

Česká zemědělská univerzita v Praze

Technická fakulta

Katedra technologických zařízení staveb



Bakalářská práce

Analýza cloudových řešení pro realizaci multiplayer hry

Jakub Hampejs

© 2022 ČZU v Praze

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Jakub Hampejs

Informační a řídicí technika v agropotravinářském komplexu

Název práce

Analýza cloudových řešení pro realizaci multiplayer hry

Název anglicky

Analysis of cloud solutions for the implementation of multiplayer games

Cíle práce

Cílem práce je analyzovat stávající nabízená cloud řešení pro tvorbu aplikace s víceuživatelským přístupem a větším datovým tokem (např. multiplayer hra). V rámci analýzy bude provedeno vyhodnocení nástrojů pro tvorbu cloudového IS na každém konkrétním typu. Závěry budou shrnuty a přehledně vyhodnoceny, včetně demonstrativní (testující) aplikace.

Metodika

1. Úvod
2. Cíl práce a metodika
3. Rozdělení a popis komerčních cloud řešení
4. Nástroje pro tvorbu IS a databází nad cloud
5. Výběr a analýza hry pro demonstraci
6. Pravidla a vazby
7. Návrh konkrétního řešení včetně ověření vícenásobného přístupu
8. Vyhodnocení a doporučení
9. Závěr

Doporučený rozsah práce

30 až 40 stran textu včetně obrázků, grafů a tabulek

Klíčová slova

cloud, bezpečnost, sdílení

Doporučené zdroje informací

Casey Hardman: Game Programming with Unity and C# : A Complete Beginner's Guide, ISBN10 1484256557, aPress, 2020

Egges Arjan: Learning C# by Programming Games, ISBN13 (EAN): 9783642365799, Springer, 2013

Gregory Jason: Game Engine Architecture, Third Edition, ISBN13 (EAN): 9781138035454, Taylor&Frances, 2018

JIROVSKÝ, V. Vademecum správce sítě. Praha: Grada, 2001. ISBN 80-7169-745-1.

SOSINSKI, B.: Mistrovství – počítačové sítě, Computer Press. 2010, ISBN 978-80- 251-3363-7

Předběžný termín obhajoby

2021/2022 LS – TF

Vedoucí práce

Ing. Zdeněk Votruba, Ph.D.

Garantující pracoviště

Katedra technologických zařízení staveb

Elektronicky schváleno dne 3. 2. 2021

doc. Ing. Jan Maláták, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 10. 2. 2021

doc. Ing. Jiří Mašek, Ph.D.

Děkan

V Praze dne 31. 03. 2022

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Analýza cloudových řešení pro realizaci multiplayer hry" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 31.3.2022

Poděkování

Rád bych touto cestou poděkoval Ing. Zdeňkovi Votrubovi, Ph.D. za konzultace a cenné rady.

Analýza cloudových řešení pro realizaci multiplayer hry

Abstrakt

Práce se zaměřuje na služby cloud computingu a jejich poskytovatele. Popisuje distribuční modely a modely nasazení. Poté uvádí cloudové leadery a srovnává jejich aktuálně nejvýhodnější bezplatné nabídky na pronájem virtuálních strojů. Seznamuje s běžně používanými typy digitálních databází a zaměřuje se na jejich cloudové poskytovatele a rovněž srovnává jejich bezplatné nabídky. Jsou představeny nejpopulárnější nástroje pro správu relačních databází. Pozornost je také kladena na síťovou komunikaci. Především na komunikační síťové modely OSI a TCP/IP, které definují pravidla, podle kterých jsou data přenášena. Kapitola je zakončena popisem základních metod z knihoven, které se využívají pro síťovou komunikaci v programovacím jazyce C#. Následuje praktická část v podobě představení demonstrativní aplikace. Po ní pokračuje programová část, ve které je popsán použitý herní framework a návrhový vzor pro klientskou část aplikace a způsob obsluhy požadavků od klientů na serverové části aplikace. Na závěr práce jsou s pomocí demonstrativní aplikace provedeny testy, které zjišťují a následně analyzují časovou odezvu dvou nabídek poskytovatelů cloudových služeb.

Klíčová slova: Cloud computing, cloudové služby, sdílení, C#, MonoGame, multithreading, síťová komunikace, databáze

Analysis of cloud solutions for the implementation of multiplayer game

Abstract

Bachelor thesis deals with services of cloud computing and their providers. It describes the distribution models and the deployment models. There are introduced cloud leaders and the comparison of their current best free virtual machine rental offers. Introduces commonly used types of digital databases and focuses on their cloud providers, also comparing their free offerings. The most popular relation database management tools are introduced. Attention is also given to network communications. In particular, the OSI and TCP/IP network communication models, which define the rules according to which data is transferred. The chapter concludes with a description of basic methods from libraries that are used for network communication in the C# programming language. This is followed by a practical section in the form of a demonstration application. The programming part continues with a description of the game framework and design pattern used for the client part of the application and how to handle requests from clients on the server part of the application. At the end of the bachelor thesis, tests are performed using the demonstration application to detect and then analyze the time response of two cloud service providers' offerings.

Keywords: Cloud computing, cloud services, sharing, C#, MonoGame, multithreading, network communication, databases

Obsah

1	Úvod.....	1
2	Cíl práce a metodika	2
3	Cloud Computing	2
3.1	Definice	2
3.2	Distribuční modely.....	3
3.2.1	Infrastruktura jako služba (IaaS).....	4
3.2.2	Platforma jako služba (PaaS).....	4
3.2.3	Software jako služba (SaaS).....	5
3.3	Modely nasazení.....	6
3.3.1	Veřejný model nasazení.....	6
3.3.2	Privátní model nasazení.....	6
3.3.3	Hybridní model nasazení.....	6
3.4	Poskytovatelé.....	6
3.4.1	Amazon Web Services (AWS).....	7
3.4.2	Microsoft Azure.....	7
3.4.3	Google Cloud Platform (GCP).....	7
3.4.4	Oracle Cloud Infrastructure (OCI).....	8
3.4.5	IBM Cloud	8
3.4.6	Srovnání nabídek uvedených poskytovatelů	8
4	Nástroje pro databáze nad cloud	9
4.1	Definice databáze	9
4.2	Typy digitálních databází.....	9
4.2.1	Prosté datové soubory	10
4.2.2	Hierarchické databáze	10
4.2.3	Dokumentově orientované databáze	10
4.2.4	Relační databáze.....	11

4.2.5	Objektové databáze.....	11
4.2.6	Objektově relační databáze.....	12
4.3	Výběr databáze pro demonstrační hru.....	12
4.4	Poskytovatelé.....	13
4.4.1	Microsoft Azure.....	13
4.4.2	FreeMySQLHosting.....	13
4.4.3	Amazon Web Services (AWS).....	14
4.4.4	RemoteMySQL.....	14
4.4.5	DB4Free.....	14
4.5	Srovnání služeb.....	14
4.6	Nástroje pro tvorbu databáze nad cloud.....	15
4.6.1	PhpMyAdmin.....	15
4.6.2	Adminer.....	16
4.6.3	MySQL WorkBench.....	17
4.6.4	DataGrip.....	17
5	Síťová komunikace.....	18
5.1	OSI model.....	19
5.2	TCP/IP model.....	19
5.2.1	Vrstva síťového rozhraní.....	20
5.2.2	Síťová vrstva.....	20
5.2.3	Transportní vrstva.....	20
5.2.4	Aplikační vrstva.....	21
5.3	Aplikační rozhraní Sockets.....	21
5.4	Síťová komunikace v programovacím jazyce C#.....	22
5.4.1	Třída Socket.....	22
5.4.2	Třídy TcpClient a TcpListener.....	22
5.4.3	Třída UdpClient.....	23
6	Popis a způsob práce s demonstrační aplikací a serverem.....	23

6.1	Schéma složení programu	23
6.2	Spuštění serveru	24
6.3	Spuštění klienta	24
6.3.1	Registrace a přihlášení.....	25
6.3.2	Menu.....	26
6.3.3	Hlavní kanál.....	27
6.3.4	Lobby.....	28
6.3.5	Hra.....	29
7	Pravidla hry Monopoly.....	31
8	Architektura aplikace	31
8.1	Výběr frameworku	31
8.2	Model-view-controller (MVC).....	32
8.3	Thread pooling	33
9	Testování	34
9.1	Odezva serverů.....	35
9.2	Názory hráčů	36
10	Závěr	37
11	Seznam použitých zdrojů	38
12	Přílohy	45

Seznam obrázků

Obr. 1. Cloud Computing.....	3
Obr. 2. Tři úrovně distribučních modelů	4
Obr. 3. Management odpovědnosti služeb.....	5
Obr. 4. Binární strom	10
Obr. 5. Dokumentově orientovaná databáze.....	11
Obr. 6. Relační databáze.	11
Obr. 7. Loga relačních systémů.	13
Obr. 8. Ukázka prostředí PhpMyAdmin.	16
Obr. 9. Ukázka prostředí Adminer.....	16
Obr. 10. Ukázka tvorby ER diagram v prostředí MySQL WorkBench.....	17
Obr. 11. Ukázka analýzy kódu v prostředí DataGrip.....	18
Obr. 12. Vrstvy síťové komunikace.....	18
Obr. 13. struktura síťových modelů OSI a TCP/IP.....	19
Obr. 14. Vzájemná komunikace mezi komponenty.....	23
Obr. 15. Spuštění serveru z příkazové řádky.....	24
Obr. 16. Připojení k serveru.....	25
Obr. 17. Registrace nového uživatele.....	26
Obr. 18. Menu klienta.....	26
Obr. 19. Hlavní kanál.....	27
Obr. 20. Založení lobby.....	28
Obr. 21. Lobby.....	29
Obr. 22. Hra.....	30
Obr. 23. Hra po kliknutí na políčko Vítkov.....	30
Obr. 24. Herní smyčka.....	32
Obr. 25. Vzor MVC.....	33

Seznam tabulek

Tab. 1. Srovnání bezplatných nabídek cloudových služeb.....	9
Tab. 2. Srovnání nabídek poskytovatelů databází.....	15

Seznam grafů

Graf 1. Odezva serveru hostovaném na Amazonu.....	35
Graf 2. Odezva serveru hostovaném na Microsoft Azure.....	36
Graf 3. Souhrn hodnot odezev serverů.....	36

Seznam použitých zkratk

IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service)
vCPU	virtual Central Processing Unit
SQL	Structured Query Language
OQL	Object Query Language
AWS	Amazon Web Services

1 Úvod

Bakalářská práce se zabývá analýzou cloudových řešení pro realizaci multiplayerových her. Toto téma je aktuální, protože v současné době roste o problematiku cloudových řešení zájem, a to jak ze strany jednotlivců, tak firem. Hlavním cílem této práce je proto analyzovat stávající nabízená cloudová řešení pro tvorbu aplikace s víceuživatelským přístupem a větším datovým tokem (např. multiplayer hra). Za testovací klientskou aplikaci jsem si vybral realizovat deskovou hru Monopoly, a to kvůli její popularitě a také proto, že dovoluje hru pro více hráčů.

V první části se bakalářská práce zabývá popisem a analýzou aktuálních nabízených cloudových služeb. Popisuje jednotlivé cloudové typy včetně jejich využití a následně z nich vybírá ty nejlepší pro zajištění hostování relační databáze a testovacího serveru s víceuživatelským přístupem a větším datovým tokem pro demonstrační multiplayer hru.

Ve druhé části se práce zabývá samotnou realizací těchto dvou na sobě nezávislých testovacích aplikací. Server, který se spustí jako konzolová aplikace a na určitém portu bude naslouchat příchozím spojením od klientů a komunikovat s relační databází. Uživatelský klient s grafickým rozhraním, který umožní uživatelům se připojit na server, komunikovat s ostatními uživateli, zobrazit si své herní statistiky, zakládat a spouštět nové hry.

Tato část se zaměřuje především na problematiku související s výběrem vhodného frameworku, který umožňuje snadně a přitom efektivně vytvářet hry s grafickým rozhraním v programovacím jazyce C#.

Dále uvádí realizaci serverové části, se kterou souvisí problémy, jak co nejefektivněji zpracovávat požadavky klientů tak, aby server zbytečně neplýtvал svými zdroji a zároveň, aby nebyl těmito požadavky přehlcen.

V neposlední řadě je důležitá volba vhodného formátu, přes který je realizováno přijímání a odesílání dat (požadavků) pro komunikaci mezi klientem a serverem a jejich vzájemná synchronizace.

2 Cíl práce a metodika

Cílem práce je analyzovat stávající nabízená cloudová řešení pro tvorbu aplikace s víceuživatelským přístupem a větším datovým tokem (např. multiplayer hra). V rámci analýzy bude provedeno vyhodnocení nástrojů pro tvorbu cloudového IS (Information System) na každém konkrétním typu. Závěry budou shrnuty a přehledně vyhodnoceny, včetně demonstrativní (testující) aplikace.

Teoretická část práce se zaměří na analýzu aktuálně nabízených cloudových řešení. Následně z nich budou vybrána ta řešení, která budou poskytovatelé nabízet nejvýhodněji v poměru cena/výkon.

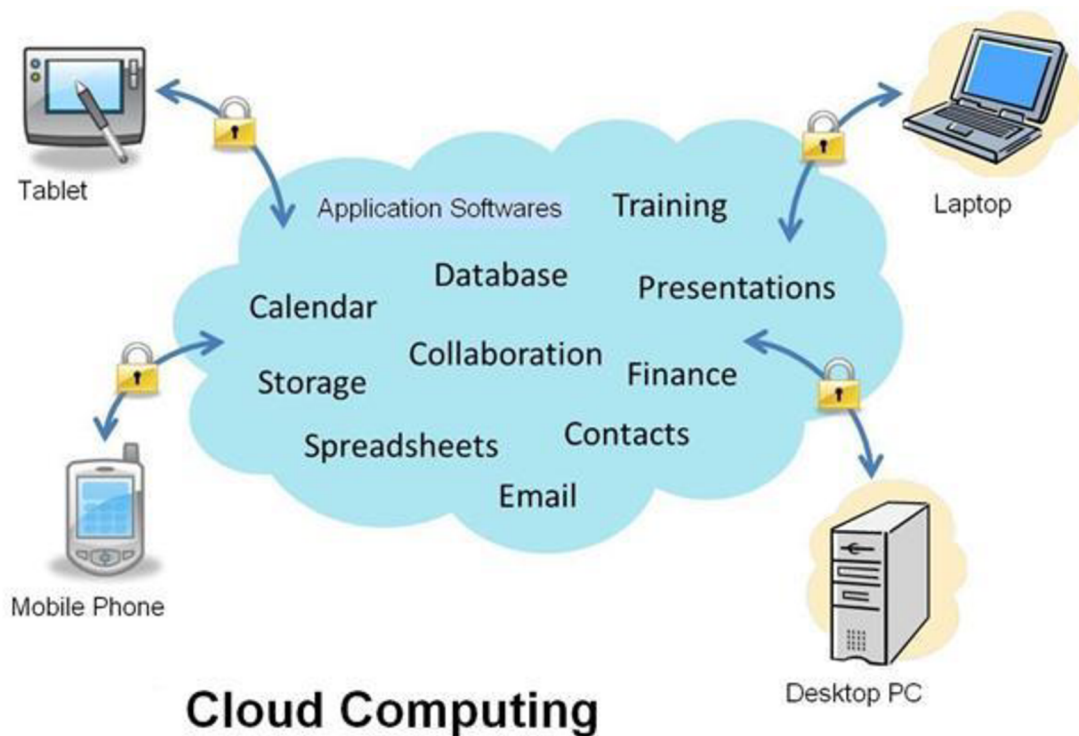
V praktické části se podrobí vybrané cloudy testování, a to za využití demonstrativní aplikace, pomocí které se zjistí jejich časová odezva. Data časových odezev budou poté přehledně vynesena do grafů a vyhodnocena.

3 Cloud Computing

V této kapitole je vysvětlen pojem cloud computing. Je popsán distribuční model a model nasazení. Tyto modely rozdělují cloud computing z hlediska poskytované služby a způsobu přístupu do cloudu. Poté jsou uvedeny nejatraktivnější bezplatné nabídky pronájmu virtuálního stroje na trhu. V závěru kapitoly je provedeno jejich srovnání.

3.1 Definice

Cloud Computing poskytuje výpočetní služby, včetně serverů, úložišť, databází, sítí, softwaru, analytických nástrojů a inteligentních funkcí. Přístup k těmto službám je běžně zajišťován vzdáleně skrze internetové připojení pomocí klientské aplikace nebo skrze webového prohlížeče (Obr. 1). Jako příklad využití těchto služeb lze uvést poštovního klienta ve webovském prohlížeči typu seznam.cz a Gmail od Google [1][2][3].



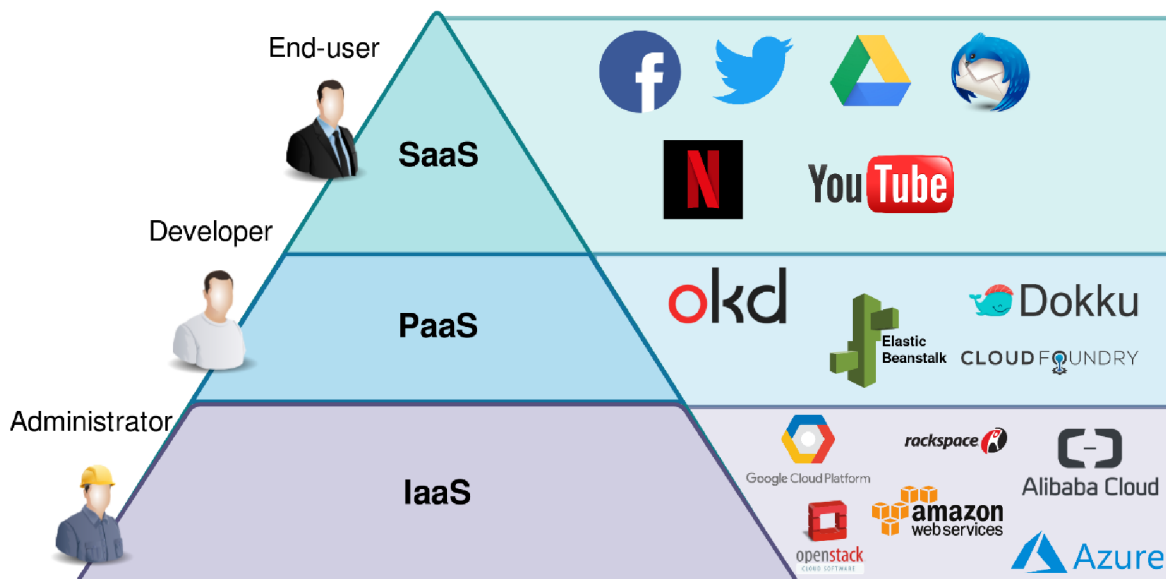
Obr. 1. Cloud Computing [4].

Mezi největší výhody cloud computingu se řadí: nízké investiční náklady (ušetření za hardware a software), výkon, rychlost, spolehlivost, produktivita a zabezpečení (např. při selhání hardwaru) [3] [5].

Podle NIST (National Institute of Standards and Technology) se cloud computing dělí podle distribučního modelu (Service model) a modelu nasazení (Deployment model). Model distribuční jej popisuje podle služby, kterou poskytuje a model nasazení (někdy též nazývaný jako architektura cloud computingu) jej popisuje z pohledu, jakým způsobem se ke službám přistupuje [1][3][6].

3.2 Distribuční modely

Pod pojmem distribuční model se rozumí rozdělení podle typu služby, kterou poskytovatel cloudu nabízí. Může se jednat o software, hardware či jejich kombinaci. Mezi základní a také nejvíce nabízené typy cloudových služeb se řadí tyto tři modely (Obr. 2) známé jako Infrastruktura jako služba (IaaS), Platforma jako služba (PaaS) a Software jako služba (SaaS). Na Obr. 3 je možno vidět rozdělení těchto tří skupin dle odpovědnosti cloudové služby [7][8].



Obr. 2. Tři úrovně distribučních modelů [9].

3.2.1 Infrastruktura jako služba (IaaS)

IaaS (Infrastructure as a Service) je typ služby Cloud Computingu, která nabízí pronájem kompletních infrastrukturních služeb nejčastěji prostřednictvím virtualizace. Těmito službami může být pronájem serverových produktů, virtuálních počítačů (VM), systémů pro zálohu dat, datových úložišť a také různých síťových prostředků, jako třeba firewallu [3][8][10].

Hlavní výhodou této služby je, že o veškerou správu a údržbu hardwaru se stará poskytovatel, který ručí v případě selhání serveru či jiné komponenty za rychlou nápravu. Další výhodou služby IaaS je okamžitá dostupnost, nízké náklady a plynulé dynamické škálování kapacitních limitů a možností (např. nároky uživatele na vlastní hardware poklesnou a může pak uvolnit jeden z více pronajatých počítačů) [10][11].

Nevýhodou IaaS může být, že data jsou uložena u poskytovatele cloudových služeb, a tudíž nad nimi uživatel nemá plnou kontrolu [12].

3.2.2 Platforma jako služba (PaaS)

PaaS (Platform as a Service) je dalším distribučním modelem cloud computingu. Jedná se o hostování platformy pro vývoj aplikací skrz virtualizaci. Poskytovatel uživateli poskytne cloudovou platformu pro vývoj, spouštění a běh aplikací. Přístup do platformy je skrze grafické uživatelské rozhraní (GUI), ve kterém vývojáři mohou spolupracovat na

veškeré své práci v celém životním cyklu aplikace, včetně kódování, integrace, testování, doručování, nasazení a zpětné vazby [13][14].

Výhodou je, že model PaaS osvobozuje uživatele od nutnosti pořizovat vlastní hardware a software a umožňuje více vývojářům současně pracovat na jednom projektu [13][14].

Nevýhodou modelu PaaS je, že každý poskytovatel používá své vlastní vývojové jazyky a lokální služby, a tudíž se aplikace stává závislou na daném poskytovateli. V důsledku toho je složitější přejít k jinému poskytovateli bez ztráty dat [13][15].

3.2.3 Software jako služba (SaaS)

SaaS (Software as a Service) je nejpoužívanější distribuční model, ve kterém jsou aplikace hostovány a nabízeny jako služba, která je dostupná přes web. Správa a podpora softwaru i hardware je zajišťována poskytovatelem a uživateli proto odpadají náklady spojené s údržbou a zabezpečením softwaru [8][15].

IaaS vs. PaaS vs. SaaS

● Odpovědnost zákazníka ● Odpovědnost poskytovatele

IaaS	PaaS	SaaS
Servery	Servery	Servery
Uložiště	Uložiště	Uložiště
Vytváření sítí	Vytváření sítí	Vytváření sítí
Virtualizace	Virtualizace	Virtualizace
OS	OS	OS
Middleware	Middleware	Middleware
Runtime	Runtime	Runtime
Aplikace	Aplikace	Aplikace
Data	Data	Data

Obr. 3. Management odpovědnosti služeb [zdroj: vlastní].

3.3 Modely nasazení

Model nasazení určuje u cloud computingu způsob přístupu do cloudové služby. Model nasazení může být veřejný, privátní anebo hybridní [16].

3.3.1 Veřejný model nasazení

Jedná se o nejužívanější typ nasazení cloud computingu. V tomto případě poskytovatel cloudové služby pronajímá uživatelům z řad široké veřejnosti své vlastní prostředky (tj. servery a uložště). Poskytovatel tyto prostředky spravuje a hostuje ve svých datových centrech. Uživatelé si následně pronajímají příslušný prostor. Tyto prostory jsou od sebe oddělené pomocí virtualizačního softwaru čili sdílí výpočetní prostředky (procesory a paměť) s ostatními uživateli. Výhodou je, že se snadno navyšuje či ubírá výkon dle potřeby uživatele [17][18].

3.3.2 Privátní model nasazení

Privátní model nasazení označuje cloudové prostředky, které jsou využívány pouze jednou organizací. Ve většině případů jsou tyto cloudové prostředky fyzicky umístěny v datovém centru společnosti, případně je hostuje jiný poskytovatel služeb. Důležité pro privátní cloudy je, že se spravují v privátní síti, tím se i zvyšuje bezpečnost dat. Hardware a software jsou explicitně vyhrazeny pro danou organizaci. Nevýhodou může být, že je potřeba dopředu znát jaký výkon bude organizace potřebovat a podle toho pořídit potřebný počet serverů [17][19].

3.3.3 Hybridní model nasazení

Hybridní model nasazení kombinuje veřejný a privátní model nasazení. Tyto hybridní modely dovolují přesun dat mezi veřejnými a privátními cloudovými prostředky organizace. Privátní cloudy jsou užitečné zejména pro citlivá data organizace. To přináší lepší flexibilitu, více možností nasazení a větší zabezpečení [17].

3.4 Poskytovatelé

Společností, které nabízejí cloud computing existuje celá řada. Zde se zaměřuji na společnosti, které poskytují převážně zdarma cloud computing s veřejným modelem nasazení a s distribučním modelem IaaS. U těchto společností (Amazon, Google, Oracle,

Microsoft a IBM) srovnávám jejich konkurenceschopnost především podle jejich nabídek poměru cena/výkon. Ty nejvhodnější dle parametrů poté následně prakticky otestuji v praktické části.

3.4.1 Amazon Web Services (AWS)

AWS je v současnosti leader v oblasti cloud computingu (s tržbami přes 20 miliard dolarů). K pronájmu nabízí své výpočetní virtuální stroje nazývané Elastic Compute Cloud a označované jako EC2. AWS nabízí bezplatnou modifikaci EC2 po dobu 12 měsíců po 750 provozních hodinách v měsíci. K dispozici je jak distribuce operačního systému Linux, tak i Windows. Dále nabízí operační paměť o velikosti 1 GB, 1 vCPU a paměť s kapacitou 30 GB [20][21].

3.4.2 Microsoft Azure

Microsoft Azure novým klientům do začátku přidá kredit v hodnotě 200 \$, který uživatel může po dobu 30 dnů využívat. Za tento kredit se dá na měsíc (30 dnů) pořídit instance virtuálního stroje s operačním systémem Windows/Linux s počtem 2 vCPU, 8 GB RAM a úložišti o kapacitě 50 GB. Microsoft Azure také nabízí po dobu 12 měsíců bezplatný virtuální stroj s operačním systémem Windows/Linux s 1 vCPU, 1 GB RAM a úložišti o kapacitě 5 GB. Microsoft Azure pro studenty nabízí k dispozici kredit o hodnotě 100 \$, který je možno využívat po dobu jednoho roku. [20][22].

3.4.3 Google Cloud Platform (GCP)

Podobně jako Microsoft i Google svým novým klientům přidá do začátku kredity, a to konkrétně 300 \$ po dobu tří měsíců. Google u placených instancí si účtuje poplatek za užívání operačního systému Windows. Pro volbu 2 vCPU, 8 GB RAM, 30 GB paměti s operačním systémem Windows pro provozní dobu 24 hodin v měsíci s datovým centrem ve Frankfurtu vyjde měsíční cena na 131 dolarů. S uvedeným kreditem pronájem této cloudové služby vystačí na dva měsíce. S operačním systémem Linux (Ubuntu, Debian, CentOS, CoreOS, BYOS) pak měsíční cena pronájmu klesne pouze na 63 dolarů. Google nabízí bezplatný virtuální stroj nazývaný F1-micro, jehož instance zahrnuje operační systém Windows/Linux, 1 vCPU, 0,6 GB RAM a 30 GB úložného prostoru. Počet provozních hodin by měl odpovídat počtu hodin v měsíci. Nevýhodou ovšem může být, že tato bezplatná

instance bude umístěna v datových centrech ve Spojených státech amerických, proto může být omezená šířka pásma [20][23].

3.4.4 Oracle Cloud Infrastructure (OCI)

Oracle novým uživatelům poskytne na dobu 30 dnů kredit v hodnotě 300 dolarů. Zdarma na neurčitou dobu nabízí pronájem instance VM.Standard.E2.1.Micro o 2 vCPU, 1 GB RAM s operačním systémem Linux (na výběr jsou tyto distribuce: Ubuntu, CentOS, Oracle Linux) a interní paměť o velikosti 20 GB; datové centrum mají v Nizozemsku v Amsterdamu [24].

3.4.5 IBM Cloud

IBM novým uživatelům nabídne po dobu 30 dnů kredit ve výši 200 dolarů. IBM nabízí konfiguraci virtuálního stroje označovaného IBM-centos s 2 vCPU, 8 GB RAM a 100 GB uložení. Tato konfigurace s operačním systémem Windows vychází měsíčně na 144 dolarů. S distribucí Linuxu poté na 78 dolarů. Nejbližší datové centrum má IBM ve Frankfurtu [25][26].

3.4.6 Srovnání nabídek uvedených poskytovatelů

V následující tabulce *Tab. 1* jsou shrnuty jednotlivé varianty nabídek od společností, které bezplatně poskytují cloud computing virtuálních strojů.

Z hlediska poskytovaného výkonu je na tom nejlépe společnost Google, která s distribučním operačním systémem Linux nabízí po dobu 90 dnů 2 jednotky vCPU, 8 GB operační paměti a 30 GB úložného prostoru.

Z hlediska dlouhodobosti největší výkon nabízí společnost Oracle, která poskytuje pronájem virtuálního stroje s 2 jednotkami vCPU, 1 GB RAM, interním uložištěm 20 GB s operačním systémem Linux. Zajímavá je i nabídka od Amazonu, který virtuální stroj nabízí po dobu jednoho roku s 1 jednotkou vCPU, 1 GB RAM, 30 GB úložného prostoru a s volbou operačního systému Linux nebo Windows.

Tab. 1. Srovnání bezplatných nabídek cloudových služeb [zdroj: vlastní].

Společnost	vCPU	RAM [GB]	Uložiště [GB]	OS	Doba [dny]
AWS	1	1	30	Win/Linux	365
Azure	2	8	50	Win/Linux	30
Azure	1	1	5	Win/Linux	365
Google	2	8	30	Win	60
Google	2	8	30	Linux	90
Google	1	0,6	30	Win/Linux	
Oracle	2	1	20	Linux	
IBM	2	8	100	Win/Linux	30

4 Nástroje pro databáze nad cloud

Na základě analýz shromážděných dat se dosahuje nových poznatků, které pomáhají v mnoha odvětvích jako například u korporací, které díky nim zlepšují obchodní strategii. Tato data je potřeba ukládat a organizovat do databází.

4.1 Definice databáze

Databáze je jakákoliv uspořádaná kolekce dat. Může se vyskytovat v digitální, ale i ve fyzické podobě. Příkladem fyzických podob je kartotéka u lékaře, ve které jsou karty pacientů vzestupně seřazeny dle abecedy. Ve většině digitálních databázích nad daty běží databázový server, který řídí databázi a umožňuje přístup k uloženým datům. Na tento server se posílají dotazy na přidání, úpravu, načtení, nebo na vymazání dat. Server dotazy provede a výsledky poté odešle nazpátek, a tudíž se v digitálních databázích nikdy s daty nepracuje napřímo.

4.2 Typy digitálních databází

Digitální databáze se člení podle způsobu jakým v nich jsou data ukládána a uspořádávána. Mezi nejběžnější typy digitálních databází se řadí: prosté databázové

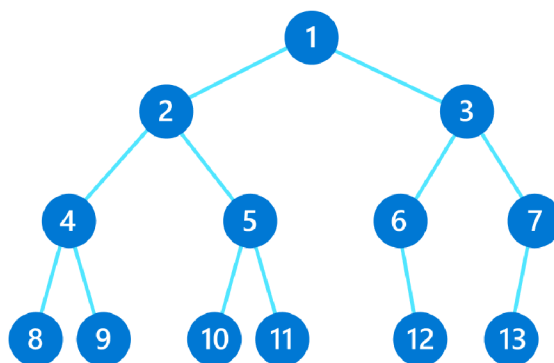
soubory, hierarchické databáze, dokumentově orientované databáze, relační databáze, objektové databáze a objektově relační databáze [27].

4.2.1 Prosté datové soubory

V těchto prostých databázových souborech jsou data ukládána sekvenčně, běžně v textovém formátu (např. .txt, .ini, .conf). Výhodami těchto souborů jsou, že se dají relativně jednoduše vytvářet a přidávat do nich další data. Nevýhodou je zdlouhavé vyhledávání dat v těchto souborech, mohou se v nich vyskytovat redundantní data a snadno se mohou poškodit. Příkladem využití tohoto databázového typu je karetní hra Solitaire, která výsledky jednotlivých hráčů zapisuje do textového souboru [27].

4.2.2 Hierarchické databáze

Hierarchické databáze využívají k ukládání dat stromovou strukturu (Obr. 4), na principu vztahů rodič-dítě. Tyto typy databází umožňují velice rychlé vyhledávání. Jejich nevýhodou může být obtížné procházení dat v případě neznalosti použitých vztahů a časem se může stát i obtížná údržba vztahů mezi daty. Příkladem použití této databáze může být registr v systému Windows [27].

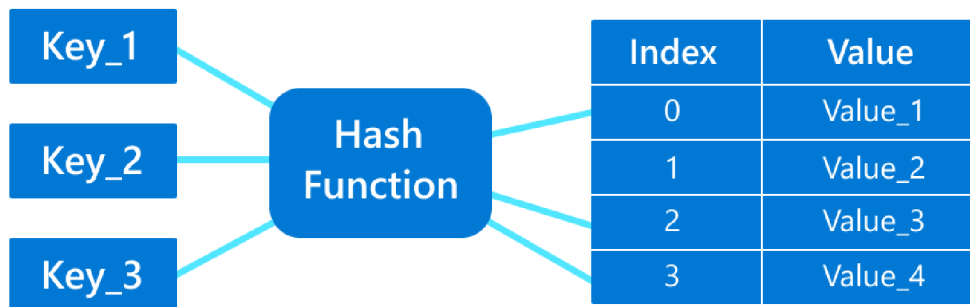


Obr. 4. Binární strom [28].

4.2.3 Dokumentově orientované databáze

V dokumentově orientovaných databázích (Obr. 5) jsou data ukládána ve volné podobě nejčastěji ve formátu JSON, BSON nebo XML a přitom jsou indexována pomocí klíčových či hashovacích hodnot. Mezi výhody tohoto typu patří dobrá škálovatelnost dat

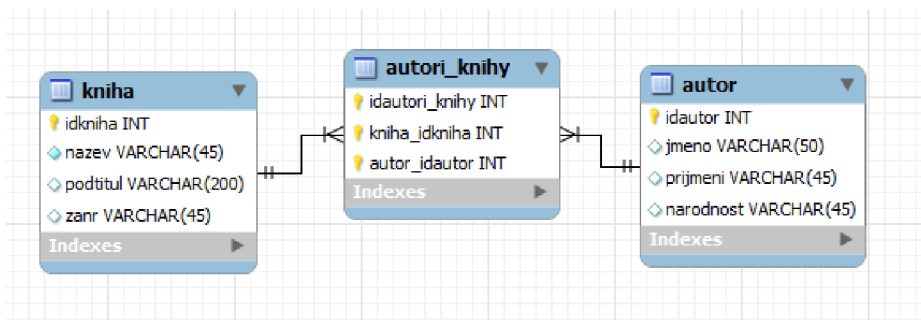
v rozsáhlých sítích. Jejich nevýhodami jsou podobně jako u prostých databázových souborů výskyt redundantních dat, mezi daty nejsou udržovány žádné vztahy a vyhledávání dat může být pomalé. Příklady databázových systémů využívajících tento typ databází jsou Redis a CouchDB [27].



Obr. 5. Dokumentově orientovaná databáze [28].

4.2.4 Relační databáze

V relačních databázích jsou data uspořádána do tabulek, které jsou tvořeny pomocí řádků a sloupců (Obr. 6). Jedná se o nejpoužívanější typ digitálních databází, který umožňuje indexování velkých objemů dat a rychlé vyhledávání. Nevýhodou pak je, že relace (vztahy) mezi jednotlivými tabulkami mohou být velice složité. Relační databáze se standardně vytváří pomocí dotazového jazyka SQL (Structured Query Language). Typickými příklady použití této databáze je pro tvorbu webového fóra nebo nabídka v e-shopech [27][29].



Obr. 6. Relační databáze [30].

4.2.5 Objektové databáze

V objektově orientovaných databázích jsou data ukládána (podobně jako je tomu v objektově orientovaných programovacích jazycích) do objektů. Objektové databáze

umožňují využívat dědičnost, zapouzdření a polymorfismus, což jsou základní stavební pilíře každého objektově orientovaného programovacího jazyka. S pomocí zapouzdření je možné v objektu dosáhnout ke skrývání dat. Jedná se o techniku kdy je možno objekt používat bez toho, aby uživatel musel vědět nebo se starat o to jak objekt uvnitř funguje. Polymorfismus s využíváním dědičnosti umožňuje objektům při volání metod zastupovat své rodiče. Atributy objektů nemusí být v objektové databázi pouze primitivními datovými typy, ale mohou být dále strukturované jako např. objekt (pomocí reference) nebo jako kolekce. Oproti relačním databázím jsou objektové databáze náročnější na návrh a implementaci, ale jejich samotné používání je jednodušší a intuitivnější. Pro práci s objektovou databází se používá jazyk OQL (Object Query Language), který je podobný jazyku SQL pro relační databáze. Příklady databázových systémů, které využívají tento typ databáze jsou ObjectDB a ObjectStore [31].

4.2.6 Objektově relační databáze

Objektově relační databáze rozšiřují rysy relačních a objektových databází. Tyto databáze přidávají do relačních tabulek objektovost. A to tak, že všechna data jsou ukládána do tabulek, nicméně některé položky mohou obsahovat rozmanitější datovou strukturu, nazývanou jako abstraktní datové typy (ADT). Pro dotazy nad databází se používá jazyk SQL [32][33].

4.3 Výběr databáze pro demonstrační hru

Každý typ digitální databáze má jiné vlastnosti. Výběr vhodného typu digitální databáze souvisí na povaze dat, která je potřeba ukládat. Je třeba specifikovat počet uživatelů, kteří budou aplikaci používat, jaký typ entit se bude ukládat, nebo jestli se povaha záznamů nebude časem lišit a spoustu dalších detailů [29].

Obecně platí, že nerelační databáze se hodí tam, kde není předem znám typ a velikost dat, která se budou ukládat do digitální databáze. Naopak relační databáze se obecně používají pro data, která mají pevně danou strukturu a lze-li na jejich základě navrhnout přívětivou architekturu pro databázi [29].

Protože demonstrační aplikace má pevně stanovenou strukturu dat, rozhodl jsem se pro použití relační databáze. Existuje více relačních systémů, které se navzájem od sebe liší implementovanými standardy pro SQL a téměř každý si přidává vlastní konstrukce, které nejsou součástí standardu. Jako jsou například: SQLite, MySQL, Oracle, Microsoft SQL

nebo MariaDB, jež vznikla jako alternativní větev MySQL. Na Obr. 7 jsou zobrazeny známá loga některých relačních systémů. [27] [29][34].

V této práci jsem se rozhodl pro demonstrační aplikaci používat relační systém MySQL. Jedná se o druhý nejrozšířenější relační systém (první místo zaujímá SQLite), který vlastní společnost Oracle. Používá se převážně při programování webových aplikací. Mým hlavním důvodem pro volbu MySQL je jeho široká nabídka hostingů [27][34].



Obr. 7. Loga relačních systémů [35–37].

4.4 Poskytovatelé

Zde se práce zaměřuje na bezplatné nabídky pro hostování MySQL databází, které nabízejí pro jejich správu vzdálený přístup. Konkrétně se zaměřuji na tyto společnosti: Microsoft Azure, FreeMySQLHosting, Amazon Web Services (AWS), RemoteMySQL a DB4Free.

4.4.1 Microsoft Azure

Microsoft Azure nabízí uživatelům po dobu 12 měsíců využít zdarma hosting MySQL databáze na 750 hodin (přibližně měsíc) výpočetního výkonu na virtuálním počítači SKU B1ms, který disponuje 1 vCPU, 2 GB RAM a pamětí s kapacitou 32 GB [38][39].

4.4.2 FreeMySQLHosting

FreeMySQLHosting nabízí uživatelům zdarma hosting jedné MySQL databáze o kapacitě 5 MB. Výhodou je rychlá registrace, v níž stačí zadat pouze e-mailovou adresu. Na tu se následně pošle odkaz na jejich webovskou stránku, na které si uživatel zvolí heslo a není tak potřeba zadávat osobní informace jako jsou třeba údaje o platební kartě, místa bydliště apod. Po přihlášení si lze již snadno databázi založit. Pro umístění serveru jsou na výběr tyto lokace: Asie, Střední Evropa, Velká Británie, Severní

Amerika a Jižní Amerika. Po založení databáze se přihlašovací údaje k databázi odešlou na zaregistrovaný e-mail, ale jsou uvedeny i na webové stránce [40][41].

4.4.3 Amazon Web Services (AWS)

Amazon nabízí novým uživatelům po dobu jednoho roku službu Amazon Relational Database Service (RDS) na instanci db.t2.micro s paměťovou kapacitou 20 GB a dalších 20 GB pro měsíční zálohu dat. V rámci této služby je k dispozici hosting pro šest databází: Amazon Aurora, PostgreSQL, MariaDB, Oracle Database, SQL Server a samozřejmě i MySQL databáze [41][42].

4.4.4 RemoteMySQL

RemoteMySQL nabízí MySQL databázi s kapacitou paměti 100 MB. Registrace je rychlá. Nicméně pro založení databáze je ještě potřeba vyplnit krátký osobní dotazník [41][43].

4.4.5 DB4Free

DB4Free nabízí hosting vždy nejnovější MySQL verze, a proto se hodí převážně pro testovací účely. Například při přecházení na novější verzi serveru nebo na vyzkoušení nových funkcí. Existuje možnost ztráty dat a výpadků, proto není vhodná pro komerční použití. Nabízí paměť o velikosti 200 MB. Registrace je velmi rychlá, stačí pouze vyplnit informace o názvu databáze a uživatele a poté na e-mail se zašlou potřebné informace ke vzdálenému připojení [41][44][45].

4.5 Srovnání služeb

Z níže uvedené Tab. 2 se jako nejlepší nabídka jeví od společnosti Amazon, která nabízí 20 GB úložného prostou po dobu jednoho roku.

Tab. 2. Srovnání nabídek poskytovatelů databází [zdroj: vlastní].

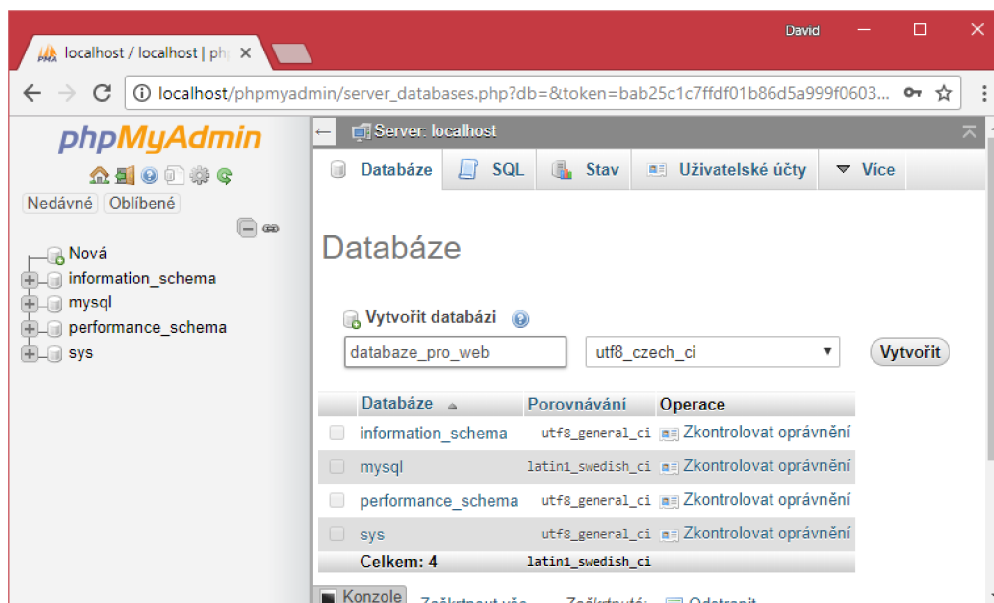
Společnost	Uložiště [MB]	Doba [dny]
Microsoft Azure	32 000	31
FreeMySQLHosting	5	
AWS	20 000	365
RemoteMySQL	100	
DB4Free	200	

4.6 Nástroje pro tvorbu databáze nad cloud

Tyto nástroje (ve většině případů s grafickým uživatelským rozhraním) se připojují na vzdálené databázové servery a snaží se usnadnit jejich administraci pomocí vizualizace dat, což může být někdy rychlejší než kdyby uživatel psal SQL dotazy na databázový server ručně. V následující části zmiňuji tyto nástroje, které jsou určeny především pro správu MySQL databází. Konkrétně: PhpMyAdmin, Adminer, MySQL WorkBench a DataGrip [46].

4.6.1 PhpMyAdmin

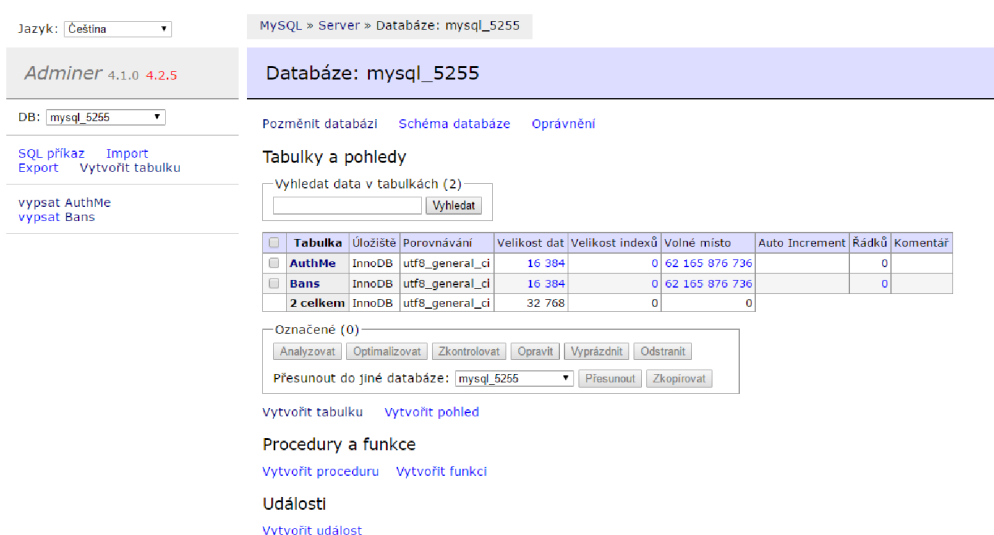
PhpMyAdmin je nejpoužívanější bezplatný softwarový nástroj určený ke správě vzdálených MySQL a MarianDB databází z prostředí webového prohlížeče (ukázka Obr. 8). Vyskytuje se téměř na každém webhostingu a je přeložen do 72 jazyků včetně češtiny. Poskytuje přehledné intuitivní grafické uživatelské rozhraní. Samozřejmostí pak je i možnost zadávání SQL dotazů. [47][48].



Obr. 8. Ukázka prostředí PhpMyAdmin [49].

4.6.2 Adminer

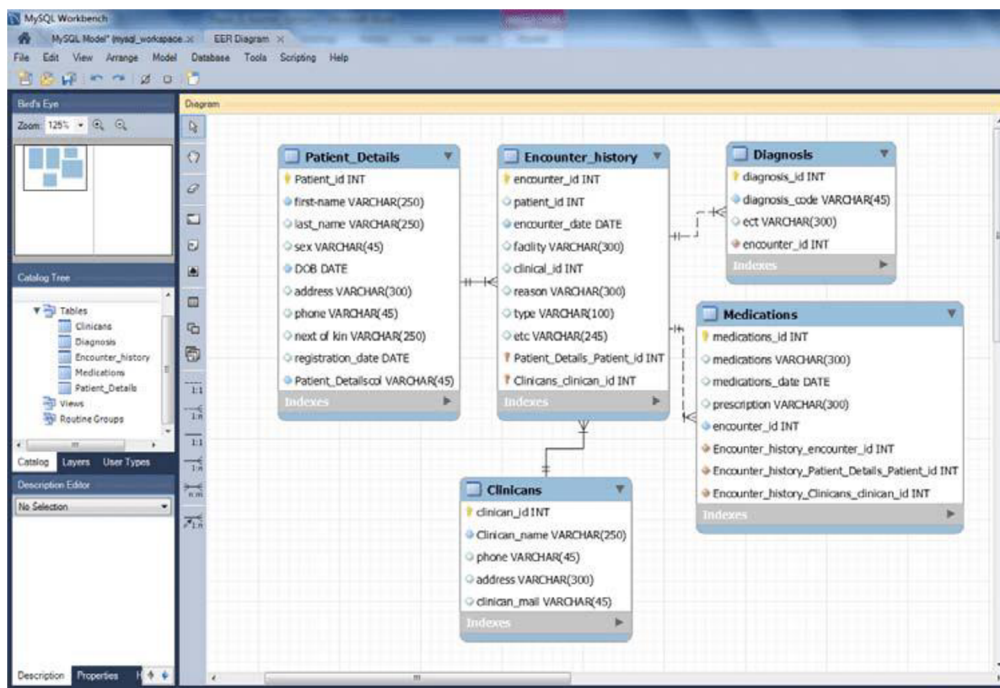
Adminer je českým nástrojem, který podporuje správu více typů databází včetně MySQL. Databázi spravuje podobně jako je tomu v nástroji PhpMyAdmin v prostředí webového prohlížeče a je k dispozici v 57 jazycích včetně češtiny (ukázka Obr. 9). Oproti MyphpAdminu je Adminer rychlejší pro správu databází, například tím, že uživateli dovoluje se přihlásit současně na dva databázové servery [50–53].



Obr. 9. Ukázka prostředí Adminer [54].

4.6.3 MySQL WorkBench

MySQL WorkBench je bezplatný nástroj s grafickým uživatelským rozhraním, který je poskytován pro vzdálenou správu MySQL databází společností Oracle. Umožňuje také vizuálně navrhnout model databázového schématu tzv. ER diagram a poté vygenerovat jeho SQL kód (ukázka na Obr. 10) [55, 56].



Obr. 10. Ukázka tvorby ER diagram v prostředí MySQL WorkBench [57].

4.6.4 DataGrip

DataGrip (ukázka na Obr. 11) je profesionální nástroj umožňující pokročilou správu 8 typů relačních databází včetně MySQL vyvíjen společností JetBrains. DataGrip má zabudované inteligentní funkce, pomocí kterých se zrychluje psaní SQL dotazů. Dále tento nástroj analyzuje kód a díky tomu odhaluje pravděpodobné chyby v syntaxi před kompilací [58].

```

SELECT
  actor_id,
  first_name,
  last_name,
  last_updat
FROM actor;

```

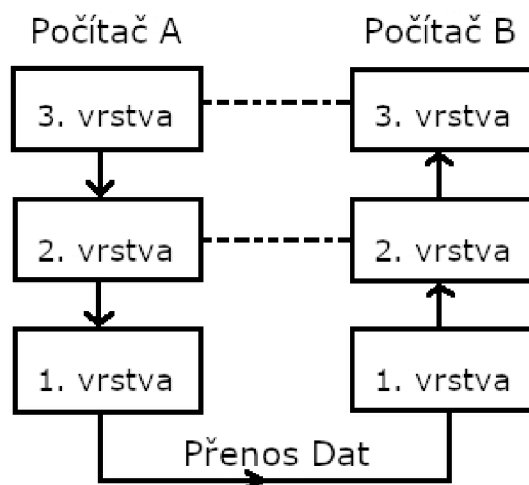
Unable to resolve column 'last_updat' more... (⌘F1)

Obr. 11. Ukázka analýzy kódu v prostředí DataGrip [58].

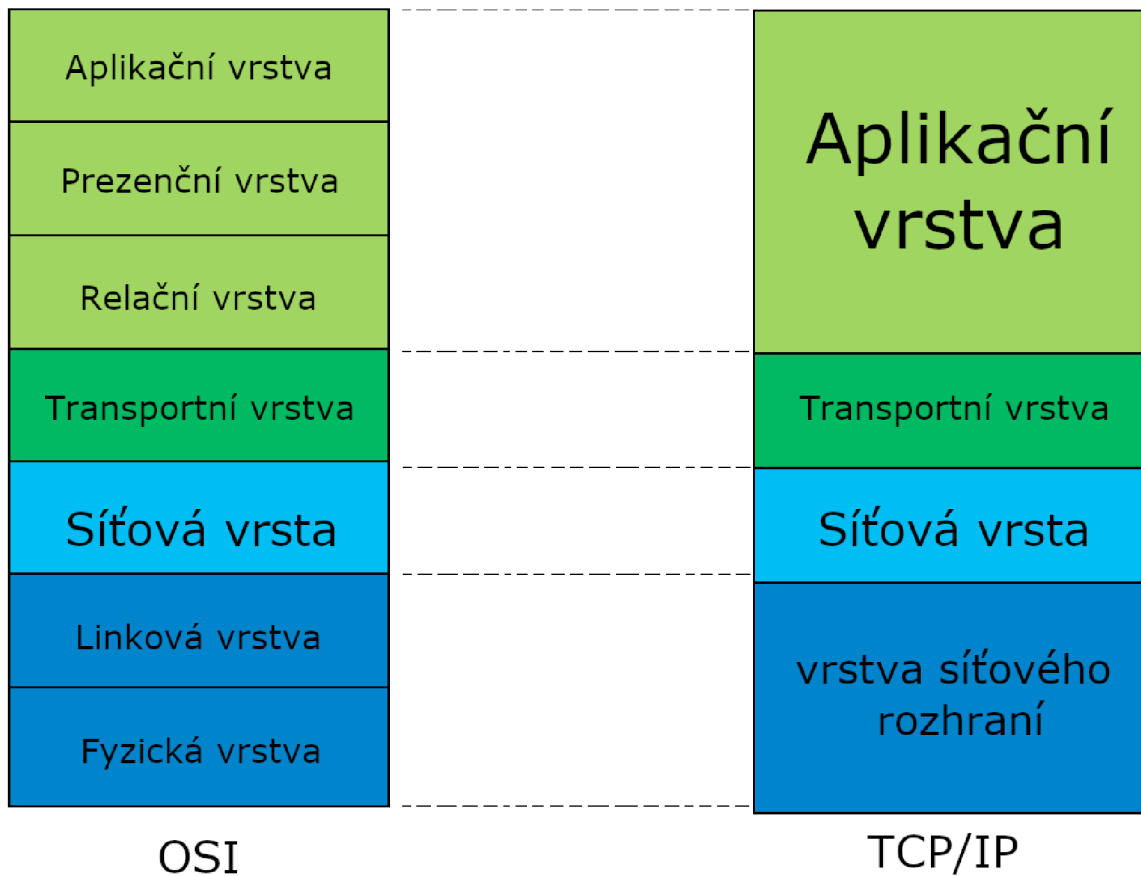
5 Síťová komunikace

Přenos dat síťovým médiem z aplikace běžící na jednom počítači na aplikaci běžící na jiném vzdálenějším počítači (v našem případě mezi klientem a serverem) je popsán pomocí síťových modelů. Tyto modely jednoznačně popisují strukturu dat. Definiují jak mají být data upravena pro přenos a následně jak mají být na druhé straně přečtena. Protože problematika síťové komunikace je velmi široká, tak se síťové modely skládají z určitého počtu vrstev, kde je každé této vrstvě přiřazen na starost určitý úkol a ostatní úkoly v logické návaznosti přenechávají na starost následujícím vrstvám. Tím je zajištěno, že při přenosu dat síťovým médiem na obou stranách spolu vertikálně (nepřímo) komunikují stejné vrstvy (viz. Obr. 12). Na těchto vrstvách běží komunikační protokoly, které řeší jakým způsobem má každá vrstva řešit své úkoly. Definiují pravidla podle kterých jsou data mezi zařízeními přenášena a jejich implementace je nezávislá na programovacím jazyku [59].

Nejrozšířenějšími síťovými modely pro komunikaci jsou OSI (Open System Interconnection) a TCP/IP [59].



Obr. 12. Vrstvy síťové komunikace [zdroj: vlastní].



Obr. 13. struktura síťových modelů OSI a TCP/IP [zdroj: vlastní].

5.1 OSI model

OSI model se skládá ze sedmi různých vrstev (viz. Obr. 13). Je navržený pro komunikaci otevřených systémů. Přináší velkou spolehlivost při přenosu dat ovšem na úkor jejich přenosové rychlosti. Jedná se o konceptuální model pro který nejsou specifikovány konkrétní protokoly nebo služby pro jeho jednotlivé vrstvy a proto se v praxi moc nepoužívá [60].

5.2 TCP/IP model

TCP/IP model vychází z OSI modelu. Důraz je zde kladen především na rychlý přenos dat. Přenášená data jsou nešifrovaná. Jedná se v současné době o nejpoužívanější síťový model. Používá se např. v nejrozsáhlejší světové síti Internet. Model TCP/IP se skládá ze čtyř vrstev (viz. Obr. 13) [61].

5.2.1 Vrstva síťového rozhraní

První vrstva v síťovém modelu TCP/IP se nazývá Network Access, někdy také Network Interface Layer (česky: vrstva síťového rozhraní). Jedná se o nejnižší vrstvu tohoto modelu, která má přístup k fyzickému přenosovému médiu. Má na starosti příjem a odesílání datových paketů. V rámci TCP/IP modelu není tato vrstva blíže specifikována a taktéž pro ni nejsou definované žádné protokoly. To je dáno tím, že je závislá na použitém hardwaru a přenosové technologii jako je např. Ethernet nebo Token ring [62–64].

5.2.2 Síťová vrstva

Vyšší vrstvou, nad vrstvou síťového rozhraní, je síťová vrstva (anglicky: Internet Layer), někdy též IP vrstva. Primárním úkolem této vrstvy je pro datové pakety najít v nejkratším čase nejvhodnější cestu mezi libovolnými dvěma uzly v síti. Tato vrstva nezajišťuje spolehlivost doručení přenášených dat, tu zajišťují vyšší vrstvy. Nejdůležitějším protokolem této vrstvy je protokol IP (Internet Protocol), který se stará o přenos datových paketů tak, aby se dostaly od odesílatele k příjemci. K tomu se využívá tzv. IP datagramů, které v záhlaví nesou IP adresu příjemce. Všechna síťová rozhraní mají v celé síti přiřazenou jedinečnou IP adresu, je tedy jednoznačně dáno, kam mají být dané datagramy doručeny [62–66].

5.2.3 Transportní vrstva

U TCP/IP síťového modelu je třetí vrstva označovaná jako transportní vrstva (anglicky: Transport Layer), nebo také jako TCP vrstva. Hlavním úkolem této třetí síťové vrstvy je zajištění přenosu dat mezi dvěma aplikačními programy, které jsou v rámci daného počítače identifikovány pomocí portů. Dle jejich nároků a požadavků transportní vrstva reguluje obousměrně tok dat, spolehlivost přenosu dat a též mění v síťové vrstvě charakter komunikace na spojitou a nespojitou. Zde jsou důležité dva protokoly, a to protokol TCP (Transmission Control Protocol) a protokol UDP (User Datagram Protocol). Protokol UDP se zaměřuje především na rychlost přenosu dat na úkor spolehlivosti doručení dat. Protokol UDP nenavazuje s druhou stranou spojení a nemáme tudíž ověřeno jestli data došla příjemci v pořádku. Používá se především v real time přenosech, kde nejsou případné chyby v přenesených datech tak patrné. Například u videokonferencí, kdy se případné ztráty

nějakého datagramu projeví třeba bliknutím obrazovky nebo například na určitý okamžik výpadkem zvuku či šumem. Ve většině případů se dnes využívá protokol TCP, který se zaměřuje více na spolehlivost přenesených dat vůči přenosové rychlosti. Protokol TCP realizuje spojitou komunikaci a datové pakety jsou na rozdíl od protokolu UDP doručovány ve správném pořadí, ve kterém byly odeslány. Společně s daty je také posílán kontrolní součet pomocí kterého se zjišťují případné ztráty datových paketů. V tomto případě si TCP protokol vyžádá jejich nové odeslání. Tento protokol se ideálně hodí pro realizaci síťového modelu klient/server [62–65, 67, 68].

5.2.4 Aplikační vrstva

Čtvrtou a zároveň též nejvyšší vrstvou modelu TCP/IP je aplikační vrstva (anglicky: Application Layer). Do této vrstvy už přímo zasahují jednotlivé aplikace. Pro tuto vrstvu je vytvořena celá řada protokolů jako jsou například protokoly POP3, IMAP a SMTP, které jsou typické pro práci s příjmem a odesíláním e-mailů. Dalšími protokoly jsou třeba: FTP, HTTP, DNS, Telnet [62, 63, 67].

5.3 Aplikační rozhraní Sockets

Pro implementaci síťového modelu TCP/IP do operačních systémů je potřeba vytvořit aplikační rozhraní, které předdefinuje metody pomocí kterých budou aplikace přistupovat ke službám protokolů ze síťového modelu TCP/IP [69].

Kolem roku 1982 bylo na univerzitě v Berkeley vyvinuté pro Unixové systémy první aplikační rozhraní tzv. BSD Socket. Přestože v dnešní době již toto rozhraní není zcela dokonalé (obsahuje jen blokující operace a není objektově orientované), bylo velmi významné. Využívá vysokou míru abstrakce, proto je toto rozhraní možné používat i na jiné struktury než pouze na TCP/IP (např. na: ISO, AppleTalk nebo IBM SNA atd.). Stalo se základem pro téměř všechny další implementace aplikačních síťových rozhraní pro různé platformy [69].

Postupem času tedy vznikala další rozhraní pro různé platformy, z nichž za zmínku určitě stojí uvést aplikační programové rozhraní (zkráceně API) Winsock, které sloužilo pro síťovou komunikaci pro systémy Windows [69].

Pro síťovou komunikaci v některých programovacích jazycích už API není potřeba, protože požadované komponenty pro práci se sockety jsou již přímo zabudované v jejich

knihovnách. Příkladem toho je programovací jazyk Java nebo C#. A právě těmto knihovnám v programovacím jazyce C# se budu dále v této práci zabývat.

5.4 Síťová komunikace v programovacím jazyce C#

Knihovny, které se v programovacím jazyce C# zabývají sítíovou komunikací nalezneme ve jmenném prostoru System.Net. API pro práci se sockety je pak implementováno ve jmenném prostoru System.Net.Sockets. Mezi hlavní třídy nacházející se v tomto jmenném prostoru se řadí třídy: Socket, TcpClient, TcpListener a udpClient. Tyto třídy umožňují přímé odesílání nebo naslouchání síťových požadavků na určitém portu [70].

5.4.1 Třída Socket

Třída Socket implementuje rozhraní Berkeley Sockets a umožňuje nejvyšší úroveň řízení a dovoluje manipulovat téměř se všemi typy spojení. Instance této třídy umožňují přístup do nižších vrstev síťového modelu[70].

5.4.2 Třídy TcpClient a TcpListener

Třídy TcpClient a TcpListener nabízejí uživateli jednoduché metody pro navázání spojení a posílání dat mezi dvěma koncovými body pomocí komunikačního protokolu TCP. Koncový bod je určen IP adresou a číslem portu. Za předpokladu, že se neimplementuje obvyklá služba (jako je např. protokol HTTP, který obvykle běží na portu 80), měla by se čísla portů volit větší než 1024. Tyto třídy jsou využívány i v demonstrativní multiplayerové hře [70].

Třída TcpClient zapouzdřuje připojení TCP a poskytuje vlastnosti pro správu připojení, včetně ukládání dat do vyrovnávací paměti[70].

Třída TcpListener naslouchá příchozím spojení sítíových klientů TCP. Mezi její nejdůležitější metody patří: Start(), AcceptTcpClient(), Stop(). Pomocí metody Start() začne naslouchat příchozím TCP spojení. Po přijetí požadavku o spojení se používá metoda AcceptTcpClient(), která vrátí objekt typu TcpClient. Naslouchání se ukončí zavoláním metody Stop() [70].

5.4.3 Třída UdpClient

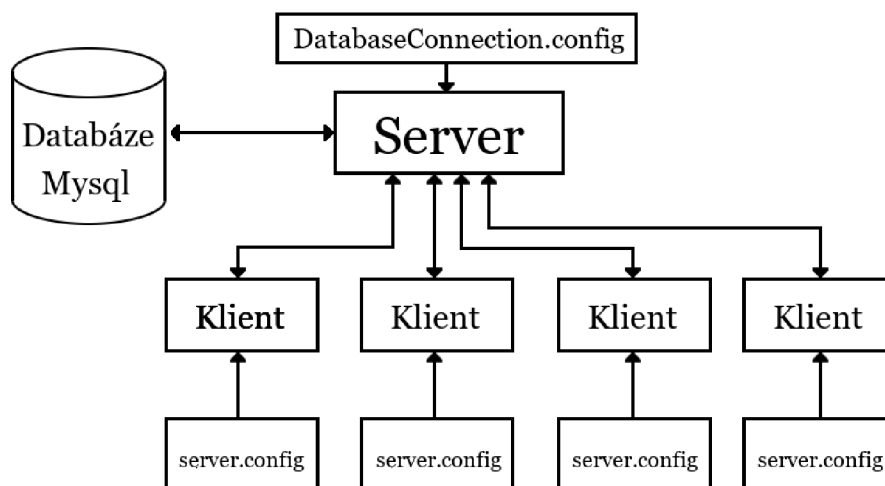
Třída UdpClient zjednodušuje práci se sockety a nabízí jednodušší rozhraní. Pro síťovou komunikaci využívá protokol UDP. Jelikož je UDP nespojitým protokolem, tak třída neobsahuje žádnou metodu, která by vracela síťový proud pro čtení a zápis. Datagramy se posílají pomocí metody Send() a přijímají přes metodu Receive() [70].

6 Popis a způsob práce s demonstrativní aplikací a serverem

Demonstrativní aplikace se skládá ze tří samostatných částí: databáze, serveru a klienta. V této kapitole je popsán způsob práce s těmito částmi.

6.1 Schéma složení programu

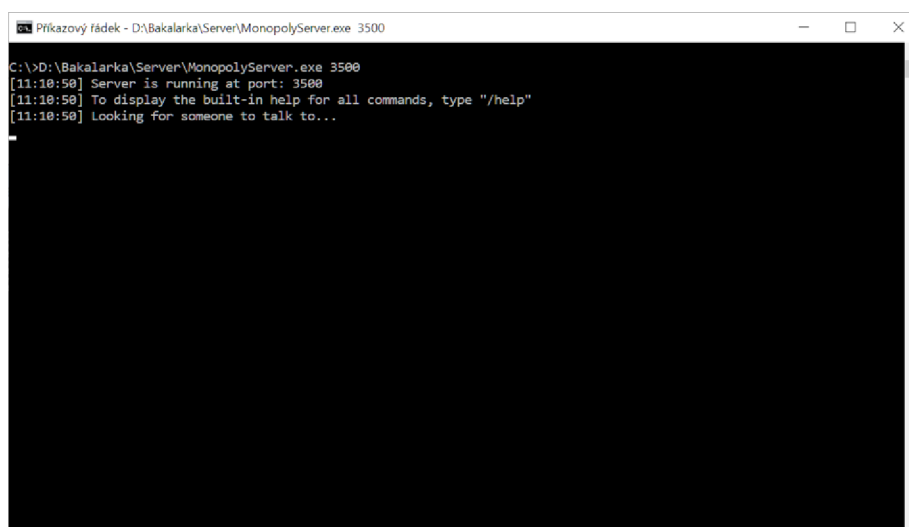
Na Obr. 14 je zobrazeno schéma vzájemné komunikace mezi jednotlivými komponenty projektu. Server se stará o komunikaci mezi MySQL databází jejíž informace potřebné k navázání spojení se nacházejí v souboru DatabaseConnection.config. Dále server zajišťuje obsluhu požadavků od klientů. Klient načítá data ze souboru server.config. Jak již název souboru napovídá tato data se týkají informací o serverech (jako jsou jejich IP adresy, porty a názvy), které slouží k navazování spojení s těmito servery.



Obr. 14. Vzájemná komunikace mezi komponenty.

6.2 Spuštění serveru

Server je možno spustit dvěma způsoby. Jeden způsob spočívá ve spuštění spustitelného souboru MonopolyServer.exe. Server bude defaultně naslouchat na portu 2050. Pokud bude vyžadováno server spustit na jiném portu, pak je potřeba mu předat tento port jako parametr, například pomocí příkazové řádky, jak je znázorněno na Obr. 15. V případě, že inicializace serveru proběhla v pořádku, je server připravený zpracovávat požadavky od klientů.

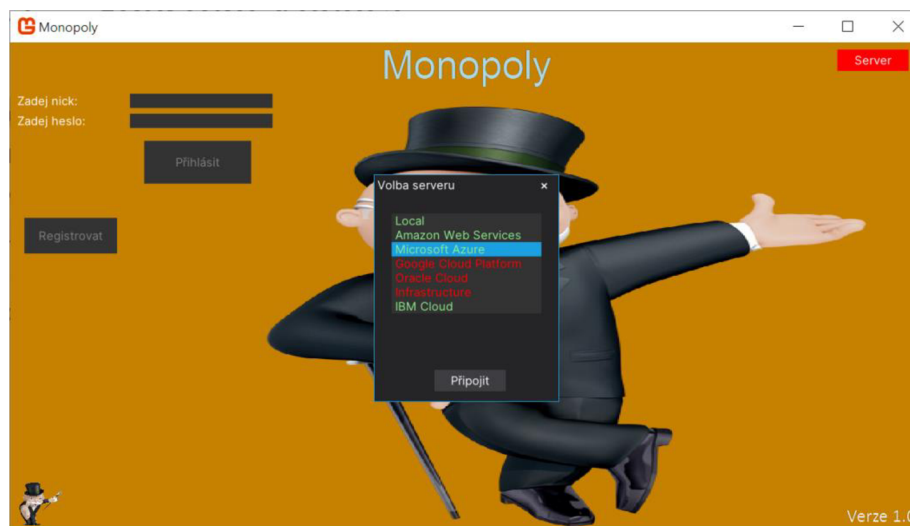


```
Příkazový řádek - D:\Bakalarka\Server\MonopolyServer.exe 3500
C:\>D:\Bakalarka\Server\MonopolyServer.exe 3500
[11:10:50] Server is running at port: 3500
[11:10:50] To display the built-in help for all commands, type "/help"
[11:10:50] Looking for someone to talk to...
```

Obr. 15. Spuštění serveru z příkazové řádky [zdroj: vlastní].

6.3 Spuštění klienta

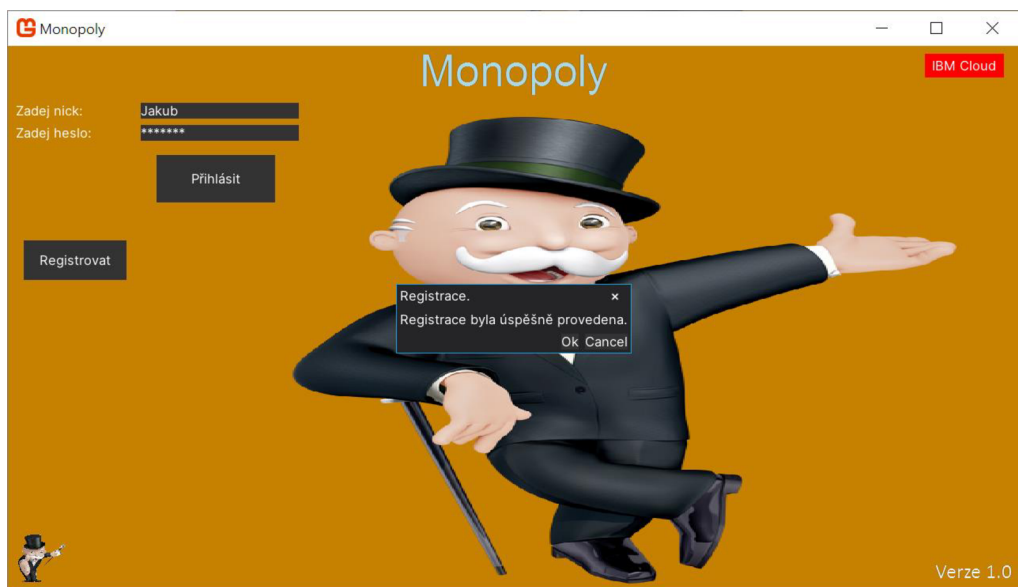
Spuštění klienta (demonstrativní aplikace) je možné pomocí spustitelného souboru Monopoly.exe. Po spuštění se uživatel dle jeho volby připojí na jeden server ze seznamu. Tento seznam serverů lze modifikovat v souboru server.config, který se nachází ve stejné složce s klientem. Barvy uživatele informují o stavu serveru. Zelená barva informuje, že server je aktivní a je možno úspěšně navázat spojení. Naopak červená barva informuje, že nebylo možné se serverem navázat spojení a klient uživateli nedovolí se na něj připojit (Obr. 16).



Obr. 16. Připojení k serveru [zdroj: vlastní].

6.3.1 Registrace a přihlášení

Po úspěšném připojení k serveru je potřeba, aby se uživatel přihlásil na svůj účet. V případě, že účet ještě na daném serveru nemá, je nutná nejprve registrace. Pro zjednodušení aplikace stačí do textových polí zadat jen přezdívku pod kterou chce uživatel vystupovat a heslo. Potom již stačí kliknout jen na tlačítko registrace, kterým pošle požadavek na server. Ten následně pomocí SQL dotazů na databázi ověří, zda přezdívka již není zabraná jiným uživatelem a pokud není, tak úspěšně uživatele zaregistruje jako je tomu na Obr. 17. V opačném případě server požádá uživatele o volbu jiné přezdívky a proces opakovat. Díky tomu je zajištěná unikátnost přezdívek uživatelů na serveru. Pomocí tlačítka přihlášení se uživatel přihlásí na svůj účet. Případně server se dá změnit po kliknutím na horní pravý červené tlačítko, které vyvolá nabídku serverů jako je tomu na Obr. 16.



Obr. 17. Registrace nového uživatele [zdroj: vlastní].

6.3.2 Menu

V menu (Obr. 18) si uživatel může vybrat mezi třemi položkami. Po stisknutí tlačítka Hrát se zobrazí nové okno hlavního kanálu. Po stisknutí tlačítka Statistiky se zobrazí nové okno se statistikami historií ze všech odehraných zápasů. Tlačítko Odejít slouží k ukončení klienta.



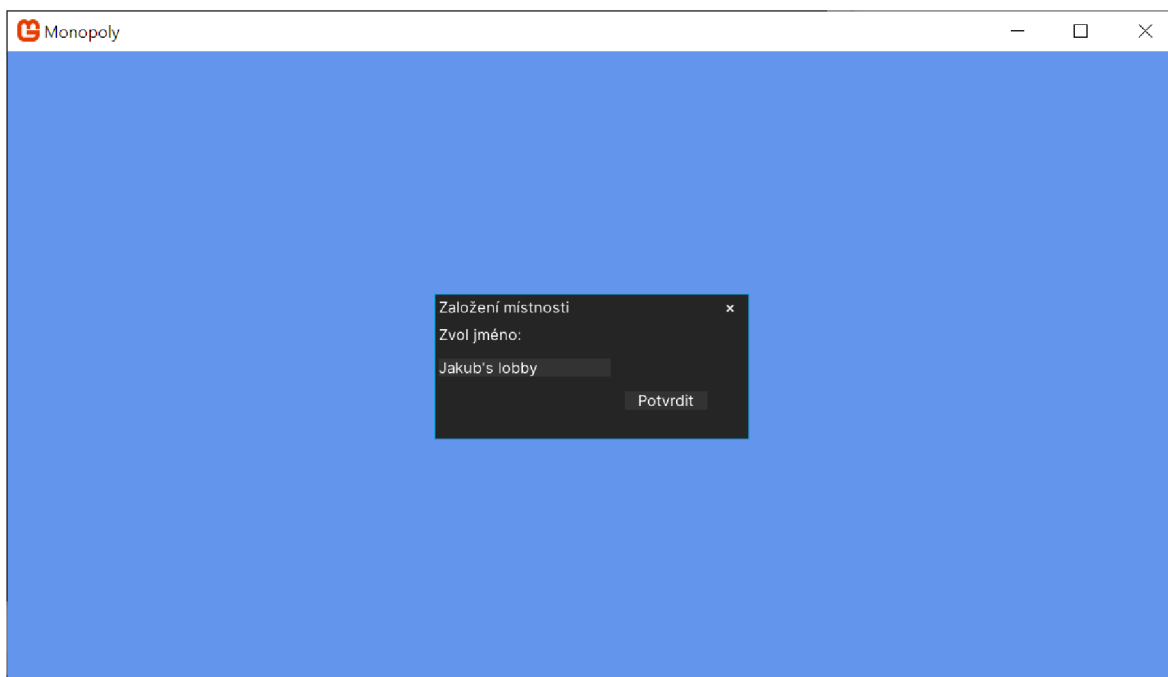
Obr. 18. Menu klienta [zdroj: vlastní].

6.3.3 Hlavní kanál

Uprostřed v okně hlavního kanálu (*Obr. 19*) je k vidění seznam všech hráčů, kteří se nacházejí na serveru. Jména hráčů jsou zobrazena bíle, nebo červeně. Červeně napsaní hráči informují klienta o tom, že tyto hráči jsou momentálně buď ve hře anebo v lobby. Bílá barva u hráčů naopak značí, že se nacházejí v hlavním kanálu a jsou k dispozici pro hru. Na pravé straně okna je umístěn chat, přes který si hráči mohou psát. Zajímavá je pak funkce pro posílání soukromých zpráv. To se hodí pokud hráč chce napsat jinému hráči, který se momentálně nachází v lobby nebo ve hře anebo pokud nechce, aby jeho zprávu viděli ostatní hráči. Docílí toho, když v chatu napíše */w jméno_hráče zpráva*. Stejného efektu může docílit i dvojitým klikem v seznamu hráčů na jméno hráče, kterému chce napsat zprávu a tím se do chatu automaticky tento příkaz umístí. Dále je možno v tomto kanále vidět všechny založené lobby. Každá lobby má přiřazen název a počet hráčů v ní připojených. Připojit se do lobby je možné buď dvojitým kliknutím na danou lobby nebo pomocí označení lobby a následného kliknutí na tlačítko připojit. Také je možné lobby založit; k tomu slouží tlačítko Založit, kde je následně uživatel vyzván ke zvolení názvu lobby (*Obr. 20*). Pomocí tlačítka Opustit se klient vrátí do Menu.



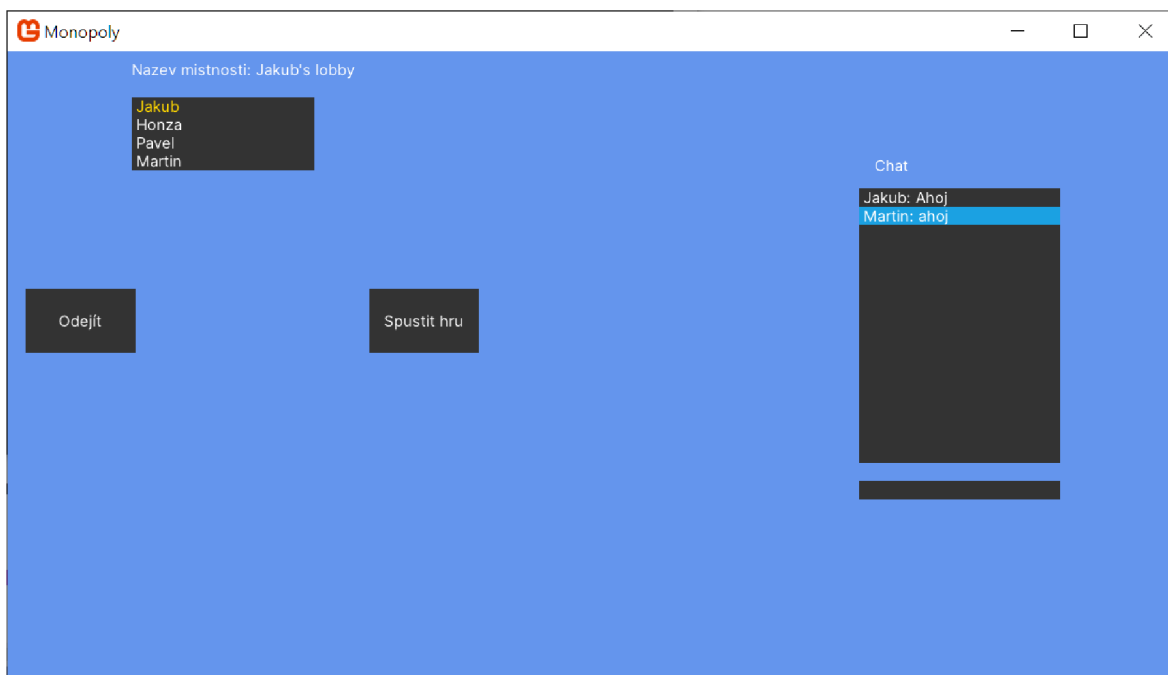
Obr. 19. Hlavní kanál [zdroj: vlastní].



Obr. 20. Založení lobby [zdroj: vlastní].

6.3.4 Lobby

Po vytvoření nebo připojení do lobby se klient přesune do nově vytvořeného kanálu (Obr. 21). V tomto kanále uživatel vidí seznam hráčů, kteří jsou připojeni v této lobby. Dále má lobby chat, přes který si hráči v lobby mohou psát. Spustit hru může pouze master, který je v seznamu hráčů označený žlutou barvou. Master připadne hráči, který je nejdéle připojený v lobby, proto při založení lobby se master stává hráč, který vytvořil danou lobby.



Obr. 21. Lobby [zdroj: vlastní].

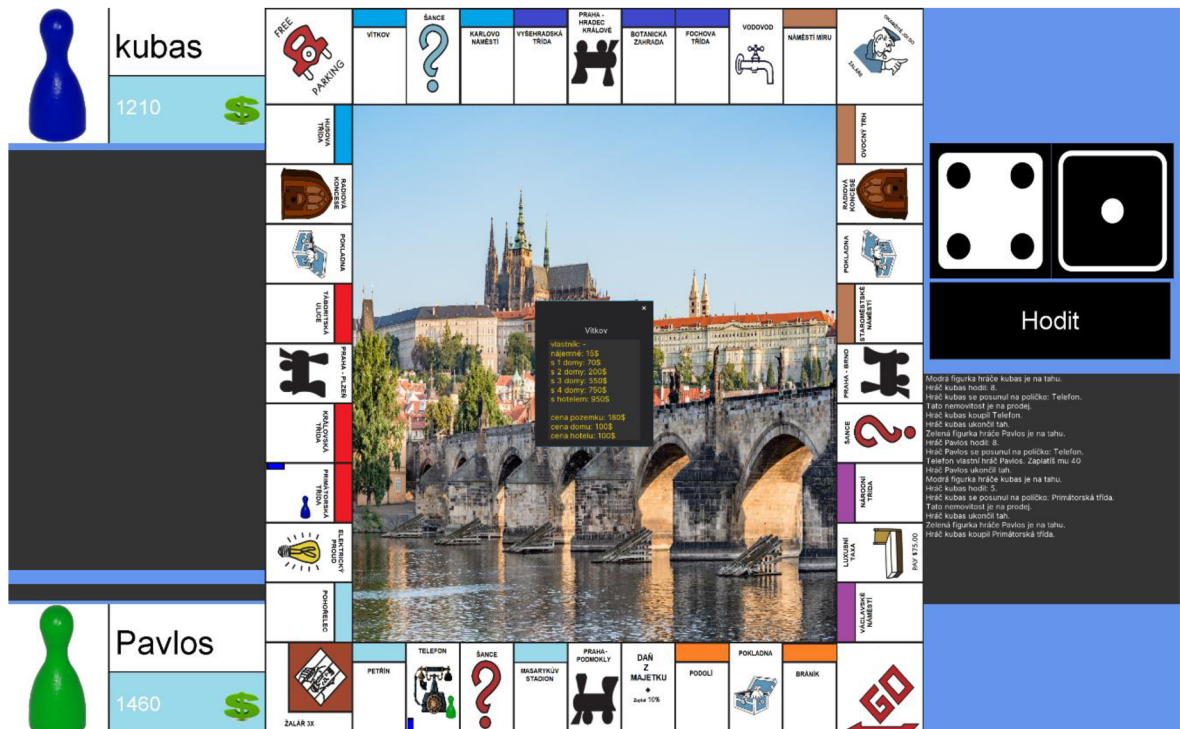
6.3.5 Hra

Po spuštění hry se objeví nové okno s inicializovanou hrou (Obr. 22). Každému hráči je náhodně přiřazena jedna z následujících barev: modrá, červená, žlutá, zelená. Touto barvou je vykreslena hráčova figurka a následně jsou i touto barvou označovány jeho zakoupené nemovitosti a domky. Pod přezdívkou hráče je zobrazen stav jeho financí. Na levé straně je umístěn chat, přes který si hráči mohou psát. Hráči se dle pravidel Monopoly střídají v tahu. Když je hráč na tahu, je vyzván k hození kostkami. Po kliknutí na tlačítko Hodit server vygeneruje dvě čísla, která se zobrazí na kostkách. Figurka se na hracím plátně začne posouvat a zastaví se na daném políčku. Poté je hráč vyzván k akci v závislosti na události daného políčka. Například na Obr. 22 modrá figurka vstoupila na políčko šance. Server náhodně vybral z určité kolekce karet šancí jednu kartu, kterou provedl. Průběh hry se promítá do pravého informativního seznamu. Dále je možné na hracím plátně kliknout na kterékoliv ze 36 herních políček. V takovém případě se zobrazí malé okénko s podrobnějšími informacemi o daném políčku například pokud hráč klikne na políčko Vítkov (Obr. 23), zobrazí se informativní okénko o cenách pozemku. Hráč končí hru v okamžiku, když už nemá dostatek financí k zaplacení pokut (včetně hodnot jeho pozemků). Vítězem se stává hráč, kterému se podaří vydržet ve hře co nejdéle. Po ukončení

hry server zapíše výsledek hráče do databáze jako je například čas strávený ve hře, počet kol, hodnota majetku. Klient se následně vrátí do menu.



Obr. 22. Hra [zdroj: vlastní].



Obr. 23. Hra po kliknutí na políčko Vítkov [zdroj: vlastní].

7 Pravidla hry Monopoly

Monopoly je patrně jedna z komerčně nejúspěšnějších deskových her. Hra byla patentována roku 1943 Charlesem Darrowem a dodnes se těší velké oblibě [71].

Pravidla hry jsou vcelku jednoduchá a mnohému jistě známa. Hra obsahuje celkem 36 políček, kde každé políčko reprezentuje nějakou událost. Hráči se pravidelně střídají v tazích. Hráč, který je na tahu hodí kostkami a o jejich součet se hráčova figurka posune na dané políčko a hráč provede příslušnou akci, která k danému políčku patří jako je například koupě pozemku, domu, hotelu, platba nájemného vlastníkovi pozemku či provedení příslušné akce na kartách šance a pokladna. Hra končí v okamžiku, kdy dojde k finančnímu úpadku ostatních hráčů [72].

8 Architektura aplikace

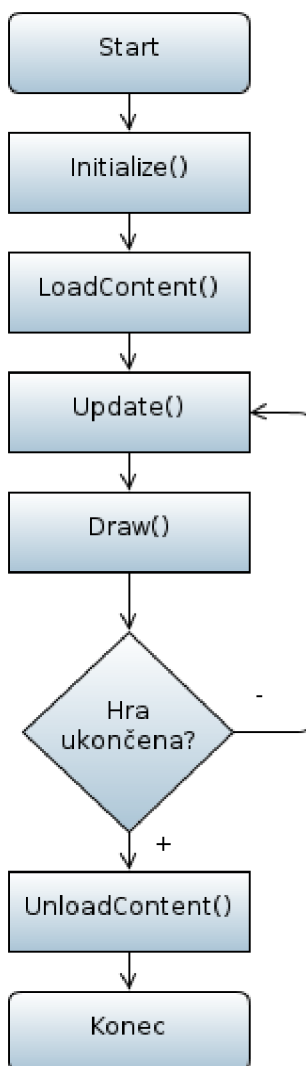
Pro snadnější implementaci hry Monopoly bylo nutné na klientské části vybrat vhodný herní framework a návrhový vzor. Na serverové části bylo nutné vyřešit problém obsluhy požadavků od klientů v nejkratším čase.

8.1 Výběr frameworku

Celá hra je napsána v programovacím jazyku C# využívající framework MonoGame. Hlavním důvodem této volby bylo, že nabízí jednoduché bezplatné rozhraní pro tvorbu her. Za zmínku stojí, že MonoGame není herní engine což znamená, že neobsahuje modul pro fyziku a kolize. Jedná se o open source herní framework, který je určen pro vývoj 2D a částečně i pro 3D her. Je volným pokračovatelem frameworku XNA, který byl do roku 2013 vyvíjen společností Microsoft. Zajišťuje podporu široké škály platforem: Windows, Linux, PS4, Xbox One, Switch, iOS, Android, PSVita a MacOS [73].

Základ hry je tvořen herní smyčkou, která je uvedena na Obr. 24. V prvním kroku při spuštění hry dojde k zavolání metody Initialize(). Tato metoda má na starosti vytvoření instancí všech potřebných objektů. Poté je volána metoda LoadContent(), která má na starost načtení obsahu jako jsou např. sprity, zvuky, textury a modely. Pak program vstupuje do nekonečného herního cyklu, ve kterém jsou volány metody Update() a Draw(). Metoda Update() implementuje logiku hry v reálném čase. Ve většině případů se jedná o indikování stavu kláves na ovladači jako je klávesnice nebo myš. Dále se stará o kolize a pohyb objektů ve hře. Metoda Draw() se stará o vykreslování herní scény. Obě tyto metody jsou programem

volány šedesátkrát za vteřinu. Při ukončení hry je zavolána metoda UnLoadContent(), která vyřeší uvolnění zdrojů hry [74].



Obr. 24. Herní smyčka [74].

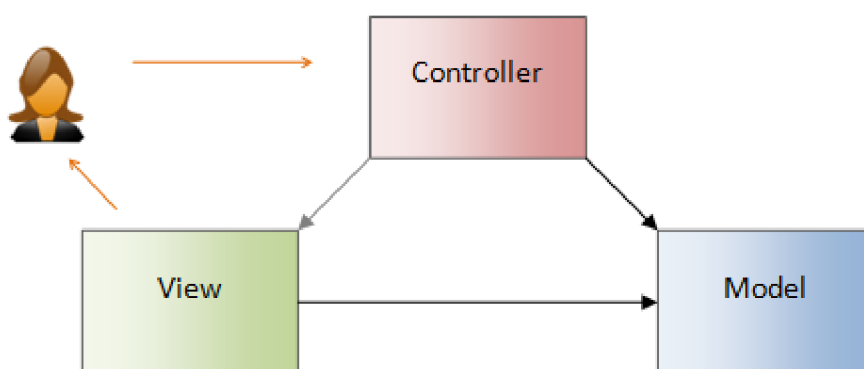
8.2 Model-view-controller (MVC)

V implementaci hry byl využit návrhový vzor model-view-controller zkráceně MVC. Jedná se o rozdělení aplikace na tři logicky samostatné komponenty (třídy). Označovány jako Model, View a Controller (Obr. 25) [75].

Model se stará o logiku hry jako jsou například různé výpočty, nebo databázové dotazy. Konkrétně v demonstrativní hře jsou v modelu definovány kromě jiného i všechny vlastnosti jednotlivých políček na hracím plátně jako jsou například jejich typy. Tím je myšleno jestli se jedná o políčko šance, ulice, vlak apod [75].

View má na starost grafický výstup dat. V demonstrativní hře je to například zobrazení hracího plátna a pohybu figurky hráče [75].

Controller propojuje komponenty model a view. Zároveň s touto komponentou komunikuje i uživatel. Controller tak zajišťuje funkčnost celé aplikace. Má na starosti aktualizaci modelu na jehož základě se překreslí View komponenta. Například v demonstrativní aplikaci jsou v controlleru definovány všechny možné stavy hráče. Jeden z nich se například týká toho zda je daný hráč na řadě a v závislosti na tom mu hra dovolí hodit kostkami [75][76].



Obr. 25. Vzor MVC [76].

8.3 Thread pooling

Server na daném portu naslouchá na připojení od klientů. Naslouchání požadavků od klientů je blokující operace. To znamená, že program zamrzne do té doby, než nějaký klient naváže spojení nebo nedojde k vypnutí programu. Proto by naslouchání mělo probíhat ve vlastním vlákne. Vytváření nových vláken ovšem vyžaduje určitý čas.

Nastává proto problém, jakým způsobem obsloužit co nejvíce klientů tak, aby doba od přijetí požadavku do jeho vyřízení byla co nejmenší a zároveň tak, aby se efektivně využívaly zdroje serveru [70].

Jedno možné řešení tohoto problému by bylo pro každého klienta vytvořit vlastní vlákno, ve kterém by probíhala daná obsluha. V praxi by to znamenalo, že by pro každé přijetí nového spojení bylo vytvořeno nové vlákno, kterému by za parametr bylo předáno toto spojení. Tohle řešení by bylo postačující pro menší aplikace s menším datovým tokem, tj. pro aplikace, na které by se nepřipojovalo mnoho klientů; bylo by i jednoduché na implementaci, ovšem přináší s sebou i řadu nevýhod [70].

V situacích, kdyby se na server začalo připojovat mnoho klientů by počet vláken rychle narůstal a celková odezva by začala klesat. Je také potřeba poznamenat, že počet vláken je operačním systémem omezen. Další nevýhodou tohoto řešení by mohlo být v okamžiku, kdyby na serveru bylo připojeno relativně málo klientů, ale docházelo by často k navazování a ukončování spojení. Jak již je zmíněno výše, vytváření nových vláken a jejich následné uvolňování je časově náročná operace [70].

Dalším, a to už lepším řešením obsluhy klientů je tzv. thread pooling. Požadavky od klientů se přidávají na konec fronty, a to v závislosti na pořadí, ve kterém přišli. Před samotným spuštěním serveru by bylo dobré vytvořit určitou množinu vláken, které by přistupovaly do fronty a obsloužily by vždy první požadavek a následně jej z fronty odebraly a přesunuly se na další. Zároveň je důležité zajistit, aby celý tento proces probíhal synchronizovaně tak, aby se nestalo, že dvě vlákna budou z fronty obsluhovat totožný požadavek. Tento způsob řešení částečně odstraňuje problém s neustálým vytvářením a uvolňováním vláken po připojení nového klienta. Tudíž klienti se mohou libovolně na server připojovat a odpojovat a množina vláken zůstane stejná [70].

Není potřeba, aby si uživatel v C# sám vytvářel takovou množinu vláken. Vše je již vytvořeno v oboru názvů `System.Threading` ve třídě `ThreadPool`, která slouží k jejich řízení. Tato třída mění v množině počet vláken dle potřeby, a to případně až ke zvolenému maximu. V takovém případě se nové požadavky od klientů udržují ve frontě a čekají na okamžik, kdy některé vlákno dokončí předchozí úlohu. Požadavek k obsluze se vloží do fronty jako parametr u volání statické metody `QueueUserWorkItem()` této třídy [70].

9 Testování

Klienti se připojovali na servery běžící na virtuálních strojích hostované od společností Microsoft Azure a Amazon s konfiguracemi, které jsou uvedeny v Tab. 1. Konkrétně od poskytovatele Microsoft Azure byla využita jejich bezplatná nabídka, která zahrnovala po dobu 30 dnů 2vCPU, 8 GB RAM, uložení o 50 GB s operačním systémem Windows. Od společnosti Amazon byla využita jejich bezplatná nabídka s konfigurací 1 vCPU, 1 GB RAM s operačním systémem Windows, poskytovanou na dobu jednoho roku.

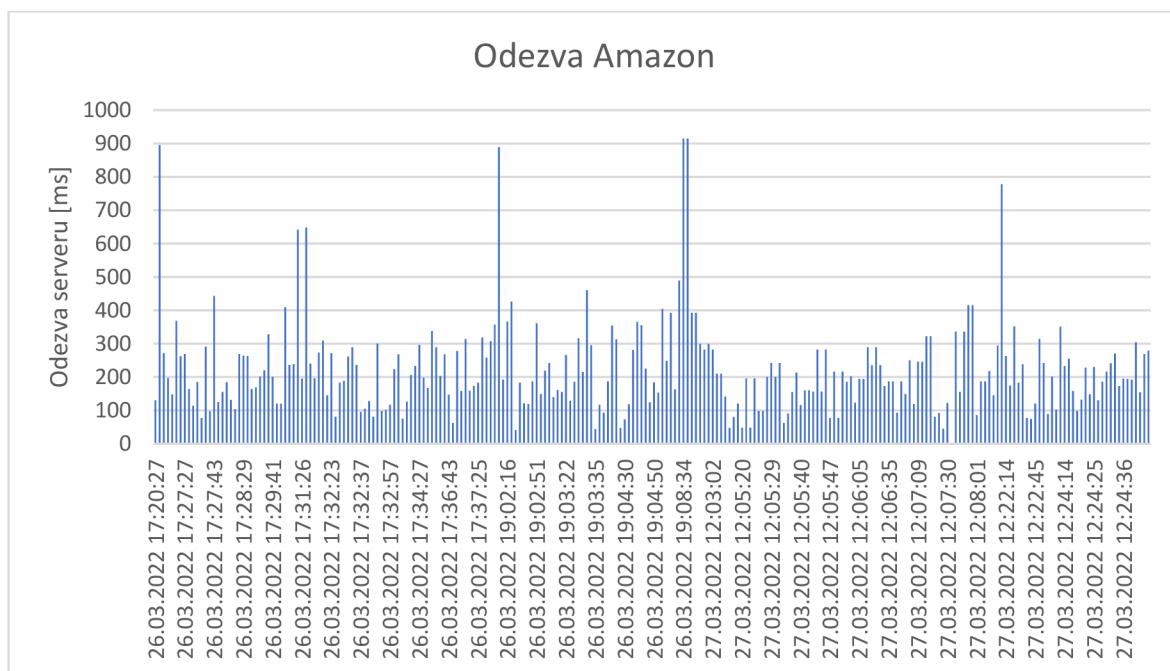
Oba virtuální stroje byly hostovány v datovém centru ve Frankfurtu tak, aby byly zajištěny pokud možno shodné podmínky. Tyto bezplatné nabídky od cloudových společností byly zvoleny záměrně tak, aby se provedlo srovnání mezi poskytovatelem cloudové služby, který krátkodobě z bezplatných nabídek na trhu poskytuje nejvyšší výkon

a poskytovatelem, který nabízí řádově nižší výkon ovšem na delší dobu. Zjišťovala se především odezva serverů na vyřízení požadavků od klientů.

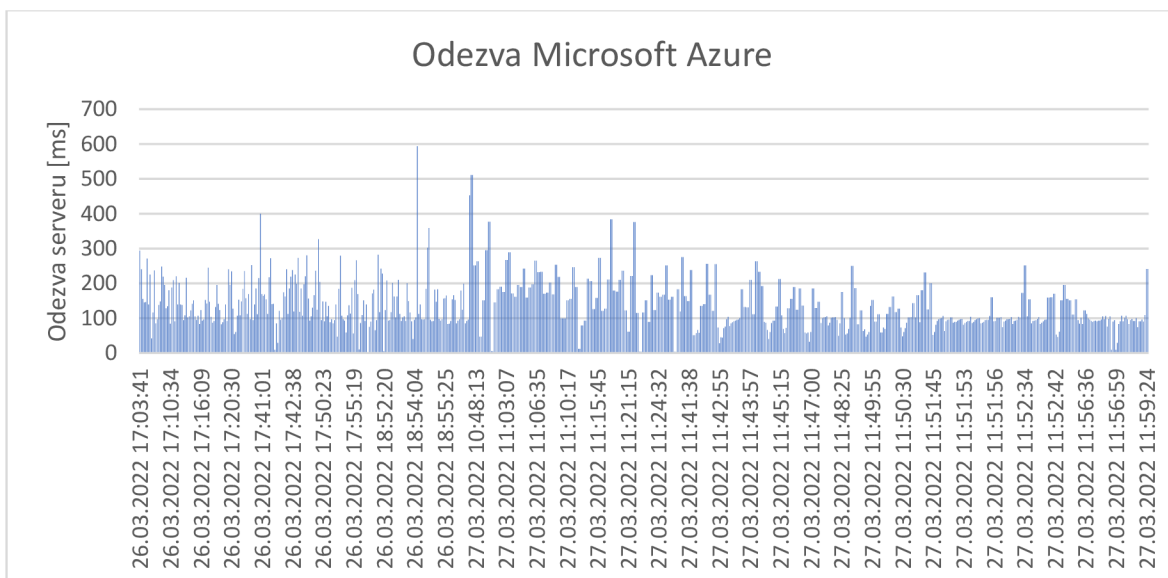
Pro hosting MySQL databáze pro servery byla zvolena společnost Amazon, která nabízí podle Tab. 2. **Chyba! Nenalezen zdroj odkazů.** nejlepší bezplatnou nabídku.

9.1 Odezva serverů

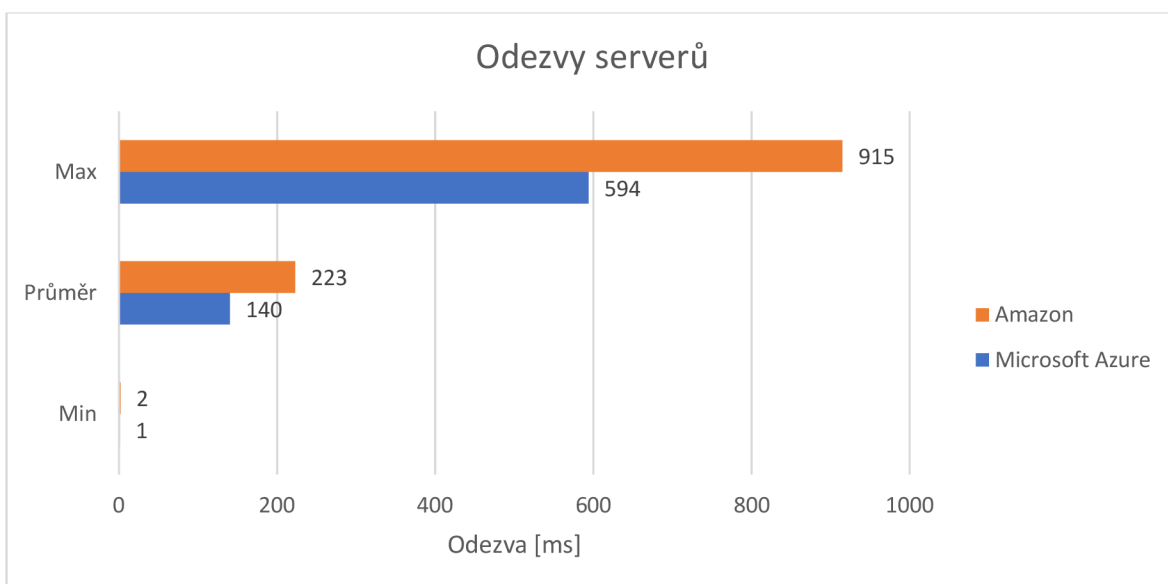
Graf 1 a Graf 2 ukazují naměřené odezvy serverů (Graf 1 pro Amazon a Graf 2 pro Microsoft Azure) pocházející od více hráčů (klientů); jak je z těchto grafů patrné, testování demonstrační aplikace probíhalo ve dnech 26.3.2022 od 17 hodin do 19 hodin a poté další den, tj. 27.3.2022 zhruba od 10 do 12 hodin. Graf 3 ukazuje ucelenější data, ze kterých jasně vyplývá, že s odezvou je na tom lépe server běžící na výkonnějším virtuálním stroji od společnosti Microsoft Azure.



Graf 1. Odezva serveru hostovaném na Amazonu.



Graf 2. Odezva serveru hostovaném na Microsoft Azure.



Graf 3. Souhrn hodnot odezev serverů.

9.2 Názory hráčů

Hráči si především stěžovali na vysokou nestabilitu klientské aplikace. Ve hře docházelo u jednotlivých hráčů k náhodnému mizení grafického uživatelského rozhraní jako je chat, informační pole a tlačítko na hození kostek. To také způsobovalo problémy s ukončením a zamrznutím aplikace. Často přihlašování hráči poukazovali na to, že si aplikace sama nedoplňuje do textových polí jejich přihlašovací údaje použité z poledního přihlášení. Další výčitky hráčů byly na nemožnost psaní diakritiky do chatu, nevhodné rozložení barev, omezenou délku přezdívky hráče na sedm znaků, absenci instalačního

souboru, špatnou viditelnost domků ve hře a na vykreslování textu mimo tlačítko nacházející se na přihlašovací okně. Pro dobrý prožitek ve hře jeden hráč doporučuje monitor s poměrem stran 16:9.

10 Závěr

Cílem této bakalářské práce bylo analyzovat stávající nabízená cloud řešení pro realizaci multiplayer hry. Byla popsána rozdělení cloudových služeb a analyzovány bezplatné nabídky cloudových leaderů pro pronajmutí virtuálního stroje a relační databáze. Dále byl popsán komunikační model TCP/IP a knihovny s ním spojené v .NET frameworku pro programovací jazyk C#.

Testující aplikace se skládá ze tří samostatných částí. Databáze, ve které jsou uložena data o hráčích a o proběhnutých hrách. Část server je konzolová aplikace, která se stará o obsluhu požadavků od klientů a komunikuje s relační databází MySQL. Třetí část demonstrativní aplikace je klient s grafickým uživatelským rozhraním s implementovanou deskovou hrou Monopoly, který navazuje spojení se serverem.

V praktické části byly otestovány dva cloudové hostiny, které poskytovaly cloudové řešení pro testovací server s víceuživatelským přístupem a větším datovým tokem pro demonstrační multiplayer hru a relační databázi. Výsledný příslušný program, který byl jedním z cílů bakalářské práce, je uveden v příloze a současně je nahraný na připojeném CD.

11 Seznam použitých zdrojů

- [1] *Co je to cloud computing? - Správa.sítě.eu* [online]. [vid. 2022-01-01]. Dostupné z: <https://www.sprava-site.eu/cloud-computing/>
- [2] *Cloudové služby: data i počítače v oblacích | Ikaros* [online]. [vid. 2022-01-01]. Dostupné z: <https://ikaros.cz/cloudove-sluzby-data-i-pocitace-v-oblacich>
- [3] *Co je cloud computing? Průvodce pro začátečníky | Microsoft Azure* [online]. [vid. 2022-01-01]. Dostupné z: <https://azure.microsoft.com/cs-cz/overview/what-is-cloud-computing/#benefits>
- [4] *cloud_computing.jpg (628×403)* [online]. [vid. 2022-01-01]. Dostupné z: https://m4rgot.com/wp-content/uploads/2015/02/cloud_computing.jpg
- [5] *Cloud Computing: Co to je a komu se vyplatí | Algotech.cz* [online]. [vid. 2022-01-01]. Dostupné z: <https://www.algotech.cz/novinky/2020-04-21-cloud-computing-co-to-je-a-komu-se-vyplati>
- [6] *Final Version of NIST Cloud Computing Definition Published | NIST* [online]. [vid. 2022-01-01]. Dostupné z: <https://www.nist.gov/news-events/news/2011/10/final-version-nist-cloud-computing-definition-published>
- [7] *Cloud - ManagementMania.com* [online]. [vid. 2022-01-01]. Dostupné z: <https://managementmania.com/cs/cloud-computing>
- [8] *Tři modely služeb Cloud Computing | OTEVŘENO* [online]. [vid. 2022-01-01]. Dostupné z: <https://www.openintl.com/the-three-service-models-of-cloud-computing/>
- [9] *cloudcomputing-2.png (2951×1463)* [online]. [vid. 2022-01-01]. Dostupné z: <https://www.cetic.be/IMG/png/cloudcomputing-2.png>
- [10] *Co je to IaaS? - Správa.sítě.eu* [online]. [vid. 2022-01-01]. Dostupné z: <https://www.sprava-site.eu/iaas/>
- [11] *IaaS, PaaS a SaaS aneb V čem se liší služby „as a Service“* [online]. [vid. 2022-01-01]. Dostupné z: <https://www.master.cz/blog/iaas-paas-a-saas-aneb-v-cem-se-lisi-sluzby-as-a-service/>

- [12] *Internet věci ve výuce* [online]. [vid. 2022-01-01]. Dostupné z: <https://www.vovcr.cz/odz/tech/229/page13.html>
- [13] *IaaS vs. PaaS vs. SaaS | IBM* [online]. [vid. 2022-01-01]. Dostupné z: <https://www.ibm.com/cloud/learn/iaas-paas-saas>
- [14] *Co je PaaS? Platforma jako definice služby [Příklady v ceně]* [online]. [vid. 2022-01-01]. Dostupné z: <https://searchcloudcomputing.techtarget.com/definition/Platform-as-a-Service-PaaS>
- [15] *SaaS vs PaaS vs IaaS: Jaký je rozdíl a jak si vybrat – BMC Software | blogy* [online]. [vid. 2022-01-01]. Dostupné z: <https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/>
- [16] *Definice cloudu - MetaCentrum* [online]. [vid. 2022-01-01]. Dostupné z: https://wiki.metacentrum.cz/wiki/Definice_cloudu
- [17] *Veřejný cloud versus privátní cloud versus hybridní cloud | Microsoft Azure* [online]. [vid. 2022-01-01]. Dostupné z: <https://azure.microsoft.com/cs-cz/overview/what-are-private-public-hybrid-clouds/#deployment-options>
- [18] *Co je veřejný cloud? - Správa.sítě.eu* [online]. [vid. 2022-01-01]. Dostupné z: <https://www.sprava-site.eu/verejny-cloud/>
- [19] *Cloud hosting - co to je a čím se liší od VPS - wpmax.cz* [online]. [vid. 2022-01-01]. Dostupné z: <https://www.wpmax.cz/cloud-hosting-public-co-to-je-a-cim-se-lisi-od-vps/>
- [20] *6 nejlepších zcela ZDARMA cloudových hostingových služeb [AKTUALIZACE 2022]* [online]. [vid. 2022-01-02]. Dostupné z: <https://www.websiteplanet.com/blog/best-free-cloud-hosting-services/>
- [21] *Amazon Web Services (AWS) Review 2022 – Is It Overrated?* [online]. [vid. 2022-01-02]. Dostupné z: <https://www.websiteplanet.com/web-hosting/amazon-aws/#overview>
- [22] *Cenová kalkulačka | Microsoft Azure* [online]. [vid. 2022-01-02]. Dostupné z: <https://azure.microsoft.com/cs-cz/pricing/calculator/#virtual-machines>

- [23] *Google Cloud Pricing Calculator* [online]. [vid. 2022-01-02]. Dostupné z: <https://cloud.google.com/products/calculator#id=c70eca6e-f81a-4396-a389-d1a73eef54c0>
- [24] *Cloud Price List | Oracle* [online]. [vid. 2022-02-12]. Dostupné z: <https://www.oracle.com/cloud/price-list.html>
- [25] *IBM Cloud Infrastructure* [online]. [vid. 2022-01-02]. Dostupné z: <https://cloud.ibm.com/vpc-ext/provision/vs>
- [26] *[2022 Best] 6 Cloud Computing Services to Get Free VM - Super Easy* [online]. [vid. 2022-01-02]. Dostupné z: <https://www.supereasy.com/100-free-6-best-cloud-service-providers/>
- [27] TIMOTHY BORONCZYK. *MySQL Okamžitě - Ovládněte MySQL za vikend*. B.m.: CPress, 2016.
- [28] *Co jsou databáze | Microsoft Azure* [online]. [vid. 2022-02-18]. Dostupné z: <https://azure.microsoft.com/cs-cz/overview/what-are-databases/>
- [29] *Relační databáze vs. nerelační databáze: Čím se liší? | MasterDC* [online]. [vid. 2022-02-18]. Dostupné z: <https://www.master.cz/blog/relacni-databaze-nerelacni-databaze-jake-jsou-rozdily/>
- [30] *4-0.png (586×199)* [online]. [vid. 2022-02-18]. Dostupné z: <https://docplayer.cz/docs-images/18/665006/images/4-0.png>
- [31] *Objektové databáze - Linux E X P R E S* [online]. [vid. 2022-03-21]. Dostupné z: <https://www.linuxexpres.cz/business/objektove-databaze>
- [32] Přednáška základy databází VOTRUBA, Zdeněk. *Databáze*
- [33] *Rozdíl mezi RDBMS a ORDBMS | Porovnejte rozdíl mezi podobnými podmínkami - Technologie - 2022* [online]. [vid. 2022-03-21]. Dostupné z: <https://cs.strephonsays.com/rdbms-and-vs-ordbms-545>
- [34] *Kterou SQL databázi použít?* [online]. [vid. 2022-02-18]. Dostupné z: <https://www.itnetwork.cz/sql/kterou-sql-databazi-pouzit>

- [35] *10-aplikasi-database-terbaik-yang-bisa-anda-coba-mana-yang-terbaik_2.png* (340×175) [online]. [vid. 2022-02-18]. Dostupné z: https://altitudetvm.com/images/website/10-aplikasi-database-terbaik-yang-bisa-anda-coba-mana-yang-terbaik_2.png
- [36] *SQLite – Vyhledávání Google* [online]. [vid. 2022-02-18]. Dostupné z: <https://upload.wikimedia.org/wikipedia/commons/3/38/SQLite370.svg>
- [37] *MySQL-Logo.png* (3000×2000) [online]. [vid. 2022-02-18]. Dostupné z: <https://cdn.liveagent.com/app/uploads/2020/11/MySQL-Logo.png>
- [38] *Shluky B-Series – Azure Virtual Machines - Azure Virtual Machines | Microsoft Docs* [online]. [vid. 2022-02-19]. Dostupné z: <https://docs.microsoft.com/cs-cz/azure/virtual-machines/sizes-b-series-burstable>
- [39] *Azure Database for MySQL – Flexible Server: Azure Free offer coming soon | Azure updates | Microsoft Azure* [online]. [vid. 2022-02-19]. Dostupné z: <https://azure.microsoft.com/en-us/updates/azure-database-for-mysql-flexible-server-azure-free-offer-coming-soon/>
- [40] *Free MySQL Hosting* [online]. [vid. 2022-02-19]. Dostupné z: <https://www.freemysqlhosting.net/>
- [41] *Free MySQL Hosting with Remote Access - List of Providers - Joomguide* [online]. [vid. 2022-02-19]. Dostupné z: <https://www.joomguide.com/free-mysql-hosting-remote-access-list-providers/>
- [42] *Amazon RDS | Cloudová relační databáze | Webové služby Amazon* [online]. [vid. 2022-02-19]. Dostupné z: <https://aws.amazon.com/rds/>
- [43] *Free Remote MySQL* [online]. [vid. 2022-02-19]. Dostupné z: <https://remotemysql.com/>
- [44] *Top free 4 Hosts that provide Remote Access to MySQL Database - Radix Code* [online]. [vid. 2022-02-19]. Dostupné z: <https://radixcode.com/top-free-4-hosts-that-provide-remote-access-to-mysql-database>
- [45] *db4free.net - MySQL Database for free* [online]. [vid. 2022-02-19]. Dostupné z: <https://www.db4free.net/>
- [46] *Je potřeba k připojení k databázi pomocí php program mysql workbench? - poradna*

- Živě.cz* [online]. [vid. 2022-02-20]. Dostupné z: <https://www.zive.cz/poradna/je-potreba-k-pripojeni-k-databazi-pomoci-php-program-mysql-workbench/sc-20-cq-664794/default.aspx?consultanswers=1>
- [47] *Lekce 1 - MySQL krok za krokem: Úvod do MySQL a příprava prostředí* [online]. [vid. 2022-02-19]. Dostupné z: <https://www.itnetwork.cz/mysql/mysql-tutorial-uvod-a-priprava-prostredi>
- [48] *phpMyAdmin* [online]. [vid. 2022-02-19]. Dostupné z: <https://www.phpmyadmin.net/>
- [49] *Lekce 2 - MySQL krok za krokem: Vytvoření databáze a tabulky* [online]. [vid. 2022-02-19]. Dostupné z: <https://www.itnetwork.cz/mysql/mysql-tutorial-vytvoreni-databaze-a-tabulky>
- [50] *phpMyAdmin nebo Adminer? - Parse-error.cz - nikdy nevíš, kdy to přijde* [online]. [vid. 2022-02-19]. Dostupné z: <https://www.parse-error.cz/clanek/1/phpmyadmin-nebo-adminer>
- [51] *Adminer - to chceš - Kutáč.cz* [online]. [vid. 2022-02-19]. Dostupné z: <https://www.kutac.cz/weby-a-vse-okolo/adminer-to-chces>
- [52] *Adminer - Čím je lepší než phpMyAdmin?* [online]. [vid. 2022-02-19]. Dostupné z: <https://www.adminer.org/cs/phpmyadmin/>
- [53] *Pozor na zneužití nástroje Adminer – BLOG GIGASERVER.CZ* [online]. [vid. 2022-02-19]. Dostupné z: <https://blog.gigaserver.cz/pozor-na-zneuziti-nastroje-adminer/>
- [54] *chrome_2016-12-05_18-52-22.png (1184×635)* [online]. [vid. 2022-02-19]. Dostupné z: https://wiki.hicoria.com/wp-content/uploads/2016/12/chrome_2016-12-05_18-52-22.png

- [55] *Modelování MySQL databáze pomocí MySQL Workbench* [online]. [vid. 2022-02-20]. Dostupné z: <https://www.abclinuxu.cz/clanky/modelovani-mysql-databaze-pomoci-mysql-workbench#!/-1/>
- [56] *MySQL :: MySQL Workbench Manual :: 1 General Information* [online]. [vid. 2022-02-20]. Dostupné z: <https://dev.mysql.com/doc/workbench/en/wb-intro.html>
- [57] *MySQL-workbench-80-CE-EER-diagram-for-a-local-patient-database-see-online-version-for.ppm (751×514)* [online]. [vid. 2022-02-20]. Dostupné z: <https://www.researchgate.net/profile/Ebtsam-Adel-2/publication/341170171/figure/fig12/AS:902214281334799@1592116055699/MySQL-workbench-80-CE-EER-diagram-for-a-local-patient-database-see-online-version-for.ppm>
- [58] *DataGrip pro MySQL* [online]. [vid. 2022-02-20]. Dostupné z: <https://www.jetbrains.com/datagrip/features/mysql.html>
- [59] STEPHEN J. BIGELOW. *Mistrovství v počítačových sítích*. B.m.: Computer Press, 2004.
- [60] *Jiří Peterka: Referenční model ISO/OSI* [online]. [vid. 2022-03-31]. Dostupné z: <https://www.earchiv.cz/anovinky/ai1552.php3>
- [61] *Jiří Peterka: Síťový model TCP/IP* [online]. [vid. 2022-03-31]. Dostupné z: <https://www.earchiv.cz/a92/a231c110.php3>
- [62] *Jiří Peterka: Síťový model TCP/IP* [online]. [vid. 2022-03-08]. Dostupné z: <https://www.earchiv.cz/a92/a231c110.php3>
- [63] *Model TCP/IP* [online]. [vid. 2022-03-08]. Dostupné z: http://ijs2.8u.cz/index.php?option=com_content&view=article&id=14&Itemid=120
- [64] *TCP/IP – Multimediaexpo.cz* [online]. [vid. 2022-03-08]. Dostupné z: <http://www.multimediaexpo.cz/mmecz/index.php/TCP/IP>
- [65] *TCP/IP* [online]. [vid. 2022-03-09]. Dostupné z: <https://ct.upce.cz/machalik/puitk-stare/site/treti.htm>
- [66] *Sada protokolů TCP/IP* [online]. nedatováno [vid. 2022-03-09]. Dostupné z: https://moodle.sspbrno.cz/pluginfile.php/6413/mod_resource/content/1/tcpip.pdf

- [67] *TCP/IP komunikace [Kalábovi]* [online]. [vid. 2022-03-09]. Dostupné z: https://kalabovi.org/pitel:isz:tcp_ip_komunikace
- [68] *Protokol UDP 1.část - Builder.cz - Informacni server o programovani* [online]. [vid. 2022-03-09]. Dostupné z: <https://www.builder.cz/rubriky/c/c--/protokol-udp-1-cast-156226cz>
- [69] *Rozhraní UNIX Sockets* [online]. [vid. 2022-03-11]. Dostupné z: <http://www.cs.vsb.cz/grygarek/PS/sockets.html>
- [70] NAGEL, Christian, Bill EVJEN, Jay GLYNN, Karli WATSON a Morgan SKINNER. *C # 2008 Programujeme profesionálně*. nedatováno. ISBN 9788025124017.
- [71] *Jak se psala historie: Monopoly* [online]. [vid. 2022-03-23]. Dostupné z: <https://www.alza.cz/hracky/article/10507.htm>
- [72] *Monopoly* [online]. [vid. 2022-03-23]. Dostupné z: <http://www.deskovehry.info/pravidla/monopoly.htm>
- [73] *MonoGame - Úvod a instalace* [online]. [vid. 2022-03-22]. Dostupné z: <https://www.itnetwork.cz/csharp/monogame/xna-game-studio-c-sharp-uvod-a-instalace>
- [74] *Lekce 1 - Struktura MonoGame hry* [online]. [vid. 2022-03-22]. Dostupné z: <https://www.itnetwork.cz/csharp/monogame/csharp-tvorba-her-monogame-tetris/xna-tutorial-struktura-hry>
- [75] *MVC architektura* [online]. [vid. 2022-03-22]. Dostupné z: <https://www.itnetwork.cz/navrh/mvc-architektura-navrhovy-vzor>
- [76] *Prezentační vzory z rodiny MVC - Zdroják* [online]. [vid. 2022-03-22]. Dostupné z: <https://zdrojak.cz/clanky/prezentacni-vzory-zrodiny-mvc/>

12 Přílohy

CD obsahující demonstrační (testující) aplikaci