

# **GYMNÁZIUM JÍROVCOVÁ 8**

## **MATURITNÍ PRÁCE**

**Vědomostní hra**

**Sára Jirkalová**

**vedoucí práce: Dr.rer.nat. Kočer Michal**

**V Českých Budějovicích 6. února 2025**

**2024/2025**

# **Prohlášení**

Prohlašuji, že jsem tuto práci vypracovala samostatně s vyznačením všech použitých pramenů.

V Českých Budějovicích dne ..... podpis .....

Sára Jirkalová

## **Abstrakt**

## **Klíčová slova**

# Poděkování

# **Obsah**

<b>I</b>	<b>Vědomostní hry v současnosti</b>	<b>2</b>
<b>1</b>	<b>Vědomostní hra</b>	<b>3</b>
<b>2</b>	<b>Dobyvatel</b>	<b>4</b>
2.1	Historie . . . . .	4
2.2	Grafické rozhraní . . . . .	5
2.3	Průběh hry . . . . .	5
<b>3</b>	<b>Sporcle</b>	<b>7</b>
3.1	Historie . . . . .	7
<b>4</b>	<b>Kahoot!</b>	<b>8</b>
4.1	Historie . . . . .	8
4.2	Průběh hry . . . . .	9
4.3	Grafické rozhraní . . . . .	9
4.4	Vliv na vzdělávání . . . . .	10
<b>II</b>	<b>Tvorba vědomostní počítačové hry</b>	<b>11</b>
<b>5</b>	<b>Cíl projektu</b>	<b>12</b>
5.1	Agilní metody . . . . .	12
5.1.1	Historie . . . . .	13
5.1.2	Extrémní programování . . . . .	13
5.1.3	Agilní metody v této práci . . . . .	14
5.2	Původní návrh . . . . .	15

<b>6</b>	<b>První verze</b>	<b>16</b>
6.1	verze 1.1 . . . . .	18
<b>7</b>	<b>Další verze?</b>	<b>19</b>
<b>8</b>	<b>Finální podoba hry</b>	<b>23</b>
8.1	Průběh hry . . . . .	23
8.2	Grafické rozhraní . . . . .	24
	<b>Bibliografie</b>	<b>29</b>

# Úvod

# Část I

## Vědomostní hry v současnosti

# 1 Vědomostní hra

Pojem vědomostní hra v této maturitní práci označuje hru založenou na odpovídání na otázky na různá téma, např. geografie, biologie, jazyky, či všeobecný přehled. Vědomostní hry lze zařadit mezi tzv. didaktické hry. To jsou takové hry, jejichž primární cíl není zábava, ale vzdělávání. Tyto hry se snaží zlepšit a zefektivnit učení za pomocí prvků herního designu, např. systém výzev a odměn, zpětná vazba atd.[4]

Videohry jako nástroj pro učení se využívají hlavně proto, že zvyšují motivaci studentů a tím zvyšují i čas, který stráví učením se.[14] Pokud jsou tyto hry navržené správně, dokáží přinášet lepší výsledky než tradiční způsoby učení. Nicméně, vědomostní a didaktické hry často nedokáží být dostatečně efektivní v zapojení a motivování studentů a vzbuzování jejich zájmu.[4]

## 2 Dobyvatel

Hra Dobyvatel je českou verzí maďarské vědomostní a strategické online hry *ConQUIZtador*, vyvíjenou společností *THX Games*.[22] V této hře proti sobě bojují tři hráči o jednotlivá území České republiky odpovídáním na dva typy otázek (*otázky s výběrem správné odpovědi* a *tipovací otázky*). Když hráč odpoví správně, nebo lépe než ostatní hráči v případě *tipovacích otázek*, má možnost zabrat nějaké území. Za každé získané území dostane hráč body. Vyhrává hráč s nejvyšším počtem bodů. [24]

Odpovědí na *tipovací otázku* je vždy celé nezáporné číslo a vyhrává ten hráč, kterého odpověď byla nejblíže té správné. Pokud je několik odpovědí stejně daleko, vyhrává ten hráč, který odpověděl jako první.

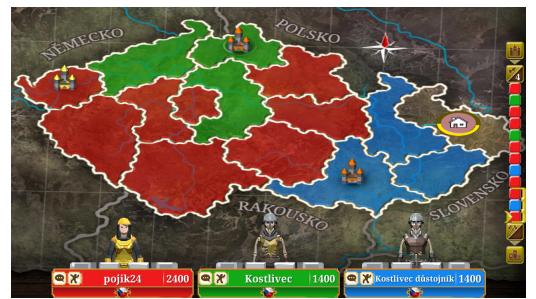
U *otázek s výběrem správné odpovědi* mají hráči vždy na výběr ze čtyř možných odpovědí, z nichž je vždy právě jedna správná. [25]

### 2.1 Historie

Hra Dobyvatel byla spuštěna v roce 2010 na portálu *TV Nova*. Již během prvních dvou let si hru vyzkoušelo více než 500 000 lidí. [33] V roce 2016 byla změněna grafická úprava hry a byly i pozměněny některé herní mechaniky. Tato změna se však nesetkala s velkým úspěchem v hrácké komunitě a vznikla dokonce petice za obnovení starého *Dobyvatele*. Tato petice však nezískala ani tisíc podpisů[13] a ”*nový Dobyvatel*” přetrval až do roku 2021. Na konci tohoto roku byla totiž webová verze hry *Dobyvatel* zrušena kvůli ukončení podpory *Adobe Flash Player*.[31] Dnes je možné si hru zahrát pouze na mobilních zřízeních. Tato mobilní verze *Dobyvatele* však není mezi hráči příliš populární a to hlavě kvůli novému 3D vizuálu.[11] Z toho důvodu začaly vznikat různé fanouškovské projekty, které se snaží simulovat původní webovou aplikaci. Jedním z takových projektů je hra *Vyzylvatel*. Autor, vlastním jménem



Obrázek 2.1: Ukázka GUI ”starého“ dobyvatele [12]



Obrázek 2.2: Ukázka GUI mobilní aplikace Dobytatel

Maroš Mečiar, někdy vystupující pod přezdívkou *Maroso*, se svou hrou snažil nejen přiblížit původnímu dobyvateli, ale i přidat něco navíc. Například si hráči mohou vybrat pouze sady otázek z určitých oborů, nebo si dokonce napsat vlastní.[23]

## 2.2 Grafické rozhraní

Grafické rozhraní hry Dobytatel se v průběhu času několikrát značně změnilo. V prvních verzích bylo trochu jednodušší, využívalo teplých barev a odpovídalo stylu své doby. Zobrazení mapy v tomto rozhraní bylo v perspektivě což ovšem změnilo velikostní poměr jednotlivých krajů a některé vypadají nepřiměřeně velké. V dalších verzích bylo grafické rozhraní čistě 2D. Všechny textury a ikony byly v kresleném stylu, který byl přehlednější a opět odpovídal své době. Nynější verze mobilní aplikace dobyvatel využívá velkého množství 3D prvků, které ale kvůli výkonu průměrných mobilních telefonů nemohou být moc detailní a působí tedy levně. Celá aplikace pak také využívá tmavších barev, což přehlednosti moc nepřidává.

## 2.3 Průběh hry

Průběh hry se liší v závislosti na tom, zda se jedná o *Běžnou hru*, nebo *Dlouhou kampaň*. *Běžná hra* má tří fáze. Při první fázi, s názvem *Stavba základny*, hra náhodně přidělí každému hráči základnu na jednom z polí. Tato pole jsou vybrána tak, aby se nedotýkala. Pole se základnou má hodnotu 1000 bodů a obsahuje tři věže. Druhá fáze se jmenuje *Expanze*. Při této fázi dostávají všichni hráči tipovací otázku. Vítězný hráč si může vybrat dvě pole, hráč který se umístil druhý jedno pole. Hráči si mohou vybrat pouze prázdná pole, a pokud takové existují, dotýkající se jejich základny nebo jejich jiného pole. Hráč dostane 200 bodů

za každé takto zabrané území. Třetí a poslední fáze s názvem *Bitva* se skládá ze čtyř částí, neboli kol, při kterých hráči postupně útočí na nepřátelská území. Pořadí hráčů je v každém kole, vyjma posledního, jiné. V posledním kole je pořadí určeno počtem bodů hráčů. Kdo má více bodů, útočí dříve. Útočník a hráč na kterého je útočeno (obránce) odpovídají na otázku s jednou správnou odpovědí. Třetí hráč je pozoruje, ale jinak se boje neúčastní. Pokud útočník neodpověděl správně, nebo na otázku nereagoval, vyhrává obránce. Pokud útočník zvolí správnou odpověď a obránce zvolí špatnou, vyhrává útočník a pokud odpoví oba hráči správně, rozhodne se bitva tipovací otázkou. Pokud útočník vyhraje, získá pole i s body, a bodová hodnota tohoto pole se nastaví na 400. Obránce tím pádem přichází nejen o území, ale taky o body, které tomuto území náležely. Pokud se ale obránci podaří území ubránit správným zodpovězením otázky, získá defenzivní bonus 100 bodů. Při útoku na základnu musí útočník rozbít všechny její věže, to znamená, že musí ve svém tahu zodpovědět jednu otázku za každou věž. Pokud se obránci podaří úspěšně ubránit, bitva končí, ale všechny již dobyté věže zůstanou dobyté a při dalším pokusu je útočník nemusí dobývat znova. Když padne poslední věž, obránce ztrácí všechno své území a body ve prospěch útočníka. Pokud poslední bitva nerozhodne o výherci celé hry, dostanou hráči ještě jednu tipovací otázkou za 10 bodů. [30]

*Dlouhá kampaň* je rozdělena do čtyř fází. První se nazývá *Stavba základny*. Během této fáze si na rozdíl od *Běžné hry* hráči vybírají umístění základny sami. Nejdříve si vybírá červený hráč, poté modrý a nakonec zelený. Barvy jsou hráčům přiděleny náhodně. Stejně jako u *Běžné hry* se základny nemohou dotýkat a každá má hodnotu 1000 bodů. Následuje fáze s názvem *Expanze*. Zde si hráči vybírají sousední území a mohou je obsadit, pokud správně odpoví na otázku s výběrem správné odpovědi. Takto získaná území mají hodnotu 200 bodů. Tato fáze má šest kola a pořadí hráčů se mění v každém kole. Třetí fáze se jmenuje *Rozdělení volných území*. V této fázi dostávají hráči tipovací otázky a výherce každé této otázky si může zabrat jedno volné území. Tato pole mají hodnocení 300 bodů. Fáze končí obsazením posledního volného území. Čtvrtá fáze, *Bitva*, probíhá téměř stejně jako u *Běžné hry*. Jediným rozdílem je to, že si hráč může dostavět padlé věže své základny zodpovězením otázky s výběrem správné odpovědi.[25]

## 3 Sporcle

*Sporcle* je webová a mobilní aplikace umožňující lidem vytvářet a hrát vědomostní kvízy.[1] Tyto kvízy jsou rozděleny do 15 hlavních kategorií: sporty, geografie, hudba, filmy, televize, jen pro zábavu, historie, literatura, jazyk, věda, gaming, zábava, náboženství, svátky a ostatní. *Sporcle* také nabízí několik formátů kvízů, např. klasický, kdy musí hráč napsat správnou odpověď, klikací, s mapu, atd. Existuje také několik módů, které může autor kvízu nastavit, například minové pole, kvíz se automaticky ukončí po zadání špatné odpovědi, pevné pořadí, hráč nemůže přeskakovat mezi otázkami, atd.[29]

### 3.1 Historie

Matt Ramme zaožil Sporcle v lednu roku 2007. První kvíz, s názvem: "Vyjmenuj prezidenty USA", byl přidán v létě téhož roku. Při zakládání předpokládal, že většina uživatelů budou starší lidé, ale aplikace se uchytila i mezi studenty. [5] Aby tento zájem ještě více vývojáři podpořili, přidali v listopadu roku 2010 žebříček univerzit podle toho, kolik jejich studentů stránku navštívilo.[38] V roce 2010 byla vyvinuta mobilní aplikace *Sporcle* pro iOS. Už v roce 2013 *Sporcle* přesáhlo jednu miliardu zahrání.

## 4 Kahoot!

*Kahoot!* je Norská online vzdělávací platforma umožňující lidem tvořit a hrát kvízy založené na otázkách s výběrem správné odpovědi.[19] Původní prototyp této platformy byl vytvořen už v roce 2006[36], ovšem k největšímu rozmachu došlo až v roce 2020 kvůli pandemii COVID-19.[17]

Cílem této platformy je zvýšení aktivního zapojení, motivace a soustředění žáků a studentů a zlepšit jejich studijní výsledky. Aplikace *Kahoot!* je také používána jako nástroj formativního hodnocení nebo jako nahrazení tradičních výukových prostředků.[37]

### 4.1 Historie

Původní prototyp který vedl až ke spuštění platformy *Kahoot!* se nazýval *Lecture Quiz*. Byla to mobilní aplikace vyvíjená Alfem Inge Wangem, Terje Øfsdahlem a Ole Kristian Mørch-Storsteinem na katedře informatiky Norské univerzity vědy a technologie, která měla lépe zapojit studenty do výuky. Poprvé byla použita v roce 2007 na univerzitě, kde byla vyvíjena, na přednášce a získala od studentů pozitivní recenze. Studenti uvedli, že se více soustředili na přednášku a hra jim přinesla přínosná pro ně i pro profesora. Hra však měla pár technických problémů, které se ale podařilo vyřešit.[36]

Druhá verze s názvem *Lecture Quiz 2.0* vyšla v roce 2011 na téže univerzitě jako verze předchozí. Autoři projektu se u druhé verze zaměřili zejména na stabilitu celého systému, zjednodušení používání pro studenty i učitele a dobrou dokumentaci usnadňující další vývoj platformy. Těchto cílů chtěli dosáhnout vytvořením webového rozhraní. Testu této vylepšené hry se zúčastnilo 21 studentů v rámci výuky a test tentokrát proběhl téměř bez technických závad. Z dotazníkového šetření uskutečněného po testu vyplývá, že cíle tohoto projektu byly dosaženy.[39]

Společnost *Kahoot!* byla založena v roce 2012 Mortenem Versvikem, Johanem Brandem a Jamiem Brookerem. Technologie této společnosti je zložena a výzkumu, jenž provedl Morten Versvik, student profesora Alfa Inge Wanga, který se také zapojil do vývoje této vědomostní hry. V roce 2013 byla tato platforma poprvé zpřístupněna pro veřejnost.[19] V už roce 2017 měla tato hra přes 40 milionů aktivních hráčů měsíčně[21] a tento rok také společnost *Kahoot!* vydala mobilní aplikaci, skrze kterou mohou učitelé zadávat žákům domácí úkoly.[28]

## 4.2 Průběh hry

Jednotlivé kvízy se nejčastěji hrají pomocí jedné velké obrazovky a mobilních telefonů pro každého hráče. Na velké obrazovce se zobrazí otázka a výběr odpovědí a hráči pomocí mobilních zařízení odpovídají. Každý hráč hraje pod zvolenou přezdívkou, takže kvíz může být anonymní, což pomáhá snižovat strach ze špatné odpovědi. Za každou správnou odpověď dostanou hráči počet bodů podle toho, jak rychle odpověděli. Mezi otázkami se vždy zobrazí na velké obrazovce správná odpověď a distribuce odpovědí, což dává zpětnou vazbu tomu, kdo kvíz vytvořil. Například, pokud většina hráčů zadala stejnou špatnou odpověď, mohla být otázka položena nepřesně. Poté se na obrazovce zobrazí přezdívky a body pěti nejúspěšnějších hráčů. Zároveň každý hráč vidí individuální zpětnou vazbu na své obrazovce.[34] Hráč s největším počtem bodů vyhrává. Tento styl hry se dá využívat například ve školách, ve firmách nebo na různých společenských akcích.

Druhá možnost jak hrát *Kahoot!* nepotřebuje velkou obrazovku. Každý hráč na svém zařízení vidí otázku i možné odpovědi a kvíz si může zahrát kdekoli a kdykoli. Tuto možnost je vhodné používat například na domácí úkoly, školení, atd. [20]

## 4.3 Grafické rozhraní

Vývojáři zvolili pro hru *Kahoot!* jednoduchý vizuál, založený na výrazných a kontrastních barvách, které dokáží snadno upoutat pozornost, ale zároveň nejsou moc křiklavé a neruší hráče. Každá odpověď je vyznačena jak barvou, tak i tvarem, což může být velmi prospěšné pro lidi trpící barvoslepotou. Jeden problém, kterého by si hráč mohl povšimnout je ten, že jedna z odpovědí, ta která je napsána na žlutém pozadí, může být na některých obrazovkách

špatně čitelná a může tím pádem znevýhodňovat některé hráče.

## 4.4 Vliv na vzdělávání

Nedostatek motivace mezi žáky může vyústit ve snížení efektivity vzdělání a negativní atmosféru ve třídě. Tento problém je ještě zřetelnější na vysokých školách s velkými třídami a málo interakce mezi přednášejícím a studenty. Výzkum ukázal, že studenti, kteří jsou aktivně zapojeni do výuky se naučí více, než pasivní studenti. Jedním ze způsobů jak žáky a studenty do výuky více zapojit jsou systémy pro odpovědi studentů (SRS - z anglického "Student response systems"). Dalším způsobem jak motivaci žáků a studentů zvýšit, je učení pomocí her. *Kahoot!* je kombinací systému pro odpovědi studentů a videohry. [37]

Většina studií porovnávající efekt aplikace *Kahoot!* a jiných výukových prostředků na výsledky vzdělávání se shoduje v tom, že používání hry *Kahoot!* při výuce způsobuje zlepšení výsledků vzdělávání oproti nejen tradičním výukovým nástrojům, ale i oproti jinému systému pro odpovědi studentů, který neměl herní prvky. Studenti kteří používali *Kahoot!* se také cítili více aktivně zapojeni do výuky a byli méně ve stresu. [32]

Další studie se zabývala rozdíly mezi dlouhodobým a krátkodobým používáním hry *Kahoot!*. Porovnávala názory studentů na několik otázek poté, co hráli Kahoot! v jedné přednášce s názory na téže otázky po používání *Kahootu* během celého semestru. Tato studie zjistila, že největší rozdíl byl v dynamice ve třídě. Studenti, kteří hráli *Kahoot!* jednou spolu během hry více komunikovali. U ostatních otázek byly rozdíly minimální.[34]

Jedna studie se také věnovala vlivu hudby a bodování, které jsou oboje součástí aplikace a mají za cíl zvýšit zájem a soutěživost mezi studenty, na motivaci, koncentraci a zájem studentů a dynamiku ve třídě. Bylo zjištěno, že přítomnost hudby měla vliv na celkovou energii studentů. Když byla hudba zapnutá, studenti více komunikovali, diskutovali a byli otevřeni dotazům vyučujícího. Bodování zvyšovalo motivaci studentů. Na druhou stranu měli však studenti bez bodování častěji pocit, že se naučili něco nového.[35]

## Část II

**Tvorba vědomostní počítačové hry**

## 5 Cíl projektu

Cílem tohoto projektu bylo naprogramovat vědomostní hru a použít přitom moderní postupy agilního programování. Tato vědomostní hra by měla být vhodná pro použití ve školním prostředí. Dále by si měl uživatel moct přidat vlastní otázky a vybrat si pouze téma, která se potřebuje naučit, nebo která ho zajímají. Dále by měl hra být pochopitelná i pro lidi, kteří nikdy počítačové hry nehráli.

Na vytvoření tohoto programu jsem se rozhodla použít programovací jazyk *Python* s rozšířením *Pygame*. Je to set modulů navržený pro tvorbu her a multimediálních programů. Pro jeho instalaci je potřeba mít nainstalovaný python.[26] Pygame instalujeme pomocí instalačního programu balíčků pro Python (zkráceně pip).[27]

### 5.1 Agilní metody

Agilní metody vývoje softwaru jsou takové metody, které následují hodnoty Manifestu Agilního vývoje software a jeho 12 principů. Autoři zvolili název *Agilní*, protože odráží důležitost adaptability a připravenosti na změnu. Změna je totiž pro Agilní vývoj softwaru velice důležitá. Jedním ze čtyř bodů Manifestu Agilního vývoje software je: Reagování na změny před dodržováním plánu. Vývojáři pracující agilně tudíž vítají změnu požadavků i během samotného vývoje. I proto je důležitý další bod manifestu: Spolupráce se zákazníkem před vyjednáváním o smlouvě. Při Agilním vývoji je tudíž nutná neustálá komunikace se zákazníkem. K tomuto se váže i jeden z dvanácti principů Agilního Vývoje, a to sice: Dodáváme fungující software v intervalech týdnů až měsíců, s preferencí kratší periody. [9] Dalším bodem manifestu je: Jednotlivci a interakce před procesy a nástroji. Jeden z rozdílů mezi Agilním a jinými přístupy k vývoji softwaru je zaměření na lidi a jejich interakce. Posledním, čtvrtým, bodem Agilního Manifesta je: Fungující software před vyčerpávající dokumentací.[8] [3]

### **5.1.1 Historie**

Za počátek Agilního vývoje se většinou považuje schůze 17 vývojářů v Utahu v roce 2001. Agilní přístup ovšem existoval i předtím, jen nebyl ucelený a neměl zatím název. Už v polovině devadesátých let vymýšleli lidé nové postupy, které se podobaly dnešnímu Agilnímu vývoji. Manifest Agilního vývoje softwaru vznikl právě na již zmíněné schůzi v Utahu. Tento manifest shrnul základní myšlenku a poprvé zde byl užit termín Agilní vývoj softwaru. V následujících měsících doplnili autoři manifest o 12 principů agilního vývoje softwaru. Na konci roku 2001 byla také vytvořena *Agile Alliance*. Jejím cílem je spojovat lidi vyvíjející software, aby mohli sdílet nápady a zkušenosti. [2]

### **5.1.2 Extrémní programování**

Jednou z agilních metod vývoje je tzv. Extrémní programování (XP). Jedná se o metodiku pro malé až střední týmy, které vyvíjejí software a mají nejasné, nebo často se měnící zadání. Cíl XP je snižovat projektová rizika, zlepšit schopnost reagovat na změny a zvýšit produktivitu.

Tato metodika je vlastně soubor několika postupů. Těmi postupy jsou:

- Plánovací hra (Rychle vytvoříme plán na další verzi, pokud je skutečnost jiná, plán zaktualizujeme)
- Malé verze (Rychle vytvoříme základní verzi projektu, kterou poté vytváříme další verze v krátkých cyklech)
- Metafora (Celý tým má stejnou představu o tom, jak má celý systém vypadat)
- Jednoduchý návrh (Systém by měl být navržen vždy co nejjednodušejí)
- Testování (Programátoři píší neustále testy jednotek, zákazník provádí testy funkcionality)
- Refakorizace (Programátoři vylepšují program bez toho, aby změnili jeho funkcionality)
- Párové programování (Všechn kód by měl být psán v párech)
- Společné vlastnictví (Všichni programátoři nesou zodpovědnost za celý kód, každý může jakoukoli část kdykoli upravit)

- Nepřetržitá integrace,
- 40-hodinový pracovní týden (Pracovní týden by neměl překročit 40 hodin a pokud je zapotřebí pracovat přesčas, nikdy takto nepracujeme dva týdny za sebou)
- Zákazník na pracovišti (zákazník by měl mít na pracovišti zástupce, kterému můžou programátoři neustále klást dotazy a který může provádět testy funkcionality)
- Standarty pro psaní zdrojového kódu (Zdrojový kód by měl mít jednotnou strukturu)

Jedním z kontroverznějších z těchto postupů je bezpochyby párové programování. Jedná se o postup, při kterém programují dva lidé zároveň na jednom počítači, s jednou klávesnicí a jednou myší. Celý proces spočívá v tom, že vždy jeden z programátorů píše a druhý ho kontroluje a zároveň přemýslí nad dalším problémem, dalším testem nebo možnou refaktORIZACÍ. Výhodou oproti tradičnímu postupu je například fakt, že při párovém programování se více chyb odhalí už při samotném psaní kódu. Další výhodou je i to, že když člověk programuje s někým, tak bude nejspíše produktivnější, protože se z úcty ke druhému nenechá rozptýlit telefonátem nebo jinou soukromou záležitostí. To nás ale přivádí k jedné z nevýhod a tou je to, že tento proces funguje pouze za předpokladu, že se tito dva lidé vzájemně snesou a mají k sobě dostatek respektu.

Podle metaanalýzy z roku 2009 nelze jednoznačně určit, zda je párové programování lepší než sólo programování, ale vždy záleží na dané situaci. Párové programování je totiž užitečnější, pokud potřebujeme jednoduchý úkol splnit co nejrychleji, i za cenu ztracené kvality, nebo potřebujeme splnit složitý úkol kvalitněji, i za vyšší cenu.[16] Párové programování také samozřejmě nebude lepší, pokud se členové týmu nerespektují, nebo se nesnesou.

### 5.1.3 Agilní metody v této práci

Pro svou práci jsem se rozhodla použít právě extrémní programování, protože má jasně dané postupy kterými jsem se mohla při psaní programu řídit. Vzhledem k povaze a velikosti projektu a vzhledem k tomu, že nepracuji v týmu, jsem se neřídila všemi postupy, ale vybrala jsem jen ty, které se k mojí práci hodí. I přesto, že jsem nepoužila všechny postupy jsem pořád postupovala agilně. Postupy, které jsem vybrala jsou: Plánovací hra, Malé verze, Metafora a Jednoduchý návrh.

Pracovala jsem tedy tak, že jsem nejdříve vytvořila velice jednoduchý prototyp mé hry a postupně jsem přidávala další funkcionality v malých verzích. Vždy když jsem jednu dokončila, udělala jsem plán té další.

## 5.2 Původní návrh

Jelikož jsem postupovala Agilně, měla jsem ze začátku pouze hrubý návrh, který jsem postupně upřesňovala. Počáteční návrh byl vědomostní hra, ve které by hráč soutěžil s počítačem, nebo jiným hráčem. Hráč by dostával otázky z různých oborů a za správnou odpověď by dostal body. Vyhrál by ten, kdo by měl nejvíce bodů. Poté, co jsem dokončila demo takovéto hry, jsem se rozhodla, že hráč bude rytíř procházející mapu a pokaždě, když vejde na nové políčko, má nějakou šanci na to, že ho napadne nějaká příšera a začne souboj. Při souboji bude hráč dostávat otázky a vždy když správě odpoví, dá příšeře nějaké poškození, jestliže odpoví špatně, poškození naopak dostane od příšery. Za každou zabitou příšeru by hráč dostal pár peněz herní měny, za které by si potom mohl kupit předměty, které by mu zvýšily šanci na výhru.

## 6 První verze

V této verzi jsem vytvořila jednoduchý prototyp mé hry. Byla to pouze terminálová aplikace pro dva hráče, kteří se střídali v odpovídání na otázky a nakonec aplikace vyhodnotila, který hráč získal více bodů.

Začala jsem importováním modulu csv, který jsem použila na manipulaci s databází otázek, která měla formu CSV souboru. Poté jsem nadefinovala třídu *Otzazka*. Tato třída má čtyři atributy, a těmi jsou: Typ, jenOtzazka, spravnaOdpoved, vsechnyOdpovedi. Její konstruktor má parametr *cisloOtzazky*. Tento konstruktor vyhledá v tabulce otázku s požadovaným číslem a přiřadí atributům třídy odpovídající proměnné.

```
1 class Otazka(object):
2     def __init__(self, cisloOtzazky):
3         with open("Otzazky.csv", mode='r', encoding='utf-8') as
4             soubor:
5                 data = csv.DictReader(soubor, delimiter=";")
6                 rows = list(data)
7                 self.typ = rows[cisloOtzazky]["Tema"]
8                 self.jenOtzazka = rows[cisloOtzazky]["Otzazka"]
9                 self.spravnaOdpoved = rows[cisloOtzazky][
10                     "SpravnaOdpoved"]
11                 self.vsechnyOdpovedi = [rows[cisloOtzazky][
12                     "SpravnaOdpoved"], rows[cisloOtzazky][
13                     "SpatnaOdpoved1"], rows[cisloOtzazky][
14                     "SpatnaOdpoved2"], rows[cisloOtzazky][
15                     "SpatnaOdpoved3"]]
```

Zdrojový kód 6.1: Konstruktor třídy *Otzazka*

Druhou třídou tohoto programu je třída DemoHrac. Má pouze dvě proměnné a těmi jsou jmeno a skore.

Třetí třídou tohoto programu je třída Demo. Má tři parametry: pocetKol, jmenoHrace1 a jmenoHrace2. Ve svém konstruktoru inicializuje dva objekty třídy Hrac, každý s jedním ze

zadaných jmen.

Dále jsem nadefinovala metodu zeptej třídy Demo. Tato metoda má jeden parametr otázka, což je objekt třídy Otázka. Metoda zeptej nejdříve zobrazí hráči otázku a výběr odpovědí v náhodném pořadí. Pro uspořádání odpovědí do náhodného pořadí jsem použila modul random a jeho metodu shuffle. Poté vyzve hráče k zadání správné odpovědi. Poté program ověří správnost odpovědi pomocí metody kontrola, která vrací True pokud je odpověď správná, v opačném případě vrací False. Pokud hráč zadal správnou odpověď, metoda zeptej vrací True a vypíše na terminál ”Správně!”. Pokud hráč zadal naopak špatnou odpověď, metoda vrátí False a vypíše zprávu ”Špatně!”.

```
1 def zeptej(self, otazka):
2     ABCD = {"A":0, "B":1, "C":2, "D":3}
3     print(otazka.jenOtazka)
4     random.shuffle(otazka.vsechnyOdpovedi)
5     print(otazka.vsechnyOdpovedi)
6     while True:
7         print("Zadej jednu z možností A, B, C, D")
8         odpoved = input("Zadej odpověď: ")
9         if odpoved in ABCD:
10             break
11     odpoved = otazka.vsechnyOdpovedi[ABCD[odpoved]]
12     if otazka.kontrola(odpoved):
13         print("Správně!")
14         return True
15     else:
16         print("Špatně!")
17         return False
```

Zdrojový kód 6.2: Metoda *zeptej*

Další metodou třídy Otazka je tah. Má jeden parametr hrac (objekt třídy Hrac). Tato metoda vybere náhodnou otázku pomocí funkce randint modulu random. Poté zavolá metodu zeptej a pokud vrátí True, připíše hráči jeden bod.

```
1 def tah(self, hrac):
2     otazka = Otazka(random.randint(1,10))
3     spravnost = self.zeptej(otazka)
4     if spravnost:
5         hrac.skore += 1
```

Zdrojový kód 6.3: Metoda *tah*

Poslední metodou třídy Demo je run. Tato metoda postupně zavolá funkci tah pro každého hráče a poté vytiskne aktuální skóre na terminál. Toto opakuje několikrát, podle toho, jaký počet kol byl zadán.

Nakonec program vytvoří instanci třídy Demo a spustí její metodu run.

```
1 if __name__ == "__main__":
2     aplikace = Demo(3, "Lubomír", "Miroslav")
3     aplikace.run()
```

Zdrojový kód 6.4: Spuštění dema

## 6.1 verze 1.1

Tato verze se od té první liší tím, že místo CSV souboru využívá SQL databázi. Pro práci s touto databází využívá modul Databaze. Tento modul má jednu třídu s názvem DbOtazek. V konstruktoru této třídy se vytvoří tabulka s názvem otazky a strukturou

Dále má tato třída tři metody: pridejOtazku, otazka a pocetotazek. Metoda pridejOtazku přidává do databáze otázky. Funkce otazka v databázi vyhledá otázku se zadaným číslem a vrátí ji jako list. Metoda pocetotazek vrací počet všech otázek v databázi.

Konstruktor třídy Otazka se tedy liší od první verze tím, že použije modul databaze k vybrání otázky z databáze. Tuto otázku poté rozbalí do několika proměnných.

```
1 import databaze
2
3 class Otazka(object):
4     def __init__(self, cisloOtazky):
5         self.cislo = cisloOtazky
6         self.otazky = databaze.Db_otazek()
7         self.celaOtazka = self.otazky.otazka(cisloOtazky)
8         self.id, self.typ, self.jenOtazka, self.spravnaOdpoved,
9             spatnaOdpoved1, spatnaOdpoved2, spatnaOdpoved3 =
10                self.celaOtazka
11         self.vsechnyOdpovedi = [self.spravnaOdpoved,
12             spatnaOdpoved1, spatnaOdpoved2, spatnaOdpoved3]
```

Zdrojový kód 6.5: Konstruktor třídy Otazka

## 7 Další verze?

Tato verze se od první výrazně liší. Nejnápadnější z rozdílů je ten, že tato verze má grafické rozhraní. Na vytvoření tohoto grafického rozhraní jsem použila set modulů pygame. Dalším rozdílem je to, že je určena pouze pro jednoho hráče.

Začala jsem vytvořením třídy App. Její konstruktor jako první inicializuje moduly Pygame, nastaví proměnnou `_running`, která určuje, zda aplikace běží na True, nadefinuje hlavní [surface] a zjistí jeho velikost. Dále nadefinuje atribut db jako databázi otázek[?].

```
1 class App:
2     def __init__(self):
3         pygame.init()
4         self._running = True
5         self._display_surf = pygame.display.set_mode((0,0),
6             pygame.HWSURFACE | pygame.DOUBLEBUF, pygame.
7                 FULLSCREEN)
8         self.sirka, self.vyska = self._display_surf.get_size()
9
10        self.db = databaze.Db_otazek()
```

Zdrojový kód 7.1: Konstruktor třídy App

První metodou třídy App je `on_execute`, která slouží ke spuštění aplikace. Pokud je proměnná `_running` rovna True, tzn. pokud má aplikace běžet, nastaví tato metoda pozadí na bílou barvu a zavolá metody `on_event` a `on_render`. V opačném případě aplikaci zavře.

```
1 def on_execute(self):
2     while self._running:
3         self._display_surf.fill((255,255,255))
4         for event in pygame.event.get():
5             self.on_event(event)
6         self.on_render()
7     pygame.quit()
```

Zdrojový kód 7.2: Metoda `on_execute`

Metoda on\_event slouží pro [kontrolu] událostí. Metoda on\_render rozhoduje, která obrazovka se hráči bude zobrazovat.

Poté jsem vytvořila grafické rozhraní pro pokládání otázek a odpovídání na ně. Byl to zatím jednoduchý návrh, který u horního okraje obrazovky vykreslil otázku a zobrazil 4 možné odpovědi. Když hráč některou z nich zvolil, program mu ukázal, zda je jeho odpověď správná, či nikoli. Na vybírání otázek z databáze jsem využila již zmíněnou třídu Otazka. Na vykreslení a funkcionality tlačítka jsem použila modul button.

Tento modul má třídu Button. Tato třída umožňuje vytvoření tlačítka a práci s ním. Má čtyři parametry: x, y, obrazek a text. Parametry x a y udávají souřadnice levého horního rohu tlačítka. Parametr obrazek [] a parametr text udává text, který bude na tlačítku vykreslen. Tento parametr není povinný a jeho [defaultní] hodnota je prázdný řetězec. Tato třída má dvě metody. První, s názvem handle\_event, slouží pro zjišťování, zda hráč na tlačítko klikl. Druhá metoda s názvem zobraz vykreslí tlačítko na zadaný [surface] a poté na toto tlačítko vypíše zadaný text.

Modul button má také třídu Textbox. Je to třída textových polí, která umožňují uživateli zadat text. Na rozdíl od třídy Button má pouze tři parametry. Chybí jí parametr text, ostatní jsou stejné. Má také metody handle\_event a zobraz. Metoda zobraz je stejná jako u třídy Button, metoda handle\_event se ale liší tím, že zjišťuje, zda má být textbox aktivní, a pokud ano, ukládá, uživatel píše, do proměnné text. Textbox je aktivní, pokud na něj uživatel klikne. Jakmile ale klikne mimo něj, textbox přestává být aktivní.

Jako další jsem přidala hlavní menu. To mělo zatím jen dvě tlačítka: Hrát a Opustit. Tato tlačítka byla opět vytvořena pomocí modulu button. Tlačítko Hrát zobrazilo hráči otázku a tlačítko Opustit zavřelo aplikaci.

Poté jsem přidala třídu Tvor a třídu Hrac. Tyto třídy nemají žádné metody a slouží tedy jen pro vytváření instancí tvorů a hráče s různými parametry. Také jsem vytvořila list se všemi druhy tvorů v této hře a přidala ho do proměnné

```

1 self.tvor = [Tvor("pavouk", 50, "obrazky/potvûrky_2.png", 15,
2   30),
3   Tvor("komár", 20, "obrazky/komarek.png", 20, 20),
4   Tvor("ryba", 50, "obrazky/potvûrky_1.jpg", 10, 25),
5   Tvor("netopýr", 55, "obrazky/netoparek.png", 12, 30)]

```

Zdrojový kód 7.3: proměnná tvori

Spolu s nimi jsem také přidala metodu souboj, která má dva parametry a těmi jsou vyherce a porazeny. Metoda odečte poškození výherce od životů poraženého. Pokud poražený zemřel, neboli pokud mu životy klesly na nulu nebo nižší, vrátí metoda True. V opačném případě vybere další otázku a vrátí False.

```

1 def souboj(self, vyherce, porazeny):
2     porazeny.zivoty = porazeny.zivoty - vyherce.utek
3     if porazeny.zivoty <= 0:
4         self.zobrazuj_mapu = True
5         self.zobrazuj_otazku = False
6         return True
7     else:
8         self.vyber_otazky()
9         return False

```

Zdrojový kód 7.4: proměnná tvori

Jako další jsem přidala mapu. Mapa byla původně tvořena čtvercovou sítí o velikosti 20x20, z čehož mohl hráč vidět při pohybu po mapě 7x7 okolí kolem něho. Nakonec jsem se ale rozhodla použít síť pouze 10x10 a hráči zobrazovat pouze oblast 3x3. Pohyb po mapě je tak jasnější, obzvláště díky větším políčkům. Mapa je definována v proměnné mapa třídy App jako slovník, ve kterém je ke každé souřadnici přiřazen typ políčka.

```

1 self.mapa = {(1,1):"L", (2,1):"L", (3,1):"L", (4,1):"V", (5,1):
2   "V", (6,1):"L", (7,1):"L", (8,1):"L", (9,1):"L", (10,1):"L",
3   (1,2):"L", (2,2):"L", (3,2):"V", (4,2):"V", (5,2):"O",
4   (6,2):"L", (7,2):"J", (8,2):"C", (9,2):"L", (10,2):"L",
5   (1,3):"V", (2,3):"V", (3,3):"V", (4,3):"V", (5,3):"C",
6   (6,3):"L", (7,3):"L", (8,3):"C", (9,3):"L", (10,3):"H",
7   (1,4):"J", (2,4):"C", (3,4):"C", (4,4):"C", (5,4):"C",
8   (6,4):"C", (7,4):"C", (8,4):"C", (9,4):"H", (10,4):"H",
9   (1,5):"B", (2,5):"B", (3,5):"C", (4,5):"B", (5,5):"B",
10  (6,5):"C", (7,5):"C", (8,5):"H", (9,5):"H", (10,5):"H",
11  (1,6):"L", (2,6):"B", (3,6):"C", (4,6):"B", (5,6):"J",
12  (6,6):"V", (7,6):"C", (8,6):"H", (9,6):"H", (10,6):"H",
13  (1,7):"L", (2,7):"L", (3,7):"C", (4,7):"L", (5,7):"L",
14  (6,7):"C", (7,7):"C", (8,7):"H", (9,7):"H", (10,7):"H",
15  (1,8):"L", (2,8):"L", (3,8):"C", (4,8):"L", (5,8):"L",
16  (6,8):"C", (7,8):"C", (8,8):"H", (9,8):"H", (10,8):"H",
17  (1,9):"L", (2,9):"L", (3,9):"C", (4,9):"L", (5,9):"L",
18  (6,9):"C", (7,9):"C", (8,9):"H", (9,9):"H", (10,9):"H",
19  (1,10):"L", (2,10):"L", (3,10):"C", (4,10):"L", (5,10):"L",
20  (6,10):"C", (7,10):"C", (8,10):"H", (9,10):"H", (10,10):"H"

```

```

8     (6,7) : "L" , (7,7) : "L" , (8,7) : "V" , (9,7) : "J" , (10,7) : "H" ,
9     (1,8) : "V" , (2,8) : "V" , (3,8) : "L" , (4,8) : "L" , (5,8) : "J" ,
10    (6,8) : "L" , (7,8) : "V" , (8,8) : "V" , (9,8) : "V" , (10,8) : "H" ,
11    (1,9) : "V" , (2,9) : "V" , (3,9) : "L" , (4,9) : "L" , (5,9) : "H" ,
12    (6,9) : "H" , (7,9) : "V" , (8,9) : "V" , (9,9) : "H" , (10,9) : "H" ,
13    (1,10) : "V" , (2,10) : "L" , (3,10) : "J" , (4,10) : "H" , (5,10) : "H"
14    , (6,10) : "H" , (7,10) : "H" , (8,10) : "H" , (9,10) : "H" , (10,10) : "H"
15    ,
16 }
```

Zdrojový kód 7.5: Definice mapy

Poté jsem přidala obchod a třídu Predmet. Tato třída má tři parametry a těmi jsou název, obrazek, cena. Tato třída má jednou metodu pouzit. Tato metoda bud' přidá hráči pět poškození, pokud se předmět jmenuje Lektvar síly, anebo hráče vylečí a přidá 20 životů, pokud se jmenuje Lektvar zdraví.

```

1 def pouzit(self):
2     if self.nazev == "Lektvar síly":
3         hrac.utok = hrac.utok + 5
4     elif self.nazev == "Lektvar života":
5         hrac.zivoty = hrac.max_zivoty
6         hrac.max_zivoty = hrac.max_zivoty + 20
```

Zdrojový kód 7.6: metoda pouzit

Když je hráč v obchodě, vidí dvě tlačítka, každé sloužící k nákupu jednoho z lektvarů. Dále se hráči zobrazuje, kolik goldů mu zbývá. V levém horním rohu obrazovky se nachází tlačítko zpět, díky kterému se může hráč dostat zpět na mapu.

Nakonec jsem přidala možnost přidat otázku do databáze. Na hlavní menu jsem přidala tlačítko Přidat otázku, na které když hráč klikne, zobrazí se mu rozhraní pro přidání otázky. Toto rozhraní obsahuje šest textových polí vytvořených pomocí třídy Button, každé pro jednu část otázky. Dále je na obrazovce tlačítko zpět, které vrátí hráče na hlavní menu a tlačítko přidat otázku, které přidá zadanou otázku do databáze.

## 8 Finální podoba hry

### 8.1 Průběh hry

Když uživatel hru spustí, jako první se mu zobrazí hlavní menu. To obsahuje tři tlačítka. Spodní tlačítko s nápisem quit zavře aplikaci. Prostřední tlačítko s nápisem přidat otázku otevře rozhraní, pomocí něhož může hráč zadat všechny části otázky a přidat ji do databáze. Poslední, horní, tlačítko s nápisem hrát spustí samotnou hru.

Když spustí hru, vidí hráč mapu. Po té se může pohybovat pomocí kláves W,A,S a D. Mapa je tvořena čtvercovými políčky, které jsou uspořádány v síti o velikosti 10x10. Každé políčko má přiřazený jeden ze sedmi typů. Těmito typy jsou: obchod, cesta, les, bažina, hory, jeskyně a voda. Hráč se může volně pohybovat po cestách, ale jakmile odbočí, zaútočí na něj nějaký tvor. Tvorů je celkem pět druhů (pavouk, ryba, komár, netopýr a [leshen]) a ke každému typu políčka je přiřazen jiný druh (les - pavouk, voda - ryba, bažina - komár, jeskyně/[hory] - netopýr). Když tedy hráč vejde na jiné políčko než cesta nebo obchod, zaútočí na něj příslušný tvor. S tímto tvorem se poté hráč musí utkat ve vědomostním souboji.

V horní části obrazovky se hráči zobrazí otázka a pod ní čtyři tlačítka s možnými odpověďmi. Nalevo od těchto tlačítek je vyobrazen hráč a pod ním [má health bar nějaký překlad?]. Na pravé straně je poté vykreslen tvor, který hráče napadl. Když hráč správně odpoví na otázku, dá poškození tvorovi. Když ale naopak odpoví špatně, dostane sám poškození. Pokud se hráči podaří zabít tvora, přičte se mu skóre a goldy.

Goldy jsou herní měna, kterou může hráč utratit v obchodě. Když hráč vejde do obchodu, zobrazí se mu v horní části obrazovky tlačítko zpět, pomocí něhož se může hráč vrátit zpět na mapu. Ve spodní části obrazovky je napsáno, kolik goldů má hráč. Dále se zde zobrazují

dvě tlačítka, pomocí nichž může hráč koupit jeden ze dvou lektvarů. Lektvar zdraví doplní hráče ztracené životy a přidá mu 20 životů navíc. Lektvar sily hráči zvýší poškození, které dává tvorům. Efekt lektvarů začne být aktivní, jakmile si ho hráč koupí.

Hra také počítá celkové skóre hráče a čím je vyšší, tím náročnější je hra. Každému tvoru se k počátečnímu poškození přidá setina hráčova skóre a k životům se mu přičte desetina hráčova skóre.

Pokud v nějakém souboji hráč zemře, zobrazí se mu na obrazovce zpráva ”Zemřel jsi”, celkové dosažené skóre a dvě tlačítka. Pomocí jednoho může hráč hru ukončit, pomocí druhého hrát znovu. Hráči se vynuluje skóre a goldy a hráči i tvorům se vrátí poškození a životy na počáteční hodnoty.

## 8.2 Grafické rozhraní

Rozhodla jsem se pro jednoduchý, černobílý vizuál. Všechny obrázky použité v této hře jsem kreslila ručně na papír, poté skenovala a upravovala v programu Malování. V celé hře jsem volila dobře čitelná, bezpatková písma. Pro všechna tlačítka jsem využila písmo Arial a pro zobrazení otázky písmo Calibri, protože písmo Arial bylo moc široké a některé otázky by se nevešly na obrazovku.

# Závěr

# Bibliografie

1. *About us.* 2025. Dostupné také z: <https://www.sporcle.com/about/>.
2. AGILEALLIANCE. *A Short History of Agile.* 2025. Dostupné také z: <https://agilealliance.org/a-short-history-of-agile/>.
3. AGILEALLIANCE. *AGILE 101: What is Agile?* 2025. Dostupné také z: <https://agilealliance.org/agile101/>.
4. ARNOLD, Brian; KOEHLER, Matthew; GREENHALGH, Spencer. Design Principles for Creating and Maintaining Immersive Experiences in Educational Games. In: 2016.
5. AUCOIN, Don. *On top of the world* [online]. 2009. Dostupné také z: [https://web.archive.org/web/20100824073333/http://www.boston.com/lifestyle/articles/2009/06/16/visitors\\_to\\_website\\_sporcle\\_learn\\_geography\\_and\\_other\\_subjects\\_and\\_have\\_fun](https://web.archive.org/web/20100824073333/http://www.boston.com/lifestyle/articles/2009/06/16/visitors_to_website_sporcle_learn_geography_and_other_subjects_and_have_fun).
6. BECK, Kent. *Extrémní programování.* Praha: Grada, 2002. ISBN 80-247-0300-9.
7. BECK, Kent. *Test-driven development: By example.* 20. printing. Addison-Wesley, 2015. ISBN 0321146530.
8. BENNEKUM; ANDREW HUNT; KEN SCHWABER; ALISTAIR COCKBURN; RON JEFFRIES; JEFF SUTHERLAND; WARD CUNNINGHAM; JON KERN; DAVE THOMAS; MARTIN FOWLER; BRIAN MARICK, Kent Beck; James Grenning; Robert C. Martin; Mike Beedle; Jim Highsmith; Steve Mellor; Arie van. *Manifest Agilního vývoje software* [online]. 2001. Dostupné také z: <https://agilemanifesto.org/iso/cs/manifesto.html>.
9. BENNEKUM; ANDREW HUNT; KEN SCHWABER; ALISTAIR COCKBURN; RON JEFFRIES; JEFF SUTHERLAND; WARD CUNNINGHAM; JON KERN; DAVE THOMAS; MARTIN FOWLER; BRIAN MARICK, Kent Beck; James Grenning; Robert C. Martin; Mike Beedle; Jim Highsmith; Steve Mellor; Arie van. *Principy stojící za*

*Agilním Manifestem*. 2001. Dostupné také z: <https://agilemanifesto.org/iso/cs/principles.html>.

10. CRISPIN, Lisa; GREGORY, Janet. *Agile testing: A practical guide for testers and agile teams*. 1. ed., 10. print. Upper Saddle River [u.a.]: Addison-Wesley, 2014. A @Mike Cohn signature book. ISBN 978-0-321-53446-0. Literaturverz. S. 501 - 508.
11. *Dobyvatel*. 2025. Dostupné také z: <https://play.google.com/store/apps/details?id=com.thxgames.triviador&hl=cs&pli=1>.
12. *Dobyvatel (1.0)*. 2013. Dostupné také z: <https://mujsoubor.cz/prohlizecove-hry/dobyvatel>.
13. DOMBROVSKÝ, Jiří. *Petice za obnovení staré verze hry Dobyvatel*. 2016. Dostupné také z: <https://e-petice.cz/petitions/petice-za-obnoveni-hry-dobyvatel.html>.
14. DONDLINGER, Mary. Educational Video Game Design: A Review of the Literature. *Journal of Applied Educational Technology*. 2007, roč. 4.
15. DYBÅ, Tore; DINGSØYR, Torgeir. Empirical studies of agile software development: A systematic review. *Information and Software Technology*. 2008, roč. 50, č. 9–10, s. 833–859. ISSN 0950-5849. Dostupné z DOI: [10.1016/j.infsof.2008.01.006](https://doi.org/10.1016/j.infsof.2008.01.006).
16. HANNAY, Jo E.; DYBÅ, Tore; ARISHOLM, Erik; SJØBERG, Dag I.K. The effectiveness of pair programming: A meta-analysis. *Information and Software Technology*. 2009, roč. 51, č. 7, s. 1110–1122. ISSN 0950-5849. Dostupné z DOI: [10.1016/j.infsof.2009.02.001](https://doi.org/10.1016/j.infsof.2009.02.001).
17. HANOA, Eilert. *Kahoot! delivers record growth with big global shift to remote learning*. 2020. Dostupné také z: <https://kahoot.com/blog/2020/07/02/kahoot-delivers-record-growth-global-shift-remote-learning>.
18. HEALD, Mark. *Countdown to Four Billion: Wrap-Up* [online]. 2021. Dostupné také z: <https://www.sporcle.com/blog/2021/04/countdown-to-four-billion-wrap-up>.
19. KAHOOT! *About us*. 2025. Dostupné také z: <https://kahoot.com/company>.
20. KAHOOT! *What is Kahoot!?* 2025. Dostupné také z: <https://kahoot.com/what-is-kahoot/>.

21. KEANE, Jonathan. *Norwegian edtech company Kahoot! reaches 1 billion players, 40 million MAUs*. 2017. Dostupné také z: <https://tech.eu/2017/03/06/kahoot-1-billion-players>.
22. MACÍCH, Jiří. *Nova spustila online hru Dobytatel.cz, mutaci strategie Conquistador* [online]. 2010. Dostupné také z: <https://www.lupa.cz/clanky/nova-spustila-online-hru-dobytatel-cz>.
23. MEČIAR, Maroš. *Vyzvateľ.com*. 2025. Dostupné také z: <https://vyzvateľ.com>.
24. *O hře* [online]. 2017. Dostupné také z: <https://web.archive.org/web/20170617023135/https://thxgames.zendesk.com/hc/cs/articles/207966549-0-he>.
25. *Oficiální pravidla hry* [online]. 2018-10. Dostupné také z: <https://web.archive.org/web/20181031133118/https://thxgames.zendesk.com/hc/cs/articles/207970149-Pravidla-hry>.
26. PYGAME. *About - Wiki*. 2026. Dostupné také z: <https://www.pygame.org/wiki/about>.
27. PYGAME. *GettingStarted — wiki*. 2026. Dostupné také z: <https://www.pygame.org/wiki/GettingStarted>.
28. RAVIPATI, Sri. *Homework Game Changer Kahoot! Launches Mobile App*. 2017. Dostupné také z: <https://thejournal.com/articles/2017/09/14/homework-game-changer-kahoot-launches-mobile-app.aspx>.
29. Sporcle Quiz Categories. 2025. Dostupné také z: <https://www.sporcle.com/categories/>.
30. THXGAMES. *Game mode: Short Campaign*. 2023. Dostupné také z: <https://help.triviador.com/en/support/solutions/articles/66000215532-game-mode-short-campaign>.
31. THXGAMES. *Nová verze Dobytatele je dostupná pouze na mobilních zařízeních*. 2021. Dostupné také z: <https://web.archive.org/web/20211121231019/https://dobyvatel.triviador.net>.
32. TURAN, Zeynep; MERAL, Elif. Game-Based versus to Non-Game-Based: The Impact of Student Response Systems on Students' Achievements, Engagements and Test Anxieties. *Informatics in Education*. 2018, roč. 17, č. 1, s. 105–116.

33. TVNOVA. *Dobyvatel.cz slaví dva roky od svého spuštění!* 2012. Dostupné také z: <https://tn.nova.cz/zpravodajstvi/clanek/251791-dobyvatel-cz-slavi-dva-roky-od-sveho-spusteni>.
34. WANG, Alf. The wear out effect of a game-based student response system. *Computers and Education*. 2015, roč. 82, s. 217–227. Dostupné z DOI: 10.1016/j.compedu.2014.11.004.
35. WANG, Alf Inge; LIEBEROTH, Andreas. The effect of points and audio on concentration, engagement, enjoyment, learning, motivation, and classroom dynamics using Kahoot. In: *European conference on games based learning*. Academic conferences international limited, 2016, sv. 20, s. 738–746.
36. WANG, Alf Inge; ØFSDAHL, Terje; MØRCH-STORSTEIN, Ole Kristian. Lecture quiz-a mobile game concept for lectures. In: *Proceedings of the 11th IASTED International Conference on Software Engineering and Application (SEA '07)*. 2007, s. 305–310.
37. WANG, Alf Inge; TAHIR, Rabail. The effect of using Kahoot! for learning – A literature review. *Computers and Education*. 2020, roč. 149, s. 103818. ISSN 0360-1315. Dostupné z DOI: <https://doi.org/10.1016/j.compedu.2020.103818>.
38. WOODFORD, Antonia. *Sporcle ranks colleges based on use*. 2011. Dostupné také z: <https://yaledailynews.com/blog/2011/02/18/sporcle-ranks-colleges-based-on-use/>.
39. WU, Bian; WANG, Alf Inge; BØRRESEN, Erling Andreas; TIDEMANN, Knut Andre. IMPROVEMENT OF A LECTURE GAME CONCEPT - Implementing Lecture Quiz 2.0. In: *Proceedings of the 3rd International Conference on Computer Supported Education*. SciTePress - Science, and Technology Publications, 2011, s. 26–35. Dostupné z DOI: 10.5220/0003331300260035.

## **Seznam tabulek**