dědičnosti je založený na rodičovské třídě, ze které můžeme dědit, a třídě potomka, který dědí z rodičovské třídy. Díky tomuto konceptu jsme schopni opakovaně využít společnou logiku, a dokonce můžeme v rodičovské třídě určit, které metody a vlastnosti bude moci dědit třída potomka. Díky syntaxi ES6 je dnes velmi jednoduché vytvořit dědičnost, a to pomocí klíčového slova *extends* [27]. Poslední termín je polymorfismus, který nám umožňuje přepis zděděných metod.

Další důležitou částí z hlediska principu OOP v Javascriptu je metoda konstruktoru. Jedná se o speciální metodu, která je volána ihned po vytvoření instance třídy. V případě, kdy metoda konstruktoru neexistuje, je zavolán výchozí konstruktor, který nic nedělá [27].

Při implementaci logiky třídy se často pracuje s metodami *getteru* a *setteru*. Jedná se o velmi jednoduché metody, kdy *getter* vrací hodnotu některé z vlastností a *setter* naopak hodnotu vlastnosti nastavuje.

## 3.2 Návrh aplikace

UML je mechanismus, který slouží pro modelování softwarových systémů. Jedná se o standard sloužící k zachycení požadavků nových systémů, ale také může pomoci porozumět již funkčním systémům a aplikacím. Modely vytvořené za pomoci jazyka UML mohou usnadnit tvorbu dokumentace a scénářů pro rozsáhlé softwarové systémy. Celkově lze využít 14 typů unikátních modelů [28]. Současně s vytvářením UML diagramů vzniká také grafický návrh aplikace za pomoci UX/UI designu a wireframů.

### 3.2.1 Účely využití UML

Dnes existuje 5 základních účelů pro využití jazyka UML v oblasti softwarového inženýrství [28].

Primárním účelem je vizualizace, díky které jsme schopni lépe zachytit požadavky, procesy, navržené řešení a architekturu [28].

Druhý důvod, proč je důležité používání jazyka UML, je zkvalitnění popisu výsledného softwaru [28]. Při komunikaci mezi zákazníky a vývojáři může docházet často k nesrovnalostem, ale s pomocí UML a vizualizace diagramů lze odstranit mnohá nedorozumění, čímž dochází k upřesnění specifikací pro výsledný produkt.

Třetí výhodou jazyka UML je možnost vygenerování zdrojového kódu přímo na základě vytvořených modelů [28]. Tento přístup může být výhodný pro vývojáře, kteří nemají dostatečné znalosti v oblasti programování, ale umí navrhnout adekvátní model, který vytvoří kód místo nich.

Čtvrtým účelem je vytvoření dokumentace [28]. Při vytváření dokumentace s využitím jazyka UML dochází k podrobnému popisu všech částí vývoje, což vede opět k odstranění nejasností mezi klientem a vývojáři.

Poslední částí je údržba stávajícího systému. V momentě, kdy již máme hotový produkt a potřebujeme v něm libovolnou část změnit, jsou UML modely velmi významnou pomocí pro vývojáře. Díky těmto modelům jsou vývojáři schopni rychle identifikovat správná místa, kde je potřeba udělat změny, a také pochopit vliv změny na celkový chod systém [28].

### 3.2.2 Analýza požadavků

Analýza a následný zisk systémových požadavků slouží k popisu funkcionality, kterou má aplikace vykonávat, určení služeb, jež má výsledný software poskytovat a také definici omezení systému. U požadavků je důležité, aby symbolizovaly potřeby zákazníků, využívané za určitým účelem, jako zadávání objednávek, změna údajů uživatele nebo možnost zrušení objednávky. Hledání optimálních požadavků nazýváme analýzou, na kterou navazuje dokumentace, testování služeb a omezení daného produktu. Požadavky můžeme rozdělit dvěma způsoby, kde první způsob dělení je na požadavky uživatelské a systémové. Druhá varianta rozdělení je na funkční a nefunkční požadavky [29].

Uživatelské požadavky jsou zaznamenány v přirozeném jazyce a následně doplněny za pomoci diagramů, které zachycují, jaké služby by měl systém za určitých omezení poskytovat jeho uživatelům. Důležité je, že mohou poskytovat určitou abstrakci, kterou poté řeší samotní vývojáři. Uživatelské požadavky jsou doplněny pomocí systémových požadavků, které slouží k zachycení detailnějšího popisu funkcí softwaru, jeho služeb a omezení. Dokument systémových požadavků by měl zajistit přesnou definici toho, co se má u výsledného produktu realizovat [29].

Druhou možností je rozdělení požadavků na funkční a nefunkční. Funkční požadavky udávají u systému, jakými službami bude vybaven, jakým způsobem by měl reagovat na konkrétní vstupy a jak by se měl chovat v určitých situacích. Obecně lze říci, že funkční požadavky popisují to, co by měl systém dělat [29]. Nefunkční požadavky poté slouží pro omezení služeb a funkcí, které jsou poskytovány vyvíjeným systémem. Jejich součástí je

řešení systémových problému typu časového omezení na reakci, omezení pomocí příslušných norem nebo spolehlivosti systému. Optimalizace nefunkčních požadavků je považována za více kritickou než optimalizace požadavků funkčních. Také je daleko náročnější testovat, zda systém nefunkční požadavky splňuje [29].

### 3.2.3 UC model

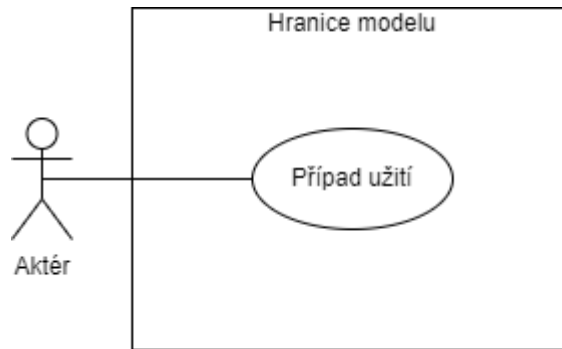
UC diagram slouží k poskytnutí celkového přehledu požadavků na systém v dané problémové oblasti. Součástí diagramů jsou aktéři, samotné případy užití, vztahy mezi případy a také poznámky. Výsledný model ukazuje, do které oblasti zapadají jednotliví aktéři a jakým způsobem mohou systém využívat. Do vytváření modelu jsou často zapojeni zákazníci, díky čemuž dochází k upřesnění požadavků [28].

Při sestavování UC diagramu dochází standardně ke stanovení hranice systému. Tyto hranice se zapisují pomocí obdélníkového rámečku, který slouží k oddělení aktérů od případů užití. Případy užití se nachází uvnitř obdélníku, a naopak aktéři mimo něj. I přesto, že hranice není nutnou podmínkou pro vytvoření UC modelu, je vhodné ji vytvořit pro zpřehlednění systému [28].

Součástí UC modelu mohou být i poznámky, které se využívají pro dovysvětlení vztahů mezi jednotlivými případy, poskytnutí podrobnějších informací o aktérech a také zdůraznění důležitosti podstatnějších případů užití [28]. Pro jejich zápis se využívá obdélník, který je v rohu seříznutý. Poznámky nemusí být nutně implementovány v diagramu, ale často přispívají k jeho lepšímu porozumění.

Jednou z hlavních složek, při vytváření UC diagramu, jsou aktéři. Jde o prvek, který reprezentuje role různých uživatelů využívajících vyvíjený systém. Aktéři nemusí nutně zobrazovat roli, která pracuje přímo se systémem, ale mohou být použiti jako rozhraní pro přístup k jinému externímu systému nebo zařízení [28]. Prvek aktéra je v UML diagramech zobrazen jako panáček.

Společně s aktéry jsou nezbytným prvkem UC modelu také případy užití, které slouží ke znázornění interakcí mezi aktéry a systémem. Pro zachycení detailů těchto interakcí se využívá scénářů. Případy užití, které jsou v grafech zobrazeny pomocí elipsy, se využívají k tomu, aby pokrývaly jednotlivé funkční požadavky. Každý případ musí pokrývat minimálně jeden požadavek a zároveň nesmí existovat požadavek, který by neřešil alespoň jeden případ užití [28].



Obrázek 8 Jednoduchý UC model

### 3.2.4 Vztahy uvnitř UC modelu

Vztahy lze rozdělit na 3 hlavní typy, a to mezi dvěma aktéry, mezi aktérem a případem užití a mezi dvěma případy užití. U těchto vzájemných vztahů mohou nastat 4 základní vazby a to asociace, dědičnost, include a extend.

Základní vazbou, která může uvnitř UC modelu nastat, je asociace. Jedná se o vztah zajišťující vzájemnou komunikaci mezi aktérem a případem užití [28]. Asociace se v diagramu zachycuje za pomoci přímky.

Další důležitou vazbou je dědičnost, která může nastat u dvou variant. Prvním variantou je dědičnost mezi aktéry, kdy nadřazený aktér předává svou funkcionalitu podřízenému [28]. Výhodou je, že nemusíme stejné případy užití vytvářet vícekrát. Druhou možností je dědičnost mezi dvěma případy užití [28]. Dědičnost, které se také říká zobecnění, se zobrazuje v UC diagramu za pomoci šipky.

Dvě vazby, které jsou specifické pouze pro případy užití, jsou include a extend. Vztah include se využívá v situaci, kdy součástí implementace chování nějakého případu užití je část, která se často opakuje. Z této části je tedy vytvořen nový rozšiřující případ užití, který je ale stále nutný pro správnou funkčnost původního hlavního případu [28]. V grafu zobrazujeme vazbu include jako šipku směřovanou od hlavního případu užití k případu, který ho rozšiřuje. Druhou vazbou je extend, vyjadřující vztah, který může, ale nemusí nastat při volání hlavního případu užití [28]. Tento vztah je opět zobrazen šipkou, ovšem tentokrát jde šipka od rozšiřujícího případu užití směrem k hlavnímu.

### 3.2.5 Scénáře

UC diagram slouží k zobrazení přehledu všech případů, které nám systém poskytuje. Pro upřesnění konkrétních případů užití se ale používají scénáře. Na začátku každého scénáře by měl být jeho název a stručný popis, který slouží pro charakteristiku hlavní funkce případu

užití. Další prvek, který by měl být součástí každého scénáře, je seznam aktérů. Tento seznam slouží pro zachycení všech možných aktérů, kteří se mohou případu užití zúčastnit, včetně systému samotného. Pro to, aby mohl být scénář vůbec realizovaný, musí být splněna vstupní podmínka. Jedná se o prvek, který není povinný, ale pokud je podmínka součástí scénáře a není naplněna, případ užití nemůže být uskutečněn. Nejdůležitější částí celého scénáře je základní a alternativní tok. Základní tok zachycuje jednotlivé interakce mezi aktéry a případy užití. Aktéři a systém se při interakcích střídají a postupně je tak za pomoci jednotlivých kroků vytvořen kompletní scénář, který by měl být zakončený splněním případu užití. Základní tok slouží pouze pro bezchybný průběh. V případě, že by nějaká neočekávaná situace u daného scénáře mohla nastat, musí být vytvořen alternativní tok, který slouží k řešení výjimek a chyb. Alternativní tok je vždy přiřazen k určité interakci ve scénáři. Po projití všech interakcí základního nebo alternativního toku může dojít k ukončení scénáře nebo ke kontrole pomocí dokončovací podmínky [30].

### 3.2.6 Návrh UX/UI designu

Prvek, který je nezbytný pro úspěšné navržení softwaru, je UX/UI design. I přesto, že oba pojmy UX i UI mají určité podobnosti, existuje spousta rozdílů mezi oběma přístupy návrhu [31].

Návrh UX se zabývá vývojem částí, které nejsou na první pohled viditelné uvnitř aplikace. Jeho hlavní funkcí je vytvoření návrhu produktu tak, aby splňoval veškeré uživatelské požadavky, které byly specifikovány. Jedná se o pojem, který se nemusí nutně vyskytovat pouze v informatice, ale můžeme se s ním setkat např. u výroby aut nebo elektroniky [32]. Při tvorbě nového softwaru využívají UX designéři analýzu požadavků a na základě jejího vyhodnocení vytvoří různá řešení, která by měla pokrýt potřeby cílové skupiny uživatelů, kteří budou produkt využívat [32].

Kromě navržení UX designu dochází také k vytváření návrhu UI, což je proces, při kterém se vývojáři zaměřují na viditelné části aplikace, především na vzhled a vizuální styl. Cílem UI designu je navržení rozhraní, které bude pro uživatele přívětivé, intuitivní a příjemné pro používání [33]. Součástí návrhu je vytváření čtyř základních prvků, a to navigace, vstupů, informačních komponent a kontejnerů.

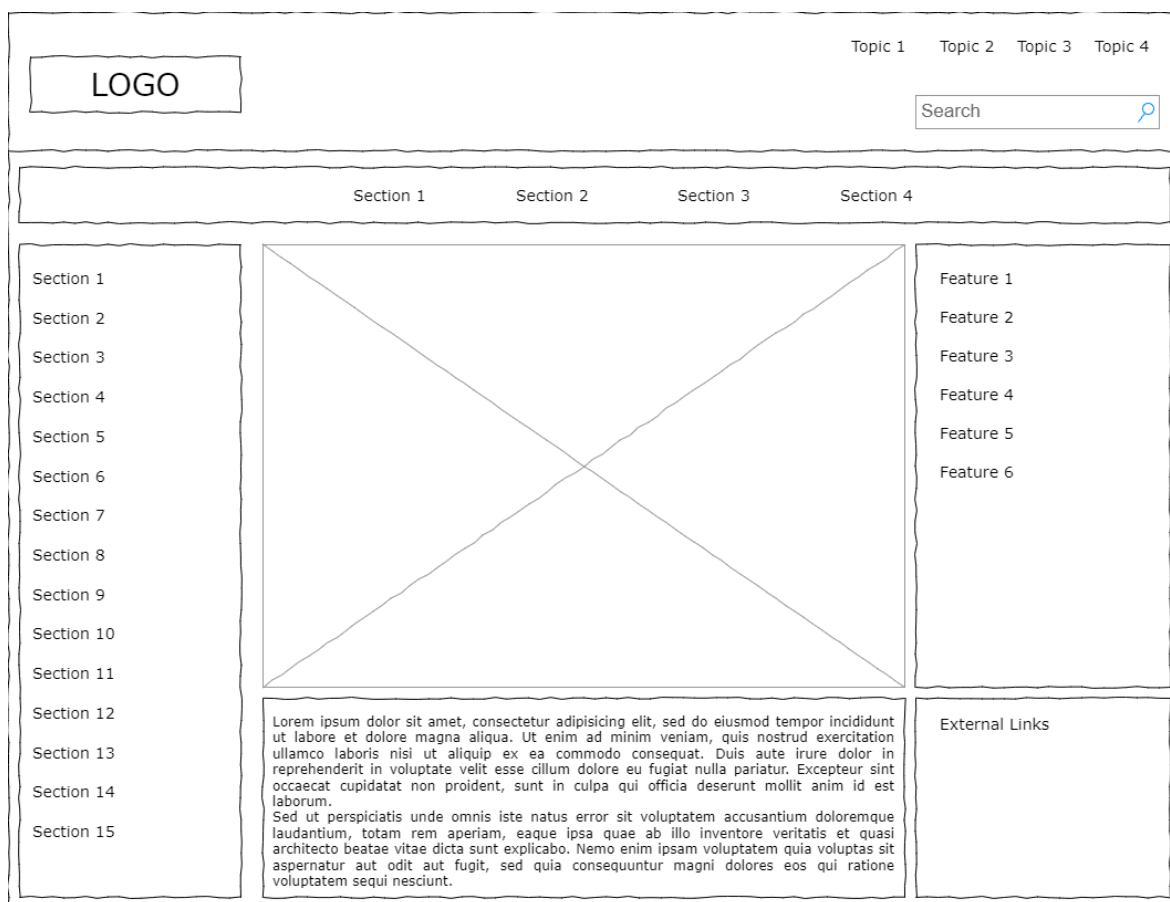
První prvek se zaměřuje na tvorbu navigací, které slouží pro lepší orientaci [31]. V případě aplikací se standardně jedná o navigační menu, zpětné šipky, posuvníky nebo vyhledávací pole.

Pro možnost zadávání informací od uživatele jsou využívány ovládací prvky pro vstupy [31]. Jedná se o běžné složky aplikací jako tlačítka, textová pole, checkboxy nebo kalendáře.

Třetím prvkem, využívaným při návrh UI designu, jsou informační komponenty. Jejich hlavní význam spočívá ve sdělení informací uživateli [31]. Příkladem takové komponenty může být ukazatel času nebo aktuální průběh videa [31].

Poslední základní složkou UI designu jsou kontejnery [31]. Jejich hlavní účel je organizace obsahu a jednotlivých částí aplikace do sekcí.

Pro implementaci UX/UI designu se při vývoji aplikací často využívá wireframů. Jedná se o návrh rozhraní systému, který se skládá z velmi jednoduchých vizuálních komponent a jejich rozložení v aplikaci [34]. Na počátku návrhu se využívá často pouze černobílých komponent, které by měly zachycovat jednotlivé interakce softwaru (viz Obrázek 9).



Obrázek 9 Jednoduchý wireframe

Poté, co zákazník schválí základní GUI, dochází k doplnění vizuálních efektů jako barev, textů a obrázků do návrhu wireframu. Postupně se tento proces několikrát zopakuje a během jednotlivých procesů dochází k úpravám vizuálních efektů a přidávání nových komponent, dokud nedojde k uspokojení potřeb zákazníka [34].



### 3.3 Systémy pro správu verzí

VCS jsou velmi důležitou součástí vývoje veškerých aplikací a využívá je většina vývojářů při tvorbě projektu. Hlavní účelem VCS je efektivní správa zdrojového kódu a poskytnutí přístupu ke všem předchozím verzím projektu [35]. Vytvoření nového softwaru vyžaduje spolupráci všech členů vývojového týmu, což VCS usnadňují. Mezi nejznámější VCS patří CMVC, Git, Apache Subversion, Azure DevOps, GitHub a Bitbucket [36]. Těchto systému lze na trhu nalézt daleko více.

Kromě toho, že systém umožňuje efektivní práci v týmu, zajišťuje také zálohování systému. Navíc máme kromě zálohy také vždy k dispozici nejaktuálnější verzi aplikace. Další výhodou je větvení a slučování projektu. Větev je kopie původního programu, který je nazýván hlavní linií [36]. Díky možnosti vytváření kopií lze docílit toho, aby každý vývojář pracoval sám a po dokončení práce může větev opět sloučit s hlavním programem. V případě, kdy je v systému nalezena chyba, může se vývojářský tým kdykoliv vrátit k předchozí verzi a snadněji určit co chybu způsobuje. Z těchto důvodů jsou dnes VCS velmi populární a považují se za standart při vývoji aplikací.

### 3.4 Vývojové prostředí

IDE je software, který slouží programátorům pro usnadnění vývoje aplikací. V dnešní době patří mezi standartní funkce IDE testování, ladění aplikace, doplňování kódu, vytváření virtuálního serveru a další [37]. V základní implementaci IDE je obvykle editor pro vytváření zdrojového kódu, nástroje pro debugování a systém pro sestavení aplikace. Programátor si poté do IDE může sám přidávat za pomoci pluginů nové funkce, které umožňují vývojářům rychlejší a efektivnější psaní zdrojového kódu. V dnešní době lze na trhu nalézt velkou spoustu populárních IDE jako např. Visual Studio Code, IntelliJ IDEA, PyCharm nebo Xcode [38].

Visual Studio Code je z dlouhodobého hlediska jedno z nejpoblárnějších IDE pro vývoj aplikací a bylo také zvoleno při vývoji této diplomové práce. Populární je především díky velké podpoře programovacího jazyka Javascript [37]. Navíc je to velmi jednoduché a výkonné IDE, které obsahuje velkou sbírku rozšíření, díky čemuž si jej vývojáři mohou upravit dle svých požadavků. Další jeho výhodou je integrace se systémem Git, což umožňuje efektivní správu verzí projektu. Součástí implementace Visual Studio Code je také velmi snadné ladění [38].

### 3.5 Umělá inteligence

Počítačové systémy, vykazující lidské kognitivní funkce a chování, nazýváme umělou inteligencí. Tyto systémy jsou založené na postupném učení, při kterém zpracovávají obrovské množství dat. Postupně jsou tak systémy schopné vytvořit vzorce, dle kterých se mohou rozhodovat. Do oblasti umělé inteligence spadají generativní nástroje pro tvorbu textů, obrázků nebo videí, chytrí asistenti, samořídící auta, senzory v nositelných zařízeních nebo vizuální filtry, které lze využívat prostřednictvím sociálních sítí [39]. Umělá inteligence se stala v posledních letech velmi populární, především díky generativním nástrojům jako ChatGPT, Gemini nebo Claude [39].

## **II. PRAKTICKÁ ČÁST**

## 4 NAVRŽENÍ ÚLOH NA ZÁKLADĚ KURIKULA SŠ

Ještě předtím, než se začala webová aplikace vytvářet, došlo k několika návrhům, jak by hra mohla vypadat. V první fázi návrhu se uvažovalo o hře, která by byla založená na otevřeném světě, ve kterém by uživatel mohl libovolně pochodovat a plnit různé zadané úlohy. Tato hra, která měla být inspirována žánrem RPG, zpočátku působila jako ideální nápad, jelikož principy RPG dokáží zabavit a uchvátit velké množství hráčů. Hlavním cílem hry ovšem nebylo co nejvíce pobavit osoby, které si hru zahrají, ale naučit je co nejvíce informací ze základů počítačových sítí. Z důvodu upřednostnění výuky před pobavením, a také díky přihlídnutí ke kurikulu SŠ složeného z témat, která by se velmi těžko implementovala do hry založené na otevřeném světě s žánrem RPG, byl původní nápad zavržen a nahrazen novým konceptem založeném na vytvoření několika interaktivních miniher. Díky tomuto principu lze daleko efektivněji implementovat jednotlivá témata počítačových sítí a z hlediska výuky by hra měla být mnohem efektivnější, protože uživatel se bude věnovat pouze danému tématu a nebude se zdržovat chozením po otevřeném světě. Zároveň budou součástí implementace dva herní módy.

První mód bude určený pro výuku začátečníků v oblasti počítačových sítí. Uživatel, který si tento mód zvolí, projde postupně všechny úrovně, které budou ve hře k dispozici. Cílem bude projít celou hrou a splnit tak veškeré interaktivní hry. Navíc vždy před každým začátkem minihry budou uživateli zobrazeny informace sloužící k pochopení dané problematiky počítačových sítí, což povede k průběžnému vzdělávání hráče. Název tohoto herního módu bude kampaň.

Ve druhém herním módu si uživatel bude moci spustit libovolnou minihru. Tento princip umožňuje opakované spuštění her, ve kterých se chce hráč zlepšovat. Navíc si uživatel bude schopen vyzkoušet i části hry, do kterých se v kampani ještě nedostal.

### 4.1 Průzkum kurikula SŠ

Ještě předtím, než se začaly jednotlivé interaktivní hry navrhovat, byl proveden průzkum školního vzdělávacího plánu pro předmět sítě a počítačové systémy, který je určený pro žáky Obchodní akademie v Kroměříži studující obor informačních technologií. Vzhledem k tomu, že obsah učiva daného předmětu je velmi rozsáhlý, byla pro tuto diplomovou práci použita pouze část vzdělávacího plánu určená pro 1. pololetí 1. ročníku.

Učivo obsažené ve školním vzdělávacím plánu pro 1. pololetí se dělí do pěti základních sekcí, kterými jsou topologie počítačových sítí, komunikace v síti, pasivní prvky, aktivní prvky a adresace v síti (viz Obrázek 10).

Učivo	ŠVP výstupy
<b>1. Topologie sítí (9)</b> – pojmy počítačová síť, server, pracovní stanice, Internet – členění sítí podle rozsahu, topologie, přenosového média, role uzlů apod. – přenosová média (druhy, principy, výhody/nevýhody) – fyzické, logické a geografické členění sítí	<b>Žák</b> – vysvětlí pojem počítačová síť a přínos zapojení počítačů do sítě – chápe specifika práce v síti (včetně rizik), využívá jejich možnosti a pracuje s jejími prostředky – klasifikuje síť podle různých kritérií – zná výhody a nevýhody jednotlivých přenosových médií a zvolí vhodné médium pro daný účel
<b>2. Komunikace v síti (10)</b> – přístupové metody k médiu – referenční model ISO/OSI – síťový model TCP/IP – základní protokoly	– rozpozná principy komunikace na síti – zná základní druhy přístupových metod k médiu, na jakém principu pracují, jejich výhody a nevýhody – zná jednotlivé vrstvy RM ISO/OSI a co zajišťují – zná jednotlivé vrstvy SM TCP/IP, dovede porovnat s RM ISO/OSI – definuje základní komunikační protokoly
<b>3. Pasivní prvky sítí (10)</b> – kabely (druhy, značení, parametry) – konektory (druhy, značení, parametry) – zásuvky, racky, patch panely – datový rozvaděč a jeho vybavení	– je schopen vysvětlit rozdíl mezi pasivními a aktivními prvky sítě – rozezná typy kabelových vedení a jejich parametry – zná základní druhy pasivních síťových prvků, jejich účel – zvolí vhodné prvky pro danou síť – zrealizuje jednoduchou strukturovanou kabeláž
<b>4. Aktivní prvky sítí (10)</b> – síťová karta – repeater – hub, bridge – switch, router, gateway – jejich typy a parametry	– zná základní druhy aktivních síťových prvků, na jaké vrstvě ISO/OSI pracují, jejich účel, princip činnosti – zvolí vhodné prvky pro danou síť a jejich konfiguraci
<b>5. Adresace v síti (15)</b> – MAC adresa, její formát, způsob zajištění jedinečnosti – IP adresa, IP aritmetika – třídy adres – subsítě, VLSM, CIDR – port a socket – překlad adres – IPv6 – unicast, broadcast, multicast, anycast – DHCP a NAT	– orientuje se v adresaci poč. sítí – zná pojmy MAC adresa a IP adresa, jejich účel, vzájemnou vazbu – zná rozdělení IP adres do tříd, veřejné a privátní adresy, dělení a slučování sítí – je schopen vysvětlit možnosti řešení nedostatku veřejných IP adres – použije funkci DHCP služby – použije funkci NAT

Obrázek 10 Část školního vzdělávacího plánu použitá pro návrh

## 4.2 Vytvoření kategorií

Po získání základních informací ze školního kurikula následovala fáze kategorizování her. Vzhledem k časové náročnosti pro návrh a implementaci her nebylo reálné uskutečnit pro každé téma hru přímo na míru. Témata tedy byla rozřazena do pěti základních kategorií.

První kategorie úloh s označením K1 je určena pro pojmy počítačových sítí, u kterých je velmi důležitá znalost definice. Spadají zde také termíny, které nelze zachytit za pomoci

obrázku nebo by byly pro začátečníky v oblasti počítačových sítí velmi obtížné na uhodnutí. Ideu této hry lze použít pro většinu témat, která jsou součástí školního vzdělávacího plánu.

Druhá oblast se týká úloh s využíváním obrázků a jejich přiřazení. Úloha byla inspirována geografickým členěním sítí, kde uživatel může přidělovat obrázek k jeho definovanému geografickému rozsahu. Koncept této hry lze kromě geografického členění využít také u členění na základě topologií a přiřazení jednotlivých aktivních a pasivních prvků. Tato kategorie má označení K2.

Pojmy z počítačových sítí, které lze zobrazit za pomoci obrázku, ale není nutné je členit, budou zařazeny do další kategorie úloh s označením K3. Na počátku této hry bude obrázek, který bude překrytý tak, aby nebyl vidět. K účelu překrytí obrázku bude určeno 16 bloků. Pokud uživatel na blok klikne tak ho odstraní a zároveň se zobrazí oblast nacházející se za tímto blokem. Jedná se o kategorii her, která může být využita pro většinu témat počítačových sítí.

Čtvrtá kategorie se týká přiřazování více slov, která spolu souvisí. Úloha byla navržena na základě modelu ISO/OSI složeného ze sedmi vrstev a modelu TCP/IP složeného ze čtyř vrstev. Uplatnění těchto miniher lze ale využít i u jiných částí počítačových sítí, jako protokolů, tříd adres nebo rozdělení kabelů na základě jejich přenosové rychlosti. Pro tuto kategorii bylo použito označení K4.

Poslední oblastí úloh budou výpočty, označené kategorií K5. Princip hry je inspirováný výpočtem podsítí, převodem číselných soustav a určením IP adresy. Minihra bude využívána pro všechny úlohy, kde je zapotřebí využívat matematiku pro získání výsledku.

### **4.3 Shrnutí úloh na základě kurikula**

Po vytvoření základních pěti kategorií bylo provedeno rozřazení konkrétních témat ze školního vzdělávacího plánu a díky tomu mohly vzniknout jednotlivé úlohy do hry. Pro tento účel byla vytvořena Tabulka 1, jejíž součástí jsou jednotlivé kategorie označené zkratkami K1, K2, K3, K4 a K5. Uvnitř Tabulky 1 lze nalézt také čísla herních módů a názvy jednotlivých témat počítačových sítí. Čísla herních módů udávající úroveň, ve které se uživatel může s danou problematikou setkat. V případě herního módu kampaně je nutné projít postupně všechny části a úlohy počítačových sítí bez toho, aby uživatel vynechal některé téma. Naopak u jednotlivých úrovní bude využíváno pseudonáhodné generace z více

možností, aby uživatel při hraní určité minihry dopředu nevěděl, jaký dostane pojem nebo obrázek.

Tabulka 1 Seznam úloh pro implementaci

Témata z kurikula	Mód kampaň	Mód úrovně	K1	K2	K3	K4	K5
Pojem bit	1	1	+				
Pojem byte	2	1	+				
Binární soustava	3	2					+
Desítková soustava	4	3				+	
Počítačová síť	5	4	+				
Server	6	4	+				
Pracovní stanice	7	4	+				
Internet	8	4	+				
Členění dle rozsahu	9	5		+			
Topologie sběrnice	10	6			+		
Topologie kruh	11	6			+		
Topologie hvězda	12	6			+		
Topologie strom	13	6			+		
Páteřní topologie	14	6			+		
Topologie neomezená	15	6			+		
Přenosová média	16	7				+	
Client server	17	8	+				
Peer to peer	18	8	+				
Model ISO/OSI	19	9				+	
Model ICP/IP	20	10				+	
Fyzická vrstva	21	11	+				
Aplikační vrstva	22	11	+				

Témata z kurikula	Mód kampaň	Mód úrovně	K1	K2	K3	K4	K5
Síťová vrstva	23	11	+				
Relační vrstva	24	11	+				
Prezentační vrstva	25	11	+				
Transportní vrstva	26	11	+				
Linková vrstva	27	11	+				
Přístupové metody	28	12				+	
SMTP	29	13	+				
POP3	30	13	+				
IMAP	31	13	+				
FTP	32	13	+				
HTTP	33	13	+				
DNS	34	13	+				
ARP	35	13	+				
Protokoly	36	14				+	
Koaxiální kabel	37	15			+		
Kroucená dvojlinka	38	15			+		
Optická vlákna	39	15			+		
Pasivní prvky	40	16		+			
Optický kabel porucha	41	17				+	
RJ-45	42	18	+				
Síťová karta	43	19			+		
Router	44	19			+		
Hub	45	20	+				
Switch	46	20	+				



Témata z kurikula	Mód kampaň	Mód úrovně	K1	K2	K3	K4	K5
Repeater	47	20	+				
Gateway	48	20	+				
Aktivní prvky ISO/OSI	49	21				+	
MAC adresa	50	22	+				
IP adresa	51	22	+				
Výpočet IP adresy	52	23					+
Výpočet síťové adresy	53	24					+
Třídy adres	54	25				+	
VLSM	55	26	+				
CIDR	56	26	+				
Port	57	26	+				
Socket	58	26	+				
Hierarchie domén	59	27				+	
Výpočet subsítě	60	28					+
Přiřazení čísla portu	61	29				+	
Veškerá problematika	62	30	+				
Veškerá problematika	63	30	+				
Veškerá problematika	64	30	+				
Veškerá problematika	65	30	+				

Z Tabulky 1 vytvořené pro rozřazení daných témat do konkrétních kategorií je zřejmé, že herní mód kampaň bude mít celkově 65 úloh, které musí uživatel pro dokončení splnit. V případě herního módu, kde si hráč volí libovolnou úroveň, bude k dispozici 30 základních úloh.

#### 4.4 Společný princip všech her

Jednotlivé minihry budou tvořeny pomocí unikátních částí, ale také prvků, které jsou společné pro všechny hry. Po spuštění libovolné minihry bude zobrazena obrazovka, která se bude skládat z popisu hry, zpětného tlačítka odkazujícího pro odchod do menu, tlačítka start pro zahájení hry, nápovědy, ovládání, textu symbolizujícího danou úroveň a názvu učiva. Po kliknutí na tlačítka start dojde ke spuštění interaktivní minihry. Současně s tím se spustí časovač, který symbolizuje, kolik vteřin hráči ještě zbývá na dokončení úrovně. Pokud se časovač dostane na hodnotu 0, hra bude ukončena. Naopak pokud uživatel dokončí úkol v daném limitu, bude mu zobrazen počet zbývajících vteřin, který v této hře symbolizuje zároveň i skóre. Skóre bude možné ovlivnit i pomocí chybných odpovědí, kdy u každé špatné odpovědi dojde k odečtení o určitý počet bodů v závislosti na dané hře. Hráč si může nápovědu zobrazit i během spuštěné hry, ale skóre mu bude nadále odečítáno. To samé platí pro zobrazení návodu, jakým stylem se daná minihra má ovládat.

#### 4.5 Guess the Word

Minihra *Guess the Word* bude určena pro úlohy z kategorie K1. Po spuštění bude uživateli zobrazena nápověda, která popisuje nějaký pojem z počítačových sítí. Současně s tím bude vykreslena i část zobrazující počet písmen daného pojmu. Důležitým prvkem bude zbývajících počet pokusů, který je také součástí implementace každé minihry. Poslední částí je česká abeceda, která v sobě obsahuje 41 tlačítek s jednotlivými písmeny. Pokud hráč klikne na některé písmeno, zobrazí se mu odpověď, zda se jednalo o písmeno, které se ve slově nachází nebo ne. V případě volby správného písmene se nic nestane a uživatel může kliknout na další písmeno. Pokud uživatel vybere špatné tlačítko, dojde k upozornění, že bylo zvoleno špatné písmeno a současně s tím uživatel přijde o jeden pokus a určitý počet bodů ze skóre. Pokud uživateli zbývá poslední pokus a zvolí špatné písmeno, hra je ukončena a automaticky prohraje se skórem 0. V momentě, kdy uživatel uhodne všechna písmena daného pojmu z počítačových sítí, dochází k výhře a hráči je zobrazeno jeho dosažené skóre.

#### 4.6 Picture Match

Další minihrou, která bude součástí implementace webové aplikace, je hra *Picture match*. Jedná se o princip přiřazování, kdy uživateli bude zobrazeno 6 obrázků, které se dají přetahovat a fungují na principu *drag and drop*. Hráči bude navíc zobrazeno dalších 6 položek, které symbolizují pozice, kam uživatel může umístit dané obrázky. Součástí

implementace, určené pro úlohy z kategorie K2, bude opět počet zbývajících pokusů a skóre, které s narůstajícím časem stráveným ve hře klesá. Při správném přiřazení obrázku dojde k tomu, že jej již nebude možné přesunout zpět a jeho původní místo ztmavne. Zároveň obrázek zůstane na místě správné položky. Pokud hráč přiřadí všechny správné obrázky, dojde k jeho výhře a zobrazení dosaženého skóre. Naopak hráč může prohrát, pokud při přiřazování udělá maximální možný počet chyb nebo mu vyprší čas. V obou případech získává skóre s hodnotou 0.

#### 4.7 Guess the Picture

Pro úlohy z kategorie K3 bude vytvořena hra *Guess the Picture*. Na počátku hry bude vygenerován obrázek, který následně překryjí bloky tak, aby nebyl vidět. Tyto bloky budou vypočítány na základě velikosti obrázku a jejich celkový počet bude 16. Pokud uživatel na blok klikne, bude odstraněn a tím se hráči zobrazí patřičná část obrázku. Současně s každým kliknutím dojde k odečtení skóre. Další podstatnou částí hry bude zobrazení klávesnice, vstupů, skóre a počet zbývajících pokusů, který se ovšem bude odečítat pouze v případě špatně zvoleného písmene. Kliknutí na blok a následné zobrazení části obrázku nebude považováno za chybu. Hra může být ukončena výhrou po uhodnutí obrázku na pozadí nebo prohrou po vypršení časového intervalu, či využití všech pokusů na dokončení.

#### 4.8 Put the Words

*Put the Words* bude hra založená na přiřazování více slov k jejich krátkým definicím. Princip minihry bude založen na zamíchání slov v levém sloupci a ponecháním původního pořadí ve sloupci pravém. Uživatel pak bude muset správně přiřadit slova z levé části za pomoci principu *drag and drop* k jejich popisu zobrazeném v pravé oblasti hry. Při chybném přiřazení bude uživateli sebrán jeden pokus, a pokud hráči dojdou veškeré pokusy nebo čas, hra bude ukončena. Naopak pokud přiřadí všechny části hry správně, bude mu zobrazen text s gratulací k výhře a dosažené skóre. Tato hra je určena pro úlohy z kategorie K4.

#### 4.9 Count the Number

Poslední minihrou bude *Count the Number* určená pro veškeré úlohy z kategorie K5, které jsou založeny na matematických výpočtech nebo na převodu číselných soustav. Princip hry je podobný jako u *Put the Words* s tím rozdílem, že uživateli jsou pseudonáhodně vygenerována čísla, která nemají být přiřazena. Díky tomu může hráč vybírat z více čísel,

která nemusí s příkladem vůbec souviset. Hru bude možné vyhrát pomocí správného doplnění všech čísel.

## 5 NAVRŽENÍ WEBOVÉ APLIKACE

Součástí návrhu webové aplikace pro výuku základů počítačových sítí je sběr požadavků společně s jejich analýzou, vytvoření modelu případů užití, sepsání scénářů a návržení aplikace pomocí wireframů.

### 5.1 Sběr a analýza požadavků

Před zahájením programování bylo nutné u webové aplikace provést sběr a analýzu požadavků. Vzhledem k tomu, že aplikace je primárně určena pro žáky, kteří se chtějí vzdělávat v oblasti počítačových sítí, bylo s žáky a učiteli před vytvořením webové hry provedeno několik slovních rozhovorů, které byly následně shrnuty v uživatelských požadavcích.

#### 5.1.1 Uživatelské požadavky

Hra bude v českém jazyce a primárně bude určena pro počítačovou platformu. Součástí implementace bude menu, rozdělené do čtyř základních sekcí, mezi které budou patřit oba herní módy, tabulka osobního rozvoje a zobrazení ovládání hry. Uživatel si bude moci zvolit hraní pomocí kampaně nebo jednotlivých úrovní. V případě, že si hráč zvolí herní mód kampaně, projde postupně veškeré herní úrovně a po dokončení mu bude zobrazeno celkové skóre, které se současně uloží do tabulky rozvoje s aktuálním datem. Druhou možností bude zvolení herního módu pomocí jednotlivých úrovní, kde si hráč bude moci zlepšit dovednosti v konkrétní oblasti počítačových sítí. Každá úroveň bude jinak náročná a bude se zaměřovat na jinou oblast počítačových sítí. Pokud bude uživatel chtít hru opustit, bude mu to umožněno v libovolném čase, ale v případě kampaně musí být zachován aktuální progres. Součástí každé minihry bude možnost zobrazení skóre, nápovědy, ovládání a počtu pokusů na dokončení hry.

#### 5.1.2 Funkční požadavky všech her

- FR01: Zobrazení kampaně
- FR02: Zobrazení všech úrovní
- FR03: Zobrazení tabulky rozvoje
- FR04: Zobrazení obecného ovládání
- FR05: Možnost návratu do menu

- FR06: Zobrazení výukového materiálu
- FR:07 Zobrazení hry guess the word
- FR:08 Zobrazení hry picture match
- FR:09 Zobrazení hry guess the picture
- FR10: Zobrazení hry put the words
- FR11: Zobrazení hry count the number
- FR12: Zobrazení nápovědy k řešení
- FR13: Zobrazení ovládání konkrétní hry
- FR14: Zobrazení prohry
- FR15: Zobrazení výhry
- FR16: Možnost opuštění hry v kterýkoliv moment

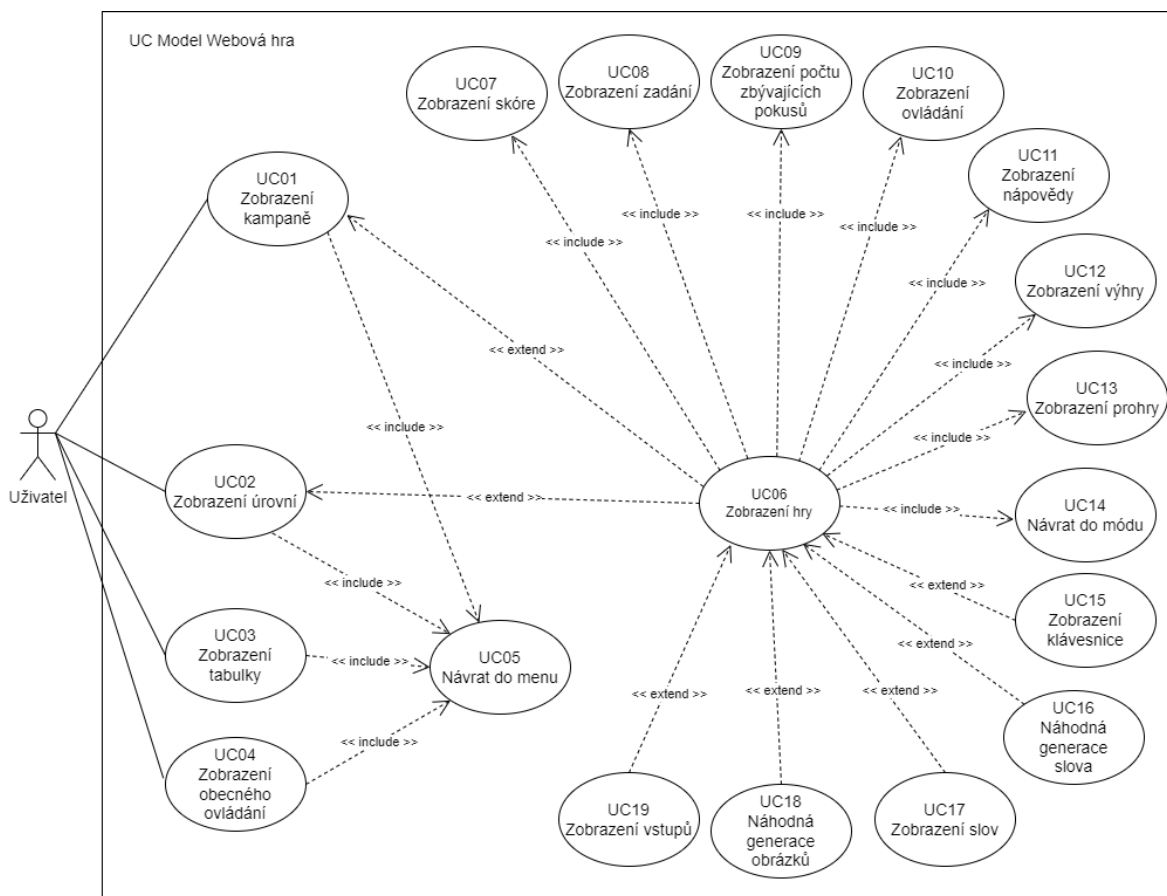
### 5.1.3 Nefunkční / systémové požadavky

- NFR:01 Hra nesmí způsobovat chyby, které by vedly k nefunkčnosti aplikace
- NFR:02 Hra bude určena primárně pro počítačovou platformu, ale bude tvořena pomocí responzivního designu, takže si ji uživatelé budou moci zahrát v různých rozlišeních
- NFR:03 Hra musí běžet nepřetržitě až na výjimečné situace, které se budou týkat vylepšení nebo aktualizace systému
- NFR:04 Hra musí být k dispozici na prohlížečích Google Chrome, Mozilla Firefox, Safari a Microsoft Edge
- NFR:05 Základní menu hry musí být na webové stránce načteno maximálně do tří sekund od okamžiku, kdy uživatel přistoupí k webovému rozhraní

## 5.2 Model případů užití

Po sepsání veškerých funkčních a nefunkčních požadavků byl navržen UC model, jehož součástí je jeden aktér zastupující uživatele hrajícího webovou hru. Diagram zobrazený na Obrázku 11 je složený z jednotlivých případů užití, které mají sepsány své scénáře složené

z hlavních a alternativních toků. Celkový počet vytvořených případů užití je 19 a slouží k pokrytí veškerých funkčních požadavků.



Obrázek 11 Diagram případů užití

Vzhledem k tomu, že nemůže existovat požadavek, který by nebyl zastoupen alespoň jedním případem užití, byly vytvořeny Tabulky 2 a 3 symbolizující matici pokrytí. Tato matice pokrytí zachycuje veškeré funkční požadavky a zároveň případy užití sloužící k jejich pokrytí. Znak plus znamená, že UC slouží k naplnění funkčního požadavku. Díky Tabulkám 2 a 3 lze potvrdit, že každý požadavek je zastoupen alespoň jedním případem užití.





Tabulka 3 Pokrytí funkčních požadavků 2

UC/FR	FR09	FR10	FR11	FR12	FR13	FR14	FR15	FR16
UC01								
UC02								
UC03								
UC04								
UC05								
UC06	+	+	+					
UC07	+	+	+					
UC08	+	+	+					
UC09	+	+	+					
UC10					+			
UC11				+				
UC12							+	
UC13						+		
UC14	+	+	+					+
UC15	+							
UC16								
UC17		+	+					
UC18	+							
UC19	+							

### 5.3 Scénáře

Zásadní oblastí návrhu jakéhokoliv softwaru je sepsání scénářů jednotlivých případů užití. Scénáře umožňují jasné zachycení situací, ke kterým by v systému mohlo dojít a díky tomu je následná implementace pro vývojáře daleko snadnější než bez nich. Navíc umožňují zachycení podmínek a cyklů. Nezbytnou vstupní podmínkou, která je nutná pro naplnění všech scénářů, je připojení k internetu.

Uživatel, který se rozhodne pro hraní herního režimu kampaně, bude muset kliknout na tlačítko v hlavním menu. Při procesu dojde k načtení dat o jeho aktuálním skóre a úrovni, ve které se bude nacházet. Součástí načteného obsahu bude poté tlačítko sloužící k návratu zpět do menu, popis problematiky dané úrovně a také možnost přesunutí uživatele do konkrétní hry, která má prověřit jeho znalosti získané v aktuální úrovni (viz Tabulka 4).

Tabulka 4 Scénář – zobrazení kampaně

<b>Označení případu užití</b>	UC01
<b>Název scénáře</b>	Zobrazení kampaně
<b>Popis scénáře</b>	Tento scénář popisuje proces zobrazení herního módu kampaně po kliknutí na tlačítko v hlavním menu.
<b>Aktéři</b>	Uživatel, systém
<b>Vstupní podmínka</b>	Uživatel musí být v hlavní navigaci hry.
<b>Základní tok</b>	<ol style="list-style-type: none"> <li>1. UC začíná poté, co uživatel klikne na tlačítko kampaň</li> <li>2. Systém skryje obsah webové stránky pomocí animace ztmavení</li> <li>3. Systém zajistí načtení obsahu kampaně během animace</li> <li>4. Systém vykoná animaci zesvětlení a zobrazí část kampaně, ve které se uživatel na daném zařízení nachází</li> <li>5. Systém zobrazí tlačítko pro návrat do menu</li> </ol>
<b>Alternativní toky</b>	3a. Uživatel kampaň ještě nespustil <ol style="list-style-type: none"> <li>1. Systém vygeneruje novou kampaň</li> <li>2. Systém pokračuje v bodě 4</li> </ol>
	3b. Uživatel kampaň již dohrál

	<ol style="list-style-type: none"> <li>1. Systém odstraní původní kampaň</li> <li>2. Systém vygeneruje novou kampaň</li> <li>3. Systém pokračuje v bodě 4</li> </ol>
--	--

Druhý herní mód bude umožňovat uživatelům hrát úroveň, kterou si zvolí. Díky tomu si hráč bude moci zopakovat oblíbené části hry nebo se zlepšit v oblastech počítačových sítí, ve kterých si není tak jistý. Pro přesun do tohoto módu bude v hlavním menu zobrazeno tlačítko s levely, které hráči změní mód a zobrazí mu možnost volby libovolné úrovně (viz Tabulka 5).

Tabulka 5 Scénář – zobrazení úrovní

<b>Označení případu užití</b>	UC02
<b>Název scénáře</b>	Zobrazení úrovní
<b>Popis scénáře</b>	Scénář je založený na operaci zobrazení herního módu jednotlivých úrovní poté, co uživatel klikne na tlačítko levely.
<b>Aktéři</b>	Uživatel, systém
<b>Vstupní podmínka</b>	Uživatel musí být v hlavní navigaci hry.
<b>Základní tok</b>	<ol style="list-style-type: none"> <li>1. UC začíná poté, co uživatel klikne na tlačítko levely</li> <li>2. Systém skryje obsah webové stránky pomocí animace ztmavení</li> <li>3. Systém zajistí načtení tlačítek pro přesun do jednotlivých úrovní během animace</li> <li>4. Systém vykoná animaci zesvětlení a zobrazí veškeré úrovně, které jsou součástí hry</li> <li>5. Systém zobrazí tlačítko pro návrat do menu</li> </ol>

Další možností, kterou bude mít k dispozici uživatel nacházející se v menu, bude zobrazení tabulky svých dosavadních výsledků. V tabulce budou uloženy data o kampaních, které byly v minulosti dokončeny. Hráč si tak bude moci porovnávat své získané výsledky z minulosti s těmi aktuálními. Mezi atributy, které budou součástí tabulky patří ID, datum dokončení kampaně a skóre (viz Tabulka 6).

Tabulka 6 Scénář – zobrazení tabulky

<b>Označení případu užití</b>	UC03
<b>Název scénáře</b>	Zobrazení tabulky
<b>Popis scénáře</b>	Scénář začíná po kliknutí na tlačítko tabulka, nacházejícího se v hlavním menu a sloužícího pro zobrazení osobních výsledků z minulosti.
<b>Aktéři</b>	Uživatel, systém
<b>Vstupní podmínka</b>	Uživatel musí být v hlavní navigaci hry.
<b>Základní tok</b>	<ol style="list-style-type: none"> <li>1. UC začíná poté, co uživatel kline na tlačítko tabulka</li> <li>2. Systém skryje obsah webové stránky pomocí animace ztmavení</li> <li>3. Systém zajistí načtení dat z kampaní, které byly již v minulosti dokončeny na daném zařízení</li> <li>4. Systém vykoná animaci zesvětlení a zobrazí statistickou tabulku</li> <li>5. Systém zobrazí tlačítko pro návrat do menu</li> </ol>
<b>Alternativní toky</b>	<p>3a. Uživatel zatím nemá dokončenou kampaň</p> <ol style="list-style-type: none"> <li>1. Systém vrátí prázdnou tabulku a vypíše informaci o tom, že kampaň ještě nebyla dokončena</li> </ol>

V případě, kdy hráč hru ještě nebude znát, bude mít možnost zobrazení obecného ovládání. Tento případ uživateli zobrazí informace o tom, jak minihry probíhají, čím jsou ovládány a co vše je jejich součástí (viz Tabulka 7).

Tabulka 7 Scénář – zobrazení obecného ovládání

<b>Označení případu užití</b>	UC04
<b>Název scénáře</b>	Zobrazení obecného ovládání
<b>Popis scénáře</b>	Scénář je vyvolán kliknutím na tlačítko ovládání a slouží pro zobrazení základních informací o společných částech jednotlivých miniher.
<b>Akteři</b>	Uživatel, systém
<b>Vstupní podmínka</b>	Uživatel musí být v hlavní navigaci hry.
<b>Základní tok</b>	<ol style="list-style-type: none"><li>1. UC začíná poté, co uživatel klikne na tlačítko ovládání</li><li>2. Systém skryje obsah webové stránky pomocí animace ztmavení</li><li>3. Systém zajistí načtení informací o ovládání</li><li>4. Systém vykoná animaci zesvětlení a zobrazí uživateli informace o základním ovládání hry</li><li>5. Systém zobrazí tlačítko pro návrat do menu</li></ol>

Návrat do menu je společný scénář, který bude nutnou součástí UC01, UC02, UC03 a UC04. Všechny případy užití budou založeny na využití tlačítka umožňujícího návrat do hlavního menu aplikace, díky čemuž se může uživatel interaktivně pohybovat v aplikaci mezi jednotlivými částmi (viz Tabulka 8).

Tabulka 8 Scénář – návrat do menu

<b>Označení případu užití</b>	UC05
<b>Název scénáře</b>	Návrat do menu
<b>Popis scénáře</b>	Jedná se o situaci, kdy se uživatel chce dostat zpět do hlavního menu. K tomuto účelu slouží v aplikaci tlačítko zpět.
<b>Aktéři</b>	Uživatel, systém
<b>Vstupní podmínka</b>	Uživatel se musí nacházet v kampani, úrovních, tabulce nebo obecném ovládní.
<b>Základní tok</b>	<ol style="list-style-type: none"><li>1. UC začíná poté, co uživatel klikne na tlačítko zpět</li><li>2. Systém skryje obsah webové stránky pomocí animace ztmavení</li><li>3. Systém vykoná animaci zesvětlení a zobrazí uživateli hlavní menu hry</li></ol>

Nejpodstatnějším scénářem celé aplikace bude zobrazení hry. Jde o scénář, který rozšiřuje herní módy kampaně a jednotlivých úrovní. Poté, co uživatel zvolí hru za pomoci libovolné úrovně nebo se rozhodne pokračovat v kampani za pomoci tlačítka hrát, bude hráči načtena minihra. Součástí každé hry jsou společné prvky skóre, zadání, počet zbývajících pokusů, možnost zobrazení ovládnutí a nápovědy, tlačítko zpět a v případě dohrání hry může být uživateli zobrazeno, zda vyhrál nebo prohrál. Navíc může hra být rozšířená o další funkcionality, jako zobrazení klávesnice, vygenerování náhodných slov a obrázků nebo zobrazení vstupů (viz Tabulka 9).

Tabulka 9 Scénář – zobrazení hry

<b>Označení případu užití</b>	UC06
<b>Název scénáře</b>	Zobrazení hry
<b>Popis scénáře</b>	Scénář začíná po kliknutí na konkrétní level nebo po spuštění hry uvnitř módu kampaně.
<b>Akteři</b>	Uživatel, systém
<b>Vstupní podmínka</b>	Uživatel se musí nacházet v kampani nebo úrovních.
<b>Základní tok</b>	<ol style="list-style-type: none"> <li>1. UC začíná poté, co uživatel klikne na tlačítko libovolné úrovně nebo na tlačítko hrát uvnitř kampaně</li> <li>2. Systém skryje obsah webové stránky pomocí animace ztmavení</li> <li>3. Systém načte data a části pro možné hraní konkrétní zvolené minihry</li> <li>4. Systém vykoná animaci zesvětlení a zobrazí uživateli vygenerovanou hru</li> </ol>

Zobrazení skóre je scénář, který bude součástí veškerých miniher. Jeho hodnota se mění v závislosti na dané hře a úrovni. Princip skóre bude založený na čase, kdy na počátku hry bude čas nastavený na určitou hodnotu a poté se každou vteřinu odečte o hodnotu jednoho bodu. Výše skóre se bude měnit také s každou chybou uživatele (viz Tabulka 10).

Tabulka 10 Scénář – zobrazení skóre

<b>Označení případu užití</b>	UC07
<b>Název scénáře</b>	Zobrazení skóre
<b>Popis scénáře</b>	Po spuštění hry je uživateli zobrazeno skóre, které je na základě časového intervalu pravidelně snižováno.
<b>Aktéři</b>	Uživatel, systém
<b>Vstupní podmínka</b>	Uživatel musí spustit libovolnou hru.
<b>Základní tok</b>	<ol style="list-style-type: none"> <li>1. UC začíná poté, co uživatel spustí hru</li> <li>2. Systém vygeneruje počáteční skóre</li> <li>3. Systém spustí časovač na základě zvoleného intervalu</li> <li>4. Systém zobrazí uživateli skóre</li> </ol>

Každá minihra bude mít definováno zadání, kde budou instrukce k tomu, jak by měl uživatel ve hře postupovat, aby dokázal zvítězit (viz Tabulka 11).

Tabulka 11 Scénář – zobrazení zadání

<b>Označení případu užití</b>	UC08
<b>Název scénáře</b>	Zobrazení zadání
<b>Popis scénáře</b>	Po spuštění hry je uživateli zobrazeno zadání, které hráči popisuje, co by měl ve hře udělat pro vítězství.
<b>Aktéři</b>	Uživatel, systém
<b>Vstupní podmínka</b>	Uživatel musí spustit libovolnou hru.
<b>Základní tok</b>	<ol style="list-style-type: none"> <li>1. UC začíná poté, co uživatel spustí hru</li> <li>2. Systém zobrazí text, který popisuje, jak by měl uživatel ve hře postupovat</li> </ol>



Součástí miniher bude zobrazení zbývajících počtu pokusů. Vzhledem k tomu, že uživatel se může při hře mýlit, bude mít každá minihra určený počet chybných pokusů. Pokud uživatel vyplývá všechny pokusy, hra bude ukončena (viz Tabulka 12).

Tabulka 12 Scénář – zobrazení počtu zbývajících pokusů

<b>Označení případu užití</b>	UC09
<b>Název scénáře</b>	Zobrazení počtu zbývajících pokusů
<b>Popis scénáře</b>	Po spuštění hry je uživateli zobrazen počet možných chyb, které může během hraní minihry udělat.
<b>Aktéři</b>	Uživatel, systém
<b>Vstupní podmínka</b>	Uživatel musí spustit libovolnou hru.
<b>Základní tok</b>	<ol style="list-style-type: none"> <li>1. UC začíná poté, co uživatel spustí hru</li> <li>2. Systém zobrazí počet zbývajících pokusů</li> </ol>

Veškeré hry budou mít specifické ovládání, které nemusí uživatel při prvním pokusu ihned pochopit. Z tohoto důvodu bude součástí každé minihry možnost zobrazení popisu ovládání za pomoci tlačítka, které uživateli zobrazí dialog s objasněním principu dané hry a poté může hráč pokračovat v jejím dokončení (viz Tabulka 13).

Tabulka 13 Scénář – zobrazení ovládání

<b>Označení případu užití</b>	UC10
<b>Název scénáře</b>	Zobrazení ovládání
<b>Popis scénáře</b>	Po kliknutí na tlačítko ovládání je uživateli zobrazen dialog s detailním popisem principu dané hry.
<b>Aktéři</b>	Uživatel, systém
<b>Vstupní podmínka</b>	Uživatel musí ve hře kliknout na tlačítko ovládání.
<b>Základní tok</b>	<ol style="list-style-type: none"> <li>1. UC začíná poté, co uživatel ve hře klikne na tlačítko ovládání</li> <li>2. Systém zobrazí dialog s detailním popisem ovládání konkrétní hry</li> </ol>

V případě, kdy si uživatel nebude jistý danou problematikou, ale už se rozhodl spustit hru, bude mít k dispozici tlačítko pro nápovědu. Tento prvek zobrazí dialog, jehož součástí bude probíraná problematika spuštěné minihry (viz Tabulka 14).

Tabulka 14 Scénář – zobrazení nápovědy

<b>Označení případu užití</b>	UC11
<b>Název scénáře</b>	Zobrazení nápovědy
<b>Popis scénáře</b>	Po kliknutí na tlačítko nápovědy je uživateli zobrazen dialog s popisem problematiky, které se daná hra týká.
<b>Aktéři</b>	Uživatel, systém
<b>Vstupní podmínka</b>	Uživatel musí ve hře kliknout na tlačítko nápovědy.
<b>Základní tok</b>	<ol style="list-style-type: none"><li>1. UC začíná poté, co uživatel ve hře klikne na tlačítko nápovědy</li><li>2. Systém zobrazí dialog s rozбором problematiky právě spuštěné hry</li></ol>

Po spuštění hry mohou nastat čtyři situace, kterými lze hru ukončit. První možností bude vítězství ve hře, kterého bude uživatel moci dosáhnout na základě zvolené minihry doplněním slova, přidělením všech obrázků, uhodnutím daného obrázku, přidělením všech slov nebo výpočtem. Pokud se to hráči podaří, bude mu zobrazena obrazovka s gratulací k výhře, dosažené skóre a možnost zopakovat si hru nebo pokračovat v kampani (viz Tabulka 15).

Tabulka 15 Scénář – zobrazení výhry

<b>Označení případu užití</b>	UC12
<b>Název scénáře</b>	Zobrazení výhry
<b>Popis scénáře</b>	Poté, co uživatel splní všechny části minihry, je mu zobrazena nová obrazovka s gratulací a možností si hru zopakovat, případně pokračovat v kampani.
<b>Aktéři</b>	Uživatel, systém
<b>Vstupní podmínka</b>	Uživatel musí zvítězit v libovolné minihře.
<b>Základní tok</b>	<ol style="list-style-type: none"> <li>1. UC začíná poté, co uživatel vyhraje minihru</li> <li>2. Systém zobrazí obrazovku, která ho informuje o dosaženém skóre a je mu zobrazeno tlačítko pro pokračování v kampani nebo zopakování hry</li> </ol>

Další dvě možnosti ukončení hry nastanou v momentě, kdy uživatel prohraje. K situaci může dojít ve dvou případech (viz Tabulka 16) a to, když hráči dojde čas, k čemuž dojde ve chvíli, kdy se mu zobrazí skóre 0 nebo v momentě využití všech pokusů. Obě události vedou k zobrazení obrazovky, která uživatele informuje o tom, že v minihře prohrál a bude mu zobrazeno skóre s hodnotou 0. Současně s informacemi bude uživateli zobrazeno také tlačítko, díky kterému může hru spustit znovu.

Tabulka 16 Scénář – zobrazení prohry

<b>Označení případu užití</b>	UC13
<b>Název scénáře</b>	Zobrazení prohry
<b>Popis scénáře</b>	Poté, co uživatel ztratí všechny pokusy nebo mu dojde čas na splnění minihry, dojde k zobrazení obrazovky s možností zopakování dané hry a také informaci, že hráč prohrál.
<b>Aktéři</b>	Uživatel, systém
<b>Vstupní podmínka</b>	Uživatel musí prohrát v libovolné minihře.
<b>Základní tok</b>	<ol style="list-style-type: none"> <li>1. UC začíná poté, co uživatel prohraje minihru</li> <li>2. Systém zobrazí obrazovku, která ho informuje o prohře a současně vytvoří tlačítko pro možné zopakování hry</li> </ol>

Poslední možnost, jak opustit již spuštěnou hru, bude s pomocí zpětného tlačítka. Pokud se uživatel bude nacházet v módu kampaně, jeho aktuální skóre se uloží a poté, co minihru znovu zapne, mu bude znovu načteno. V případě opuštění hry v módu jednotlivých úrovní bude skóre při příštím zapnutí znovu nastaveno na defaultní skóre dané minihry. Po kliknutí na tlačítko bude tedy hráč přenesen buď do kampaně, nebo do sekce pro výběr jednotlivých úrovní (viz Tabulka 17).

Tabulka 17 Scénář – návrat do módu

<b>Označení případu užití</b>	UC14
<b>Název scénáře</b>	Návrat do módu
<b>Popis scénáře</b>	V momentě, kdy si uživatel spustí minihru a chce ji opustit, bude vrácen zpět do kampaně nebo do volby úrovní.
<b>Akteři</b>	Uživatel, systém
<b>Vstupní podmínka</b>	Uživatel musí spustit libovolnou minihru.
<b>Základní tok</b>	<ol style="list-style-type: none"> <li>1. UC začíná poté, co uživatel klikne na tlačítko zpět při spuštěné hře</li> <li>2. Systém skryje obsah webové stránky pomocí animace ztmavení</li> <li>3. Systém zjistí, jaký mód je aktuálně zapnutý</li> <li>4. Systém vykoná animaci zesvětlení a zobrazí uživateli kampaň nebo sekci s jednotlivými úrovněmi</li> </ol>

Součástí miniher budou i prvky, které nebudou obsaženy ve všech hrách. Mezi tyto komponenty patří klávesnice, vygenerování náhodných slov a obrázků, zobrazení vstupů nebo zobrazení slov včetně jejich jednoslovných definicí.

Prvním prvkem bude klávesnice, která bude součástí dvou her, a to *Guess the Word* a *Guess the Picture*. Jedná se o zobrazení jednotlivých znaků české abecedy v podobě tlačítek, na které uživatel bude moci kliknout. Celkově se tedy zobrazí 41 komponent (viz Tabulka 18).

Tabulka 18 Scénář – zobrazení klávesnice

<b>Označení případu užití</b>	UC15
<b>Název scénáře</b>	Zobrazení klávesnice
<b>Popis scénáře</b>	Scénář nastává v momentě spuštění specifické hry a slouží pro zobrazení české klávesnice.
<b>Akteři</b>	Uživatel, systém
<b>Vstupní podmínka</b>	Uživatel musí spustit hru <i>Guess the Word</i> nebo <i>Guess the Picture</i> .
<b>Základní tok</b>	<ol style="list-style-type: none"> <li>1. UC začíná poté, co uživatel spustí hru <i>Guess the Word</i> nebo <i>Guess the Picture</i></li> <li>2. Systém zobrazí 43 tlačítek, kde každé písmeno symbolizuje jedno písmeno abecedy</li> </ol>

Další specifickou komponentou bude náhodná generace slov, která bude využívána pouze u hry *Guess the Word*. Po spuštění hry systém získá pole termínů s jejich definicemi a pomocí pseudonáhodné generace zobrazí jedno slovo společně s jeho popisem (viz Tabulka 19).

Tabulka 19 Scénář – náhodná generace slova

<b>Označení případu užití</b>	UC16
<b>Název scénáře</b>	Náhodná generace slova
<b>Popis scénáře</b>	Scénář nastává v momentě spuštění hry <i>Guess the Word</i> , kdy systém vygeneruje jedno slovo z pole včetně jeho definice.
<b>Aktéři</b>	Uživatel, systém
<b>Vstupní podmínka</b>	Uživatel musí spustit hru <i>Guess the Word</i> .
<b>Základní tok</b>	<ol style="list-style-type: none"><li>1. UC začíná poté, co uživatel spustí hru <i>Guess the Word</i></li><li>2. Systém načte pole slov s definicemi</li><li>3. Systém pseudonáhodně vybere jedno slovo z pole</li><li>4. Systém zobrazí definici slova, které je určené pro hádání</li></ol>

Třetí rozšíření se týká her *Put the Words* a *Count the Number*. Tyto hry budou založeny na tom, že uživatel spojuje pojmy nebo doplňuje čísla do výpočtu. Po zahájení her dojde k zamíchání pole a jeho následného zobrazení (viz Tabulka 20).

Tabulka 20 Scénář – zobrazení slov

<b>Označení případu užití</b>	UC17
<b>Název scénáře</b>	Zobrazení slov
<b>Popis scénáře</b>	Scénář nastává v momentě spuštění specifické hry a slouží pro zamíchání možných slov v poli a zobrazení tak, aby uživatel měl pokaždé slova jinak uspořádána.
<b>Aktéři</b>	Uživatel, systém
<b>Vstupní podmínka</b>	Uživatel musí spustit hru <i>Put the Words</i> nebo <i>Count the Number</i> .
<b>Základní tok</b>	<ol style="list-style-type: none"><li>1. UC začíná poté, co uživatel spustí hru <i>Put the Words</i> nebo <i>Count the Number</i></li><li>2. Systém načte pole slov s definicemi</li><li>3. Systém zamíchá pole</li><li>4. Systém zobrazí slova ze zamíchaného pole</li></ol>



V případě miniher *Picture Match* a *Guess the Picture* bude využíváno rozšiřující komponenty pro náhodnou generaci obrázků. Jedná se o podobný případ užití jako UC16 s tím rozdílem, že nebude generovat definici, ale odkaz na obrázek, který slovo vystihuje (viz Tabulka 21).

Tabulka 21 Scénář – náhodná generace obrázků

<b>Označení případu užití</b>	UC18
<b>Název scénáře</b>	Náhodná generace obrázků
<b>Popis scénáře</b>	Scénář nastává v momentě spuštění specifické hry a slouží k pseudonáhodnému vygenerování obrázku společně se slovem, které ho definuje.
<b>Aktéři</b>	Uživatel, systém
<b>Vstupní podmínka</b>	Uživatel musí spustit hru <i>Picture Match</i> nebo <i>Guess the Picture</i> .
<b>Základní tok</b>	<ol style="list-style-type: none"> <li>1. UC začíná poté, co uživatel spustí hru <i>Picture Match</i> nebo <i>Guess the Picture</i></li> <li>2. Systém načte data typu JSON</li> <li>3. Systém vygeneruje odkaz na obrázek společně se slovem, které ho definuje</li> <li>4. Systém zobrazí obrázek</li> </ol>

Posledním rozšiřujícím prvkem ve hře bude zobrazení vstupů. V případě, kdy uživatel bude hrát hru s využitím klávesnice, bude potřebovat k úplné funkčnosti zároveň i zobrazení vstupů. Vstupy budou definovány na základě hádaného slova, kde místo jeho každého jednotlivého písmene je zobrazeno podtržítka (viz Tabulka 22).

Tabulka 22 Scénář – zobrazení vstupů

<b>Označení případu užití</b>	UC19
<b>Název scénáře</b>	Zobrazení vstupů
<b>Popis scénáře</b>	Scénář nastává v momentě spuštění specifické hry a slouží pro zobrazení délky hádaného slova.
<b>Aktéři</b>	Uživatel, systém
<b>Vstupní podmínka</b>	Uživatel musí spustit hru <i>Guess the Word</i> nebo <i>Guess the Picture</i> .
<b>Základní tok</b>	<ol style="list-style-type: none"> <li>1. UC začíná poté, co uživatel spustí hru <i>Guess the Word</i> nebo <i>Guess the Picture</i></li> <li>2. Systém spočítá délku hádaného slova</li> <li>3. Systém na základě délky slova vygeneruje potřebný počet podtržitek</li> <li>4. Systém zobrazí podtržítka</li> </ol>

## 5.4 Navržení aplikace pomocí wireframů

Po vytvoření všech digramů a sepsání scénářů bylo provedeno základní navržení aplikace s pomocí wireframů. Jedná se o techniku, která zachycuje základní komponenty a jejich umístění ve hře. Vzhledem k tomu, že jedním z nefunkčních požadavků je responzivní design, byly wireframy navrženy pro uživatele počítačových i mobilních zařízení.

První wireframe, který byl při navrhování aplikace vytvořen, byl určen pro úvodní stránku aplikace (viz Obrázek 12). Tato webová stránka bude uživateli zobrazena ihned po navštívení webu, tudíž by měla uživatele co nejvíce zaujmout. Na druhou stranu, jedním z klíčových nefunkčních požadavků je, že hlavní menu, které bude součástí úvodní stránky, musí být uživateli zobrazeno do tří vteřin od příchodu na stránku. Z tohoto důvodu byl wireframe navržen velmi jednoduše tak, aby na něm nebyly zbytečné prvky, které by načítání webu zpomalily.



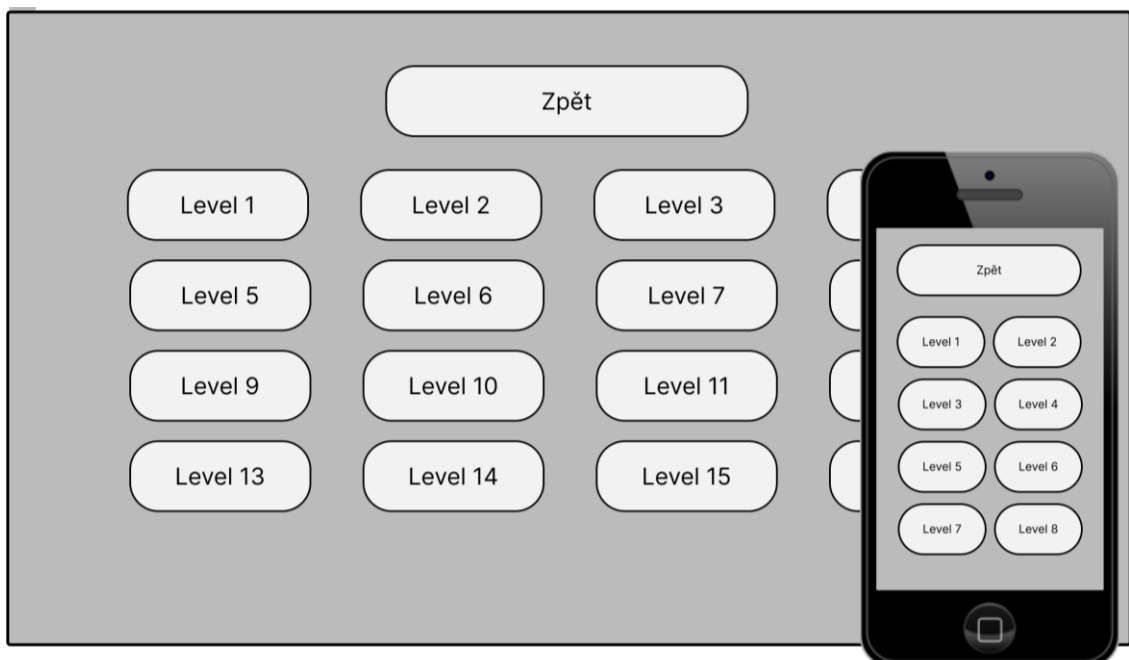
Obrázek 12 Wireframe – úvodní stránka

Pro případ, který nastane po zvolení herního módu kampaně, byl vytvořen wireframe, který je složený z tlačítka pro návrat do menu, obrázku, tlačítka pro spuštění hry, a především výukového textu, který uživatele připraví na úspěšné projití další úrovně (viz Obrázek 13). Součástí text budou vždy důležité informace, které hráč bude muset znát pro to, aby zvládl další část hry.



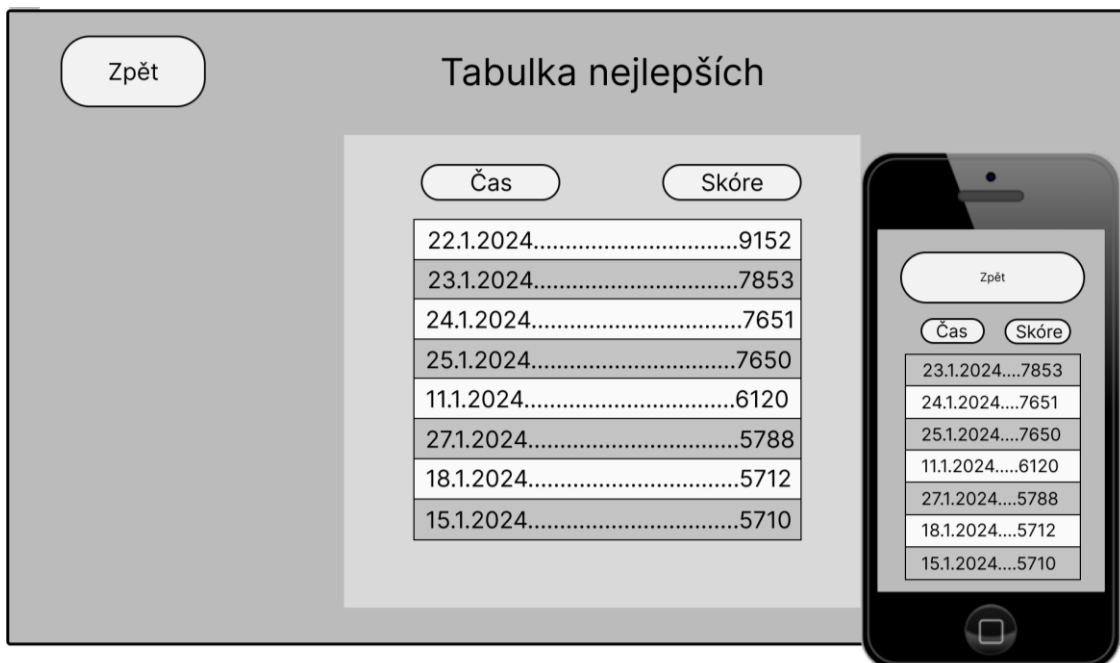
Obrázek 13 Wireframe – kampaň

Další oblast základního UI designu byla věnována hernímu módu, který umožňuje hrát libovolnou úroveň. V tomto případě bylo hlavním cílem zobrazení veškerých úrovní pohromadě za pomoci tlačítek tak, aby si uživatel mohl zvolit minihru kterou chce (viz Obrázek 14).



Obrázek 14 Wireframe – rozdělení úrovní

Uživatel, který již v minulosti dohrál kampaň si bude moci zobrazit statistickou tabulku svých výsledků (viz Obrázek 15). Součástí tabulky bude také filtrace na základě časového údaje a získaného skóre.



Obrázek 15 Wireframe – tabulka

Pátý wireframe je určený pro nové uživatele webových stránek. Součástí wireframu je obrázek a text zachycující obecné principy miniher, včetně jejich ovládání (viz Obrázek 16). Poté, co si uživatel informace přečte, může se vrátit zpět do menu za pomoci tlačítka.



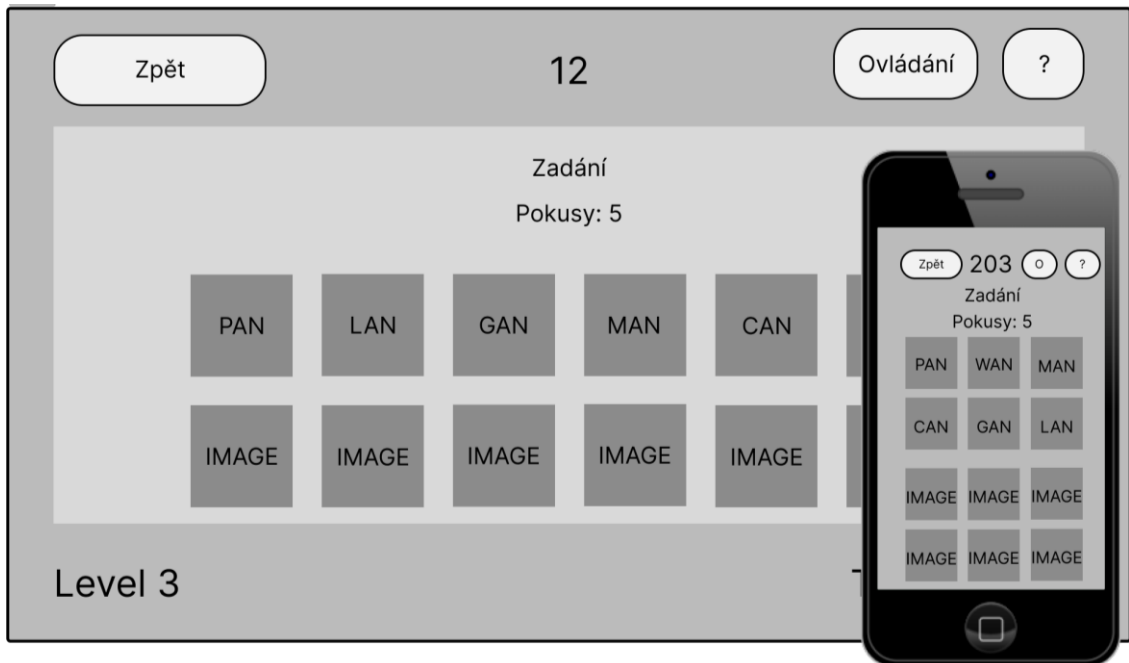
Obrázek 16 Wireframe – ovládání

První hra, pro kterou byl vytvořen wireframe, se jmenuje *Guess the Word*. U tohoto návrhu lze vidět rozdělení do vnější a vnitřní sekce, což je zachyceno různými barevnými odstíny. Vnější sekce bude stejná pro všechny hry s tím rozdílem, že se v ní budou pouze střídat texty. Vnitřní oblast bude poté u každé hry složená z jiných komponent. V případě minihry *Guess the Word* budou těmi komponentami definice slov, vstupy pro doplnění písmen, počet zbývajících pokusů a klávesnice složená z tlačítek (viz Obrázek 17).



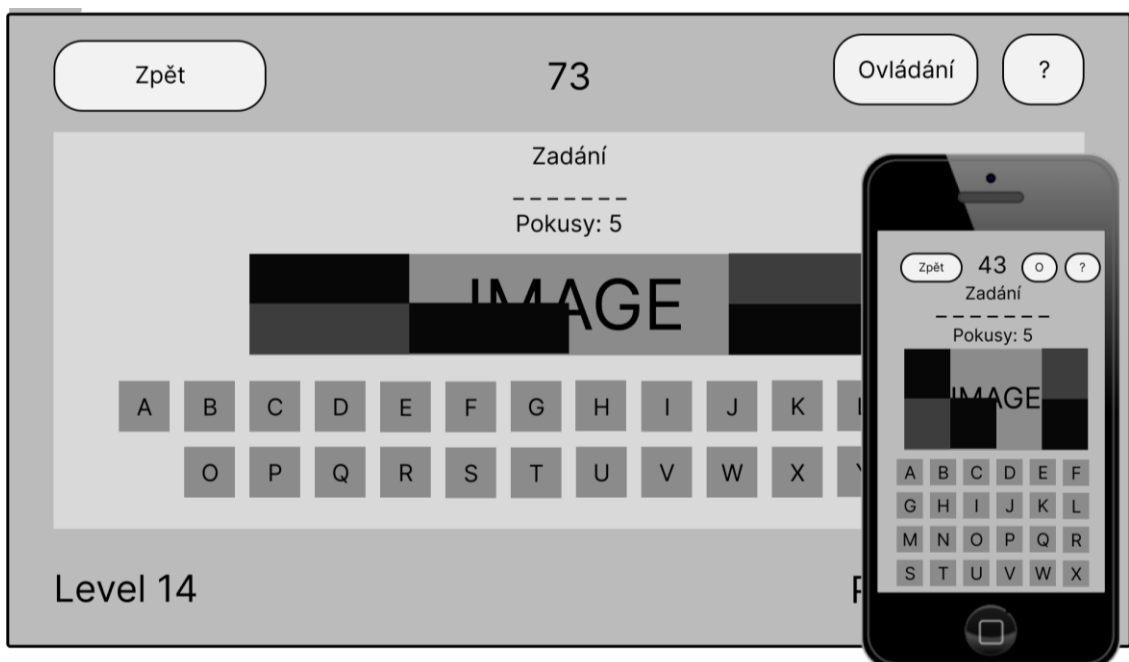
Obrázek 17 Wireframe – Guess the Word

Součástí návrhu minihry *Picture Match* je popis zadání toho, co má uživatel ve hře dělat, počet pokusů, který má na dohrání hry, 6 pseudonáhodně vygenerovaných obrázků a 6 vstupů, ke kterým budou obrázky přidělovány (viz Obrázek 18).



Obrázek 18 Wireframe – Picture Match

Wireframe vytvořený pro hru *Guess the Picture* je složený ze zadání, vstupů pro doplnění písmen, počtu zbylých pokusů, vygenerovaného obrázku a klávesnice. Obrázek bude na začátku hry celý překrytý (viz Obrázek 19).



Obrázek 19 Wireframe – Guess the Picture

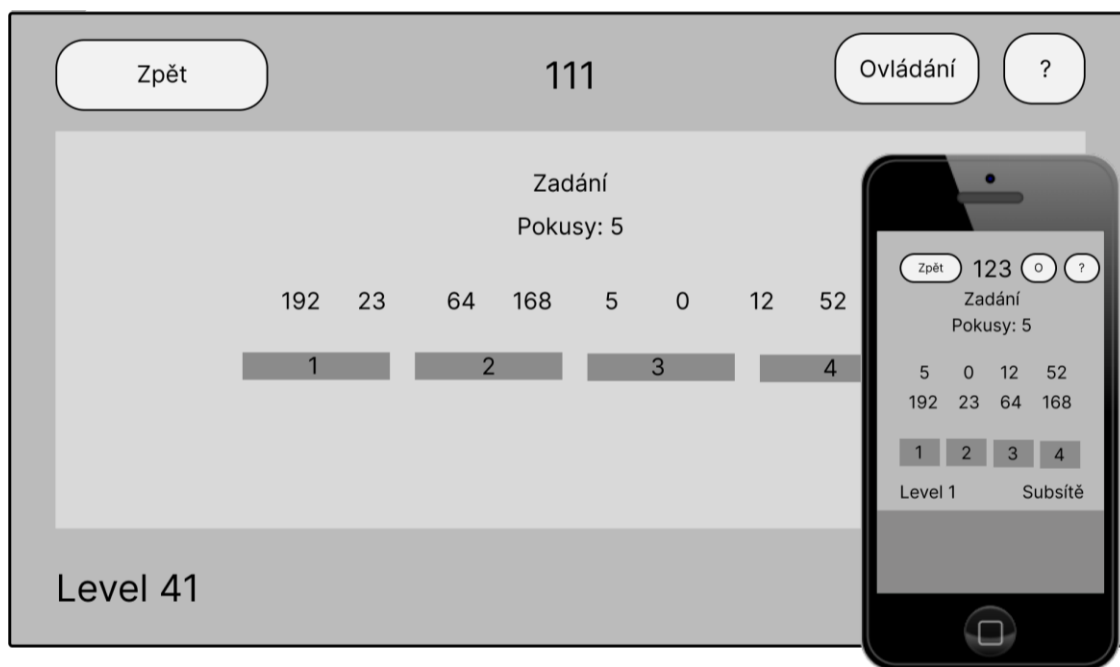
Návrh další minihry s názvem *Put the Words* je složený z prvků obecného popisu zadání, počtu pokusů a termínů, které bude moci uživatel přidělit k jejich definicím. Pro vítězství ve hře bude uživatel muset přiřadit správně všechny pojmy (viz Obrázek 20).



Obrázek 20 Wireframe – Put the Words



Pátá minihra, která se jmenuje *Count the Number*, má wireframe velmi podobný s hrou *Put the Words* s tím rozdílem, že tato hra bude založena na počítání a uživatel bude mít k dispozici více vygenerovaných čísel než míst k jejich přidělení. Na rozdíl od hry *Put the Words* uživatel nebude muset využít všechna čísla (viz Obrázek 21).



Obrázek 21 Wireframe – Count the Number

Posledním návrhem je dialog, který bude zobrazen po kliknutí na nápovědu nebo ovládání uvnitř spuštěné hry. Součástí dialogu bude text, obrázek a tlačítko pro jeho opuštění a návrat do hry (viz Obrázek 22).



Obrázek 22 Wireframe – dialog pro nápovědu

## 6 NAPROGRAMOVÁNÍ WEBOVÉ APLIKACE

Po ukončení navrhovací fáze došlo k zahájení vývoje webové aplikace s využitím programovacích jazyků HTML, CSS a především Javascriptu. V rámci souborů aplikace jsou k dispozici tři adresáře, a to *css*, *scripts* a *src* obsahující veškeré fonty a obrázky. Mimo tyto adresáře se poté nachází ještě soubor *index.html*.

### 6.1 Skript menu.js

Po spuštění webové stránky je uživateli jako první zobrazeno menu složené ze čtyř tagů sloužících pro přesun do dalších oblastí aplikace. Všechny tagy, které jsou součástí hlavní navigace (viz Obrázek 23), mají implementované své vlastní unikátní ID a atribut *onclick*, který zajišťuje spuštění funkcí *goToCampaign*, *goToLevels*, *goToTable* a *goToSettings* při kliknutí na oblast daného HTML elementu. Implementace těchto funkcí je součástí skriptu *menu.js*, jehož hlavním účelem je zajištění pohybu napříč hrou, tvorba animací a interaktivita.

```
<!--Menu block-->
<div class="nav" id="main-menu">
  <h1 class="main-title">Digitální svět</h1>
  <div class="btn" id="start" onclick="goToCampaign()">Kampaň</div>
  <div class="btn" id="levels" onclick="goToLevels()">Levely</div>
  <div class="btn" id="tab" onclick="goToTable()">Tabulka</div>
  <div class="btn" onclick="goToSettings()">Ovládání</div>
</div>
```

Obrázek 23 HTML kód hlavního menu

Pro optimální chod hry jsou zásadní dvě funkce, a to *clearBlock* a *loadBlock*. Součástí obou funkcí je poté jeden parametr, kterým je unikátní ID libovolného tagu. Po spuštění funkce *clearBlock* dojde k načtení elementu s pomocí funkce *getElementById* a následně je změněna jeho CSS vlastnost *display* na *none*. U funkce *loadBlock* je proces stejný s tím rozdílem, že CSS vlastnost *display* je nastavena na *block*. Tyto funkce nám tedy slouží pro zobrazení nebo zneviditelnění určitých sekcí aplikace.

Důležitými prvky webové hry jsou animace, které neplní pouze roli efektů, ale současně jsou důležité pro získání času při načítání jednotlivých miniher a sekcí aplikace. V rámci hry je použito velké množství obrázků, ikon a textů, jejichž načtení potřebuje určitý časový interval. Tento interval nám zajišťují funkce *darkenScreen* a *lightenScreen*. První funkcí je *darkenScreen*, která vytvoří nový element pomocí funkce *createElement*, přidá mu třídu

*dark-overlay* a poté jej vloží do tagu *body* pomocí metody *appendChild*. Následně nastaví jeho CSS vlastnost *opacity* na hodnotu 1, čímž zajistí postupné ztmavení obrazovky. Tento efekt trvá 600 ms po jejichž uplynutí je spuštěna funkce *lightenScreen*, která obrazovku opět zesvětlí a zobrazí uživateli nový obsah webové stránky (viz Obrázek 24). V rámci aplikace jsou funkce využívány společně pro přechody mezi jednotlivými sekcemi hry.

```
function lightenScreen() {
  var overlay = document.querySelector(".dark-overlay");
  overlay.style.opacity = 0;
  setTimeout(function () {
    document.body.removeChild(overlay);
  }, 600);
}
```

Obrázek 24 JS Funkce *lightenScreen*

Během vývoje aplikace musela být po provedení základního otestování přidána funkce *blockButtons*. Tato funkce po kliknutí na libovolné tlačítko v aplikaci deaktivuje určitým elementům CSS vlastnost *pointerEvents* změnou hodnoty na *none*. Díky tomu nemůže uživatel při animaci ztmavení a zesvětlení znovu použít tlačítko, což by vedlo k pádu aplikace. Po provedení animace dojde k obnovení původních vlastností všech tlačítek.

Z důvodu, aby nedocházelo k psaní redundantního kódu, vznikla funkce *clearAllWithouBlock*. Její hlavní role je deaktivace všech nepotřebných elementů a načtení pouze jednoho hlavního bloku na základě ID, které je získáno z parametru.

U herního módu kampaně má uživatel k dispozici zobrazení svého dosavadního dosaženého skóre. Aby byla hodnota skóre vždy aktuální byla vytvořena funkce *refreshScore*, která načítá data z *localStorage* a vkládá je do všech potřebných elementů aplikace (viz Obrázek 25).

```
function refreshScore() {
  const scores = document.querySelectorAll(".campaign-menu-score");

  scores.forEach((score) => {
    score.textContent = "Celkové skóre: " + localStorage.getItem("score");
  });
}
```

Obrázek 25 JS Funkce *refreshScore*

Pokud uživatel zvládne projít celou kampaní, může si do tabulky svých osobních výsledků uložit hodnotu dosaženého skóre. V implementaci funkce *endCampaign* dochází k načtení

dat z *localStorage* a následného vytvoření nového záznamu do tabulky, který se skládá z ID, data dokončení kampaně a dosaženého skóre. Po vložení dat do tabulky dojde k odstranění dosaženého celkového skóre, levelu a aktuálního skóre z *localStorage* (viz Obrázek 26). Uživatel si poté může tabulku zobrazit v hlavním menu a řadit data na základě kalendářního dne, kdy byla kampaň dokončena nebo hodnoty získaného skóre. Zdrojový kód s funkcemi pro řazení dat je součástí skriptu *table.js*.

```
function endCampaign() {
  clearAllWithoutBlock("main-menu");
  if (localStorage.getItem("score")) {
    let tableData = localStorage.getItem("tableData");
    tableData = tableData ? JSON.parse(tableData) : [];

    let newEntry = {
      attempt: tableData.length + 1,
      date: new Date().toLocaleDateString(),
      score: localStorage.getItem("score"),
    };

    tableData.push(newEntry);

    localStorage.setItem("tableData", JSON.stringify(tableData));

    localStorage.removeItem("score");
    localStorage.removeItem("level");
    localStorage.removeItem("actualScore");
  }
}
```

Obrázek 26 JS funkce endCampaign

Každá minihra, která je součástí webové aplikace, má svou vlastní funkci pro její spuštění. Tyto funkce jsou spouštěny pomocí tlačítek a slouží k načtení minihry a odstranění nepotřebných elementů na stránce.

Součástí skriptu *menu.js* jsou také funkce zajišťující přesun z hlavního menu do dalších částí aplikace. První funkcí je *goToCampaign*, která na počátku nastaví správný herní mód a následně zjistí, zda kampaň byla již v minulosti spuštěna. Pokud ano, dojde k načtení dosavadního skóre a úrovně, na kterou se uživatel na daném zařízení dostal. V případě, kdy uživatel herní mód ještě nikdy nespustil nebo už jej v minulosti dohrál, je uživateli vygenerováno nové počáteční skóre s hodnotou 10000 bodů a nastavena úroveň na

hodnotu 1. Další část zdrojového kódu je složená ze dvou switchů, které hráči na základě aktuálního levelu zobrazí odpovídající blok elementů.

V rámci menu je implementována také funkce *goToLevels*, která obsahuje podobnou logiku jako *goToCampaign* s tím rozdílem, že neuchovává aktuální úroveň a hodnotu skóre.

Tlačítka tabulka a ovládání mají také své funkce, a to *goToTable* a *goToSettings*. *GoToTable* slouží pro načtení dat o dosavadních výsledcích dosažených na daném zařízení a jejich následnému zobrazení v přehledné tabulce. Funkce *goToSettings* je určena pro uživatele, kteří jsou na webové stránce poprvé a potřebují znát alespoň základní informace o tom, jakým způsobem hru ovládat.

## 6.2 Třídy jednotlivých her

Hlavní částí webové aplikace je 7 skriptů, které tvoří logickou stránku všech miniher. V rámci každého skriptu se poté nachází vždy zdrojový kód jedné třídy.

### 6.2.1 Třída Game

Základním prvkem implementace herního systému je skript *game.js*, ve kterém se nachází třída *Game*. Tato třída obsahuje konstruktor a metody *init*, *blockButtons*, *startGame*, *stopGame*, *updateTimer*, *buttonBackClick*, *generateHeartsHtml*, *initiateGame*, *clear* a *show* (viz Obrázek 27).

Konstruktor je složený ze sedmi parametrů minihry, mezi které patří čas na dokončení, unikátní ID, název a popis minihry, text pro prohru, počet správných řešení na vítězství a maximální množství chyb, které hráč může udělat. V rámci konstruktoru dochází také k získání instance třídy *Timer*, zavolání metody *init* a k nadefinování textů pro tlačítka (viz Obrázek 27).

```
class Game {
  constructor(
    time,
    id,
    gameName,
    gameDescription,
    loseText,
    countToWin,
    loseCount
  ) {
    this.id = id;
    this.gameName = gameName;
    this.gameDescription = gameDescription;
    this.loseText = loseText;
    this.countToWin = countToWin;

    this.winCount = 0;
    this.loseCountFixed = loseCount;
    this.loseCount = 0;

    this.score = time;
    this.startScore = time;
    this.timer = new Timer(() => this.updateTimer(), 1000);
    this.init();

    this.endGameText = "Konec hry, vaše skóre: 0";
    this.startGameButtonText = "Start";
    this.restartGameButtonText = "Restart";
    this.winningText = "Gratulujeme vyhrál jste!<br />Vaše skóre: ";
  }
}
```

Obrázek 27 Konstruktor třídy Game

Metoda *init* byla navržena tak, aby umožnila vytvoření obrazovky zachycující herní prostředí, ve kterém se bude hráč nacházet těsně před spuštěním dané minihry. Tato metoda uživateli zároveň zobrazí základní popis hry a tlačítko, po jehož stisknutí dojde k animaci a zavolání metody *initiateGame*. Aby se zabránilo neustálému používání metody *init* při opakovaném spuštění stejné minihry, byla navržena dodatečná metoda *startGame*, která nastavuje pouze nezbytné prvky, přičemž ostatní prvky jsou již součástí instance minihry.

*GenerateHeartsHtml* je metoda, která slouží pro vygenerování počtu pokusů, jež má uživatel k dispozici na dokončení minihry. Uvnitř metody se nachází cyklus, který vygeneruje na základě atributu *loseCountFixed* potřebný počet pokusů na dokončení, které jsou reprezentovány srdíčky. Na začátku hry má uživatel k dispozici vždy pouze plná srdce, ale

pokud udělá chybu, tak je mu jedno srdce zprava zaměněno za prázdné. Na konci jsou pokusy v podobě srdcí vloženy do HTML elementu s využitím metody *innerHTML* (viz Obrázek 28).

```
generateHeartsHtml() {
  let heartsHtml = "";
  const fullHeartSrc = "./src/img/srdce/heart_full.png";
  const emptyHeartSrc = "./src/img/srdce/heart_empty.png";
  for (let i = 0; i < this.loseCountFixed; i++) {
    heartsHtml += `<img class="heart" src='${{
      i < this.lossCount ? fullHeartSrc : emptyHeartSrc
    }}' alt='${{
      i < this.lossCount ? "Plné srdce" : "Prázdné srdce"
    }}' style='width: 20px; height: 20px;'>`;
  }

  this.userInputSection.innerHTML = "";
  this.userInputSection.innerHTML += `<div id='${{this.gameName}}-chanceCount
  -${{this.id}}'>${{heartsHtml}}</div>`;
}
```

Obrázek 28 Metoda pro generaci životů

Metody *clear* a *show* slouží pro vyčištění uživatelského rozhraní tak, aby se zamezilo psaní redundantního kódu. Jde o jednoduché metody, které vkládají nebo odebírají HTML elementům třídu *hide*, která má CSS vlastnost *display: none*. Další dvě jednoduché metody jsou *stopGame* a *blockButtons* pro zastavení časovače a vynulování skóre. Tyto metody jsou často překryté a rozšířené v dalších třídách, které dědí od třídy *Game*.

Velmi důležitou metodou je *initiateGame*, která je volána poté, co se uživatel rozhodne spustit minihru. Na počátku zdrojového kódu této minihry dojde k nastavení počtu správných řešení na 0, což slouží k identifikaci, že uživatel ještě nedosáhl požadovaného počtu správných řešení k vítězství. Dále zde dojde k odstranění všech nepotřebných textů a elementů z obrazovky, zobrazení prvků dané minihry, vygenerování srdíček a spuštění časovače. Součástí této metody je také ověření, zda se jedná o herní mód kampaň a v případě, kdy se skutečně o kampaň jedná, dojde u dané úrovně k načtení aktuálního skóre ze stavu minihry uloženého v paměti.

V rámci implementace třídy *Game* je také metoda *updateTimer*. Tato metoda je volána každých 1000 ms a dochází v ní vždy ke snížení skóre o hodnotu 1. Současně s tím je kontrolováno, zda skóre nedosáhlo již hodnoty 0 a pokud ano, tak je minihra ukončena, dojde k vynulování časovače, zobrazení textu s prohrou a vykreslení tlačítka pro možné



zopakování hry s textem *restart*. Zajímavou částí je také ověření, zda skóre hráče není již menší nebo rovné desíti. Pokud by tomu tak bylo, dojde ke změně animace a barvy skóre na červenou (viz Obrázek 29). V metodě také dochází ke kontrole herního módu a v případě, kdy se jedná o kampaň, dojde k odečtení patřičného počtu bodů od celkového získaného skóre.

```
if (timerElement) {
  this.score--;
  timerElement.innerText = this.score;
}

if (this.score <= 10) {
  timerElement.style.color = "rgb(255, 68, 68)";
  timerElement.classList.add("blink");
} else {
  timerElement.style.color = "white";
  timerElement.classList.add("pulse");
}
```

Obrázek 29 Zdrojový kód udávající barvu skóre

Poslední metoda, která se nachází v rámci třídy *Game*, je *buttonBackClick*. Po jejím spuštění dochází k zastavení a vynulování časovače a následnému vyhodnocení výsledku hry. Pokud uživatel dokázal získat dostatek bodů k vítězství je mu zobrazen vítězný text, počet získaných bodů ve hře a tlačítko pro zopakování hry nebo pokračování v kampani. Druhou možností je, že uživateli již vypršel počet pokusů na dokončení a bude mu zobrazeno skóre 0 společně s textem o prohře. Poslední možností je opuštění hry pomocí tlačítka pro přesun zpět do menu. V případě, kdy dojde k přesunu do menu a uživatel je v herním módu kampaně, dochází k uložení aktuálního skóre. Pokud se následně hráč rozhodne v kampani pokračovat bude mu na počátku hry nastaveno uložené skóre namísto výchozího.

### 6.2.2 Třída *DragableGame*

Součástí webové aplikace jsou také minihry založené na principu *drag and drop*. Pro tyto minihry byl vytvořen skript *dragablegame.js* obsahující třídu *DragableGame*. Třída má své specifické metody, jako *shuffleArray*, *createDraggableElement*, *createDroppableElement*, *generateRandomItemsArray*, *dragStart*, *dragEnter*, *dragOver*, *dragLeave* a *drop*. *DragableGame* je navíc potomkem třídy *Game*, takže dědí její atributy a metody, které ve spoustě příkladů překrývá.

V rámci konstruktoru *DragableGame* dochází oproti její rodičovské třídě k rozšíření o dva parametry. Prvním parametrem je *task* sloužící k upřesnění zadání a druhým je *totalDraggableItems*, který udává celkový počet prvků, se kterými bude možno ve hře pohybovat.

Uvnitř třídy *DragableGame* lze nalézt také implementaci pseudonáhodné generace, kterou zajišťují metody *shuffleArray* a *generateRandomItemsArray*. Obě tyto metody využívají funkci *Math.random*, která slouží pro generaci pseudonáhodného čísla v rozsahu od 0 do 1. Rozdíl mezi metodami spočívá v tom, že *shuffleArray* pole pouze zamíchá na rozdíl od *generateRandomItemsArray*, která navíc na základě parametru dokáže vytvořit pole o různé velikosti (viz Obrázek 30).

```
generateRandomItemsArray(n, originalArray) {  
  let res = [];  
  let clonedArray = [...originalArray];  
  if (n > clonedArray.length) n = clonedArray.length;  
  for (let i = 1; i <= n; i++) {  
    const randomIndex = Math.floor(Math.random() * clonedArray.length);  
    res.push(clonedArray[randomIndex]);  
    clonedArray.splice(randomIndex, 1);  
  }  
  return res;  
}
```

Obrázek 30 Metoda pro vygenerování pole s náhodnými prvky

Při vytváření minihry s využitím *drag and drop* systému jsou použity dvě metody, a to *createDraggableElement* a *createDroppableElement*. V rámci jejich implementace je použita funkce *insertAdjacentHTML* s parametrem *beforeend*, která slouží pro vložení nového HTML obsahu za posledního potomka daného elementu (viz Obrázek 31). Součástí obou metod je parametr *item*, kterým je objekt obsahující informace o názvu obrázku, označení a barvě textu.

```
createDroppableElement(item) {
  this.matchingPairs.insertAdjacentHTML(
    "beforeend",
    `<div class="${this.gameName}-matching-pair">
      <span
        class="${this.gameName}-droppable"
        data-brand="${this.id}-${item.imageName}"
      >
        ${item.brandName}
      </span>
    </div>`
  );
}
```

Obrázek 31 Metoda pro vytvoření elementu ve třídě DragableGame

Pro zajištění správné funkčnosti principu *drag and drop* je využito pěti metod, a to *dragStart*, *dragEnter*, *dragOver*, *dragLeave* a *drop*. Všechny tyto metody mají parametr *event*, který reaguje na vyvolanou událost.

V momentě, kdy uživatel začne přetahovat prvek, je spuštěna metoda *dragStart*. Součástí události je nastavení dat, která musí být přenesena společně s přetahovaným prvkem. Tato data jsou uložena do klíče *text* a obsahují ID hodnotu přetahovaného prvku (viz Obrázek 32).

```
dragStart(event) {
  event.dataTransfer.setData("text", event.target.id);
}
```

Obrázek 32 Metoda dragStart

Metoda *dragEnter* nastává v momentě, kdy se přetahovaný prvek dostane nad oblast, kam je možné jej přemístit. Součástí implementace je ověření, zda se jedná o element, do kterého lze vkládat přetahovaný prvek a také zda již dříve nebyl nějaký jiný prvek vložený. Pokud jsou podmínky splněny, tak je oblast pro vložení prvku zvětšena díky přidělení patřičné třídy (viz Obrázek 33).

```
dragEnter(event) {
  if (
    event.target.classList &&
    event.target.classList.contains(`${this.gameName}-droppable`) &&
    !event.target.classList.contains("dropped")
  ) {
    event.target.classList.add(`${this.gameName}-droppable-hover`);
  }
}
```

Obrázek 33 Metoda dragEnter

Pro zablokování výchozího chování prohlížeče byla implementována metoda *dragOver*, která využívá stejnou podmínku jako *dragEnter* s tím rozdílem, že přidání třídy nahrazuje funkce *preventDefault*.

Třetí metodou využívající podmínku pro ověření, zda lze vložit přetahovaný prvek do cílového elementu, je *dragLeave*. Tato metoda je volána v momentě, kdy uživatel opustí oblast, do které lze vložit přetahovaný element. Po této události dojde k odstranění třídy, která způsobuje zvětšení oblasti cílového elementu a tomu je tak nastavena původní velikost.

Nejsložitější metodou třídy *DragableGame* je *drop*, která je spuštěna v momentě, kdy uživatel pustí přetahovaný prvek. Na počátku metody dojde k odstranění třídy, která zvětšuje oblast elementu. Pokud uživatel skutečně pustil prvek nad určenou oblastí, dojde k načtení hodnoty atributu pomocí metody *getAttribute*. Tato hodnota atributu je následně porovnána s daty přetahovaného prvku, jež byla získána pomocí metody *getData* s parametrem *text*. V případě, kdy jsou si hodnota atributu a dat rovny, dojde ke změně stylů pomocí přidání třídy tak, aby se přetahovaný prvek již nemohl znovu použít. Současně s tím zůstane přetahovaný prvek zobrazený v cílovém elementu, aby bylo jasné, že byl přiřazený správně. Následující logika metody vyhodnotí, zda uživatel již dosáhl požadovaného počtu správných přiřazení a pokud ano, dojde k ukončení hry a hráči je zobrazeno jeho dosažené skóre společně s gratulací k výhře. Pokud by se jednalo o herní mód kampaň, bude mu navíc zobrazeno tlačítko pro pokračování v kampani a dojde ke změně celkového skóre. V rámci zdrojového kódu metody je rovněž implementována logika pro situaci, kdy uživatel přesune přetahovaný prvek do jiného elementu, než je jeho cílová destinace. V tento moment se data s atributem nerovnájí a dochází ke snížení skóre o 5 bodů. Současně s tím je uživateli sebrán jeden pokus na dokončení. Pokud by se počet pokusů na dokončení rovnal hodnotě 0, dojde k ukončení hry a uživateli je zobrazen text pro prohru, skóre s hodnotou 0 a tlačítko pro

spuštění hry s textem *restart*. U herního módu kampaně navíc dojde k odečtení bodů od celkového skóre.

### 6.2.3 Třída *GuessTheWord*

Uvnitř skriptu *guesstheword.js* lze nalézt potomka třídy *Game* a to *GuessTheWord*. Tato třída má rozšířený konstruktor o parametr *options* a obsahuje metody *generateRandomValue*, *generateWord* a *createGame*. Do parametru *options* je poté předáván objekt, který je založený na principu klíče a hodnoty. Klíč symbolizuje libovolný pojem z počítačových sítí a uvnitř hodnoty je poté text s nápovědou na uhodnutí slova uloženého v klíči (viz Obrázek 34).

```
const gameSettingsLevel13 = {
  SMTP: "Protokol pro přenos elektronické pošty mezi poštovními servery",
  POP3: "Protokol pro stahování elektronické pošty z poštovního serveru na lokální počítač.",
  IMAP: "Protokol pro správu elektronické pošty na serveru a synchronizaci mezi různými zařízeními",
  FTP: "Protokol pro přenos souborů mezi počítači v síti",
  HTTP: "Protokol pro komunikaci mezi webovým prohlížečem a webovým serverem",
  DNS: "Protokol pro překlad doménových jmen na IP adresy a naopak",
  ARP: "Protokol pro mapování mezi IP adresami a fyzickými MAC adresami v lokální síti",
};
```

Obrázek 34 Objekt s daty pro hru *Guess the word*

Pro pseudonáhodnou generaci je využita metoda *generateRandomValue*, která jako parametr přebírá pole, jehož délkou násobí hodnotu získanou pomocí funkce *Math.random*. Výsledná hodnota se poté zaokrouhlí s využitím funkce *Math.floor* a je vrácena pomocí klíčového slova *return*. Tato funkcionalita je dále implementována v rámci zdrojového kódu metody *generateWord*, jež má za úkol generovat náhodné slovo z výběru. Tento výběr byl získán prostřednictvím metody *Object.keys*, která byla aplikována na objekt získaný prostřednictvím parametru konstruktoru *options*.

Velmi podstatnou metodou pro třídu *GuessTheWord* je *createGame*, která je součástí rozšířené metody *initiateGame*. Na jejím počátku se nachází metoda *generateWord* a je vytvořena abeceda. Následně dojde za pomoci cyklu *foreach* k vytvoření tlačítka pro každé písmeno české abecedy společně s logikou pro určení, zda se jedná o písmeno, které je součástí hádaného slova či nikoli. V případě, kdy uživatel při hádání zvolí správné písmeno, tlačítku je přidána CSS třída tak, aby jej nebylo možné opakovaně stisknout a dojde k vyhodnocení, zda se nejedná o poslední písmeno z hádaného slova. Pokud by se jednalo o poslední hádané písmeno, došlo by k ukončení hry a zobrazení získaného skóre. Uživatel může také zvolit nesprávné písmeno, což vede k odečtení pěti bodů ze skóre a odebrání jednoho pokusu na dokončení minihry stejně jako u her využívajících *drag and drop*

funkcionalitu. Chybnému písmenu je přidána CSS třída, která zajistí, že na tlačítko nepůjde znovu kliknout, ale bude mít rozdílnou barvu než správně uhodnuté písmeno.

```
const game13 = new GuessTheWord(  
  25,  
  "game13",  
  GTWGameName,  
  GTWLabel,  
  loseText,  
  1,  
  3,  
  gameSettingsLevel13  
);
```

Obrázek 35 Vytvoření instance třídy `GuessTheWord`

#### 6.2.4 Třída `GuessThePicture`

Třída `GuessThePicture`, která je součástí skriptu `guessthepicture.js`, má spoustu podobných metod jako `GuessTheWord`. V rámci zdrojového kódu třídy `GuessThePicture` lze nalézt ale rozdílné metody jako `generatePicture` a `hideBlock`. V konstruktoru jsou také předávána jiná data, která jsou v podobě pole objektů se čtyřmi vlastnostmi, a to název obrázku, označení, popis a barva (viz Obrázek 36).

```
const gameSettingsLevel15 = [  
  {  
    imageName: "src/img/GTP_images/coax.png",  
    brandName: "Koaxiální",  
    description: "O jaký kabel (přenosové médium) se jedná?",  
    color: "#333333",  
  },  
  {  
    imageName: "src/img/GTP_images/TP.png",  
    brandName: "Dvojlinka",  
    description: "O jaký kabel (přenosové médium) se jedná?",  
    color: "#333333",  
  },  
  {  
    imageName: "src/img/GTP_images/FO.png",  
    brandName: "Optický",  
    description: "O jaký kabel (přenosové médium) se jedná?",  
    color: "#333333",  
  },  
];
```

Obrázek 36 Objekt s daty pro hru `Guess the picture`

Metoda *generatePicture* slouží k pseudonáhodnému vygenerování jednoho obrázku, který je následně překrytý kontejnerem. Součástí kontejneru je poté 16 stejně velkých elementů vypočítaných na základě velikosti obrázku. Výsledná velikost jednoho elementu je pak rovna jedné čtvrtině celkové šířky a výšky obrázku. Jakmile dojde k vytvoření kontejneru, všem elementům je přidán naslouchač volající metodu *hideBlock* (viz Obrázek 37).

```
const overlayBlocks = document.querySelectorAll(".overlay-block");
overlayBlocks.forEach((block) => {
  block.addEventListener("click", function () {
    instance.hideBlock(this);
  });
});
```

Obrázek 37 Nastavení naslouchačů pro elementy v metodě *generatePicture*

*HideBlock* je jednoduchá metoda, jejíž úlohou je zobrazit oblast obrázku nacházející se za elementem na který uživatel klikl. V rámci implementace metody *hideBlock* dojde ke změně stylu *opacity* na hodnotu 0, což způsobí, že element ztratí viditelnost. Dále dojde ke změně vlastnosti *pointerEvents* na *none*, čímž je elementu zablokována možnost interakce. Na konci metody dojde ke snížení aktuálního skóre o 5 bodů.

```
const game15 = new GuessThePicture(
  80,
  "game15",
  GTPGameName,
  GTPLabel,
  loseText,
  1,
  3,
  gameSettingsLevel15
);
```

Obrázek 38 Vytvoření instance třídy *GuessThePicture*

### 6.2.5 Třída *PictureMatch*

Jedním z potomků *DragableGame* je třída *PictureMatch*. Zdrojový kód této třídy se nachází ve skriptu *picturematch.js* a jeho součástí jsou specifické metody *getRandomBrand*, *createGame* a také konstruktor, který je rozšířený o 6 parametrů týkajících se dat pro nastavení jednotlivých elementů (viz Obrázek 39).

```
const gameSettingsLevel5brandsPAN = [  
  {  
    imageName: "src/img/topologie/PAN_1.png",  
    brandName: "PAN",  
    color: "#333333",  
  },  
  {  
    imageName: "src/img/topologie/PAN_2.png",  
    brandName: "PAN",  
    color: "#333333",  
  },  
  {  
    imageName: "src/img/topologie/PAN_3.png",  
    brandName: "PAN",  
    color: "#333333",  
  },  
  {  
    imageName: "src/img/topologie/PAN_4.png",  
    brandName: "PAN",  
    color: "#333333",  
  },  
];
```

Obrázek 39 Objekt s daty do hry *Picture match*

Metoda *getRandomBrand*, která je specifická pouze pro třídu *PictureMatch*, má jako parametr pole, ze kterého na základě pseudonáhodně vygenerovaného čísla vybere jeden prvek. Do parametru pole jsou předávána data získaná z konstruktoru.

Druhou specifickou metodou je *createGame*. Na počátku zdrojového kódu této metody dojde k odstranění původních elementů, čehož je využíváno v případě, kdy si uživatel spouští hru opakovaně. Poté dojde k naplnění pole pomocí vygenerovaných hodnot s využitím metody *getRandomBrand*. V rámci metody je také vytvořeno pole šesti elementů, které budou seřazeny abecedně na základě jejich označení. Po získání obou polí dochází k vytvoření samotných elementů pomocí metod *createDraggableElement* a



*createDroppableElement*, které jsou v případě třídy *PictureMatch* upraveny pro optimální práci s obrázky. V poslední části metody dochází k vytvoření naslouchačů, které jednotlivým elementům přidají funkcionalitu *drag and drop* systému (viz Obrázek 40).

```
this.draggableElements.forEach((elem) => {
  elem.addEventListener("dragstart", (event) => this.dragStart(event));
});

this.droppableElements.forEach((elem) => {
  elem.addEventListener("dragenter", (event) => this.dragEnter(event));
  elem.addEventListener("dragover", (event) => this.dragOver(event));
  elem.addEventListener("dragleave", (event) => this.dragLeave(event));
  elem.addEventListener("drop", (event) => this.drop(event));
});
```

Obrázek 40 Nastavení naslouchačů pro hry využívající systém drag and drop

### 6.2.6 Třída *PutTheWords* a *CountTheNumber*

Pro poslední dvě minihry byly vytvořeny třídy *PutTheWords* a *CountTheNumber*, které vycházejí z rodičovské třídy *DragableGame*. Tyto třídy jsou rozšířeny pouze o metodu *createGame*, která není optimalizována pro práci s obrázky, ale s oblastmi textu. Třída *PutTheWords* manipuluje s daty, kde každý přesunutelný prvek má svůj určený cílový prvek, do kterého může být vložen. Jedná se tedy o vztah 1:1. Naopak ve třídě *CountTheNumber* se nacházejí i data, která nemají žádný cílový prvek pro vložení. Hráč tedy musí rozlišit, zda se jedná o data určená k přetažení nebo pouze o náhodná data, která nesouvisí s žádným cílovým elementem.

### 6.3 Třída *Timer*

Důležitým prvkem, který je součástí skriptu *timer.js*, je časovač sloužící především pro zobrazení aktuálního skóre. V rámci skriptu je definována třída *Timer*, jejíž konstruktor vyžaduje dva nezbytné parametry. Prvním z nich je funkce, která bude opakovaně prováděna po uplynutí časového intervalu specifikovaného ve druhém parametru. Pro ovládání časovače byly vytvořeny 3 metody, a to *start* pro jeho spuštění, *stop* pro jeho zastavení a *setNullTimer*, která nastavuje hodnotu atributu *timerInterval* na *null*, aby bylo jasné, zda interval existuje nebo ne.

## 6.4 Popup

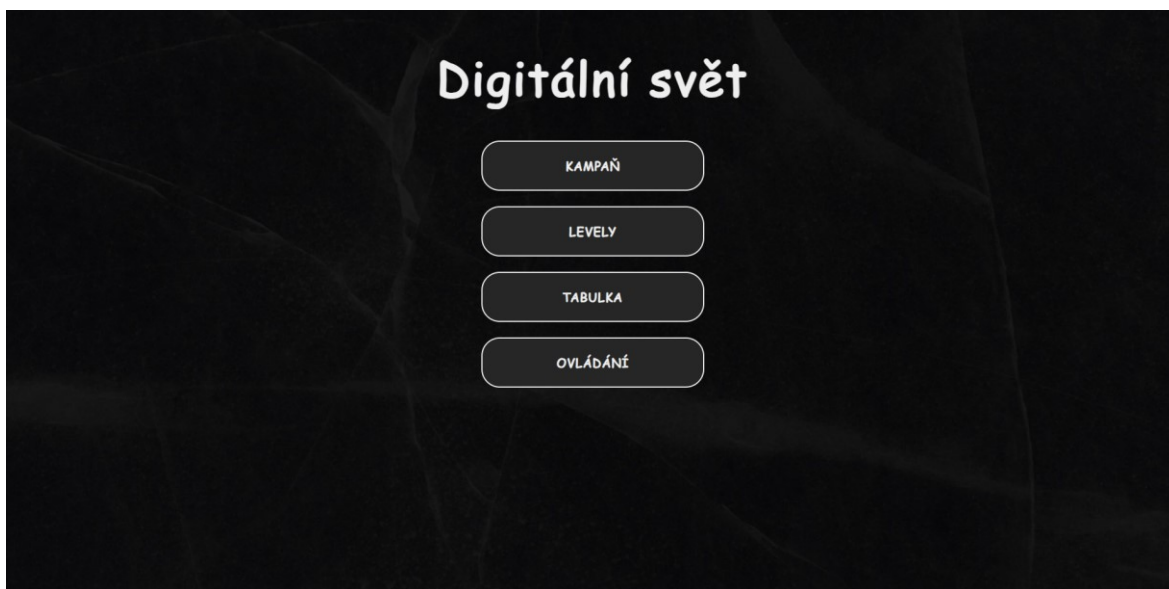
*Popup* je specifická část aplikace sloužící pro pomoc uživateli. Každá hra je opatřena tlačítky pro získání nápovědy a zobrazení ovládání dané hry. Po kliknutí na tlačítko ovládání či nápovědy dojde k postupnému zobrazení dialogového okna s textem a obrázkem, které jsou minihráč přiděleny na základě jejich ID (viz Obrázek 41). Během okénka ovšem uživateli dále utíká čas na dokončení hry. Uživatel může nápovědu nebo ovládání zrušit buď pomocí stisknutí křížku, nebo kliknutím kdekoliv mimo oblast dialogového okna.

```
{  
  gameID: "game1",  
  gameHelpText: help1,  
  gameImageHelp: "src/img/avarat/5.png",  
  gameSettingsText: GTWSettings,  
  gameImageSettings: "src/img/avarat/15.png",  
},
```

Obrázek 41 Data pro dialogová okna

## 7 POPIS APLIKACE A JEJÍHO GUI

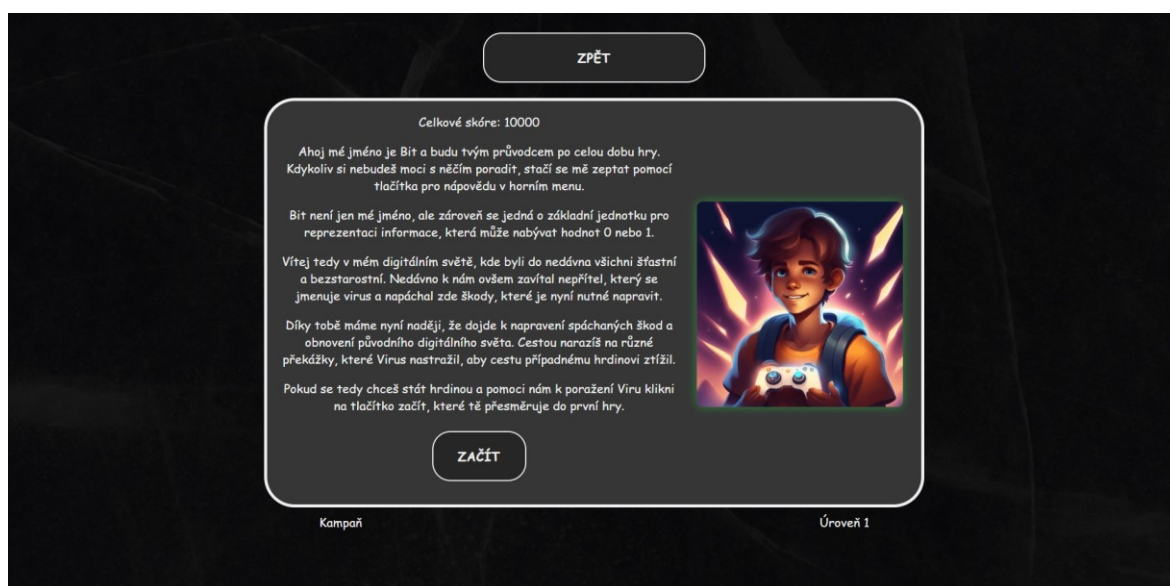
V následující kapitole bude popsáno GUI webové hry. GUI této aplikace je velmi jednoduché a snadno použitelné. Skládá se z několika tlačítek, která jsou uspořádána tak, aby uživatelům usnadnila navigaci (viz Obrázek 42). Rozložení GUI bylo vytvořeno tak, aby bylo intuitivní a uživatelé tak neměli problém s nalezením všech požadovaných funkcí.



Obrázek 42 Hlavní menu hry

## 7.1 Mód kampaň

Po stisknutí tlačítka kampaně v hlavním menu je uživateli zobrazen herní mód, který slouží k průchodu všemi úrovněmi aplikace. Tato část se skládá z několika grafických komponent, kterými jsou tlačítka pro návrat do menu, dosavadní získané skóre, text pro pochopení daného tématu, tlačítka pro pokračování do minihry, obrázek průvodce a úroveň ve které se uživatel nachází (viz Obrázek 43).



Obrázek 43 Herní mód kampaň

## 7.2 Mód levely

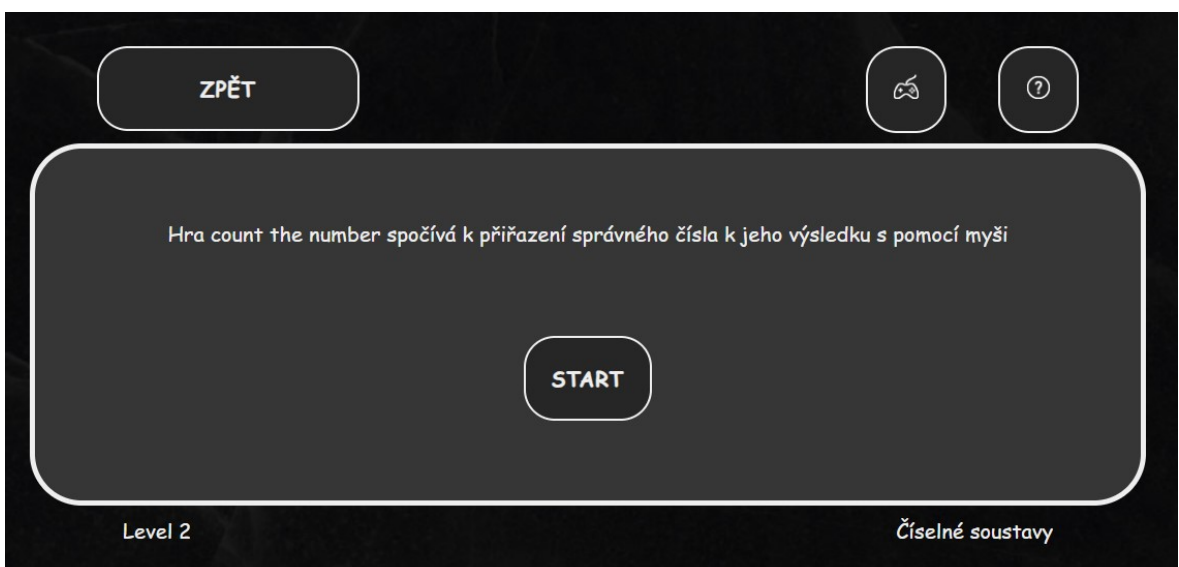
Pokud má uživatel již předchozí zkušenosti s počítačovými sítěmi nebo si chce pouze vyzkoušet různé části aplikace, může si v hlavním menu zvolit tlačítka levely a následně vybrat libovolnou úroveň, ke které by se v kampani musel dostat postupně. Součástí tohoto módu je 30 miniher, zaměřených na různé problematiky počítačových sítí (viz Obrázek 44). Samozřejmostí je tlačítka umožňující návrat do hlavního menu.



Obrázek 44 Herní mód levely

### 7.3 GUI Minihry

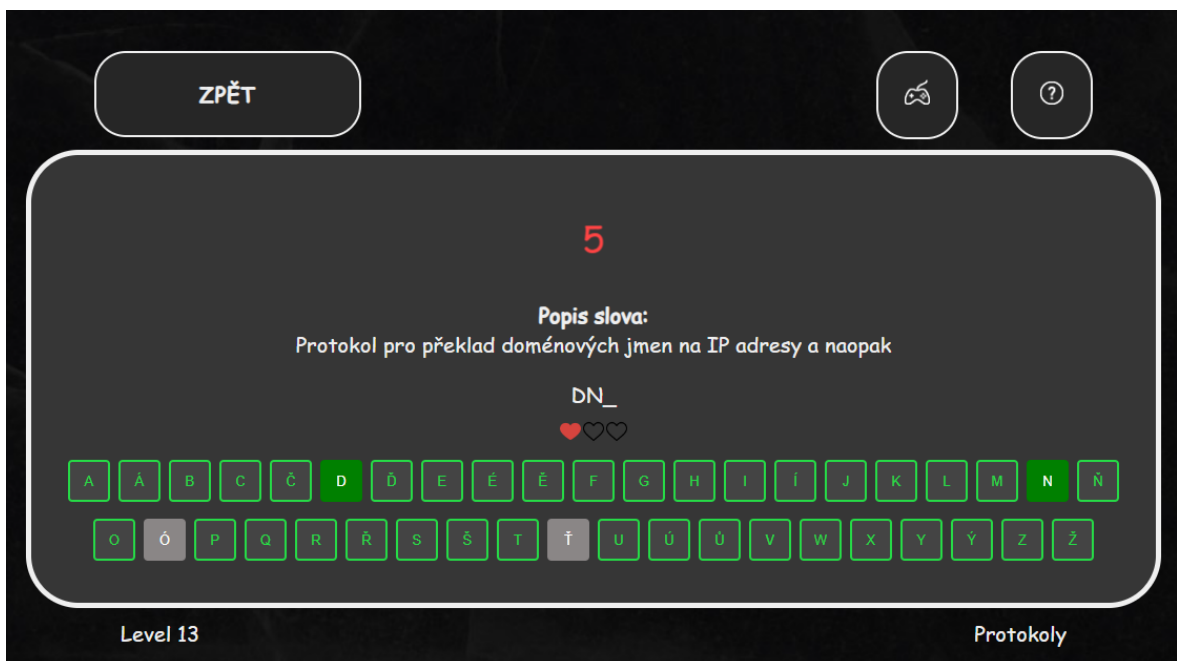
Poté, co se uživatel rozhodne pokračovat v kampani nebo si zvolí určitou úroveň z vybraných levelů, dojde k načtení obrazovky se základním popisem hry a tlačítkem, které uživatele již přenesou do samotné minihry. Dochází zde ke změně GUI, kde jsou přidány 2 tlačítka pro zobrazení ovládní a nápovědy pro danou minihru (viz Obrázek 45). Po stisknutí tlačítka *start* dojde k zobrazení společných komponent, které jsou součástí všech miniher, a také specifických komponent na základě zvolené minihry.



Obrázek 45 Menu pro spuštění konkrétní hry

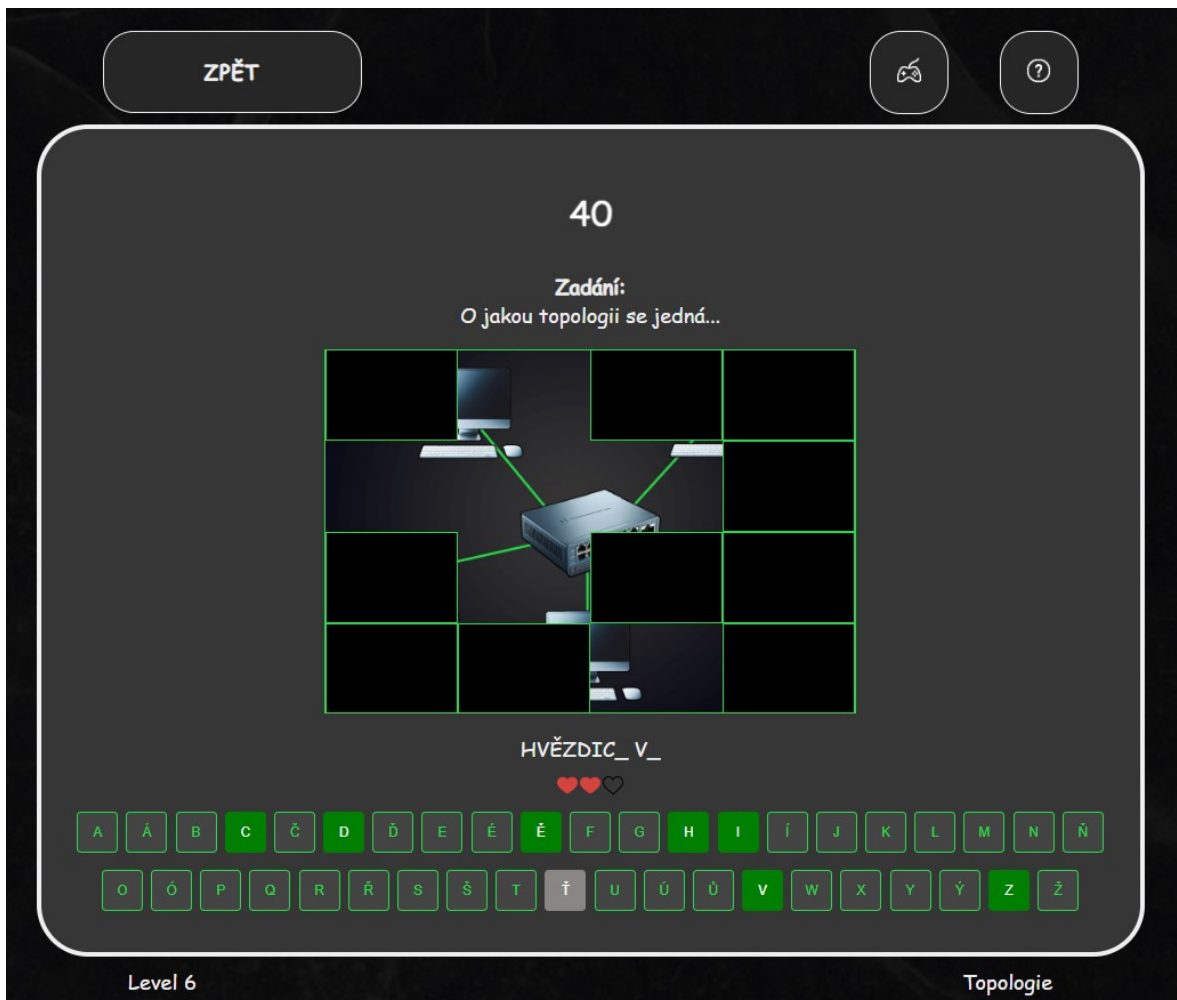
První zásadní komponentou všech miniher je ukazatel skóre, u kterého každou vteřinou dochází ke snížení o 1 bod. Jakmile se skóre dostane na hodnotu 10, změní se jeho barva na červenou, což symbolizuje, že hráči dochází čas. Dále je zde vidět popis slova a ukazatel životů, které symbolizují počet pokusů, které uživateli zbývají na dokončení. Komponenty skóre, popis a životy jsou součástí všech miniher.

*Guess the Word* je minihra, která má specifické komponenty v podobě klávesnice a oblasti hádaného slova (viz Obrázek 46). Pokud uživatel zvolí správné písmeno, bude znak zobrazen namísto podtržítka v oblasti hádaného slova, které musí uživatel pro dokončení minihry celé doplnit. V opačném případě, kdy hráč zvolí chybné písmeno, přichází o jeden pokus a bude mu odečteno 5 trestných bodů ze skóre.



Obrázek 46 Hra Guess the word

GUI minihry *Guess the Picture* je velmi podobné hře *Guess the Word* s tím rozdílem, že obsahuje navíc komponentu s obrázkem (viz Obrázek 47). Hráč má za úlohu uhodnout obrázek, který je překrytý elementy. Při kliknutí na element je zobrazena pouze část obrázku a dojde ke snížení hráčova skóre o 5 bodů.



Obrázek 47 Hra Guess the picture

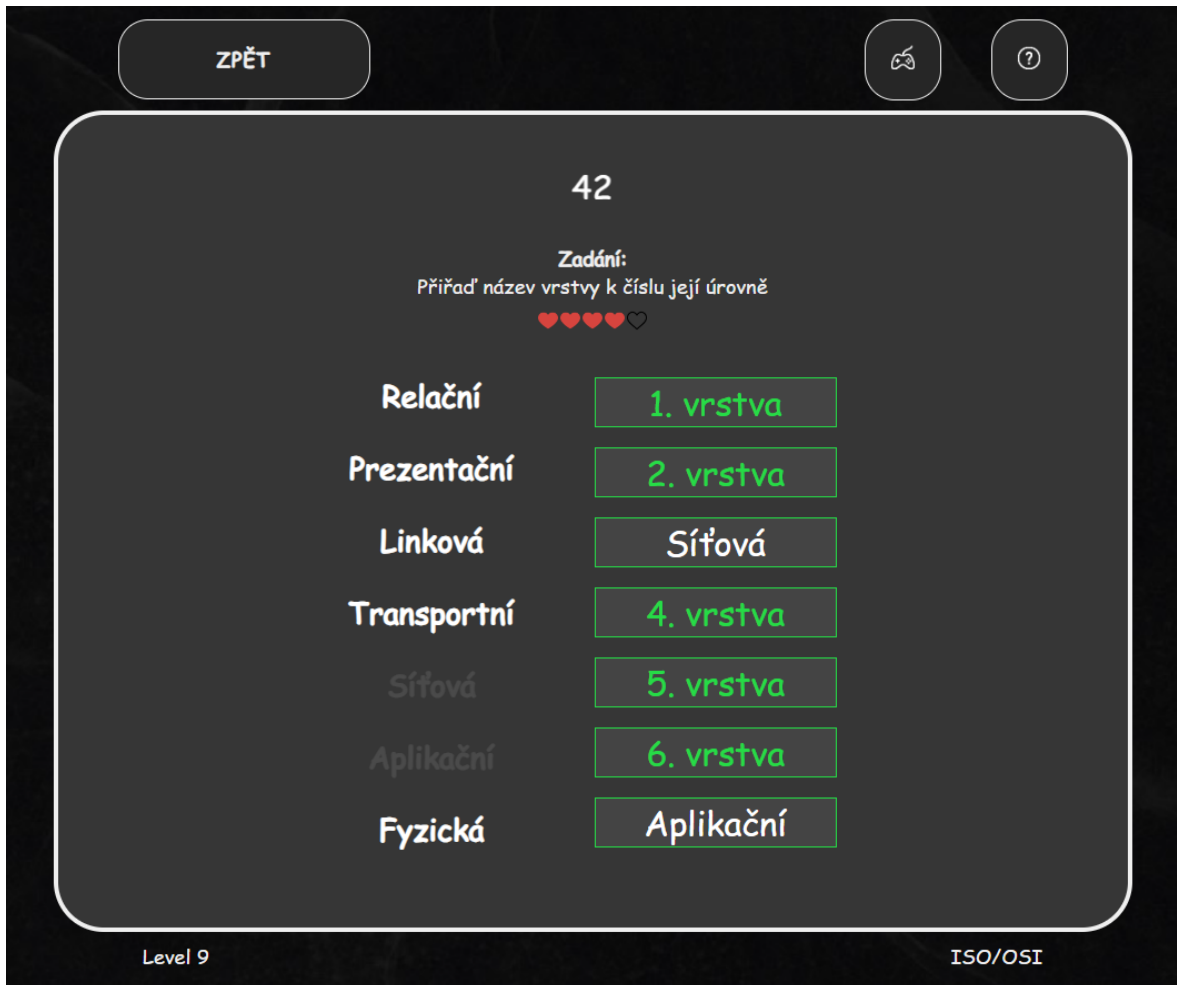
Minihra *Picture Match* využívá GUI, jejímž cílem je správné přiřazení šesti obrázků ke svým definicím. Pokud uživatel přiřadí obrázek správně, jeho původní místo bude ztmaveno a nebude jej již moci opakovaně přetáhnout (viz Obrázek 48). Současně s tím se obrázek zobrazí na místě definice. Díky tomu hráč ví, že tento pojem již správně přiřadil a může pokračovat přiřazením ostatních obrázků. Pokud dojde k přiřazení všech obrázků k jejich definicím, uživatel zvítězí ve hře.



Obrázek 48 Hra Picture match



V případě minihry *Put the Words* došlo u GUI kromě základních komponent k implementaci dvou sloupců (viz Obrázek 49). V levém sloupci jsou texty, které musí být přiřazeny do pravého sloupce. V případě této minihry musí být všechny pojmy z levého sloupce přiřazeny k textům na straně pravé.



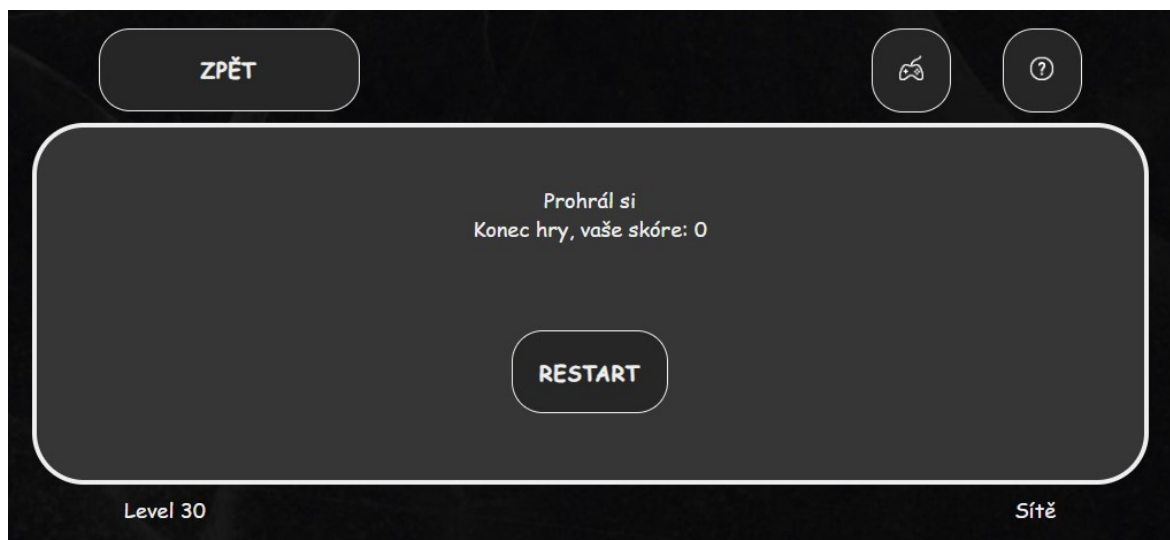
Obrázek 49 Hra Put the words

Minihra, která je určena pro témata, kde je nutné využívat matematické výpočty, je *Count the Number* s GUI velmi podobným hře *Put the words*. V případě minihry *Count the number* jsou sloupce nahrazeny řádky (viz Obrázek 50). Rozdíl mezi hrami je především v tom, že se ve vrchním řádku nachází navíc prvky, které nemají být přiřazeny a slouží pouze ke zmatení hráče.



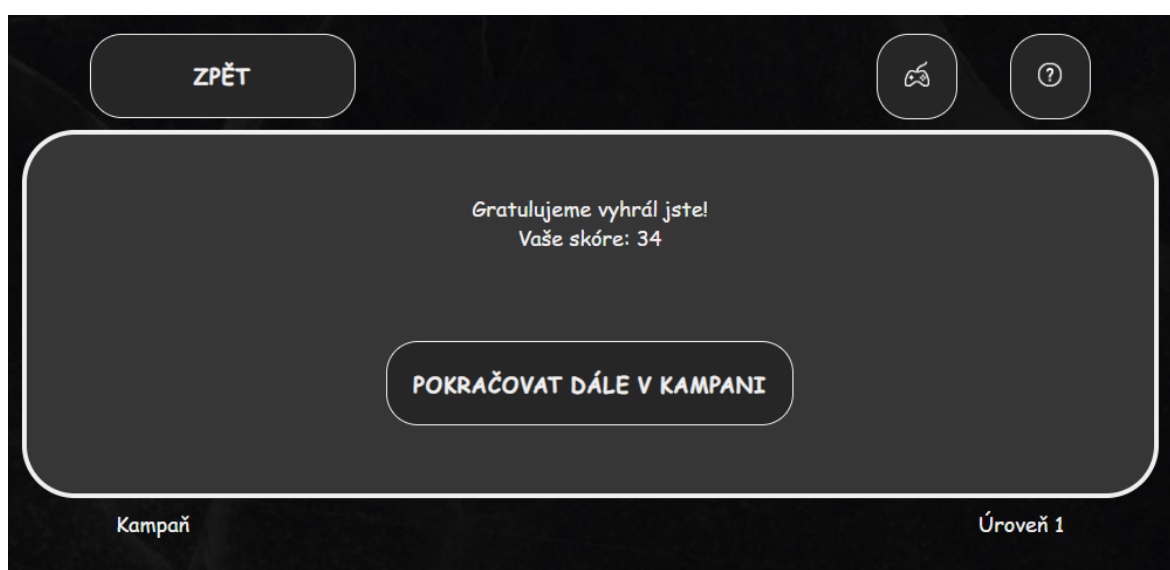
Obrázek 50 Hra Count the number

Ve všech minihrách může nastat situace, kdy uživatel prohraje. Prohra nastává v momentě, kdy již nezůstávají další životy nebo vyprší časový limit na dokončení. V obou případech je uživateli zobrazen text o prohře, dosažené skóre s hodnotou 0 a tlačítko umožňující zopakování hry (viz Obrázek 51).



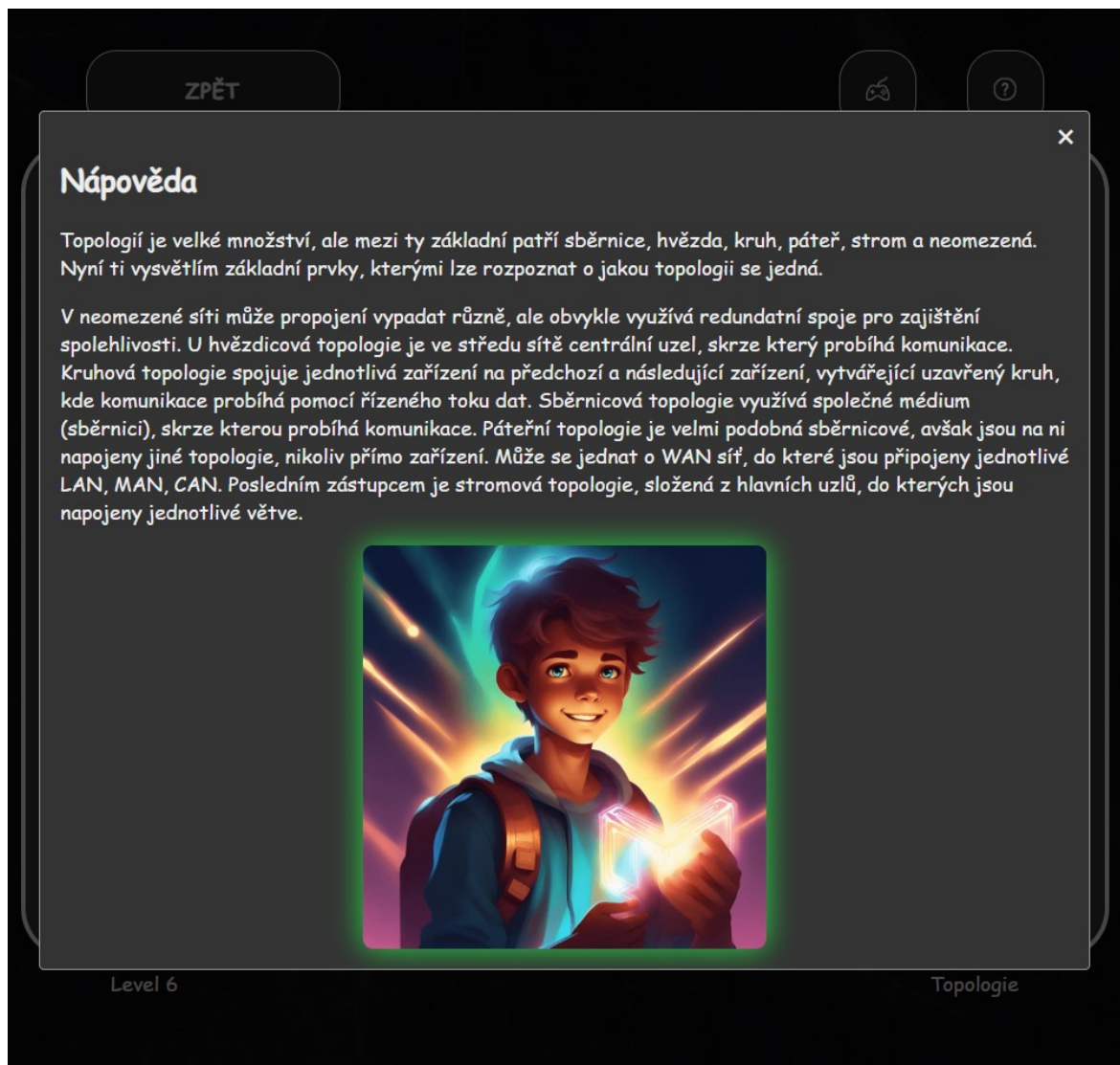
Obrázek 51 Prohra ve hře

Pokud uživatel zvítězí, což může nastat ve specifických situacích na základě zvolené minihry, je mu naopak zobrazen text s gratulací k výhře a jeho dosažené skóre. Pokud se nachází v herním módu, kde si volí libovolnou úroveň, bude mu zobrazeno i tlačítko pro zopakování minihry. V případě, kdy si hráč zvolil herní mód kampaně, bude mu zobrazeno tlačítko pro pokračování (viz Obrázek 52).



Obrázek 52 Výhra v kampani

Součástí všech miniher a jejich GUI je také možnost zobrazení nápovědy a ovládání hry. Pro tento účel jsou vždy v pravém horním rohu vytvořeny dvě tlačítka, po jejichž stisknutí dojde k zobrazení dialogového okna s textem, který by měl uživateli pomoci danou situaci vyřešit (viz Obrázek 53). V implementaci dialogu je také křížek pro jeho zrušení a obrázek průvodce.



Obrázek 53 Dialogové okno s nápovědou

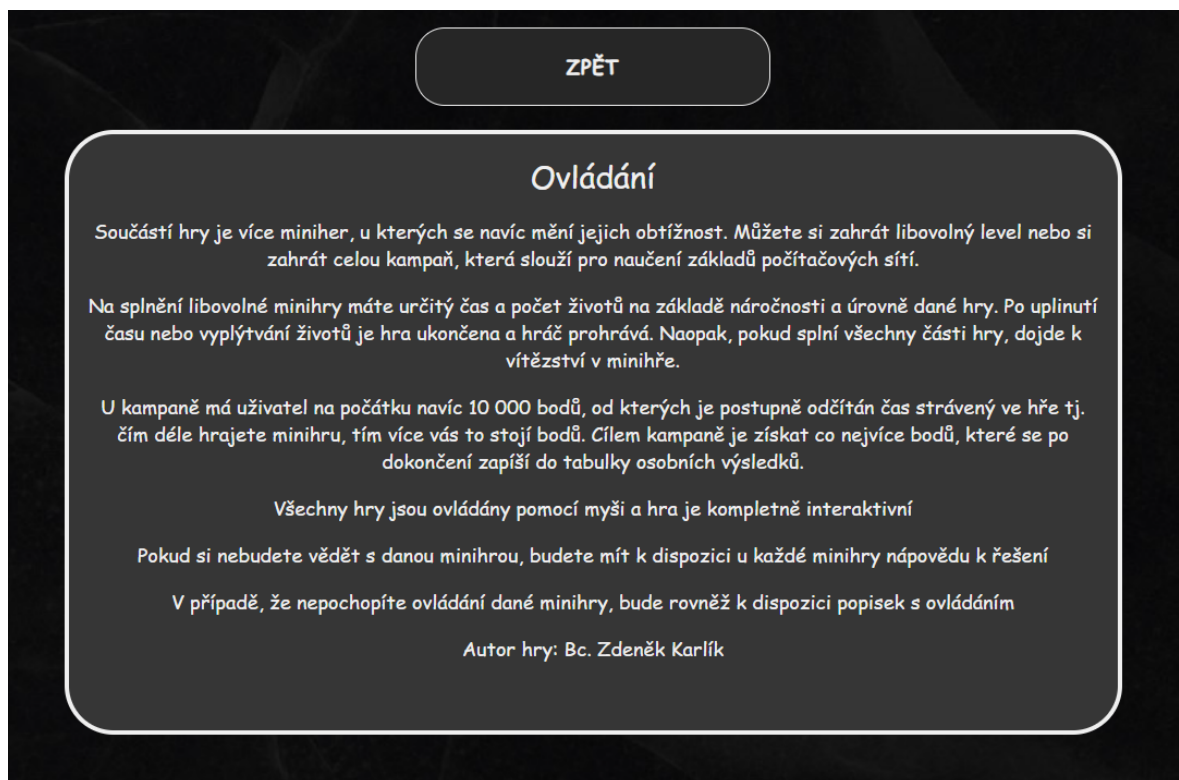
## 7.4 Tabulka a obecné ovládání

Pro zobrazení svých osobních výsledků je vytvořeno v hlavním menu tlačítko tabulka. Uvnitř této tabulky lze nalézt data z dokončených kampaní. Mezi tato data patří datum dokončení hry, skóre a ID (viz Obrázek 54). Data lze poté řadit dle všech sloupců.

Číslo pokusu	Datum	Skóre ↓
12	18. 3. 2024	9980
9	18. 3. 2024	9952
1	29. 2. 2024	9942
16	18. 3. 2024	9940
10	18. 3. 2024	9912
6	17. 3. 2024	9904
4	1. 3. 2024	9892
2	29. 2. 2024	9882
3	1. 3. 2024	9881
15	18. 3. 2024	9851

Obrázek 54 Detail tabulky osobních výsledků

Posledním prvkem GUI je obecné ovládání. K jeho zobrazení slouží poslední tlačítko v hlavním menu a jsou zde popsány základní principy ovládání webové aplikace (viz Obrázek 55).



ZPĚT

### Ovládání

Součástí hry je více miniher, u kterých se navíc mění jejich obtížnost. Můžete si zahrát libovolný level nebo si zahrát celou kampaň, která slouží pro naučení základů počítačových sítí.

Na splnění libovolné minihry máte určitý čas a počet životů na základě náročnosti a úrovně dané hry. Po uplynutí času nebo vyplývání životů je hra ukončena a hráč prohrává. Naopak, pokud splní všechny části hry, dojde k vítězství v minihře.

U kampaně má uživatel na počátku navíc 10 000 bodů, od kterých je postupně odčítán čas strávený ve hře tj. čím déle hrajete minihru, tím více vás to stojí bodů. Cílem kampaně je získat co nejvíce bodů, které se po dokončení zapíší do tabulky osobních výsledků.

Všechny hry jsou ovládány pomocí myši a hra je kompletně interaktivní

Pokud si nebudete vědět s danou minihrou, budete mít k dispozici u každé minihry nápovědu k řešení

V případě, že nepochopíte ovládání dané minihry, bude rovněž k dispozici popisek s ovládáním

Autor hry: Bc. Zdeněk Karlík

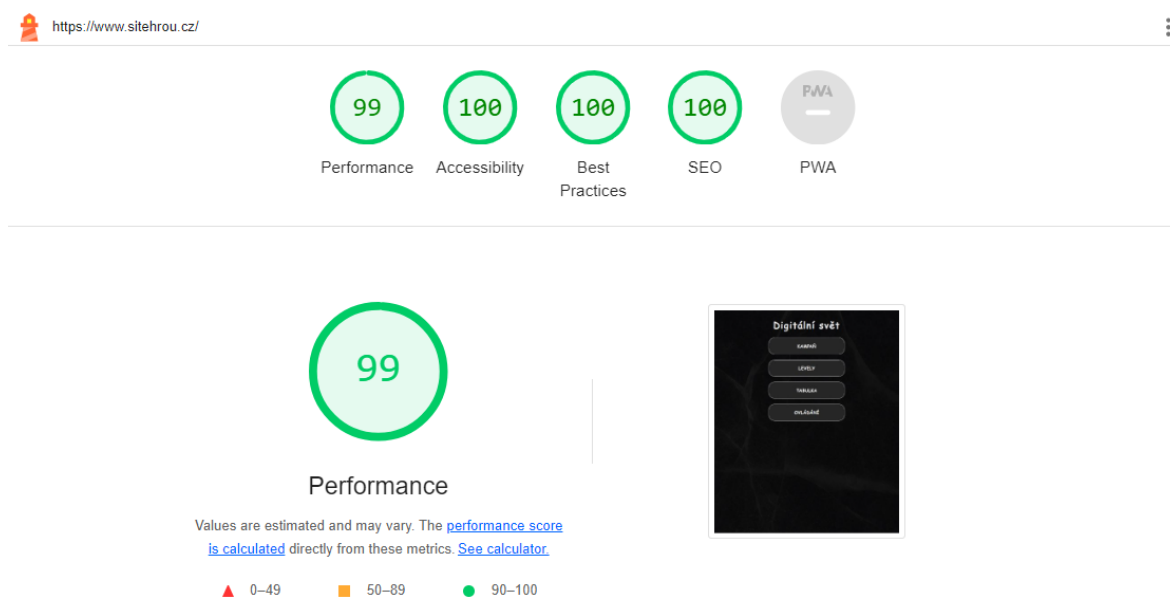
Obrázek 55 Základní ovládání hry

## 8 OPTIMALIZACE APLIKACE

Po naprogramování aplikace došlo k její optimalizaci tak, aby byla co nejvíce výkonná. Jako první došlo k optimalizaci velikosti obrázků, které byly až příliš velké. Pro tento účel byla využita webová aplikace TinyPNG, která umožňuje redukovat velikost obrázků se zachováním kvality. Zmenšení velikosti obrázků vedlo k daleko lepším číslům u nástroje Google Lighthouse, což je nástroj pro testování optimalizace webových stránek.

Následně došlo na webu k doplnění meta tagů a úpravě některých CSS komponent, které byly buď moc velké, nebo naopak malé.

Jakmile byly provedeny změny, došlo k opětovnému provedení testu optimalizace webové stránky. Výsledky z této optimalizace již dosahovaly téměř dokonalých hodnot (viz Obrázek 56).



Obrázek 56 Test optimalizace z nástroje Google Lighthouse

Poslední fází bylo nasazení beta verze hry do produkce s cílem získání dat od jednotlivých hráčů prostřednictvím formuláře.

## 9 SESTAVENÍ A VYHODNOCENÍ ANKETY

Jakmile byla aplikace nasazena na produkci, došlo k sestavení anonymní ankety, která byla nezbytná pro získání odpovědí na jednotlivé dílčí cíle. Dotazník byl primárně distribuován mezi studenty střední školy, ale díky propagaci na sociálních sítích se do ankety zapojili i jedinci z jiných sektorů. Anketa byla vytvořena prostřednictvím Microsoft Forms, což je jedna z aplikací, která je součástí balíčku Microsoft Office 365. Celkově bylo vytvořeno 17 otázek, rozdělených do tří základních sekcí, a to osobní informace, počítačové sítě a hraní her.

### 9.1 Nadefinování cílů ankety

Hlavním cílem ankety bylo zjištění, zda by respondenti měli zájem o vzdělávací hru tohoto formátu. Kromě hlavního cíle ankety byly nadefinovány ale také cíle dílčí, které jsou shrnuty v Tabulce 23.

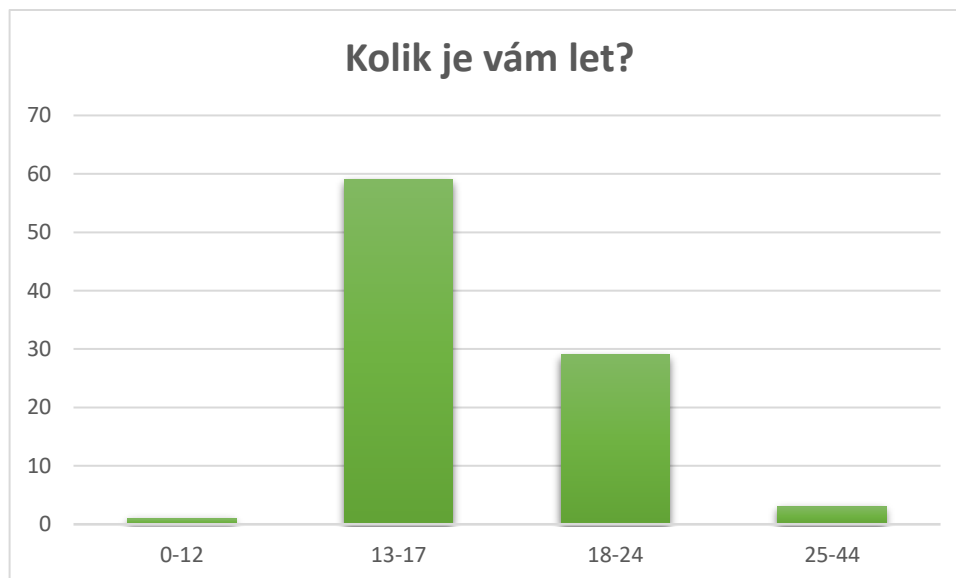
Tabulka 23 Popis dílčích cílů

Označení dílčího cíle	Popis dílčího cíle
Dílčí cíl 1	Zjištění, zda respondenti vnímají počítačové sítě a jejich výuku za důležité.
Dílčí cíl 2	Získání dat o osobních zkušenostech respondentů v oblasti počítačových sítí.
Dílčí cíl 3	Určení četnosti a zájmu o hraní her u respondentů.
Dílčí cíl 4	Získání zpětné vazby ohledně hratelnosti hry.
Dílčí cíl 5	Zhodnocení zájmu respondentů o začlenění her do výuky.
Dílčí cíl 6	Zjištění, zda by hra mohla přivést nové zájemce do oblasti počítačových sítí.

### 9.2 Osobní dotazy

Sekce s osobními dotazy byla zaměřená na věk respondentů, jejich pohlaví a pozici ve školství. Tyto informace napomáhají získat preference určitých skupin, což by mohlo být zásadní pro budoucí rozvoj aplikace.

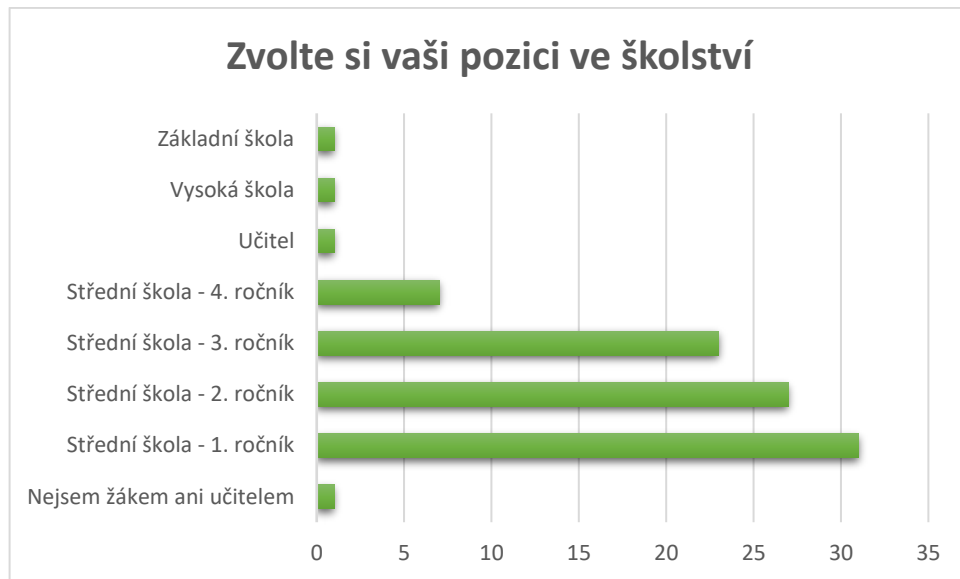
V rámci sekce s osobními dotazy byla stanovena anketní otázka „Kolik je vám let?“, kde respondenti měli možnost volby ze šesti odpovědí: „0-12, 13-17, 18-24, 25-44, 45-64, 65 a více“. Z Grafu 1 vyplývá, že nejvíce odpovídajících spadá do věkové kategorie 13-17 let, konkrétně 59 respondentů. Druhou nejpočetnější skupinou jsou odpovídající ve věku 18-24 let, zastoupení této skupiny činilo 29 respondentů (viz Graf 1).



Graf 1 Anketa – věk respondentů

Dalším kritériem pro kategorizaci respondentů byla jejich pozice ve školství. Anketa obsahovala otázku „Zvolte si vaši pozici ve školství.“, u které mohli respondenti vybírat z osmi odpovědí: „Základní škola, střední škola – 1. ročník, střední škola – 2. ročník, střední škola – 3. ročník, střední škola 4. ročník, vysoká škola, učitel a nejsem žákem ani učitelem“. Údaje zobrazené v Grafu 2 zachycují, že nejpočetnější skupinou respondentů byli žáci 1. ročníku SŠ. Celkově se ankety zúčastnilo 88 žáků SŠ.





Graf 2 Anketa – pozice ve školství

Poslední kritérium kategorizace se týkalo pohlaví respondentů. Na otázku „*Jaké je vaše pohlaví?*“ mohli odpovídající volit mezi odpověďmi: „*Žena, muž, jiné a nechci odpovídat*“. Dle Grafu 3 je zřejmé, že se ankety zúčastnili především muži, jejichž podíl odpovědí činil 87 %.

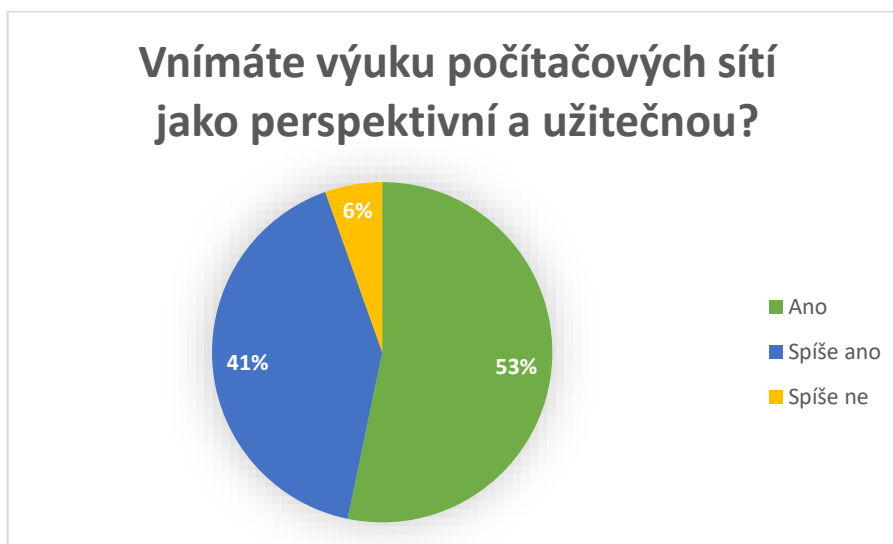


Graf 3 Anketa – pohlaví respondentů

### 9.3 Dotazy z oblasti počítačových sítí

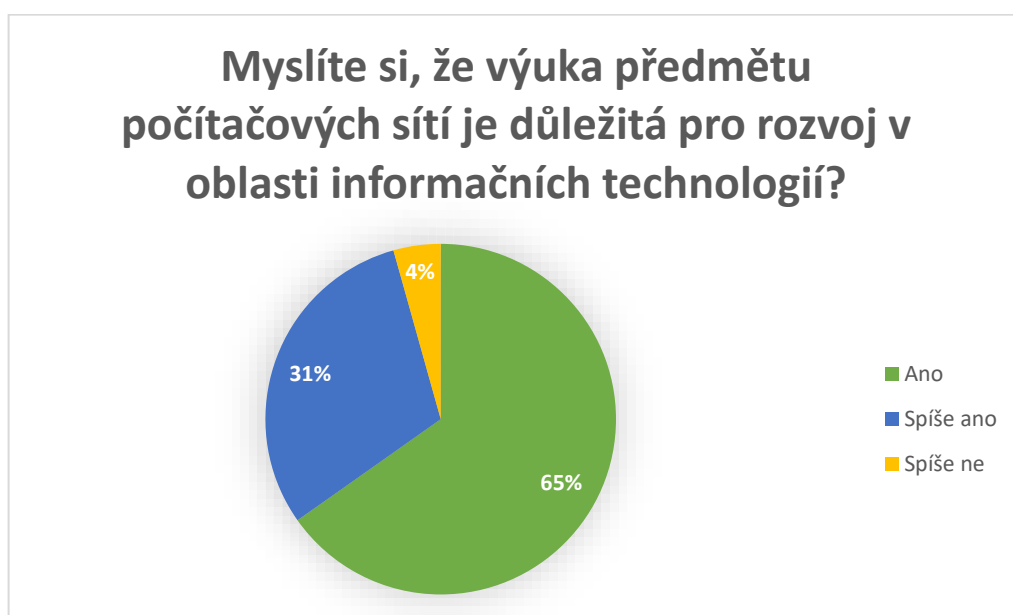
V rámci ankety byly položeny také otázky z oblasti počítačových sítí, které měly za úkol zjistit, jakým způsobem je tato sekce informačních technologií vnímána.

Pro získání odpovědi na dílčí cíl 1 byla respondentům v rámci sekce počítačových sítí položena anketní otázka „*Vnímáte výuku počítačových sítí jako perspektivní a užitečnou?*“, na kterou bylo možné odpovídat prostřednictvím čtyř odpovědí: „*Ano, spíše ano, spíše ne a ne*“. Více než polovina respondentů odpověděla „*ano*“ (viz Graf 4), dalších 41 % odpovídajících zvolilo možnost „*spíše ano*“ (viz Graf 4). Z dat je zřejmé, že více než 90 % respondentů vnímá výuku počítačových sítí jako užitečnou. Pouhých 6 % odpovídajících uvedlo, že počítačové sítě nevnímají jako příliš perspektivní (viz Graf 4). Poslední možností byla odpověď „*ne*“, kterou nezvolil ani jeden odpovídající.



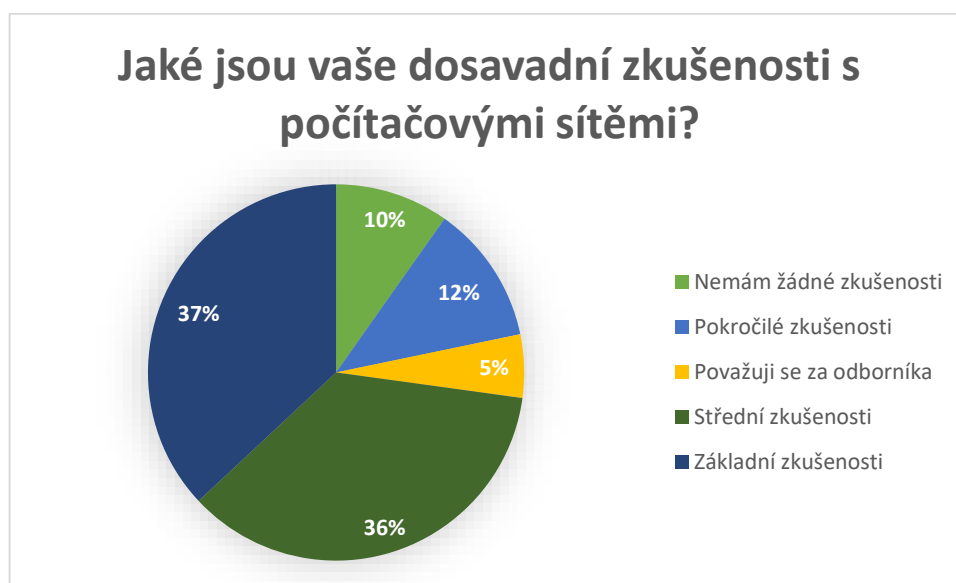
Graf 4 Anketa – užitečnost počítačových sítí

Dále byla k ověření dílčího cíle 1 stanovena anketní otázka „*Myslíte si, že výuka předmětu počítačových sítí je důležitá pro rozvoj v oblasti informačních technologií?*“, kde respondenti mohli vybírat z odpovědí: „*Ano, spíše ano, spíše ne a ne*“. Z Grafu 5 je zřejmé, že účastníci ankety považují výuku počítačových sítí za důležitou, vzhledem k tomu, že pouhé 4 % odpovídajících ji vnímají jako méně podstatnou. V případě této ankety opět nedošlo ani jednou ke zvolení odpovědi ne. Na základě Grafu 4 a 5 lze konstatovat, že vybraný vzorek respondentů vnímá počítačové sítě a jejich výuku jako užitečnou, což je odpověď na dílčí cíl číslo 1.



Graf 5 Anketa – důležitost počítačových sítí

Aby mohla být získána data pro dílčí cíl 2, který se zaměřuje na zkušenosti respondentů v oblasti počítačových sítí, byla položena otázka „*Jaké jsou vaše dosavadní zkušenosti s počítačovými sítěmi?*“ jejíž součástí bylo pět odpovědí: „*Nemám žádné zkušenosti, základní zkušenosti, střední zkušenosti, pokročilé zkušenosti a považuji se za odborníka*“. V rámci ankety byla odpověď „*střední zkušenosti*“ definována jako úroveň, která se nachází mezi odpověďmi „*základní zkušenosti*“ a „*pokročilé zkušenosti*“. Nejvíce odpovědí bylo uvedeno u možností „*základní zkušenosti*“ a „*střední zkušenosti*“, jejichž podíl činil dohromady 73 %, což by odpovídalo úrovni výuky počítačových sítí na střední škole. Naopak 10 % respondentů uvedlo, že nemají žádné zkušenosti, což může být způsobeno tím, že se ankety zúčastnily i osoby mimo obor informačních technologií. Naopak někteří respondenti uvedli, že se považují za pokročilé, či dokonce odborníky v oblasti počítačových sítí. Jednotlivé procentuální zastoupení všech kategorií lze vidět v Grafu 6 z jehož dat lze konstatovat, že respondenti mají různorodé zkušenosti.



Graf 6 Anketa – zkušenosti s počítačovými sítěmi

#### 9.4 Dotazy týkající se hraní her

Třetí část ankety se již týkala samotného hraní her. Tato sekce se zaměřuje na hlavní cíl ankety a také na dílčí cíle číslo 3, 4, 5 a 6, které slouží pro určení zájmu o hraní her u respondentů, získání zpětné vazby ohledně hratelnosti hry, možnost začlenění hry do výuky a zjištění, zda by hra mohla přivést nové zájemce do oblasti počítačových sítí.

Pro získání odpovědí na dílčí cíl 3 byla respondentům položena anketní otázka „*Jak často hrajete videohry?*“, u které byly stanoveny odpovědi: „*Každý den, několikrát za týden,*

*jednou týdně, příležitostně a videohry nehrají téměř vůbec*“. Z odpovědí uvedených v anketě bylo zjištěno, že 79 % respondentů hraje hry pravidelně během týdne. 36 odpovídajících dokonce uvedlo odpověď „každý den“, což odpovídalo 39 % ze všech odpovědí. Z toho lze usuzovat, že hraní her je u dotazovaného vzorku respondentů velmi populární. Naopak pouhých 7 % odpovídajících uvedlo, že videohry nehrají téměř vůbec (viz Graf 7).



Graf 7 Anketa – interval hraní her

Anketní otázka číslo 8 „*Máte předchozí zkušenosti s podobnými vzdělávacími hrami?*“ sloužila k získání dat o zkušenostech se vzdělávacími hrami u respondentů. U otázky respondenti volili ze tří odpovědí: „*Ano, ano ale minimální a ne*“. Z Grafu 8 je zřejmé, že celkově 83 % odpovídajících má alespoň nějaké zkušenosti s hrami zaměřenými na vzdělávání. 17 % respondentů uvedlo, že se s hrami pro podporu vzdělávání nemají žádné zkušenosti.

Na základě dat získaných z Grafu 7 a 8 lze konstatovat, že většina respondentů hraje hry pravidelně a v minulosti se již setkali s hrami pro podporu vzdělávání. Toto tvrzení slouží k odpovědi na dílčí cíl číslo 3.



Graf 8 Anketa – zkušenosti s podobnými hrami

Otázky číslo 9, 10 a 11, které jsou součástí ankety, slouží především k získání zpětné vazby od uživatelů, kteří si hru vyzkoušeli. Respondenti byly dotázáni na otázku „Byla hra intuitivní a snadno ovladatelná?“, u které vybírali ze čtyř odpovědí: „Ano, spíše ano, spíše ne a ne“. Celkem 44 % respondentů zvolilo odpověď „ano“. Dalších 41 % odpovídajících zvolilo možnost „spíše ano“. Celkově tedy 85 % odpovědí bylo pozitivních, což bylo považováno za přijatelné. Celkové výsledky ankety jsou zobrazeny v Grafu 9.



Graf 9 Anketa – intuitivnost hry

Kromě ovládní byla hra také hodnocena na základě její obtížnosti. Pro získání dat o obtížnosti hry byla vytvořena anketní otázka: „Jak hodnotíte obtížnost hry?“, jejíž součástí bylo pět možných odpovědí: „Velmi obtížná, obtížná, střední, snadná a velmi snadná“.

V rámci ankety byla odpověď „střední“ specifikována jako úroveň umístěna mezi kategoriemi „obtížná“ a „snadná“. Zde bylo dosaženo velmi dobrých výsledků, vzhledem k tomu, že 64 % účastníků zvolilo odpověď „střední“. Na základě těchto odpovědí je patrné, že obtížnost byla vhodně implementována a hra nebyla moc jednoduchá ani příliš složitá. Dalších 13 % dotazovaných uvedlo, že hra byla snadná a 22 % respondentů zvolilo možnost „obtížná“. Součástí ankety byla také možnost „velmi snadná“, kterou nezvolil žádný účastník ankety. Tyto výsledky jsou shrnuty v Grafu 10.



Graf 10 Anketa – obtížnost hry

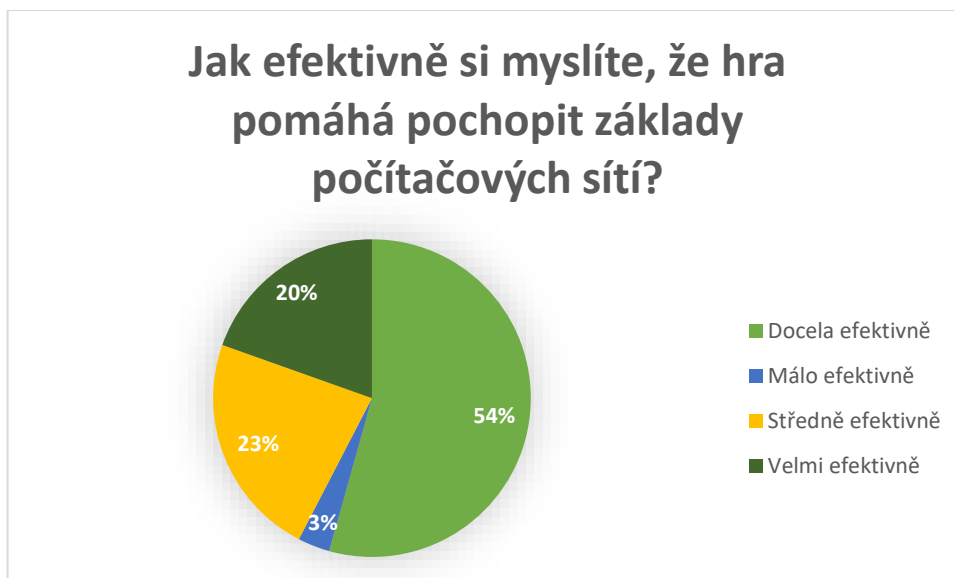
Třetí důležitá část hodnocení webové hry pro výuku počítačových sítí byla zábavnost. Pro tuto oblast byla stanovena anketní otázka „*Jak hodnotíte celkovou zábavnost hry?*“, u které měli respondenti možnost volby z pěti odpovědí: „*Velmi zábavná, zábavná, průměrná, nudná a velmi nudná*“. Z Grafu 11 lze vyčíst, že 45 % účastníků ankety shledává hru jako zábavnou a 16 % dokonce jako velmi zábavnou. V této oblasti by hra potřebovala určitá vylepšení, vzhledem k tomu, že 31 % respondentů hodnotilo hru jako průměrnou a 8 % jako nudnou. Tato skutečnost ovšem může být ovlivněna i tím, že hra byla zaměřena především na kvalitní výuku a díky tomu nelze dosáhnout ideálních čísel v oblasti zábavnosti. V anketě byla uvedena i možnost „*velmi nudná*“, kterou nezvolil žádný dotazovaný.

Na základě získaných dat z Grafu 9, 10 a 11 lze konstatovat, že hru považuje většina respondentů za intuitivní, zábavnou a středně obtížnou.



Graf 11 Anketa – zábavnost hry

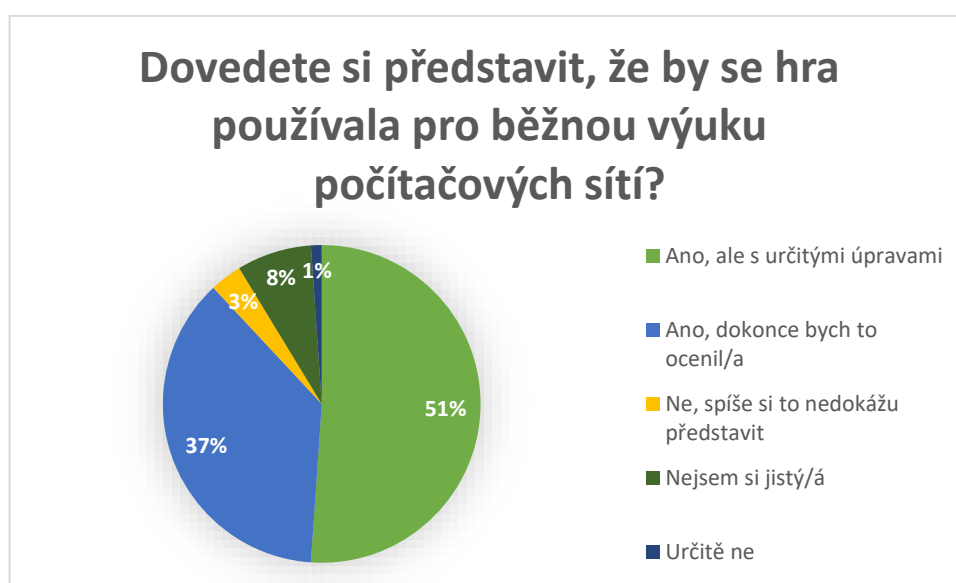
Pro získání dat ohledně efektivity hry byla v rámci ankety stanovena otázka „*Jak efektivně si myslíte, že hra pomáhá pochopit základy počítačových sítí?*“, kde respondenti volili z pěti odpovědí: „*Velmi efektivně, docela efektivně, středně efektivně, málo efektivně a neefektivně*“. Odpověď „*středně efektivně*“ byla definována jako úroveň nacházející se mezi odpověďmi „*docela efektivně*“ a „*málo efektivně*“. Na základě odpovědí respondentů, kde 20 % uvedlo „*velmi efektivně*“ a dalších 54 % „*docela efektivně*“ lze usoudit, že by hra při výuce mohla pomoci pochopit některá témata (viz Graf 12). Odpověď „*neefektivně*“ neuvedl žádný účastník dotazníku.



Graf 12 Anketa – efektivita učení počítačových sítí

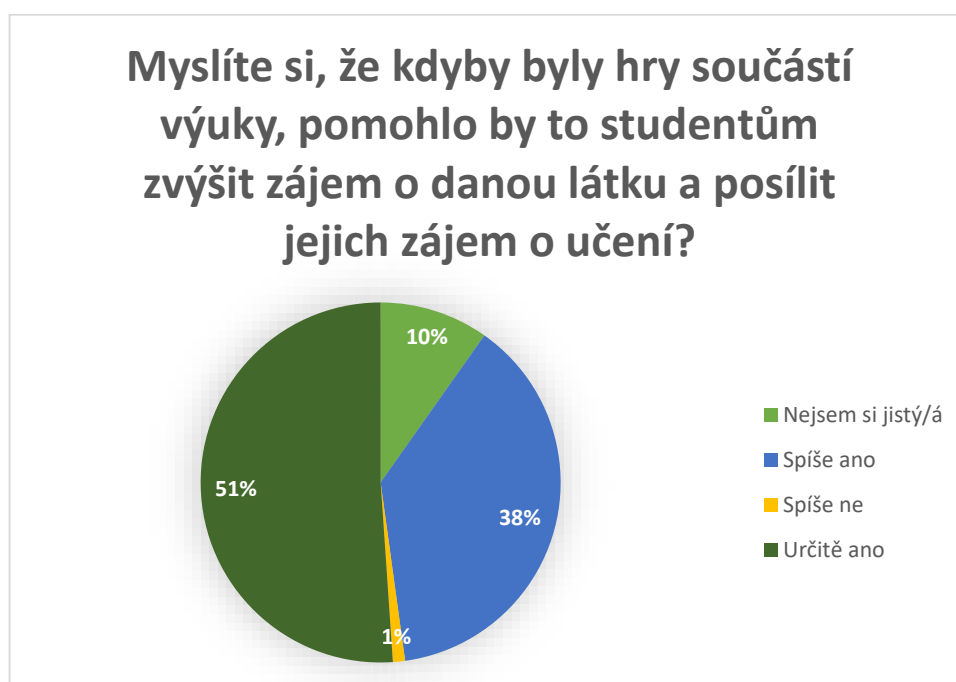


Pro dílčí cíl číslo 5 byla stanovena anketní otázka „Dovedete si představit, že by se hra používala pro běžnou výuku počítačových sítí?“, u které měli účastníci ankety k dispozici 5 odpovědí: „Ano, dokonce bych to ocenil/a, ano, ale s určitými úpravami, nejsem si jistý/á, ne, spíše si to nedokážu představit a určitě ne“. Dílčí cíl číslo 5 se zabývá začleněním hry do běžné výuky počítačových sítí. Zde došlo k velmi pozitivním odpovědím ze strany respondentů, z nichž 37 % uvedlo, že by hru dokonce ocenili. Dalších 51 % odpovídajících také souhlasilo s tím, že by ve hře došlo k určitým úpravám. Tyto dvě hodnoty dávají dohromady 88 %, díky čemuž lze konstatovat, že účastníci ankety by měli zájem o začlenění hry do běžné výuky předmětu počítačových sítí. Veškeré odpovědi jsou součástí Grafu 13.



Graf 13 Anketa – využití hry při výuce

Anketní otázka číslo 14 „Myslíte si, že kdyby byly hry součástí výuky, pomohlo by to studentům zvýšit zájem o danou látku a posílit jejich zájem o učení?“ poskytuje také data pro dílčí cíl číslo 5. Pro otázku číslo 14 bylo specifikováno 5 odpovědí: „*Určitě ano, spíše ano, nejsem si jistý/á, spíše ne a určitě ne*“. V případě odpovídajících uvedlo 51 % možnost „*určitě ano*“. Dalších 38 % bylo pro odpověď „*spíše ano*“ a 10 % respondentů si nebylo jistých. Součástí ankety byla také možnost „*určitě ne*“, kterou nezvolil žádný účastník. Na základě těchto dat, které jsou uvedeny v Grafu 14, lze pozorovat, že 89 % účastníků ankety by mělo o výuku vyšší zájem, kdyby její součástí byly hry.



Graf 14 Anketa – vliv her na zájem o vzdělávání

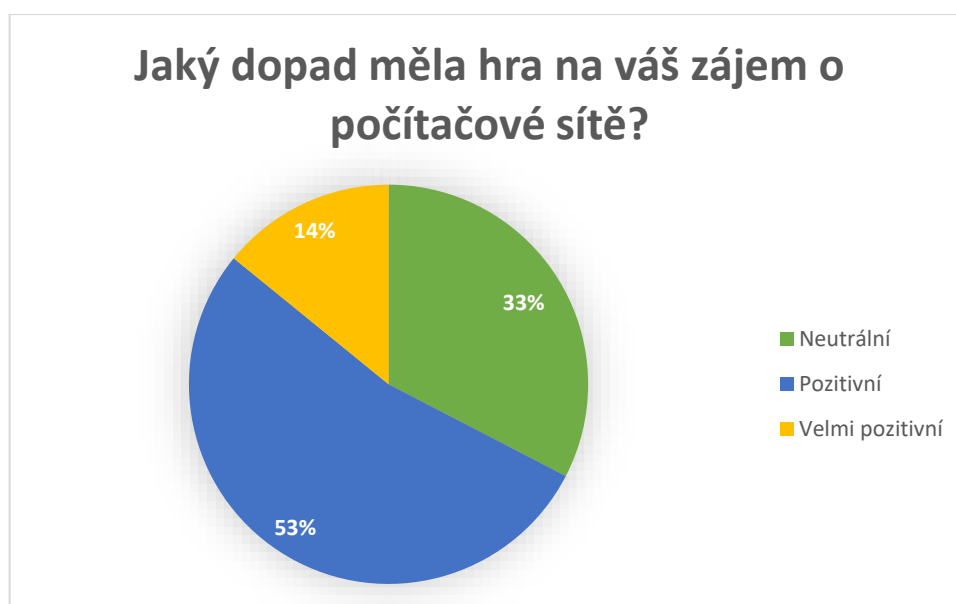
Lidé mají často problémy pochopit určitá témata. Kdyby ale témata byla vysvětlována formou her, mohlo by to mít pozitivní vliv na pochopení dané problematiky. Z tohoto důvodu byla stanovena otázka „Bylo by pro vás snazší pochopit některá témata, kdyby byla vyučována formou hry (nejen v počítačových sítích)?“, u které bylo možné zvolit z pěti odpovědí: „Určitě ano, spíše ano, nejsem si jistý/á, spíše ne a určitě ne“. Ze všech respondentů, z nichž většina byla žáků střední školy, 85 % uvedlo, že by uvítalo výuku formou her. Celkem 14 % respondentů si nebylo jistých a pouze jedna odpověď byla „spíše ne“. Z těchto informací, které jsou uvedeny v Grafu 15, je zjevné, že by zakomponování her do výuky mohlo pomoci při pochopení některých témat.

Na základě dat získaných z Grafů 13, 14 a 15 lze říci, že by zvolený vzorek respondentů ocenil využití her při výuce počítačových sítí, ale i ostatních témat.



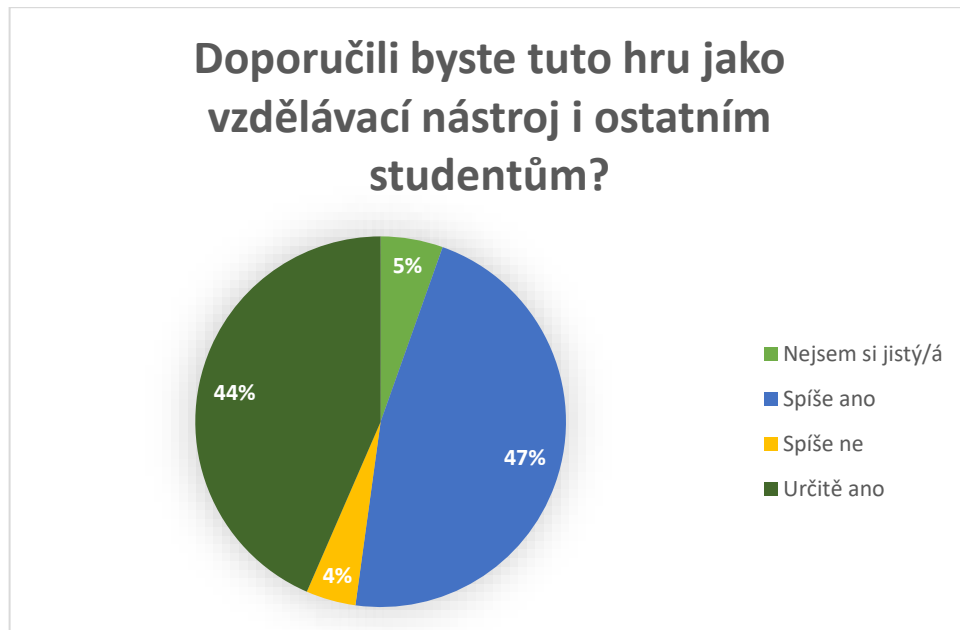
Graf 15 Anketa – výuka formou her

Pro zjištění, zda hra zvyšuje zájem o počítačové sítě, byla vytvořena anketní otázka „*Jaký dopad měla hra na váš zájem o počítačové sítě?*“, v rámci které měli dotazovaní k dispozici volbu z pěti odpovědí: „*Velmi pozitivní, pozitivní, neutrální, negativní a velmi negativní*“. Dle Grafu 16 je zřejmé, že žádný odpovídající ne zvolil odpověď „*negativní*“ a „*velmi negativní*“. Odpověď „*neutrální*“ poté zvolilo 33 % dotazovaných. Na druhou stranu hra měla pozitivní vliv na zvýšení zájmu o počítačové sítě u 67 % uchazečů ankety. Možnost „*velmi pozitivní*“ zvolilo 14 % respondentů. Na základě dat v Grafu 16 se tedy dá prohlásit, že by hra mohla přivést nové zájemce do oblasti počítačových sítí, což je odpověď na dílčí cíl číslo 6.



Graf 16 Anketa – změna zájmu o počítačové sítě po hraní hry

Poslední otázka ankety byla zaměřená na doporučení hry ostatním studentům. Anketní otázka „Doporučil byste tuto hru jako vzdělávací nástroj i ostatním studentům?“ měla k dispozici 5 odpovědí: „Určitě ano, spíše ano, nejsem si jistý/á, spíše ne a určitě ne“. Odpovědi ankety, které jsou zobrazeny v Grafu 17, byly velmi pozitivní. Ze všech odpovídajících uvedlo 44 % možnost „určitě ano“. Dalších 47 % respondentů uvedlo, že by hru spíše doporučili. Dohromady by tedy hru mohlo doporučit 91 % dotazovaných.



Graf 17 Anketa – doporučení hry

## 9.5 Vyhodnocení ankety a naplnění cílů

Během ankety bylo zjištěno, že by vybraný vzorek respondentů měl zájem o hru vyvíjenou součástí této diplomové práce, což byl zároveň i hlavní cíl průzkumu. Dílčí cíle a jejich vyhodnocení jsou poté shrnuty v Tabulce 24.

Tabulka 24 Naplnění dílčích cílů

Označení dílčího cíle	Naplnění cíle a jeho vyhodnocení
Dílčí cíl 1	Na základě Grafu 4 a 5 lze konstatovat, že respondenti vnímají počítačové sítě a jejich výuku za důležité.
Dílčí cíl 2	Data v Grafu 6 dokazují, že účastníci ankety mají různorodé zkušenosti s počítačovými sítěmi.
Dílčí cíl 3	Díky Grafům 7 a 8, bylo zjištěno, že respondenti hrají hry pravidelně a většina z nich se již v minulosti setkala se vzdělávacími hry.
Dílčí cíl 4	Dle Grafů 9, 10 a 11 lze učinit závěr, že většina dotazovaných osob považuje hru za intuitivní, zábavnou a středně obtížnou.
Dílčí cíl 5	Data z Grafů 13, 14 a 15 naznačují, že zvolený vzorek respondentů by při výuce ocenil využití her.
Dílčí cíl 6	Graf 16 dokazuje, že by hra u svých hráčů mohla zvýšit zájem o oblast počítačových sítí.

## 10 VYPRACOVÁNÍ ZÁSAD

Součástí diplomové práce byla realizace šesti zásad. První zásadou diplomové práce bylo vypracování stručné literární rešerše na téma vývoj počítačových her a použitých technologií. Tato část práce byla reflektována v kapitolách 2 a 3. Druhá zásada se zaměřovala na navržení vhodných úloh s přihlédnutím ke kurikulu SŠ. V kapitole 4 byly formulovány vhodné úlohy, které odpovídaly obsahu učebních osnov a tématům probíraných na SŠ. Třetí zásadou bylo navrhnout webovou aplikaci pro výuku základů počítačových sítí. Tato část diplomové práce byla rozpracována v rámci kapitoly 5, kde došlo ke specifikaci požadavků na aplikaci a navržení hlavních scénářů a funkcionalit. Čtvrtá zásada se zabývala naprogramováním navržené webové aplikace. V kapitole 6 byl popsán vývoj hlavních částí aplikace. Pátá zásada spočívala ve vytvoření adekvátního popisu výsledné aplikace. Tento popis byl proveden v kapitole 7 a obsahoval ukázky hlavních částí GUI. Po naprogramování hlavních funkcionalit a vytvoření adekvátního popisu došlo k nasazení aplikace na produkci. Následně byla vypracována poslední zásada, která se zaměřovala na sestavení a vyhodnocení ankety na vybraném vzorku studentů. Tato zásada byla popsána v rámci kapitoly 8.

## 11 BUDOUCNOST APLIKACE

Aplikace byla nasazena na produkci a byl proveden základní průzkum, během kterého bylo zjištěno, že by o webovou hru tohoto typu byl zájem.

Na základě odpovědí získaných z ankety by se aplikace měla začít zaměřovat více na prvky zábavnosti, čehož by mohlo být dosaženo prostřednictvím vytvoření vlastního účtu, jehož součástí by byla postava, která by si mohla postupně vylepšovat své schopnosti s pomocí herní měny získané během plnění úrovní. Také by se do hry mohly přidat ocenění.

Důležitým prvkem by mohla být také implementace dalších miniher tak, aby hry byly co nejvíce různorodé a neodrazovaly hráče přílišným opakováním.

Pro rozvoj aplikace by bylo také velmi podstatné získání nových potenciálních zájemců, což by mohlo být uskutečněno prostřednictvím oslovení jiných středních škol, jejichž součástí je obor informačních technologií s předmětem počítačových sítí.



## ZÁVĚR

Diplomová práce se zabývala podporou výuky základů počítačových sítí. Hlavním cílem práce bylo vytvoření interaktivní webové hry, která by začátečníkům a studentům pomohla s pochopením základních témat v oblasti počítačových sítí.

V úvodu teoretické části došlo k popisu hlavní motivace pro vytvoření webové hry a také k průzkumu aplikací zaměřených na problematiku počítačových sítí. Následně došlo k vytvoření první části literární rešerše, zaměřené na principy vývoje počítačových her. Druhá část literární rešerše byla zaměřena na technologie a postupy využívané v praktické části diplomové práce.

Na počátku praktické části došlo na základě kurikula SŠ k návržení vhodných úloh, které mají za cíl efektivně podporovat výuku základů počítačových sítí. Tyto úlohy byly navrženy tak, aby odpovídaly tematickému obsahu a cílům kurikula.

Po stanovení úloh na základě vzdělávacího plánu SŠ došlo v praktické části k návrhu interaktivní webové hry. Součástí tohoto návrhu byl sběr a analýza funkčních a nefunkčních požadavků na základě kterých byl poté vytvořen UC diagram složený z jednotlivých případů užití. Každému případu užití byl sepsán vlastní scénář. V závěrečné fázi návrhu aplikace došlo u hlavních částí webové hry k vytvoření wireframů.

Další část diplomové práce byla zaměřena na naprogramování a popis výsledné aplikace. V rámci této části došlo k popisu nejdůležitějších prvků zdrojového kódu webové hry. Také byl proveden popis GUI jednotlivých částí aplikace.

V rámci praktické části práce byla také sestavena anketa, které se zúčastnilo 92 respondentů. Z odpovědí na anketní otázky bylo zjištěno, že by respondenti měli zájem o hru pro podporu výuky počítačových sítí, která byla vyvíjena jako součást této diplomové práce. Toto zjištění bylo zároveň odpovědí na hlavní cíl ankety. Součástí ankety byly i dílčí cíle, mezi které patřilo zjištění, zda respondenti vnímají počítačové sítě a jejich výuku za důležité, získání dat o osobních zkušenostech respondentů v oblasti počítačových sítí a určení zájmu o hraní her u respondentů. Bylo zjištěno, že respondenti vnímají počítačové sítě jako důležité, mají různorodé zkušenosti v oblasti počítačových sítí a většina z nich hraje hry pravidelně. Další dílčí cíle byly již zaměřeny na vývoj hry. Mezi tyto cíle patřilo získání zpětné vazby o hrátelnosti vyvíjené hry a zhodnocení zájmu respondentů o začlenění her do výuky. Z odpovědí v anketě je zřejmé, že většina respondentů považuje hru za intuitivní a zábavnou. Zároveň by účastníci ankety ocenili, kdyby byly hry, včetně hry vyvíjené v rámci této

diplomové práce, součástí běžné výuky. Posledním dílčím cílem bylo zjištění, zda by hra mohla přivést nové zájemce do oblasti počítačových sítí. Na základě odpovědí z ankety byl stanoven závěr, že by hra mohla zvýšit zájem o oblast počítačových sítí.

Webová hra je funkční a byla uvedena do produkce. V rámci budoucího vývoje by mohlo dojít k implementaci dalších interaktivních miniher a především prvků, které by do hry přidaly více zábavnosti. Mezi tyto prvky by mohlo patřit oceňování hráčů, vylepšování postav a využívání herní měny.

## SEZNAM POUŽITÉ LITERATURY

- [1] *Cisco networking academy*. online. 2024. Dostupné z: <https://www.netacad.com/courses/packet-tracer>. [cit. 2024-02-13].
- [2] *Počítačové sítě*. online. In: Umíme informatiku. 2024. Dostupné z: <https://www.umimeinformatiku.cz/cviceni-pocitacove-site>. [cit. 2024-02-13].
- [3] *Computer Network Tutorials*. online. In: Google play. 2024. Dostupné z: [https://play.google.com/store/apps/details?id=cn.computernetworks.networks.networking.learn.topology.lan.wan&hl=en\\_US](https://play.google.com/store/apps/details?id=cn.computernetworks.networks.networking.learn.topology.lan.wan&hl=en_US). [cit. 2024-02-14].
- [4] QUWAIDER, Muhannad; ALABED, Abdullah a DUWAIRI, Rehab. The Impact of Video Games on the Players Behaviors: A Survey. *Procedia Computer Science*. 2019, roč. 10, č. 151, s. 1-8. ISSN 1877-0509.
- [5] *Video Game History*. online. In: History. 2017, October 17, 2022. Dostupné z: <https://www.history.com/topics/inventions/history-of-video-games>. [cit. 2024-02-17].
- [6] GREGORY, Jason. *Game engine architecture*. 3rd ed. Boca Raton: Taylor and Francis, CRC Press, 2018. ISBN 978-1138035454.
- [7] SIBONY, Joseph. *The Best Game Engines You Should Consider for 2023*. online. In: Incredibuild. 2023. Dostupné z: <https://www.incredibuild.com/blog/top-gaming-engines-you-should-consider>. [cit. 2024-02-17].
- [8] COSSU, Sebastiono M. *Game Development with GameMaker Studio 2: Make Your Own Games with GameMaker Language*. 1st ed. London, UK: Apress, 2019. ISBN 978-1484250099.
- [9] LAVIERI, Edvard. *Learning AWS Lumberyard Game Development*. 1st ed. Packt Publishing, 2016. ISBN 978-1786460868.
- [10] COLEMAN, Drew. *Beginner Game Developers, Say No to Game Engines*. online. In: Medium. 2019. Dostupné z: <https://drewcampbell92.medium.com/beginner-game-developers-say-no-to-game-engines-4667e0941241>. [cit. 2024-02-17].
- [11] SERRANO, Harold. *Can I develop games without a Game Engine?*. online. In: HaroldSerrano. 2018. Dostupné z: <https://www.haroldserrano.com/blog/canidevelopgameswithoutagameengine>. [cit. 2024-02-17].
- [12] JOHNS, Robert a SEMAH, Benjamin. *7 Best Programming Languages for Game Development in 2024*. online. In: Hackr. 2024. Dostupné z: <https://hackr.io/blog/best-programming-language-for-games>. [cit. 2024-02-17].
- [13] *Introduction to game development for the Web*. online. In: Mozilla. 2024. Dostupné z: <https://developer.mozilla.org/en-US/docs/Games/Introduction>. [cit. 2024-02-17].
- [14] NEPOR, Vladimír. *Advantages and disadvantages of web games*. online. In: Gamearter. 2019. Dostupné z: <https://www.gamearter.com/blog/advantages-disadvantages-of-web-games>. [cit. 2024-02-17].
- [15] STEFYN, Nadia. *How video games are made: the game development process*. online. In: Cgspectrum. 2022. Dostupné z: <https://www.cgspectrum.com/blog/game-development-process>. [cit. 2024-02-17].
- [16] PISKUNOV, Egor. *Your Guide On Game Development Process*. online. In: Ilogos. 2023. Dostupné z: <https://ilogos.biz/stages-of-game-development-your-guide-on-game-development-process/>. [cit. 2024-02-17].

- [17] *Top 7 Game Development Trends For 2024*. online. In: Geeksforgeeks. 2024. Dostupné z: <https://www.geeksforgeeks.org/top-game-development-trends/>. [cit. 2024-02-17].
- [18] FULLERTON, Tracy. *Game Design Workshop: A Playcentric Approach to Creating Innovative Games*. 4th ed. A K Peters/CRC Press, 2018. ISBN 978-1138098770.
- [19] ERHEL, Séverin a JAMET, Eric. The effects of goal-oriented instructions in digital game-based learning. *Interactive Learning Environments*. 2016, roč. 1, č. 24, s. 1744-1757. ISSN 1049-4820.
- [20] *9 Web Technologies Every Web Developer Must Know*. online. In: TMS. 2023. Dostupné z: <https://tms-outsourcing.com/blog/posts/web-technologies/>. [cit. 2024-02-17].
- [21] CASTRO, Elizabeth a HYSLOP, Bruce. *HTML5 a CSS3: názorný průvodce tvorbou WWW stránek*. 1. vyd. Brno: Computer Press, 2012. ISBN 9788025137338.
- [22] *HTML: HyperText Markup Language*. online. In: Mozilla. 2023. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/HTML>. [cit. 2024-02-17].
- [23] *CSS: Cascading Style Sheets*. online. In: Mozilla. 2024. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/CSS>. [cit. 2024-02-17].
- [24] TERRA, John. *Breaking Down CSS and CSS3 Differences*. online. In: Simplilearn. 2023. Dostupné z: <https://www.simplilearn.com/difference-between-css-and-css3-article>. [cit. 2024-02-17].
- [25] *What is JavaScript?*. online. In: Mozilla. 2024. Dostupné z: [https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript). [cit. 2024-02-17].
- [26] STUART, Graeme. *Introducing JavaScript Game Development: Build a 2D Game from the Ground Up*. 1st ed. Imprint: Apress, 2017. ISBN 978-1484232514.
- [27] *Object-Oriented Programming in JavaScript*. online. In: Freecodecamp. 2023. Dostupné z: <https://www.freecodecamp.org/news/object-oriented-programming-javascript/>. [cit. 2024-02-17].
- [28] UNHELKAR, Bhuvan. *Software engineering with UML*. 1st ed. Boca Raton: CRC Press, 2018. ISBN 978-1138297432.
- [29] SOMMERVILLE, Ian. *Software engineering*. 10th ed. Boston: Pearson, 2015. ISBN 978-0133943030.
- [30] HARTINGER, David. *Lekce 3 - UML - Use Case Specifikace*. online. In: Itnetwork. 2024. Dostupné z: <https://www.itnetwork.cz/navrh/uml/uml-use-case-specifikace-diagram>. [cit. 2024-02-17].
- [31] *What Is UI Design? Definition, Tips, Best Practices*. online. In: Coursera. 2023. Dostupné z: <https://www.coursera.org/articles/ui-design>. [cit. 2024-02-18].
- [32] KAŽOUSKOVÁ, Barbora. *UX a UI Design: Jak na uživatelské rozhraní webů a aplikací?*. online. In: Rascasone. 2023. Dostupné z: <https://www.rascasone.com/cs/blog/ux-design-ui-design>. [cit. 2024-02-18].
- [33] *User Interface (UI) Design*. online. In: Interaction Design Foundation. 2023. Dostupné z: <https://www.interaction-design.org/literature/topics/ui-design>. [cit. 2024-02-18].
- [34] CHEN, Jieshan; CHEN, Chunyang; XING, Zhenchang; XIA, Xin; ZHU, Liming et al. Wireframe-based UI Design Search through Image Autoencoder. *ACM Transactions on Software Engineering and Methodology*. 2020, roč. 29, č. 3, s. 1-31.

- [35] ZOLKIFLI, Nazatul; NGAH, Amir a DERAMAN, Aziz. Version Control System: A Review. *Procedia Computer Science*. 2018, roč. 135, č. 3, s. 408-415. ISSN 1877-0509.
- [36] BASUMALLICK, Chiradeep. *What Is Version Control? Meaning, Tools, and Advantages*. online. In: Spiceworks. 2022. Dostupné z: <https://www.spiceworks.com/tech/devops/articles/what-is-version-control/>. [cit. 2024-02-18].
- [37] *Top 15 Web Development IDEs & Career Tips*. online. In: Simplilearn. 2023. Dostupné z: <https://www.simplilearn.com/top-web-development-ides-and-career-tips-article>. [cit. 2024-02-18].
- [38] SHAH, Sahil. *Best ultimate IDE's for the developers in 2023*. online. In: Medium. 2023. Dostupné z: <https://medium.com/@amsahil/best-ultimate-ides-for-the-developers-in-2023-3a49b5e9603d>. [cit. 2024-02-18].
- [39] SCHROER, Alyssa. *Artificial Intelligence*. online. In: Builtin. 2024. Dostupné z: <https://builtin.com/artificial-intelligence>. [cit. 2024-02-18].

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

2D	Dvoudimenzionální
3D	Třídimenzionální
5G	Pátá generace mobilních sítí
AAA	Triple-A
API	Rozhraní pro programování aplikací
ARP	Adress Resolution Protocol
As-AnArt	Asymmetric Background Animations
C++	Programovací jazyk C++
C#	Programovací jazyk C#
CIDR	Classless Inter-Domain Routing
CMVC	Collaborative Management and Version Control
CSS	Kaskádové styly
CSS3	Kaskádové styly verze 3
DLC	Obsah ke stažení
DNS	Domain Name System
DOM	Document Object Model
ES6	ECMAScript 6 (verze JavaScriptu)
FR	Functional Requirement
FTP	File Transfer Protocol
GDD	Game Design Document
GLM	Graphics Library Mathematics
GPT	Generative Pre-trained Transformer
GTA	Grand Theft Auto
GUI	Grafické uživatelské rozhraní
HSL	Barevný model Hue, Saturation, Lightness

---

HSLA	Barevný model Hue, Saturation, Lightness, Alpha
HTTP	Hypertext Transfer Protocol
HTML	HyperText Markup Language
HTML5	HyperText Markup Language verze 5
ID	Identifikátor
IDE	Integrated Development Environment
IMAP	Internet Message Access Protocol
IP	Internet Protocol
ISO	International Organization for Standardization
JS	JavaScript
JVM	Java Virtual Machine
K1	Kategorie 1
K2	Kategorie 2
K3	Kategorie 3
K4	Kategorie 4
K5	Kategorie 5
MAC	Media Access Control
NFR	Non-Functional Requirement
OOP	Objektově orientované programování
OSI	Open Systems Interconnection
OXO	Hra Piškvorky
PNG	Portable Network Graphics
POP3	Post Office Protocol verze 3
RGB	Barevný model Red, Green, Blue
RGBA	Barevný model Red, Green, Blue, Alpha
RJ	Registered Jack

---

RPG	Role-playing game
SMTP	Simple Mail Transfer Protocol
SRC	Source code
SŠ	Střední škola
TCP	Transmission Control Protocol
UML	Unified Modeling Language
UC	Use Case
UI	Uživatelské rozhraní
UX	Uživatelská zkušenost
VCS	Version Control System
VLSM	Variable Length Subnet Masking
VR	Virtuální realita
WebGL	Web Graphics Library
WebRTC	Web Real-Time Communication



**SEZNAM OBRÁZKŮ**

Obrázek 1 Časové úseky strávené nad diplomovou prací .....	12
Obrázek 2 Síť ze simulátoru Cisco Packet Tracer .....	13
Obrázek 3 Snímek obrazovky testu na odkaze umimeinformatiku.cz.....	13
Obrázek 4 Snímek obrazovky aplikace Computer Network Tutorials .....	14
Obrázek 5 Editor herního engine Unity.....	16
Obrázek 6 GTA V – AAA hra [15] .....	21
Obrázek 7 Základní struktura HTML kódu .....	24
Obrázek 8 Jednoduchý UC model .....	30
Obrázek 9 Jednoduchý wireframe .....	32
Obrázek 10 Část školního vzdělávacího plánu použitá pro návrh.....	37
Obrázek 11 Diagram případů užití.....	47
Obrázek 12 Wireframe – úvodní stránka.....	67
Obrázek 13 Wireframe – kampaň.....	68
Obrázek 14 Wireframe – rozdělení úrovní .....	68
Obrázek 15 Wireframe – tabulka.....	69
Obrázek 16 Wireframe – ovládání.....	69
Obrázek 17 Wireframe – Guess the Word.....	70
Obrázek 18 Wireframe – Picture Match .....	71
Obrázek 19 Wireframe – Guess the Picture.....	71
Obrázek 20 Wireframe – Put the Words.....	72
Obrázek 21 Wireframe – Count the Number.....	73
Obrázek 22 Wireframe – dialog pro nápovědu.....	74
Obrázek 23 HTML kód hlavního menu.....	75
Obrázek 24 JS Funkce lightenScreen .....	76
Obrázek 25 JS Funkce refreshScore .....	76
Obrázek 26 JS funkce endCampaign .....	77
Obrázek 27 Konstruktor třídy Game .....	79
Obrázek 28 Metoda pro generaci životů.....	80
Obrázek 29 Zdrojový kód udávající barvu skóre.....	81
Obrázek 30 Metoda pro vygenerování pole s náhodnými prvky.....	82
Obrázek 31 Metoda pro vytvoření elementu ve třídě DragableGame .....	83
Obrázek 32 Metoda dragStart .....	83
Obrázek 33 Metoda dragEnter .....	84
Obrázek 34 Objekt s daty pro hru Guess the word.....	85

Obrázek 35 Vytvoření instance třídy GuessTheWord.....	86
Obrázek 36 Objekt s daty pro hru Guess the picture .....	86
Obrázek 37 Nastavení naslouchačů pro elementy v metodě generatePicture .....	87
Obrázek 38 Vytvoření instance třídy GuessThePicture.....	87
Obrázek 39 Objekt s daty do hry Picture match .....	88
Obrázek 40 Nastavení naslouchačů pro hry využívající systém drag and drop .....	89
Obrázek 41 Data pro dialogová okna .....	90
Obrázek 42 Hlavní menu hry.....	91
Obrázek 43 Herní mód kampaň .....	92
Obrázek 44 Herní mód levely .....	93
Obrázek 45 Menu pro spuštění konkrétní hry .....	93
Obrázek 46 Hra Guess the word .....	94
Obrázek 47 Hra Guess the picture .....	95
Obrázek 48 Hra Picture match.....	96
Obrázek 49 Hra Put the words.....	97
Obrázek 50 Hra Count the number .....	98
Obrázek 51 Prohra ve hře .....	99
Obrázek 52 Výhra v kampani .....	99
Obrázek 53 Dialogové okno s nápovědou .....	100
Obrázek 54 Detail tabulky osobních výsledků .....	101
Obrázek 55 Základní ovládání hry.....	101
Obrázek 56 Test optimalizace z nástroje Google Lighthouse .....	102

**SEZNAM TABULEK**

Tabulka 1 Seznam úloh pro implementaci .....	39
Tabulka 2 Pokrytí funkčních požadavků .....	48
Tabulka 3 Pokrytí funkčních požadavků 2 .....	49
Tabulka 4 Scénář – zobrazení kampaně .....	50
Tabulka 5 Scénář – zobrazení úrovní .....	51
Tabulka 6 Scénář – zobrazení tabulky .....	52
Tabulka 7 Scénář – zobrazení obecného ovládání .....	53
Tabulka 8 Scénář – návrat do menu.....	54
Tabulka 9 Scénář – zobrazení hry.....	55
Tabulka 10 Scénář – zobrazení skóre .....	56
Tabulka 11 Scénář – zobrazení zadání .....	56
Tabulka 12 Scénář – zobrazení počtu zbývajících pokusů .....	57
Tabulka 13 Scénář – zobrazení ovládání .....	57
Tabulka 14 Scénář – zobrazení nápovědy .....	58
Tabulka 15 Scénář – zobrazení výhry.....	59
Tabulka 16 Scénář – zobrazení prohry .....	60
Tabulka 17 Scénář – návrat do módu .....	61
Tabulka 18 Scénář – zobrazení klávesnice .....	62
Tabulka 19 Scénář – náhodná generace slova .....	63
Tabulka 20 Scénář – zobrazení slov .....	64
Tabulka 21 Scénář – náhodná generace obrázků.....	65
Tabulka 22 Scénář – zobrazení vstupů .....	66
Tabulka 23 Popis dílčích cílů.....	103
Tabulka 24 Naplnění dílčích cílů.....	118

**SEZNAM GRAFŮ**

Graf 1 Anketa – věk respondentů .....	104
Graf 2 Anketa – pozice ve školství .....	105
Graf 3 Anketa – pohlaví respondentů .....	105
Graf 4 Anketa – užitečnost počítačových sítí .....	106
Graf 5 Anketa – důležitost počítačových sítí.....	107
Graf 6 Anketa – zkušenosti s počítačovými sítěmi.....	108
Graf 7 Anketa – interval hraní her .....	109
Graf 8 Anketa – zkušenosti s podobnými hrami.....	110
Graf 9 Anketa – intuitivnost hry .....	110
Graf 10 Anketa – obtížnost hry.....	111
Graf 11 Anketa – zábavnost hry .....	112
Graf 12 Anketa – efektivita učení počítačových sítí.....	112
Graf 13 Anketa – využití hry při výuce .....	113
Graf 14 Anketa – vliv her na zájem o vzdělávání.....	114
Graf 15 Anketa – výuka formou her .....	115
Graf 16 Anketa – změna zájmu o počítačové sítě po hraní hry .....	116
Graf 17 Anketa – doporučení hry .....	117

## SEZNAM PŘÍLOH

Příloha P I: CD

## **PŘÍLOHA P I: NÁZEV PŘÍLOHY**

Součástí přiloženého CD je zdrojový kód aplikace