



**MATEMATICKO-FYZIKÁLNÍ
FAKULTA**
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Tadeáš Tomiška

**Vývoj mobilní aplikace a generátoru
otázek pro hru Desítka**

Ústav formální a aplikované lingvistiky

Vedoucí bakalářské práce: RNDr. David Mareček, Ph.D.

Studijní program: Informatika

Studijní obor: Informatika se specializací
Programování a vývoj software

Praha 2023

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V dne

Podpis autora

Rád bych poděkoval vedoucímu své bakalářské práce RNDr. Davidu Marečkovi, Ph.D. za jeho trpělivost, pomoc a čas, který mi při vypracování práce věnoval. Děkuji také své rodině za všeobecnou podporu při studiu.

Název práce: Vývoj mobilní aplikace a generátoru otázek pro hru Desítka

Autor: Tadeáš Tomiška

Ústav: Ústav formální a aplikované lingvistiky

Vedoucí bakalářské práce: RNDr. David Mareček, Ph.D., Ústav formální a aplikované lingvistiky

Abstrakt: Tato bakalářská práce se zaměřuje na vytvoření mobilní aplikace pro Android. Aplikace umožní hrát online verzi hry Desítka od společnosti Mindok. Součástí práce je i vytvoření otázek pro hru. Ty budou generovány pomocí webových stránek z Wikipedie. Pro získání potřebných dat bude použita technika parsování stránek. Aplikace bude napsána v jazyce Java a bude určena pro Android verze 10 a vyšší. Pro komunikaci mezi zařízeními bude použita client-server architektura. Komunikačním prostředkem bude technologie wifi. Aplikace bude mít stejná pravidla jako hra Desítka a bude podporovat 2 herní režimy. Půjde hrát v režimu online s ostatními hráči nebo v režimu friend s přáteli.

Klíčová slova: desítka generátor otázek wikipedia

Title: Development of a mobile application and question generator for the game Smart10.

Author: Tadeáš Tomiška

Institute: Institute of Formal and Applied Linguistics

Supervisor: RNDr. David Mareček, Ph.D., Institute of Formal and Applied Linguistics

Abstract: This bachelor's thesis focuses on creating a mobile application for Android that allows playing an online version of the game Smart10. The work also includes generating questions for the game, which will be generated using web pages from Wikipedia. The technique of web page parsing will be used to obtain the necessary data. The application will be written in Java and will be intended for Android versions 10 and higher. A client-server architecture will be used for communication between devices, with communication via Wifi technology. The application will have the same rules as the game Smart10 and will support two gaming modes. It will be playable in an online mode with other players or in a friend mode with friends.

Keywords: smart10 question generator wikipedia

Obsah

Úvod	3
1 Popis hry	4
1.1 Průběh hry	4
1.2 Otázky	4
2 Generování otázek	5
2.1 Nástroje pro generování	5
2.1.1 Ruční tvorba otázek	5
2.1.2 Generování otázek pomocí NLP	5
2.1.3 Generování otázek pomocí strojového učení	5
2.1.4 Zvolený nástroj	5
2.2 Zdroje dat	6
2.3 Získání dat	6
2.3.1 Nejnavštěvovanější stránky	6
2.3.2 Parsování stránek	6
2.4 Vytvoření otázek	8
2.5 Kontrola uživatelem	8
3 Analýza hry	9
3.1 Programovací jazyk	9
3.1.1 Java	9
3.1.2 Kotlin	9
3.1.3 C++	9
3.1.4 Zvolený jazyk	9
3.2 Model architektury	10
3.2.1 Peer-to-peer	10
3.2.2 Client-server	10
3.2.3 Zvolená architektura	10
3.3 Komunikace	10
3.3.1 Java Socket	10
3.3.2 GRPC	11
3.3.3 Zvolená knihovna	11
3.4 Požadavky na hru	11
3.4.1 Režimy hry a připojení	11
3.4.2 Zahájení hry	11
3.4.3 Hraní hry	11
3.4.4 Konec hry	12
3.5 Řešení technických problémů hráčů	12
3.6 Synchronizace hráčů	12
3.7 Testování hry	13
3.7.1 Ngrok	13

4	Uživatelská dokumentace	14
4.1	Menu hry	14
4.2	Hraní hry	16
4.3	Spuštění aplikace	18
4.3.1	Server	18
4.3.2	Ngrok	19
4.3.3	Mobilní aplikace	19
4.4	Vytvoření otázek pro hru	19
5	Programátorská dokumentace	21
5.1	Klient	21
5.2	Server	21
5.3	Komunikace mezi klientem a serverem	22
5.4	Question Generator	23
5.5	Aktuální nastavení hry	24
	Závěr	25
	Seznam použité literatury	26
	Seznam obrázků	27
A	Přílohy	28
A.1	Obsah elektronické přílohy	28

Úvod

Desítka je společenská vědomostní hra. V České republice byla vydána v roce 2018 vydavatelstvím Mindok. (5)

Tato bakalářské práce se zaměřuje na implementaci hratelné verze hry pro mobilní zařízení s operačním systémem Android verze 10 a vyšší. Cílem práce je také vytvořit dostatečné množství otázek k této hře, aby ji hráči mohli hrát opakovaně a vychutnat si nové a různorodé zážitky. Momentálně neexistuje žádná online verze této hry, a proto bude tato práce poskytovat novou a moderní formu pro hráče, kteří si přejí zahrát tuto oblíbenou hru na svých mobilních zařízeních kdykoli a kdekoli.

Hra bude navržena tak, aby byla co nejvíce podobná původní deskové hře. Kromě toho bude mít aplikace dva režimy hraní. Půjde hrát v režimu online s ostatními hráči nebo v režimu friend s přáteli. Pro komunikaci mezi hráči bude využita wifi a architektura aplikace bude postavena na klient-server modelu. Server bude zajišťovat správu her a hráčů, výběr otázek a kontrolu správnosti odpovědí.

1. Popis hry

Originální hra je určena pro dva až osm hráčů. Obsahuje 200 tématických okruhů, přičemž každý okruh má 10 otázek. (6) V naší implementaci bude režim online určen pro 3 hráče. V režimu friend bude moci hrát 2-5 hráčů.

1.1 Průběh hry

Hra se skládá z jednotlivých kol. V každém kole se náhodně vybere jeden okruh a hráči postupně odpovídají na otázky. Pokud hráč odpoví správně, získává bod a zůstává pro dané kolo ve hře. Odpoví-li hráč špatně, vypadává pro dané kolo ze hry a ztrácí všechny body v kole získané. Hráč se také může rozhodnout pasovat. V tom případě vypadává pro dané kolo ze hry, ale všechny body v kole získané mu zůstávají. Vítězí hráč, který jako první získá 20 bodů.

1.2 Otázky

Okruhy ve hře Desítka pokrývají široké spektrum oblastí, jako jsou například věda, historie, geografie, sport, kultura a mnoho dalších. Otázky v okruzích jsou sestaveny tak, aby byly dostatečně náročné, ale zároveň zajímavé a srozumitelné pro hráče různého věku a kulturního pozadí. Jsou formulovány tak, aby vyžadovaly určitou úroveň znalostí a logického myšlení, což umožňuje hráčům vylepšit své schopnosti a zároveň se bavit.



Obrázek 1.1: Ukázková otázka

2. Generování otázek

V této části se budeme zabývat vytvořením otázek do naší hry. Generování otázek je klíčovou částí vytváření herního obsahu a má významný vliv na zábavu a hrátelnost hry. V našem případě je cílem vytvořit dostatečné množství otázek, aby hráči mohli hrát hru opakovaně. Dalším cílem je udržet otázky přiměřeně složité, aby se udržela vyšší úroveň zábavy.

2.1 Nástroje pro generování

Existuje více možností, jak generovat otázky pro hru Desítka. My se podíváme na některé z nejčastěji používaných.

2.1.1 Ruční tvorba otázek

Jedná se o nejstarší a nejjednodušší způsob, jak získat otázky pro hru. V této metodě se otázky vytvářejí manuálně a mohou být zaměřeny na konkrétní téma. Tento způsob však vyžaduje spoustu času a úsilí.

2.1.2 Generování otázek pomocí NLP

Pomocí technik zpracování přirozeného jazyka (NLP) lze automaticky generovat otázky ke specifickému tématu. V tomto případě je klíčové vybrat správnou databázi otázek a definovat pravidla, podle kterých se otázky generují. Tento způsob umožňuje rychlé generování velkého množství otázek. Na druhou stranu, tyto otázky mohou být příliš jednoduché nebo neuspokojivé pro hráče.

2.1.3 Generování otázek pomocí strojového učení

Ke generování otázek pro hru můžeme využít i metodu strojového učení. Ta je založena na algoritmech učení, které umožňují systému „naučit se“ otázky a vytvořit otázky, které jsou přiměřeně složité a zajímavé pro hráče. Výhodou tohoto způsobu je, že systém může samostatně vytvářet nové otázky bez nutnosti ručního zadávání dat. Nevýhodou je, že strojové učení vyžaduje velké množství dat a výkonu počítače.

2.1.4 Zvolený nástroj

V našem případě použijeme ke generování otázek pro hru Desítka techniku zpracování přirozeného jazyka (NLP). Jedním z důvodů je, že potřebujeme vytvořit velké množství otázek, které budou dostatečně různorodé a přiměřeně složité, aby hra nepřestala být pro hráče zajímavá. Použití NLP technologií nám umožní rychle a efektivně zpracovat rozsáhlé množství textových dat a automaticky extrahovat klíčové informace potřebné pro generování otázek. Tento přístup také umožní vysokou míru automatizace a opakovatelnosti procesu generování otázek, což je výhodou při vytváření herního obsahu.

2.2 Zdroje dat

Existuje mnoho různých zdrojů dat, které bychom mohli použít. V úvahu připadají Wikipedie, Wikidata, Britannica atd. Nejvhodnější volbou pro naše potřeby bude Wikipedie, a to zejména díky následujícím vlastnostem.

- Rozsah - Wikipedie obsahuje obrovské množství informací o mnoha různých tématech, což nám umožní vytvořit dostatečné množství otázek pro hru Desítka.
- Dostupnost - Wikipedie je volně dostupná, což umožní snadný přístup k velkému množství informací.
- Jazyk - Wikipedie je napsána v češtině.
- Wikipedia API - Tato technologie umožňuje programátorům získat data z Wikipedie prostřednictvím speciálních dotazů. Jedná se o snadný a automatizovatelný způsob, jak získat tisíce článků a informací ke generování otázek. (4)

2.3 Získání dat

Ke generování otázek nemůžeme použít celou Wikipedii. Problémem je její velikost. Pokud bychom se snažili pracovat s každým článkem na Wikipedii, dostali bychom se k enormnímu množství informací. Ty by znamenaly obrovskou zátěž pro jakýkoliv algoritmus ke generování otázek. Zaměříme se proto pouze na nejnavštěvovanější stránky na Wikipedii. Tím snížíme množství článků, s nimiž budeme pracovat, a zároveň budeme vytvářet otázky z témat, která jsou všeobecně známá a populární mezi lidmi.

2.3.1 Nejnavštěvovanější stránky

Pro nalezení nejnavštěvovanějších stránek na české wikipedii použijeme standardizované API poskytované společností Wikimedia Foundation. Konkrétně využijeme následující **endpoint**.

Tento endpoint poskytuje seznam přibližně 1000 nejnavštěvovanějších článků na české Wikipedii pro daný den. Pro získání dat z konkrétního dne stačí do URL adresy endpointu přidat datum ve formátu YYYY/MM/DD, například pro den 2023/05/01 použijeme toto **URL**.

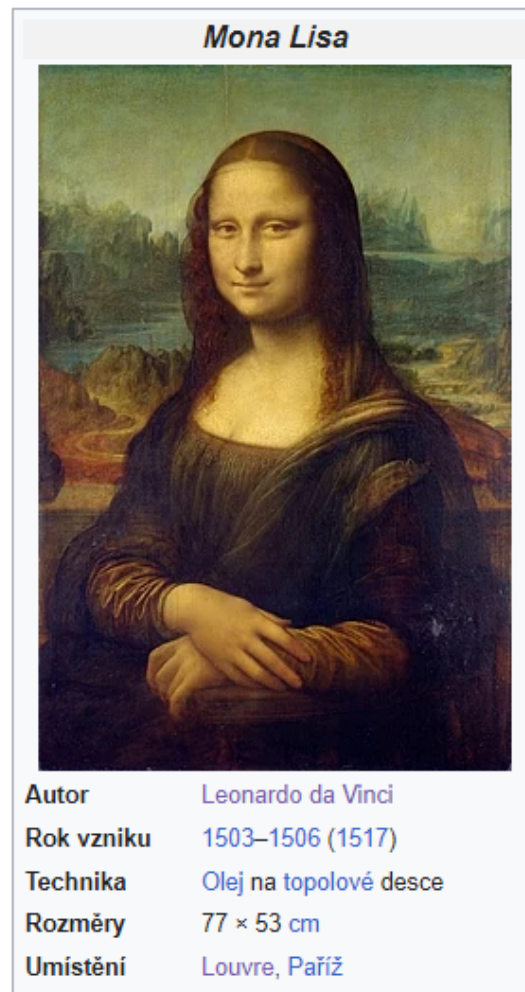
Tento přístup umožňuje rychlé a efektivní získání relevantních dat pro generování otázek. My jsme jej využili pro nalezení množiny nejnavštěvovanějších dat v dubnu v roce 2023, tedy mezi 2022/04/01 až 2023/05/01.

2.3.2 Parsování stránek

Pro přístup k jednotlivým článkům na české Wikipedii použijeme API rozhraní, které umožňuje získat obsah článku ve formátu wiki textu. Konkrétně využijeme následující **endpoint**, doplněný o název stránky (např. **Anglie**).

Pro získání relevantních informací budeme pracovat pouze s články, které obsahují Infobox tabulku. Pro získání relevantnějších dat zavedeme ještě další omezení. Aby byla dvojice klíč-hodnota naparsována, musí hodnota splňovat 2 následující podmínky:

- Hodnota odkazuje na právě jeden článek na Wikipedii
- Hodnota obsahuje maximálně tři slova



Obrázek 2.1: Ukázka infoboxu

- V tomto případě bude naparsována pouze dvojice autor: Leonardo da Vinci. Ostatní dvojice nesplňují parsovací podmínky.

Díky omezením získáme hodnoty, které jsou obecně známé, neboť mají vlastní stránku na Wikipedii. Zároveň mají hodnoty maximální délku tří slov a jsou tedy vhodné jako odpovědi na otázky.

2.4 Vytvoření otázek

Otázky vytvoříme pro taková data, jejichž klíč se vyskytne v alespoň deseti naparsovaných dvojicích. Pokud naparsujeme 10 států a každý Infobox u státu obsahuje jako klíč hlavní město a k němu validní hodnotu, vytvoříme otázku: „Jaké je hlavní město státu?“. Máme-li více naparsovaných dvojic se stejným klíčem, vytvoříme otázku pro každých 10 dvojic.

2.5 Kontrola uživatelem

Nakonec musíme vytvořené otázky manuálně zkontrolovat. Námi zkontrolované otázky jsou potom použity ve hře.

3. Analýza hry

V této části se zaměříme na to, jakým způsobem implementovat mobilní hru Desítka.

3.1 Programovací jazyk

K vývoji aplikací pro Android lze použít několik programovacích jazyků. Mezi nejpoužívanější jazyky patří Java, Kotlin a C++.

3.1.1 Java

Java byla v minulosti primárním jazykem pro vývoj aplikací pro Android. Tento jazyk je robustní, osvědčený a disponuje širokou podporou a mnoha knihovnamí, což umožňuje snadnou integraci s dalšími nástroji a systémy. V posledních letech se však začal stále více prosazovat jazyk Kotlin.

3.1.2 Kotlin

Kotlin je moderní programovací jazyk, který byl vyvinut společností JetBrains a je oficiálně podporován společností Google. Jedním z hlavních důvodů, proč se Kotlin stal populárním jazykem pro vývoj aplikací pro Android, je jeho schopnost snižovat množství opakujícího se kódu a zlepšovat bezpečnost aplikace díky vysokému typovému systému. Kotlin také nabízí mnoho moderních funkcí a knihoven, což usnadňuje práci s jazykem a umožňuje rychlý vývoj aplikací.

3.1.3 C++

C++ je další programovací jazyk, který se dá použít pro vývoj aplikací pro Android. Tento jazyk se často používá pro tvorbu výkonných aplikací a her, protože umožňuje přímý přístup k paměti a hardwaru zařízení. Nicméně, C++ vyžaduje více práce a má vyšší nároky na výkon a paměť, což může způsobit pomalejší vývoj a větší složitost kódu.

3.1.4 Zvolený jazyk

V našem případě si zvolíme Javu. Jedním z důvodů pro výběr Javy je, že se jedná o jeden z nejrozšířenějších a nejstabilnějších programovacích jazyků, který má rozsáhlou knihovnu pro vývoj mobilních aplikací. Java také disponuje silnou dokumentací a komunitou, která poskytuje podporu a pomáhá řešit případné problémy. Dalším důvodem je moje osobní zkušenost s programováním v Javě a vysoká úroveň kompatibility Javy s operačním systémem Android, která zajišťuje rychlý a efektivní vývoj aplikací.

3.2 Model architektury

Pro hraní hry musí být mobilní zařízení propojena a schopna vzájemně komunikovat. Bez komunikace by nebylo možné synchronizovat informace mezi hráči a přenášet data nutná pro hru. Existují dvě základní architektury, které se ke komunikaci používají. (7)

3.2.1 Peer-to-peer

Architektura peer-to-peer umožňuje přímou komunikaci mezi dvěma a více zařízeními bez nutnosti centrálního serveru. Každé zařízení může být současně klientem i serverem. Tento model se využívá například u aplikací pro sdílení souborů, kdy jednotliví uživatelé tvoří síť a mohou si mezi sebou přímo vyměňovat data bez nutnosti účasti centrálního serveru. Peer-to-peer architektura má výhodu v rychlosti a úspornosti při komunikaci mezi menším počtem zařízení, ale může být náročná na koordinaci větších sítí.

3.2.2 Client-server

Architektura client-server zahrnuje rozdělení aplikace do dvou částí. Jednou je klient, který požaduje informace. Druhou je server, který odpovídá na požadavky a poskytuje data. Komunikace mezi klientem a serverem probíhá pomocí sítě, často přes internet. Tento model se používá u větších aplikací, kde je potřeba zpracovat velké množství dat a udržovat je na jednom místě. Tím je zajištěna konzistence dat a usnadňuje se jejich správa. Architektura client-server má výhodu v robustnosti a spolehlivosti při práci s větším množstvím dat, ale může být náročná na síťovou infrastrukturu a správu serverového prostředí.

3.2.3 Zvolená architektura

V naší aplikaci si zvolíme architekturu client-server, díky čemuž můžeme vygenerované otázky uchovávat pouze na serveru. Klientům jsou následně otázky posílány po jedné, vždy když začne nové kolo hry. Tento přístup výrazně sníží velikost aplikace, protože není potřeba ukládat otázky na zařízení hráčů. Zároveň tím zajistíme konzistenci dat, protože všichni hráči budou mít přístup ke stejným otázkám.

3.3 Komunikace

Pro komunikaci mezi zařízeními a serverem využijeme internet. Nabízí se několik knihoven, které můžeme ke komunikaci využít. Mezi nejlepší z nich patří následující dvě.

3.3.1 Java Socket

Java Socket je knihovna pro síťové programování v jazyce Java. Tato knihovna umožňuje komunikaci mezi aplikacemi pomocí soketů, což jsou programové roz-

hraní pro síťové spojení. Java Socket podporuje jak TCP, tak UDP protokoly a poskytuje jednoduché API pro vytváření serverů a klientů. (1)

3.3.2 GRPC

GRPC je moderní open-source knihovna pro vzdálené volání procedur (RPC) vyvinutá společností Google. Tato knihovna podporuje více jazyků včetně Javy. Je postaven na protokolu HTTP/2 a používá binární formát pro přenos dat, což umožňuje rychlou a efektivní komunikaci. (2)

3.3.3 Zvolená knihovna

My použijeme knihovnu Java Socket. Ta umožňuje obousměrnou komunikaci mezi klientem a serverem, což je pro náš účel klíčové. Dále Java Socket poskytuje dobře zdokumentované API a širokou podporu vývojářské komunity.

3.4 Požadavky na hru

V této části se zaměříme na to, co od naší hry očekáváme. Také se podíváme na to, jakým způsobem server řídí hru a za které herní události je zodpovědný klient. V následujícím textu jsou pojmy hráč a klient volně zaměňovány.

3.4.1 Režimy hry a připojení

Naše aplikace podporuje dva herní režimy. Hráči mohou hrát s ostatními hráči, nebo s přáteli. Hra s ostatními hráči znamená, že hráč hraje s náhodně vybraným protihráči. Hra s přáteli znamená, že hráč vyzve ke hře své přátele, kterým pošle kód pro připojení do hry. Klient je zodpovědný za navázání komunikace se serverem a posílání požadavku pro připojení se ke hře. Server má na starost ověření požadavku a poslání zprávy o úspěšném/neúspěšném připojení zpět klientovi.

3.4.2 Zahájení hry

Zahájení hry má na starost server, který hru zahájí v okamžiku, kdy se k ní připojí poslední hráč. Není-li hra zahájena do 60 sekund od jejího vytvoření, je zrušena.

3.4.3 Hraní hry

Hra se skládá z jednotlivých kol. Průběh kola řídí server. Každé kolo můžeme rozdělit do 3 fází.

Příprava kola

Server vybere otázku a pošle ji všem hráčům v kole včetně správných odpovědí. Server také vybere začínajícího hráče. První kolo začíná hráč, který se ke hře připojil jako první. Hráči se v hraní střídají podle pořadí, ve kterém se ke hře připojili.

Hraní kola

Server pošle všem hráčům jméno hráče, který je na tahu. Hráč, který je na tahu potom pošle svoji odpověď na otázku serveru. Server odpověď vyhodnotí. Je-li odpověď špatná, hráč vypadává pro dané kolo ze hry a ztrácí všechny body v kole získané. V případě správné odpovědi zůstává hráč v daném kole a získává bod. Hráč se také může rozhodnout pasovat, v takovém případě vypadává ze hry, ale všechny body v kole získané mu zůstávají. Server nejprve pošle všem hráčům ve hře obdrženou odpověď a potom jméno dalšího hráče, který je na tahu.

Vyhodnocení kola

Kolo končí, když v něm nezbyde žádný hráč, nebo je-li zodpovězeno všech 10 otázek. Hráčům jsou zobrazeny správné odpovědi na otázky a následně průběžné skóre hry. Skóre hry má na starost server. Na konci každého kola je skóre posláno všem hráčům.

3.4.4 Konec hry

Hra končí, jakmile některý z hráčů získá 20 bodů, nebo zůstane-li ve hře méně než 1 hráč. Na konci hry jsou hráčům zobrazeny výsledky. Hráčům je také nabídnuta možnost zahrát si hru znovu. To se hodí obzvláště pro hry v režimu friend, neboť hráči nemusí znovu zadávat kód hry.

3.5 Řešení technických problémů hráčů

Pro zajištění příjemného a plynulého herního zážitku hráčů, se musíme vypořádat s technickými problémy, které na straně hráčů můžou nastat. Řešením je zavedení časového limitu na zodpovězení otázky. Pokud server neobdrží do určitého časového limitu od hráče na tahu žádnou zprávu, předpokládá, že spojení s hráčem bylo ztraceno. Tento hráč je vyřazen ze hry.

Server se také musí umět vypořádat s neaktivními hráči. To jsou takoví hráči, kteří nechají aplikaci spuštěnou, ale aktivně se hry neúčastní. V takovém případě pošle klient po uplynutí časového limitu serveru zprávu o tom, že uživatel pasuje bez aktivního zapojení se do hry. Je-li tato zpráva poslána třikrát za sebou, je hráč vyřazen ze hry.

3.6 Synchronizace hráčů

Pro plynulý zážitek ze hry je také potřeba co nejlepší synchronizace hráčů. Na straně serveru je každému hráči přiřazeno samostatné vlákno. Synchronizace je potom dosaženo tak, že hra se posouvá do další fáze teprve v okamžiku, kdy je každé vlákno připraveno. K první synchronizaci dochází při zahájení hry. K dalším dochází vždy před zahájením a vyhodnocením kola.

Na straně klienta jsme synchronizace dosáhli následujícím způsobem. Hráč na tahu pošle zprávu o svém tahu serveru. Server ji obratem pošle všem hráčům včetně odpovídajícího hráče. Tím snížíme latenci mezi doručením zprávy s

odpovědí jednotlivým hráčům. Dále využijeme toho, že klient zná správnou odpověď na otázku, neboť jsme mu před zahájením kola poslali správné odpovědi. Správnost odpovědi můžeme tedy ověřit na straně klienta, čímž zjednodušíme komunikaci mezi klienty a serverem. (Správnost odpovědi musíme vyhodnocovat i na straně serveru. Jedním z důvodů je, že musíme vědět, kolik hráčů je ještě v kole.)

3.7 Testování hry

Hru otestujeme na lokálním serveru. K propojení serveru, který běží na notebooku, a mobilních telefonů s aplikací použijeme technologii ngrok.

3.7.1 Ngrok

Ngrok je nástroj, který umožňuje vytváření veřejných tunelů k aplikacím spuštěným na místním počítači. Funguje tak, že vytvoří spojení mezi veřejnou IP adresou a aplikací spuštěnou na lokálním počítači. Tento veřejný tunel je pak k dispozici pro ostatní uživatele. Aplikace spuštěná na místním počítači se tedy stane přístupnou pro veřejnost, aniž by bylo nutné ji nasazovat na veřejný server.

(3)

4. Uživatelská dokumentace

Pro správné fungování hry je potřeba mobilní telefon s androidem verze 10 a výše, internetové připojení a notebook pro spuštění lokálního serveru, s nímž hra komunikuje.

4.1 Menu hry

Po spuštění hry se uživateli zobrazí domovská obrazovka. Tlačítko *Hrát online* je určeno pro hru s náhodně vybranými hráči. Po kliknutí na něj je uživateli zobrazena obrazovka s informacemi o vytvářené hře. Uživatel vidí jména hráčů, kteří se již připojili. Dále také na spodní obrazovce vidí čas do zahájení hry. Pokud se hra nenaplní, než vyprší čas do jejího zahájení, je zrušena a uživatel je přesměrován zpět na domovskou obrazovku.



Obrázek 4.1: Úvodní obrazovka

Kliknutím na tlačítko *Hrát s přáteli* je uživateli zobrazena obrazovka pro hru s přáteli se dvěma tlačítky.

- *Vytvořit hru* - Po stisknutí tohoto tlačítka se uživateli zobrazí obrazovka, kde si nastaví parametry hry. Uživatel zvolí počet hráčů, kteří budou hru hrát (2-5) a kliknutím na tlačítko *Vytvořit* vytvoří hru. Následně je uživateli zobrazena obrazovka s informacemi o vytvářené hře. Ty jsou stejné jako u hry s náhodně vybranými hráči, tedy jména již připojených hráčů a čas do zahájení hry. Navíc ale uživatel vidí i kód hry, který pošle svým přátelům.
- *Připojit se ke hře* - Kliknutím na toto tlačítko se uživateli zobrazí obrazovka vyžadující osmimístný kód hry. Ten byl vygenerován serverem při vytvoření hry a skládá se z malých písmen latinské abecedy. Zadá-li uživatel správný kód hry, je připojen ke hře a je mu zobrazena obrazovka s informacemi o vytvářené hře. Pokud je kód hry neplatný, uživatel je o této skutečnosti informován a hra ho přesměruje na domovskou obrazovku.



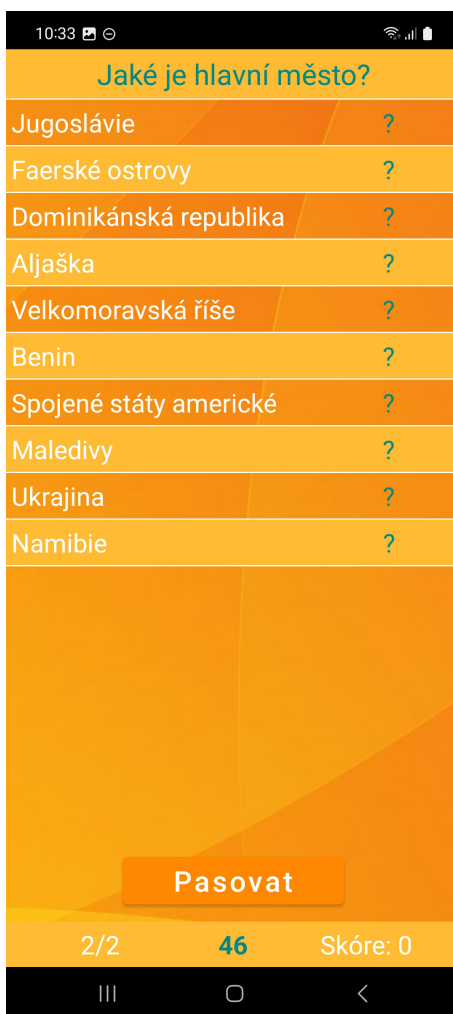
Obrázek 4.2: Příprava hry

V horní části obrazovky se nachází navigační menu aplikace. Uprostřed vidí uživatel své jméno. K jeho zadání je vyzván při prvním spuštění aplikace. V levém horním rohu je ikonka domečku. Po jejím stisknutí je uživatel přesměrován zpět na domovskou obrazovku. Po kliknutí na ikonku v pravém horním rohu je uživateli zobrazeno vyskakovací menu se 2 položkami.

- *Změnit jméno* - Po stisknutí se uživateli zobrazí obrazovka pro změnu jména hráče.
- *O hře* - Po stisknutí je uživatel přesměrován na webovou stránku se základními informacemi o hře.

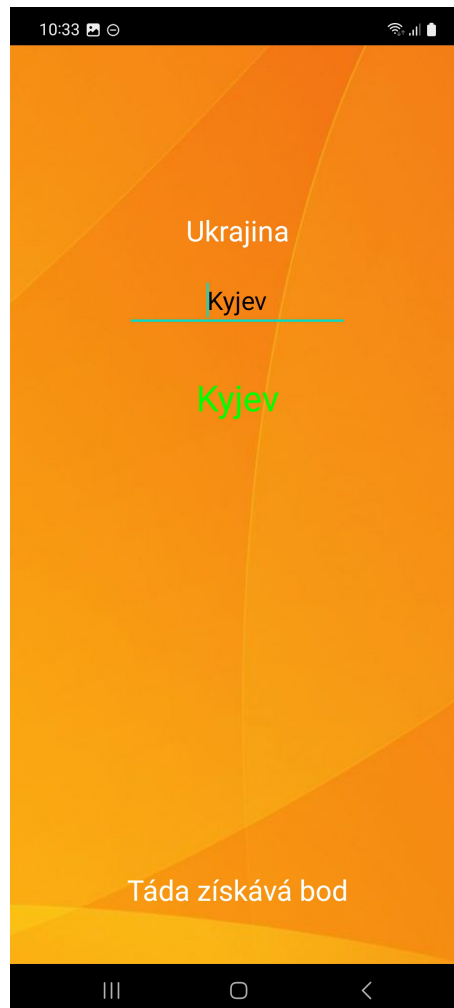
4.2 Hraní hry

Hra začíná v okamžiku, kdy se k ní připojí poslední hráč. Hra určí, který hráč je na tahu. Po 5 vteřinách je potom hráčům zobrazena otázka s jednotlivými podotázkami. V levé části dolní obrazovky vidí hráči počet hráčů, kteří jsou ještě aktivní v kole. V pravé části dolní obrazovky vidí hráč své aktuální skóre. Uprostřed dolní části obrazovky běží časový limit. Hrající hráč má 50 vteřin na odpověď. Odpoví tak, že klikne na otazník u vybrané otázky a odpověď napíše do textového pole, jež se mu zobrazí. Pokud chce hráč pasovat, klikne na tlačítko *Pasovat*. Klikne-li na tlačítko pasovat nebo na otazník u otázky hráč, jenž není na tahu, nedochází k žádné akci. Pokud hráč do 50 vteřin neodpoví na otázku a zároveň ani nepasuje, je serveru poslána zpráva s tím, že hráč pasuje, ale byl neaktivní. Je-li hráč neaktivní třikrát za sebou, je vyřazen ze hry.



Obrázek 4.3: Ukázka otázky

Hráčova odpověď je zobrazena všem hráčům ve hře. Pokud hráč pasuje, zobrazí se hráčům obrazovka s informací o tom, že hráč pasuje. Pokud hráč odpovídá na otázku, zobrazí se hráčům obrazovka s příslušnou otázkou a hráčova odpověď. Je-li odpověď správná, text se zbarví do zelena. Při špatné odpovědi je text červený.



Obrázek 4.4: Správná odpověď

Kolo končí vyřazením všech hráčů, nebo zodpovězením všech otázek. V této fázi hra zobrazí uživatelům správné odpovědi na jednotlivé otázky. Následně je hráčům zobrazeno průběžné skóre hry. Hráči vidí svůj bodový zisk a zároveň bodový zisk ostatních hráčů. Hra končí v okamžiku, kdy jeden z hráčů získá 20 bodů, nebo zůstane-li ve hře jeden a méně hráčů. V tomto okamžiku se hráčům zobrazí 2 tlačítka.

- *Hrát znovu* - Po stisknutí se uživateli zobrazí obrazovka s informacemi o vytvářené hře. Hrál-li uživatel hru s náhodně vybranými hráči, je hráč přidán do nové hry proti náhodně vybraným hráčům. Pokud hrál uživatel hru s přáteli, čeká na připojení přátel, s nimiž právě dohrál. Pokud se hráči nepřipojí do časového limitu pro zahájení hry, je hra zrušena a hráč je přeměřován na úvodní obrazovku
- *O hře* - Po stisknutí je uživatel přeměřován na domovskou obrazovku

Ztratí-li hráč během hraní internetové připojení, je ze hry odebrán a je přeměrován na domovskou obrazovku.



Obrázek 4.5: Konec hry

4.3 Spuštění aplikace

Pro hraní hry je potřeba udělat 3 základní kroky. Musíme spustit server na našem počítači. Následně musíme použít nástroj ngrok, abychom vytvořili veřejný tunel k serveru spuštěnému na našem počítači. Tento veřejný tunel použijeme v naší mobilní aplikaci pro komunikaci se serverem. Na následujícím odkazu je video o tom, jak aplikaci spustit.

4.3.1 Server

Server je maven projekt v adresáři Server. Stačí projekt otevřít v nějakém IDE a spustit. Projekt lze spustit také pomocí příkazové řádky z adresáře Server zadáním příkazu:

```
mvn compile exec:java -Dexec.mainClass="desitka.server.Main"
```

4.3.2 Ngrok

Nejprve si musíme vytvořit účet na **ngrok**. Účet je zcela zdarma. Následováním pokynů na webové stránce stáhneme nástroj ngrok. Potom již stačí v terminálu zadat příkaz **ngrok tcp 4444**.

Tím namapujeme náš lokální port 4444, na kterém běží náš server, na novou adresu. Tu vytvoří nástroj ngrok a zobrazí se v příkazové řádce. My adresu použijeme v naší mobilní aplikaci pro komunikaci se serverem. Adresa může vypadat například takto: **tcp://0.tcp.eu.ngrok.io:16497**.

4.3.3 Mobilní aplikace

V mobilní aplikaci musíme změnit adresu Socketu, kterou zadáváme při vytváření nového klienta pro komunikaci se serverem. Konkrétně se jedná o řádek 1615 v souboru Desitka.java, který se nachází v adresáři Desitka. Má-li hodnota forwarding na příkazové řádce při použití nástroje ngrok například výše uvedenou hodnotu, upravíme kód na řádce 1615 takto:

```
Socket client = new Socket("0.tcp.eu.ngrok.io", 16497)
```

Nyní je třeba aplikaci nahrát na náš mobilní telefon nebo na virtuální zařízení.

- Mobilní zařízení
 - V nastavení našeho telefonu klikneme na možnost *O telefonu*
 - Několikrát klikneme na *Verze softwaru* nebo *Číslo sestavení*
 - Po pár kliknutích se zobrazí informace o tom, že jsme se stali vývojářem
 - Vrátime se do nastavení a klikneme na *Vývojářské možnosti*
 - Zapneme možnost ladit přes USB
 - Nahrajeme aplikaci do telefonu s použitím Android Studia a USB kabelu
- Virtuální zařízení
 - V Android Studiu rozklikneme připojená zařízení nalevo od šipky spustit a klikneme na Device Manager
 - Klikneme na Create Device a nastavíme si zařízení
 - Aplikaci spustíme na virtuálním zařízení

4.4 Vytvoření otázek pro hru

Pokud chceme vytvořit nové otázky pro hru, postupujeme následovně. Nejprve doporučujeme vymazat všechny již vytvořené soubory. Potom spustíme program v adresáři QuestionGenerator. Nechceme-li používat IDE, můžeme spustit program použitím příkazové řádky z adresáře QuestionGenerator.

```
mvn compile exec:java -Dexec.mainClass="desitka.generator.Main"
```

Chceme-li vygenerovat otázky pro jiné časové období než 1.4.2023 - 1.5.2023, musíme ve třídě Main změnit počáteční a koncové datum. Jakmile program skončí, vytvoří následující soubory:

- loaded-titles - obsahuje množinu nejnavštěvovanějších článků na wikipedie v daném časové rozmezí. (Např. titul Francie)
- loaded-articles - tento adresář obsahuje články s narpárovanými dvojicemi. Dvojice klíč-hodnota splňují parsovací podmínky.

1	motto	Liberté, Égalité, Fraternité
2	hlavní město	Paříž
3	nejvyšší hora	Mont Blanc
4	obyvatelé aktuální k	2022
5	státní zřízení	poloprezidentská republika
6	vládce1	Emmanuel Macron
7	vládce2	Élisabeth Borneová

Obrázek 4.6: Načtené dvojice pro titul Francie

- loaded-data - tento adresář obsahuje csv soubory, které mají stejný klíč. Klíč je použit jako název souboru. (Např. soubor s názvem **nejvyssi_hora** bude obsahovat dvojici **Francie - Mont Blanc**)
- questions - tento adresář obsahuje vytvořené otázky ve formátu JSON

Vygenerované otázky musíme nyní projít a zkontrolovat. Nutnost kontroly můžeme vidět na získaných dvojicích pro titul Francie. Čtvrtý řádek tabulky říká, k jakému roku je platný údaj o počtu obyvatel, ale neříká již nic o samotném počtu. Tuto otázku určitě použít nechceme. Dále si můžeme všimnout, že pojem vládce1 označuje prezidenta a vládce2 je myšlen premiér.

Načtené dvojice v adresáři loaded-data můžeme využít k vytvoření dalších otázek, nebo můžeme dvojice upravit a otázky vygenerovat znovu. Nami vybrané otázky potom musíme přemístit do adresáře Server/questions. Otázky v tomto adresáři používá náš server při hře.

5. Programátorská dokumentace

Naše aplikace se skládá ze 3 částí. Máme serverovou a klientskou část aplikace. Poslední částí je poté program pro vygenerování otázek ke hře. V této části si stručně představíme všechny 3 části. Detailní popis všech tříd vygenerovaný pomocí Javadoc je součástí přílohy.

5.1 Klient

V našem případě je klientem mobilní aplikace pro android napsaná v jazyce Java. Ta se nachází v adresáři Desítka. Hlavní úkolem klienta je navázání spojení se serverem a následná interakce s ním. Ke komunikaci používáme knihovnu Java Socket.

Aplikace využívá různé layouty. Přepínání mezi nimi a interakci s herním prostředím má na starost třída Desítka.java. Desítka.java je hlavní řídicí třídou aplikace. Stará se o správu uživatelského rozhraní, komunikaci se serverem a řízení herní logiky na straně uživatele. Při zahájení hry je vytvořeno nové vlákno (thread), které má na starosti komunikaci se serverem. V důsledku toho máme poté v aplikaci dvě vlákna - jedno pro uživatelské rozhraní a druhé pro komunikaci se serverem.

5.2 Server

Serverová část aplikace Desítka je implementována jako Maven projekt napsaný v jazyce Java. Nejdůležitějšími komponentami serverové části jsou třídy Server.java a Game.java.

- Server.java - Tato třída má na starost komunikaci s klienty. Jakmile je přijato připojení od klienta, Server.java vytváří nové vlákno, které je odpovědné za obsluhu komunikace s daným klientem.
- Game.java - Tato třída má na starost samotnou logiku hry. Je také použita k synchronizaci hráčů. Hra například nezahájí další kolo, dokud každé vlákno, které reprezentuje klienta ve hře, nepotvrdí, že je připraveno na nové kolo.

5.3 Komunikace mezi klientem a serverem

Pro hraní hry naváže klient spojení se serverem. Klient pošle serveru nejprve **jméno hráče** a potom jednu z následujících zpráv:

- **onlineGame** - Hráč chce hrát hru s náhodně vybranými hráči
- **createGame** - Hráč chce vytvořit novou hru pro přátele
- **joinGame** - Hráč se chce připojit ke hře s přáteli, v takovém případě musí serveru poslat i **kód hry**, ke které se chce připojit

Server zprávu vyhodnotí, pokud chce hráč hrát hru s náhodně vybranými hráči nebo pokud chce založit hru, je mu vyhověno. Chce-li se hráč připojit ke hře, dostane jednu ze 3 následujících odpovědí:

- **joined** - Úspěšně připojen
- **nonExistingGame** - Hra se zadaným kódem neexistuje
- **nameAlreadyJoined** - Hráč s tímto jménem již ve hře je

Obdrží-li aplikace druhou nebo třetí zprávu vytvoří Alert Dialog a informuje uživatele o chybě. Je-li připojení úspěšné, pošle server nejprve všem dříve připojeným hráčům jméno nově připojeného hráče. Potom server pošle klientovi zprávu s časem, kdy byla hra vytvořena a jména všech hráčů ve hře. Hra je zahájena, pokud se naplní do 60 vteřin od jejího vytvoření. V opačném případě je hra zrušena.

Hra se skládá z jednotlivých kol. Každé kolo má stejný průběh. Server pošle hráčům otázku pro dané kolo a jméno hráče, který je na tahu. Hrající hráč potom pošle serveru id otázky, na kterou odpovídá a svoji odpověď. Pokud se hráč rozhodl pasovat, pošle zprávu **pass** a na id otázky nezáleží. Je-li hráč neaktivní, obdrží server zprávu **noActivity**, na id otázky opět nezáleží. Server přešle obdrženou zprávu všem hráčům ve hře. Následně server pošle jméno dalšího hráče, který je na tahu.

Na konci kola pošle server hráčům zprávu **roundFinished**. Potom server pošle hráčům zprávy v následujícím pořadí:

- počet hráčů ve hře
- Informaci o tom, zda hra již skončila **true/false**
- **jméno** a **skóre** hráče - Tato zpráva se opakuje pro každého hráče ve hře. Hráči jsou seřazeni podle aktuálního skóre od nejvyššího skóre k nejnižšímu.

V případě konce hry ukončí server spojení s klientem. Pro hraní další hry je navázáno nové spojení. To platí i pro odvetné hry.

5.4 Question Generator

Ke generování otázek používáme Maven projekt napsaný v jazyce Java. Generování otázek se skládá ze 4 fází. Ke každé fázi je použita jedna z následujících tříd.

- **WikiParser.java** - Tato třída je zodpovědná za získání názvů nejnavštěvovanějších článků za dané časové období. K tomu využívá Wikipedia API a knihovnu Jsoup. Názvy nejnavštěvovanějších článků jsou potom nahrány do souboru loaded-titles.
- **WikiReader.java** - Tato třída má na starost získání dvojic klíč-hodnota z Infoboxu u nejnavštěvovanějších článků. Připomeňme, že pro parsování dvojic jsme si zavedli 2 podmínky.

- Hodnota má nejvýše 3 slova
- Hodnota je odkazem na právě 1 stránku na Wikipedii.

První podmínku ověříme snadno. Druhá podmínka při použití Wikipedia API znamená, že hodnota musí být ohraničena dvěma hranatými závorkami:

```
{{Infobox - obraz
| název = Mona Lisa
| obrázek = Mona Lisa, by Leonardo da Vinci, from C2RMF
retouched.jpg
| velikost obrázku = 250px
| autor = [[Leonardo da Vinci]]
| rok vzniku = [[1503]]-[[1506]] ([[1517]])
| rozměry = 77 x 53 [[metr|cm]]
| technika = [[olejomalba|Olej]] na [[topol]]ové desce
| umístění = [[Louvre]], [[Paříž]]
}}
```

Obrázek 5.1: Infobox získaný pomocí Wikipedia API

Pro výše uvedený případ bude neparsována pouze dvojice autor: Leonardo da Vinci. V adresáři loaded-articles program potom vytvoří soubor Mona-Lisa.csv s touto dvojicí.

- **QuestionParser.java** - Tato třída seřadí neparsované dvojice podle společného klíče. Najde-li program 10 a více společných klíčů, je z nich vytvořena otázka do hry. Zároveň je v adresáři loaded-data vytvořen csv soubor se jménem klíče. Ten obsahuje dvojice **původní název článku: hodnota**. V případě Mona Lisy by se nový soubor jmenoval autor.csv a obsahoval by dvojici **Mona Lisa: Leonardo da Vinci**.
- **QuestionCreator.java** - Tato třída slouží k vytvoření otázek ve formátu JSON a jejich uložení do adresáře questions. Nabízí 2 statické metody. Jednu pro vytvoření otevřené otázky. Druhou k vytvoření otázky na seřazení.

5.5 Aktuální nastavení hry

Hru se nám bohužel nepodařilo umístit na veřejný server. Kvůli tomu jsme ji nechali v testovací verzi se změněným pravidlem pro konec hry. To je nastaveno tak, že hra končí v okamžiku, kdy jeden z hráčů získá 2 body. Toto nastavení můžeme změnit v adresáři Server v souboru Game.java úpravou proměnné MAX-SCORE.

Jelikož se hra hraje pouze do 2 bodů, používá aktuálně pouze 15 otázek. Tyto otázky jsme manuálně vybrali a zkopírovali z adresáře QuestionGenerator/questions do adresáře Server/questions. Otázky jsme záměrně nijak neupravovali, abychom ukázali, že některé z nich nepotřebují žádný zásah uživatele. Uživatel si samozřejmě může do hry přidat další otázky.

Závěr

V rámci této bakalářské práce byla úspěšně implementována mobilní verze společenské vědomostní hry Desítka pro zařízení s operačním systémem Android verze 10 a výše. Hru lze hrát s náhodně vybranými hráči i s přáteli. Vzhledem k dosavadní absenci online verze této hry nabízí tato práce jedinečnou možnost, jak si zahrát oblíbenou hru na dovolené s přáteli.

Druhým cílem práce bylo vytvořit otázky, které budou ve hře použity. Tento cíl se také podařilo splnit. Z nejnavštěvovanějších článků na Wikipedii za měsíc duben jsme vygenerovali přes 400 otázek, z nich je více než polovina vhodná pro hru. Zajímavostí také je, že se námi vygenerované otázky poměrně liší od otázek v deskové hře Desítka. Povedlo se nám tedy vytvořit nové otázky ke hře, což považuji za úspěch.

Možná rozšíření práce

Práce by si určitě zasloužila umístění na veřejný server, což bylo také v plánu. Bohužel, při bližším zkoumání se ukázalo, že veřejné servery s dostatečnou kapacitou a požadovanými funkcemi jsou často spojeny s určitými náklady, takže od tohoto záměru bylo nakonec upuštěno.

Ke generování otázek jsme použili 2 parsovací podmínky. Jejich výhodou bylo, že byly poměrně přímočaré, ale přesto vedly k narpárování dostatečného množství dat. Určitě by však mohly být nějak upraveny, abychom zajistili ještě pestřejší a rozmanitější výběr otázek. Popřípadě by otázky mohly být generovány jinou cestou, například pomocí strojového učení.

Seznam použité literatury

- [1] BAELDUNG (2023). Baeldung.com. URL <https://www.baeldung.com/a-guide-to-java-sockets>.
- [2] GRPC AUTHORS (2023). grpc.io. URL <https://grpc.io/>.
- [3] JARUS, N. (2023). ngrok.com. URL <https://ngrok.com/docs/secure-tunnels/>.
- [4] MEDIAWIKI (2023). mediawiki.org. URL https://www.mediawiki.org/wiki/API:Main_page.
- [5] MINDOK (2018). mindok.cz. URL <https://mindok.cz/hra/desitka/>.
- [6] MINDOK (2018). mindok.cz. URL https://www.mindok.cz/userfiles/files/pravidla/desitka_pravidla_web.pdf.
- [7] UNKNOWN (2017). techdifferences.com. URL <https://techdifferences.com/difference-between-client-server-and-peer-to-peer-network.html>.

Seznam obrázků

1.1	Ukázková otázka	4
2.1	Ukázka infoboxu	7
4.1	Úvodní obrazovka	14
4.2	Příprava hry	15
4.3	Ukázka otázky	16
4.4	Správná odpověď	17
4.5	Konec hry	18
4.6	Načtené dvojice pro titul Francie	20
5.1	Infobox získaný pomocí Wikipedia API	23

A. Přílohy

A.1 Obsah elektronické přílohy

- Tato práce ve formátu PDF/A
- Adresář Desitka
 - Zdrojový kód
 - Dokumentace Javadoc
- Adresář Server
 - Zdrojový kód
 - Dokumentace Javadoc
 - Otázky pro hru
- Adresář QuestionGenerator
 - Naparsované články z Wikipedie
 - Vygenerované otázky
 - Zdrojový kód
 - Dokumentace Javadoc