

EKONOMICKÁ UNIVERZITA V BRATISLAVE
FAKULTA HOSPODÁRSKEJ INFORMATIKY

Evidenčné číslo: 103003/I/2022/36124048424981508

Softvérové riešenie vybraných úloh teórie hier
Inžinierska práca

2022

Bc. Ján Zdarilek

EKONOMICKÁ UNIVERZITA V BRATISLAVE
FAKULTA HOSPODÁRSKEJ INFORMATIKY

Softvérové riešenie vybraných úloh teórie hier
Inžinierska práca

Študijný program: Informačný manažment
Študijný odbor: Informačný manažment
Školiace pracovisko: Katedra hospodárskej informatiky
Vedúci záverečnej práce: doc. Ing. Zuzana Čičková, PhD.

Bratislava 2022

Bc. Ján Zdarilek



Ekonomická univerzita v Bratislave
Fakulta hospodárskej informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Ján Zdarilek
Študijný program: informačný manažment (Jednoodborové štúdium, inžiniersky II. st., denná forma)
Študijný odbor: ekonómia a manažment
Typ záverečnej práce: Inžinierska záverečná práca
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Softvérové riešenie vybraných úloh teórie hier

Anotácia: Charakteristika teórie hier, hry v normálnom a hry v rozvinutom tvare, analýza softvérových možností riešenia hier. Ukážky riešenia zvolených typov hier.

Vedúci: doc. Ing. Zuzana Čičková, PhD.
Katedra: KOVE FHI - Katedra operačného výskumu a ekonometrie FHI
Vedúci katedry: prof. Mgr. Juraj Pekár, PhD.
Dátum zadania: 03.11.2020

Dátum schválenia: 05.11.2020

prof. Mgr. Juraj Pekár, PhD.
vedúci katedry

ČESTNÉ VYHLÁSENIE

Čestne vyhlasujem, že inžiniersku záverečnú prácu som vypracoval samostatne pod vedením vedúcej inžinierskej práce s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú uvedené v zozname použitej literatúry.

Dátum: 23.04.2022

.....

(podpis študenta)

Pod'akovanie: Ďakujem doc. Ing. Zuzane Čičkovej, PhD. za odborné vedenie a pomoc pri tvorbe inžinierskej práce.

Abstrakt

Zdarilek, Ján: *Softvérové riešenie vybraných úloh teórie hier*. – Ekonomická univerzita v Bratislave. Fakulta Hospodárskej informatiky; Kove FHI. – Vedúci záverečnej práce: doc. Ing. Zuzana Čičková, PhD. Bratislava: FHI EU, 2022, počet strán 52.

Inžinierska práca je zameraná na softvérové riešenia nami vybraných hier. Pri tvorbe tejto práce sme aplikovali literatúru v slovenskom, českom a aj v anglickom jazyku. Avšak kľúčové jadro práce vychádza z vlastných nápadov a spracovania. Výsledným zámerom práce je predstaviť naprogramované hry v oblasti Teórie hier. Cieľom teoretickej časti inžinierskej práce je detailné vysvetlenie pojmov, z ktorých v našej práci vychádzame. Praktická časť je zameraná na samotnú tvorbu aplikácií pomocou programovacieho jazyka Python vo vývojovom prostredí PyCharm. Okrem základných údajov o programovacom jazyku Python je táto časť práce zameraná na podrobnú analýzu vytvorených aplikácií. Detailne vysvetľuje aj zdrojový kód jednotlivých hier. Cieľom práce je analýza softvérových možností riešenia hier a aj ich spracovanie.

Kľúčové slová:

Bulo, hra pirát a Sicíľčan, teória hier

Abstract

Zdarilek, Ján: Software solutions of selected Game Theory tasks. - University of Economics in Bratislava. Faculty of Business Informatics; Kove FHI. - Thesis supervisor: doc. Ing. Zuzana Cickova, PhD. Bratislava: FHI EU, 2022, number of pages 52.

The engineering thesis is focused on software solutions for our selected games. In creating this thesis, we have applied literature in Slovak, Czech and English. However, the key core of the thesis is based on our own ideas and processing. The final aim of the thesis is to present programmed games in the field of Game Theory. The aim of the theoretical part of the engineering thesis is a detailed explanation of the concepts on which we base our work. The practical part is focused on the creation of applications using the Python programming language in the PyCharm development environment. In addition to basic data about the Python programming language, this part of the work is focused on a detailed analysis of created applications. It also explains in detail the source code of individual games. The aim of the thesis is the analysis of software options for solving games and their processing.

Keywords:

Bulo, game pirate and Sicilian, game theory

Obsah

Úvod 2

1	Súčasný stav problematiky	4
1.1	Teória hier a jej definície	4
1.2	Vznik Teórie hier	4
1.3	Koncept a základné pojmy v Teórii hier	6
1.4	Nashovo ekvilibrium	8
1.5	Konfliktné a nekonfliktné situácie	9
1.5.1	Hlavné typy konfliktných situácií.....	9
1.6	Typy hier	11
1.6.1	Hry v normálnom tvare.....	11
1.6.2	Maticové hry	12
1.6.3	Väzňova dilema	12
1.6.4	Hry v rozvinutom tvare.....	13
2	Cieľ práce	14
3	Metodika práce a metódy skúmania	16
3.1	Technológie	16
3.1.1	Hp pavilion 14-dv0003nc	16
3.1.2	Python	16
3.1.3	Popis prostredia	17
3.1.4	PyCharm	18
3.2	Definícia a príklady hier v rozvinutom a normálnom tvare	20
3.3	Analýza softvérových možností riešenia hier	21
3.3.1	Strom hry a NIM.....	22
3.3.2	Hra Buló	23
3.3.3	Hra piráta a Sicíľčana.....	24
3.4	Analýza hry pirát a Sicíľčan	24
3.4.1	Maticová hra	24
3.5	Analýza hry Buló	26
3.5.1	Strom hry Buló	26
3.5.2	Spätná indukcia.....	29
4	Výsledky práce	31
4.1	Softvérové riešenie hry pirát a Sicíľčan.....	31
4.2	Softvérové riešenie hry Buló	34

4.2.1	Analýza.....	40
4.2.2	Kód	42
Záver		47
Zoznam obrázkov.....		48
Zoznam použitej literatúry		49
Príloha A: Obsah elektronickej prílohy.....		51
Príloha B: Používateľská príručka		52

Úvod

Pod pojmom teórie hier si mnohí predstavujú kategóriu počítačových hier. Avšak táto téma skrýva oveľa viac, ako sa na prvý pohľad môže zdať. Ide o pomerne mladú vedeckú disciplínu, ale jej náznaky sa medzi ľudstvom objavili už v staroveku. Samotné hry sú dôležitou súčasťou tejto témy a poskytujú terminológiu, ktorá je zrozumiteľná pre každého človeka. Zároveň sa téma stáva atraktívnejšou a schopnou osloviť väčšiu masu ľudí, a to nie len z čisto matematického prostredia. Téma teórie hier sa týka každého z nás. Zasaňuje mnohé momenty nášho života bez toho, že by sme si to uvedomovali. Ide napríklad o politické vyjednávania, policajné výsluchy, no z bežného života je to napríklad aj vyberanie, kam ísť na večeru s priateľmi. Teória hier je hra chápaná ako konfliktná rozhodovacia situácia medzi viacerými subjektami. V princípe ide o matematický pohľad v oblasti rozhodovania sa a riešenia rozhodovacích problémov. Oblasť teórie hier je veľmi obsiahla téma a jej rozvoj, či prípadné využitie majú obrovský potenciál.

V tejto práci sa venujeme len časti z toho, čo naozaj ponúka táto krásna a zaujímavá téma. Pozrieme sa na históriu a historické osobnosti, ktoré založili a rozvinuli oblasť teórie hier. Na ich život, ale aj na publikácie, ktorými predstavili teóriu hier celému svetu. Ďalej si rozoberieme dôležité teoretické informácie, ktoré zahŕňajú všetky dôležité poznatky potrebné k hlbšiemu poznaniu a pochopeniu témy. Okrem definícií a základných údajov sa budeme venovať aj definovaniu jednotlivých pojmov, ktoré poslúžia na priblíženie témy čitateľom.

Hoci sa teória hier netýka len samotných stolových hier, v nasledujúcich stranách tejto práce práve hry tvoria dôležitý element. Hry sú zábavnou formou trávenia času, ktorá je blízka každému a je jednou zo základných ľudských činností popri učení a práci. Hra prináša jej účastníkovi uspokojenie a radosť, v prípade prehry, prirodzene, vyvoláva aj negatívne emócie. Okrem toho sú hry, najmä v detstve, školou, v ktorej sa dieťa mimovoľne učí. Tento aspekt hry sa však vzťahuje nie len na deti, ale aj na dospelých. Analýza hry Buló a taktiež hry pirát a Sicíľčan prináša hlbší pohľad na problematiku jednotlivých hier. Pozrieme sa v nej na možnosti riešenia a ich výhody, respektíve nevýhody. Na základe týchto zhrnutých poznatkov a úvah vyberieme najvhodnejšie riešenie. V časti strom hry a NIM sa venujeme aj hrám, ktoré nie sú v tejto práci softvérovo riešené, ale pomôžu nám

pri analýze nami zvolených hier. Analýzou riešení úloh si otvoríme cestu pre softvérové riešenie jednotlivých hier.

Kľúčovou časťou tejto práce je práve softvérové riešenie hier. Detailne v nej vysvetľujeme funkčnosť riešenia pre nami vybranú hru na základe uskutočnenej analýzy. Poskytujeme vysvetlenie použitých algoritmov a funkcií spolu s ukážkou kódu. Zároveň poskytujeme dokončené aplikácie, ktoré si je možné vyskúšať a zahrať.

1 Súčasný stav problematiky

1.1 Teória hier a jej definície

„Teória hier je vedná disciplína, ktorá je súčasťou operačnej analýzy a jedným z odvetví aplikovanej matematiky. Zaoberá sa modelovaním a riešením niektorých typov konfliktných situácií, v ktorých sa stretávajú záujmy ľudí, skupín ľudí alebo organizovaných systémov v najširšom zmysle slova (inštitúcia, ekonomický, politický, vojenský celok a pod.). Analyzuje predpokladané, ako aj skutočné správanie subjektov v jednotlivých typoch hier s cieľom nájsť optimálne stratégie.“ [1]

„Teória hier poskytuje všeobecné matematické metódy pre analýzu situácie, v ktorých sa dvaja jednotlivci alebo viac jednotlivcov rozhodujú tak, že ich rozhodnutia ovplyvnia stav jedného alebo druhého.“ [2]

1.2 Vznik Teórie hier

Vznik teórie hier je spojený s rokom 1928 a americkým matematikom John von Neumannom [1]. Jeho pôvodné meno bolo János Neumann. Bol maďarský matematik židovského pôvodu, ktorý značnou mierou prispel k vedným odborom ako sú kvantová fyzika, funkcionálna analýza, teória množín, ekonomika, informatika, numerická analýza, hydrodynamika, štatistika a mnoho ďalších matematických disciplín. Narodil sa v roku 1903 v Budapešti a zomrel v roku 1957 vo Washingtone D.C. Von Neumann vyrástol zo zázračného dieťaťa, v jedného z popredných svetových matematikov v dvadsiatom storočí. Bol priekopníkom teórie hier a spolu s Alanom Turingom a Claudom Shannonom bol jedným z koncepčných vynálezcov digitálneho počítača s uloženým programom [4]. Venoval sa aj hľadaniu vedeckej argumentácie *klamania* v hre poker. Popri tom spojil viaceré predošlé práce viacerých matematikov, pomocou ktorých položil základy Teórie hier ako novej aplikovanej vedy [5]. Dôležitou inováciou von Neumanna, ktorá bola základným kameňom jeho Teórie hier a aj ústredná téma jeho neskorších výskumov v tejto oblasti, bolo tvrdenie a dôkaz vety o min-maxe. V roku 1944 John von Neumann a Oskar Morgenstern vydali knihu Teória hier a ekonomické správanie. V nej riešili prípady spolupráce stratégií správania sa a v jej druhom vydaní položili axiomatické základy, a metódy teórie

hier, ktoré mali uplatnenie v ekonomických oblastiach [3]. Táto publikácia je prvou, ktorá systematicky skúmala Teóriu hier [1]. John von Neumann bol jednou z ústredných postáv v oblasti matematiky a špeciálne v oblasti Teórie hier. Jeho prínos pre problematiku teórie hier je kľúčový a neprehliadnuteľný.

Ďalšou veľkou osobnosťou bol John Forbes Nash. Narodil sa v Bluefielde v Západnej Virgínii v roku 1928 a zomrel v roku 2015. Bol veľmi talentovaný v matematike a jeden z jeho profesorov mu v jeho odporúčacom liste na vysokú školu napísal: „je to matematický génus“. Študoval na Princeton univerzite v New Jersey. Ako PhD študent dokázal existenciu ekvilibria, ktoré doteraz nosí jeho meno [6]. Medzi jeho ďalšie práce patrí vyriešenie Hilbertovho devätnásteho problému týkajúceho sa parciálnych diferenciálnych rovníc. V roku 1994 mu bola udelená Nobelova pamätná cena za ekonomické vedy, za prácu v oblasti teórie hier [6].

Teória hier je relatívne mladá vedná disciplína ale jej korene siahajú do dávnej minulosti a svojou podstatou úzko súvisia s filozofiou [3]. Teóriu hier možno nájsť už v staroveku a súvisí s bojmi a strategickým správaním. Každá vojna, ale aj jej imitovanie v hre či športe predpokladajú porozumenie teoretickému rámcu teórie hier. Veľmi zrozumiteľnou ukážkou je šach, ktorý je starší viac ako 2600 rokov. Podstatou úspechu v tejto hre je schopnosť anticipovať správanie súpera vopred a prispôbiť tomu vlastné správanie. Obidvom hráčom je jasné, že ich cieľ je vyhrať ale postupnosť krokov, ktoré ich k tomu privedú sú odlišné [3]. Prvé filozoficko-teoretické analyzovanie teórie hier však nachádzame podľa Dona Rossa v dialógu Láches a Symposium u Platóna. Ross a iní identifikujú v Sokratových úvahách o bitke z Delia analýzu možností správania vojakov v nasledujúcej bitke a potrebu obmedzenia ich možností voľby, ktorá ochromuje bojaschopnosť vojska [3].

Iným príkladom, už z novoveku, je formalizovanie základného problému teórie hier Pascalom. Pascalova stávka preukazuje obmedzené kompetencie rozumu v možnosti preukázať existenciu Boha, ale zároveň výborné kompetencie v možnosti korigovať dôsledky viery, resp. neviery v Boha. Ak totiž verím v Boha, a on nie je, moja viera obmedzí niektoré malé pôžitky, ktoré by som si inak mohol dožičiť. Ak v neho neverím a nie je, tieto pôžitky si dožičím. Ak však Boh skutočne je, a ja v neho verím, získam celú večnosť vo forme spásy. Ak však je, a ja neverím, prídem o všetko. Pascal vo svojom aforizme ukázal, že keďže nevieme o Božej existencii rozhodnúť, je pre nás výhodnejšie, ak v neho veríme,

pretože možné konzekvencie takéhoto postoja sa kryjú s našimi intenciami viac ako konzekvencie opačného postoja [3].

Míľnikov, ktoré súvisia s teóriou hier je veľké množstvo. Mnohé pomenovávajú teóriu hier priamo a niektoré nie úplne priamo. Napriek tomu môžeme povedať, že Teória hier je tu už od dávnych čias a stáva sa stále viac dôležitou.

1.3 Koncept a základné pojmy v Teórii hier

V tejto časti sa pozrieme na základné pojmy v oblasti Teórie hier. Pre jednoduchšie pochopenie Teórie hier je potrebné definovať si niektoré z nich:

Hra – Pod týmto pojmom myslíme akúkoľvek cielenú a vedomú interakciu ľubovoľného jednotlivca so svojím prostredím, a jej cieľom by malo byť uspokojenie zámerov daného jednotlivca [3]. Dôležité podmienky hry podľa Jespers Julla sú: 1. formálny systém je vystavaný na pravidlách. 2.má kvantifikovateľné a variabilné výstupy. 3. rôzne výstupy majú iné hodnoty. 4. hráči pomocou správania ovplyvňujú výstupy. 5. na výstupoch sú emocionálne zainteresovaní. 6. dôsledok je voliteľný a prenosný [7]. Z uvedených podmienok je zjavné, že hra prebieha medzi racionálnymi hráčmi, ale nie je viazaná len na správanie jedného hráča k inému hráčovi. Za hru možno považovať aj správanie jednotlivca voči systému entít, vzťahov, ktoré správne reagujú a vstupujú do uskutočnenia alebo neuskutočnenia jeho zámeru. Takýmto hrám hovoríme hry s jedným hráčom (hry s prírodou). Hráčom sa stane príroda [3]. Keď to rozoberieme do detailov tak príroda je vždy účastníkom hry. Ako príklad spomenieme už len samotnú existenciu hráčov, zákonitosti prírody zasahujúce do situácie a rôzne ďalšie. Ale pri množstve hrách, tieto pôsobenia eliminujeme a sústredíme sa hlavne na správanie ostatných aktérov [3].

Voľba a stratégia – V hre je dôležité správanie hráča. Cielené voľby hráča smerujúce k dosiahnutiu nejakého zámeru by sme mohli pomenovať ako stratégia. Hry majú rôzne povahy a podľa toho možno rozlišovať voľby spočívajúce v jednotlivom rozhodnutí v danej neopakovateľnej situácii a výbere voľby s opakovanými ťahmi. Dôležitým pojmom je stratégia. Pri stratégií ide o dlhodobejšiu činnosť, ktorá môže byť aj vopred premyslená a zohľadňuje viacero premenných v súvislosti s odlišnými dôsledkami vlastných

predchádzajúcich alebo súperových ťahov. Preto môžeme v prípade opakovaných ťahov hovoriť o stratégií [3].

Pravdepodobnosť a teória možných svetov – Možnosť naplnenia alebo nenaplnenia cieľa a existencia slobody, alebo aspoň náhody, v rozhodovaní, sú nevyhnutnými podmienkami Teórie hier. V prípade, že by to tak nebolo, tak by sme nemohli hovoriť o hre. Hra predpokladá nejednoznačnú kauzálnu späťosť medzi rozhodnutím a výsledkom, a je ju možné vyjadriť nenulovou pravdepodobnosťou dosiahnutia požadovaného stavu v závislosti od nášho rozhodnutia. A aj od rozhodnutia ostatných aktérov. Možnosť rôznych výsledkov a neistota, poukazujú na filozofický problém možných svetov. O hre môžeme teda hovoriť vtedy, keď existujú možnosti výskytu rôznych výsledných stavov [3].

Výstupy – Dôležitým pojmom pri výstupoch je úžitok. Ním máme na mysli mieru dosahovaných výstupov, pri použití určitej voľby alebo stratégie. S hrou daný pojem súvisí tak, že výsledkom hry má byť získanie čo najväčšieho úžitku. Výstup je stav, ktorý nastane ako dôsledok použitia jednotlivých stratégií. V prípade aktéra, ak sú stavy v zhode s jeho plánom, môžeme hovoriť o zisku. Ak nie sú, tak nezávisle od toho, ako to hodnotí iný hráč, ide o stratu. Z tohto vyplýva, že výstupy, ciele a úžitky sú hlavným motorom hier. Ale ak sa pozrieme na ich hodnotenie, tak zistíme, že môže byť veľmi relatívne. Preto sa na ich meranie používajú rôzne kritériá [3]: Ordinálna škála – porovnávajúcu kvalitatívne stupne stavov, napríklad známky v škole [3]. Kardinálna intervalová škála – kvantitatívne porovnávajúcu objekty navzájom, napríklad teplotu telesa [3]. Kardinálna pomerná škála – porovnávajúcu pomer kvantity posudzovaného kritéria, napríklad pomer výnosov [3]. Výsledky racionálneho rozhodovania veľmi ovplyvňuje správne kvantifikovanie ziskov a strát. Medzi jednotlivých aktérov sa rozloží možný zisk. A toto rozdelenie delíme na: hry s nulovým úhrnom a hry s nenulovým úhrnom [3]. Hra s nulovým úhrnom má vo výsledku nulový súčet ziskov a strát u všetkých účastníkov. To v jednoduchosti znamená, že ak niekto stratil, tak niekto ďalší musel získať. Čiže výška výhier a strát je rovná nule [3]. Hra s nenulovým úhrnom má vo výsledku asymetrické rozloženie výhier a strát. Sú závislé od zvolených stratégií jednotlivých aktérov. Pri hrách s opakovaním je celkový úhrn ziskov a strát závislý od súčtu jednotlivých vstupov. Aktér pri nastavení stratégie berie do úvahy celkový zisk pred čiastkovým [3].

Východiskové stavy – Základným a rozhodujúcim momentom Teórie hier je správne posúdenie tohto stavu. Stavom myslíme zadanie riešenej situácie, explicitne ale i implicitne vyjadrené vzťahy medzi prvkami rozhodovacieho procesu. Úlohou Teórie hier je opísať vzťahy medzi východiskovými stavmi, stratégiami a výstupmi, a zároveň hľadať optimálne stratégie správania sa, vedúce k čo najväčšiemu zisku v situácii, keď rozhodovanie ostatných účastníkov nemožno považovať ako známu premennú [3].

Rozhodovanie s istotou, neúplnými informáciami a s rizikom – Rozhodovanie s istotou je vtedy, ak každá z alternatív možného konania vedie k známemu jednoznačnému dôsledku. O rozhodovaní s rizikom hovoríme vtedy, ak jednotlivé alternatívy možného konania vedú k dôsledkom, ktorých výskyt možno vyjadriť prostredníctvom matematickej pravdepodobnosti. Príkladom môže byť napríklad hra Black Jack. Ak ale rozloženie pravdepodobnosti výskytu nejakého dôsledku nie je známe, ale je neurčitá jeho hodnota, hovoríme o rozhodovaní s neurčitosťou. Príkladom je napríklad výber neznámeho jedla v reštaurácii. Pri rozhodovaní s rizikom a aj s neurčitosťou si vyžadujú od aktéra použitie rôznych stratégií, ktoré by mali minimalizovať nežiadúce výsledky [3].

Racionálne rozhodovanie - Pod týmto pojmom máme na mysli taký rozhodovací proces, ktorého všetky dôvody spočívajú výlučne v rozumových operáciách. Samotné racionálne rozhodovanie je základnou podmienkou úspešnosti Teórie hier. Racionálne rozhodnutie je teda také rozhodovanie, ktorého všetky dôvody a dôsledky možno logicky a konzistentne vysvetliť. V prípade neracionálneho rozhodovania sa aktér neriadi výlučne racionálnymi dôvodmi. V Teórii hier je akékoľvek rozhodovanie založené na iných ako výlučne racionálnych dôvodoch, považované za neracionálne rozhodovanie. V bežnom živote človeka sa racionálne a neracionálne rozhodovanie prekrýva. Neracionálny hráč je, ak je ľahostajný k výsledku, alebo nesleduje jeho naplnenie [3].

1.4 Nashovo ekvilibrium

Optimálnu stratégiu hráčov v konfliktnej situácii môžeme nájsť pomocou Nashového ekvilibria. Nashovo ekvilibrium alebo inak Nashova rovnováha je riešenie, v ktorom platí, že pokiaľ sa niektorý z hráčov nebude držať optimálnej stratégie a jeho súper áno, jeho výplata sa zníži. Môžeme teda jednoducho povedať, že kto sa odchýli od optimálnej stratégie, nemôže si polepšiť [14].

Definície:

Profil stratégií s^* je Nashovým ekvilibriom hry G v čistých stratégiách, ak pre každého hráča i a pre všetky $s_i \in S$, kde S je množina strategických profilov a s_i sú najlepšie odpovede, platí:

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*)$$

Nashovo ekvilibrium s^* v čistých stratégiách je ostré, ak pre každého hráča i a pre všetky $s_i \in S$ také, že $s_i \neq s_i^*$ platí:

$$u_i(s_i^*, s_{-i}^*) > u_i(s_i, s_{-i}^*)$$

John Forbes Nash dokázal, že každá konečná hra má aspoň jedno takéto riešenie a takýmto bodom hovoríme rovnovážne stavy [3]. Nashova rovnováha nemusí byť nutne Pareto-optimálna. Riešenie môže priviesť k výsledku, ktorý aspoň pre jedného z hráčov priniesie nižší zisk ako niektorý iný výsledok hry. Zároveň, Nashova rovnováha nemusí byť v hre jedinečná. Môže ich byť aj viac [14].

1.5 Konfliktné a nekonfliktné situácie

V tejto časti sa pozrieme na konfliktné a nekonfliktné situácie v Teórii hier. Potom ich detailne zanalyzujeme v nasledujúcej časti. Teória hier sa zaoberá riešením nekonfliktných a konfliktných situácií. Nekonfliktnou situáciou nazývame takú, ktorej sa zúčastňuje len jeden účastník. Zároveň pozná množinu všetkých svojich možných rozhodnutí a vie ich oceniť. Konfliktnou situáciou sa nazýva taká, kde je počet účastníkov rozhodovacej situácie konečný a všetci poznajú množinu všetkých svojich rozhodnutí, respektíve stratégií, a množiny rozhodnutí svojich súperov. Zároveň vedia oceniť svoje rozhodnutie vzhľadom na rozhodnutie svojich súperov [10].

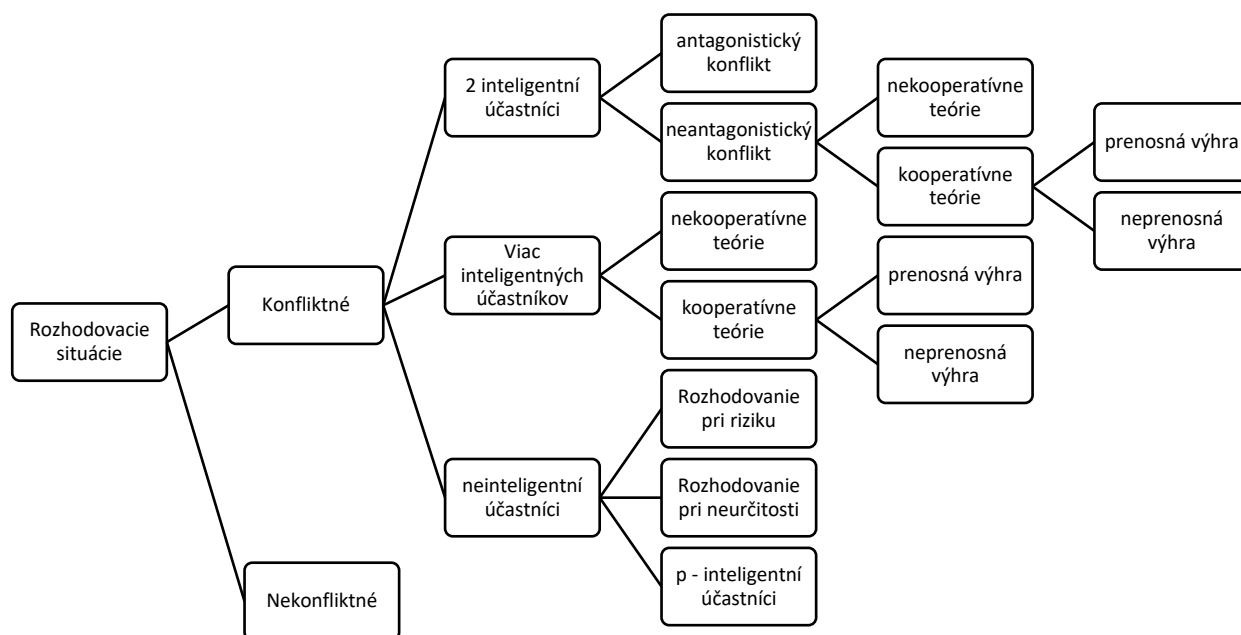
1.5.1 Hlavné typy konfliktných situácií

Aby sme vedeli podať uspokojivú definíciu optimálneho rozhodnutia, budeme musieť konfliktné situácie ešte ďalej rozdeľovať a to podľa rôznych hľadísk.

Najskôr si zoberme konfliktné situácie s dvoma účastníkmi a uvažujme nad nimi. Ak sú obaja inteligentní a ak jeden účastník stráca práve to, čo druhý získava, hovoríme o tzv.

antagonistickom konflikte. V antagonickom konflikte nemá dohadovanie a spolupráca medzi účastníkmi význam, pretože sú ich záujmy úplne protikladné. Príkladom môže byť vojenský konflikt, kde je úplne bežné, že nastane tento typ konfliktu. V istom zmysle je aj najjednoduchším typom konfliktu. Definícia optimálneho rozhodnutia v tomto type nerobí problémy [9]. Často sa môžeme stretnúť s prípadmi, keď si síce každý účastník sleduje vlastné záujmy, ale tie nemusia byť v priamom protiklade so záujmami druhého účastníka. Vtedy hovoríme o neantagonistických konfliktoch. Tie môžeme ďalej rozdeliť na kooperatívne a nekooperatívne hry. Kooperatívna hra je situácia, keď sa uzavrie dohoda s druhým účastníkom konfliktu pred rozhodnutím, akú voľbu spraví. Ak to možné nie je, hovoríme o nekooperatívnej hre. V kooperatívnej teórii hier je dôležitý prípad, keď sa hráči môžu, napríklad zmluvne dohodnúť, že účastník, ktorému výsledok prináša podstatne lepši výsledok, dá druhému účastníkovi nejakú náhradu respektíve kompenzáciu za to, že mu k lepšiemu výsledku dopomohol vhodnou voľbou rozhodnutí. Vtedy hovoríme o prípade prenosných výhod. Ak to nie je možné, tak hovoríme o neprenosnej výhre. Rozdelenie hier na kooperatívne a nekooperatívne platí aj v prípade, že sú v konflikte viac ako dvaja účastníci.

V prípade kooperatívnej teórie treba uvažovať nad tým, že niektorí účastníci budú medzi sebou spolupracovať, čiže vytvoria spojenectvo, aby získali výhody, ktoré by bez neho nemali. V tomto type konfliktu je teda podstatná otázka, s kým vytvoriť spojenectvo a v závislosti od toho dáva zmysel uvažovať o výhodnosti jednotlivých rozhodnutí. Rovnako môžeme uvažovať aj pre prípady kde sú prenosné a neprenosné výhry. Ak nie je prístupné tvorenie koalícií, tak je teória konfliktu viac než dvoch účastníkov len priamym zovšeobecnením prípadu dvoch účastníkov. V prípade, že ide o konfliktné situácie, kde vplyvy nedovoľujúce určiť jednoznačný dôsledok rozhodnutí majú náhodnú povahu. Druhého účastníka konfliktu považujeme potom za istý druh náhodného mechanizmu, ktorý volí svoje rozhodnutia podľa nejakého pravdepodobnostného rozloženia. Toto rozloženie prvý účastník buď pozná a vtedy hovoríme o rozhodovaní pri riziku alebo nepozná. Vtedy potom hovoríme o rozhodovaní pri neurčitosti. Niekedy sa druhý účastník môže javiť prvému ako spojenie inteligentného účastníka a náhodného mechanizmu. Vtedy hovoríme o p-inteligentných účastníkoch [9].



Obr. 1.1: Klasifikácia hier [9]

1.6 Typy hier

Ďalej sa pozrieme na rôzne typy hier. Špeciálne sa pozrieme na hry v normálnom tvare a rozvinutom tvare. V tejto časti budeme vychádzať zo zdrojov [3] a [9].

1.6.1 Hry v normálnom tvare

Ak si prvý hráč volí svoju voľbu nezávisle od voľby protihráča alebo aj protihráčov a ostatní hráči volia svoje rozhodnutia nezávisle od voľby prvého hráča, hovoríme o hre v normálnom tvare. Dôležitou časťou uskutočnenia voľby je to, že ju hráči vykonávajú súčasne a o svojich voľbách nie sú navzájom informovaní [3]. Konfliktné rozhodovacie situácie vznikajú napríklad pri hraní hier ako karty, šach a podobne. Sú to hry z kategórie salónnych. V začiatkoch matematického rozboru v tejto oblasti sa tento rozbor týkal práve týchto hier. Až v roku 1944 si J. von Neumann a O. Morgenstern všimli zhodnej štruktúry konfliktných situácií vznikajúcich pri salónnych hrách a pri napríklad vojenskom alebo ekonomickom rozhodovaní [9].

1.6.2 Maticové hry

Jedná sa o konečnú hru dvoch inteligentných hráčov v normálnom tvare. Každý z hráčov volí svoje stratégie nezávisle od druhého. Hra má antagonistický konflikt, čo znamená, že zisk jedného z hráčov predstavuje stratu u druhého. Hra je reprezentovaná maticou. Matica obsahuje platby hráčov, pri použití jednotlivých stratégií [23]. Maticovým hrám sa budeme venovať detailnejšie v časti analýz softvérových možností riešenia hier.

1.6.3 Väzňova dilema

Väzňova dilema označuje vzorový príklad hry s neurčitou a s nenulovým súčtom výplat. Jej podstatou je objasnenie optimálnej stratégie eliminujúcej riziká, vyplývajúce z nemožnosti vopred určiť rozhodnutia protihráča, pričom rozhodnutie protihráča bezprostredne ovplyvní výšku výplaty prvého hráča. Počiatky väzňovej dilemy siahajú do roku 1950. V tom roku Merrill Flood a Melvin Dresher vytvorili myšlienkový experiment opisujúci racionálne správanie hráčov, ktorí nezávisle od seba sledujú svoje vlastné záujmy. Albert W. Tucker bol matematik na Princetonskej univerzite a zároveň učil Johna Nasha. W. Tucker formalizoval Dresherov experiment a vymyslel mu príhodu, podľa ktorej má dilema svoj názov. V pôvodnom zadaní problému ide o nasledujúcu situáciu [3]:

„Dvaja členovia zločineckej skupiny sú zadržaní a uväznení. Každý väzeň je osamote bez možnosti komunikovania alebo výmeny správ s druhým. Polícia priznáva, že nemá dostatok dôkazov na ich odsúdenie za hlavný trestný čin, ale má dostatok dôkazov na odsúdenie na rok väzenia za spáchanie menších skutkov. Simultánne ponúkne každému väzňovi faustovský obchod. Ak bude jeden z nich svedčiť proti druhému, svedok bude prepustený, kým spoluväzeň dostane tri roky za hlavný zločin. Obidvaja sú informovaní o tom, že druhému väzňovi bola ponúknutá tá istá dohoda. V prípade, že obidvaja väzni budú svedčiť proti sebe navzájom, budú odsúdení na dva roky vo väzení. Väzňom je ponechaný nejaký čas na rozmyslenie, ale v žiadnom prípade sa nemôžu dozvedieť, ako sa rozhodol ten druhý.“ [13]

Tu nastáva otázka, že ako sa má každý z väzňov rozhodnúť? Keďže sa navzájom nemôžu ovplyvňovať, každý by sa mal starať o svoje vlastné blaho [3]. Ak si odmyslíme

všetky sociálne, psychologické a morálne aspekty problému, a obmedzíme sa len na čisto racionálne rozhodnutia aktérov, dospejeme k tomu, čo nazývame tabuľka možných riešení. Sociálnymi, morálnymi a psychologickými aspektami myslíme napríklad možnú pomstu medzi aktérmi, reakcie okolia a podobne. Ak sa prvý hráč rozhodne zostať lojálny k spolupáchateľovi a nezradí ho, výsledok možného trestu úplne závisí len od rozhodnutia druhého hráča. Ak sa druhý hráč zachová rovnako, dochádza k ich vzájomnej spolupráci, ktorej výsledkom je rok väzenia pre každého z uväznených. V prípade, že protihráč zradí svojho spolupáchateľa, výsledkom bude, že prvého hráča uväznia na tri roky. Spolupracovať teda znamená istotu väzenia a to buď na rok alebo tri. V prípade, že prvý hráč sa rozhodne spolupracovať s vyšetrovateľom a zradí spolupáchateľa, tak je opäť výsledok závislý od správania druhého hráča. Ak aj on zradí a usvedčí prvého hráča, tak obaja dostanú dva roky väzenia. Ak však druhý hráč zvolí stratégiu, že spolupáchateľa neudá a odmietne spolupracovať, výsledkom bude beztrestnosť prvého hráča a jeho sloboda. Prvý hráč teda pri nespolupráci so spolupáchateľom môže získať 2 roky väzenia alebo slobodu [3].

1.6.4 Hry v rozvinutom tvare

V tejto časti sa pozrieme na hry v rozvinutom tvare v Teórii hier. Okrem samotného názvu, ako hry v rozvinutom tvare, ich tiež poznáme ako hry v explicitnom tvare alebo iterované hry. Jednoducho povedané sú tieto typy hier o postupnosti strategických ťahov, kde hráči reagujú na predchádzajúce ťahy protihráča. Čo sa týka ukážok hier, ktoré by sme sem mohli zaradiť, tak sú to mnohé salónové hry. Napríklad šach, dáma a mnohé ďalšie. V teórii hier ich najčastejšie znázorňujeme špeciálnym grafom, ktorý má tvar stromu. Môžeme ho pomenovať ako strom hry [16].

2 Cieľ práce

Hlavným cieľom tejto inžinierskej práce sú softvérové riešenia nami zvolených hier. Na to aby sme dosiahli tento cieľ je najskôr potrebné spracovať teóriu v tejto oblasti a charakterizovať teóriu hier. Vďaka tomu dostaneme všeobecný prehľad v tejto oblasti, ktorý nám pri ďalšom postupe môže pomôcť. Ďalším dôležitým, avšak menším cieľom je pozrieť sa na dva typy hier, ktoré sa nachádzajú v zadaní práce. Jedná sa o hry v normálnom a hry v rozvinutom tvare. Pri nich si rozoberieme špecifiká jednotlivých hier ako aj existujúce hry pre oba tvary. V tejto práci máme za cieľ vypracovať jednu hru z hier v normálnom tvare a aj jednu hru v rozvinutom tvare. Avšak k tomu, aby sme dosiahli cieľ, ktorým je hotový program, tiež potrebujeme kvalitnú analýzu možností riešenia týchto hier. Preto sme si charakterizovali tieto hry. Tento pohľad nám posluží na to aby sme si jednotlivé hry rozeanalyzovali a pozreli sa bližšie na možné komplikácie pri vývoji aplikácií. Zároveň nám to poskytne víziu, akým spôsobom vyvíjať jednotlivé algoritmy a ktoré funkcie využiť.

Samotný vývoj aplikácie prinesie konkrétne reálne výsledky. Tie budú vychádzať zo spomínaného rozboru jednotlivých hier. Prvotným cieľom je pozrieť sa na podčasti hier. V nich zistiť potencionálne problémy, ktorým sa môžeme neskôr vyhnúť a zároveň nám môžu pomôcť pri pochopení rôznych detailov hry. Okrem problémov napríklad so spôsobom realizácie hier je to aj výber správnych algoritmov a postupov pre efektívny beh programu. Okrem efektívnosti je potrebné sa pozrieť na samotnú funkčnosť kódu a aj výhody použitia jednotlivých modulov alebo funkcií. Po odpovedaní si na tieto jednotlivé otázky dosiahneme stav, v ktorom sme pripravený na samotnú realizáciu hier. Keďže sa jedná o rozličné hry tak aj samotná realizácia sa bude líšiť. Avšak aj napriek rozdielom budú v sebe obsahovať prvky, ktoré budú rovnaké alebo s malými úpravami podobné. Úlohou ich bude identifikovať a neskôr aplikovať pri oboch programoch. Úlohou pre nás bude teda aj využitie už použitých prvkov. Tým môžeme dosiahnuť zrýchlenie procesu vývoja aplikácií ako aj ich jednotnosť, čo sa týka dizajnu, použitia a tak ďalej. Toto opätovné použitie bude možné hlavne pri vývoji druhej zo zvolených hier bez ohľadu na to, ktorú si zvolíme ako prvú. Po vývoji prvej aplikácie dosiahneme koncept hry, z ktorého môžete čerpať poznatky pre našu druhú aplikáciu.

V prípade hier v normálnom tvare sme si dali za cieľ spraviť hru s názvom pirát a Sicíľčan. Z hier v rozvinutom tvare sme vybrali hru s názvom Bulo. Pre tieto hry chceme rovnako pripraviť aj rozbor odohratej hry pre hráčov, ktorý sa bude dať zobrazit' na konci hry. Analýza pre hru pirát a Sicíľčan bude obsahovať maticu s popisom a druhá hra bude obsahovať strom hry.

Po splnení týchto cieľov dosiahneme použiteľné programy. Tie si môžeme zahrať a zároveň dávame každému možnosť si vyskúšať úlohy v teórii hier. Zároveň otvárame možnosť ďalšiemu vývoju týchto hier.

3 Metodika práce a metody skúmania

3.1 Technológie

Keďže sa v tejto práci venujeme aj softvérovému riešeniu hier, tak je dôležité definovať technológie použité pri samotnom programovaní a príprave riešenia. Pôjde o opis počítača, jazyka a vývojového prostredia.

3.1.1 *Hp pavilion 14-dv0003nc*

Keďže ide o riešenie, ktoré bude bežať na počítači, treba predstaviť aj prístroj, na ktorom je dané riešenie vytvárané a testované. Ide o notebook značky Hp. K jeho vybaveniu patrí procesor od spoločnosti Intel, konkrétne Intel Core i5 jedenástej generácie. Jeho rýchlosť frekvencie je 2.4 GHz. Ďalším špecifikom je pamäť RAM. Toto zariadenie má 16 GB operačnej pamäte. Čo sa týka grafiky, notebook obsahuje grafiku Intel iris xe graphics, pričom ide o vstavaný typ. Posledným dôležitým parametrom, ktorý je vhodné spomenúť, je operačný systém, na ktorom sú dané riešenia vytvárané. Windows 11 od spoločnosti Microsoft predstavuje najnovšiu verziu operačného systému na trhu.

3.1.2 *Python*

Python je moderný a veľmi populárny programovací jazyk. Jeho tvorcom je Guido van Rossum. Používajú ho firmy ako napríklad Uber, PayPal a Google. Ide o jazyk, ktorý je interpreter, a odlišuje sa teda od kompilačných jazykov, akými sú napríklad Pascal, C a C++ [11]. Zaujímavé je porovnanie populárnych jazykov Python a Java. Programy naprogramované v Pythone bežia priemerne 3- až 5-krát pomalšie ako rovnocenné programy v Jave. Tento pomalší beh má viacero dôvodov. Jedným z nich je, že v Pythone nie je potrebné definovať dátové typy. Tým sa však predlžuje čas spustenia. Vezmime si príklad sčítania dvoch premenných. Ak by sme ich sčítavali v Pythone, tak by sa najskôr musel zistiť ich dátový typ a až následne sa vykoná operácia, pričom Java dané typy pozná a môže danú operáciu vykonať hneď. Avšak výhodou Pythonu je rýchlosť vývoja aplikácií. Už len fakt, že nie je nutné definovať dátové typy, uľahčuje a zrýchľuje proces vývoja [12].

Hlavné vlastnosti jazyka Python sú [11]:

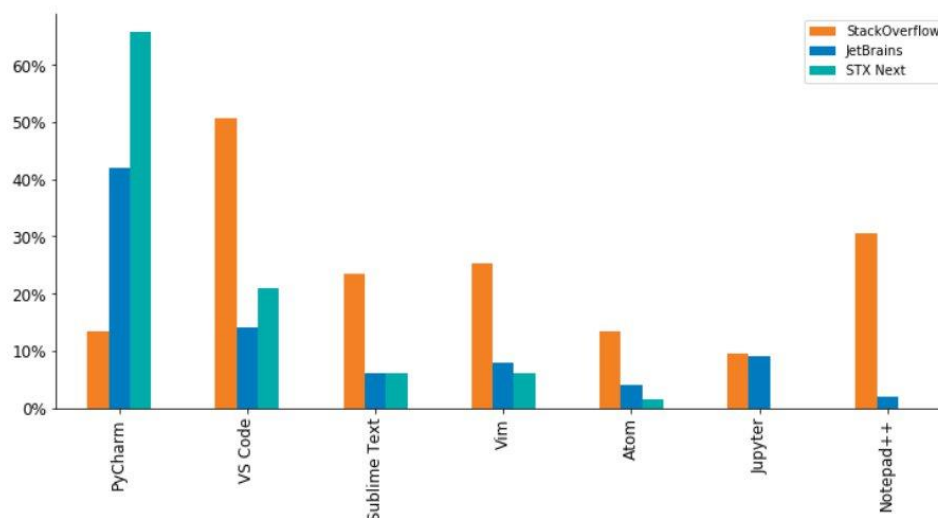
- „veľmi jednoduchá a dobre čitateľná syntax, a keďže Python je aj vysoko interaktívny, je vhodný aj na vyučovanie programovania“
- „na rozdiel od staticky typovaných jazykov, pri ktorých je treba dopredu deklarovať typy všetkých dát, je Python dynamicky typovaný, čo znamená, že žiadne deklarácie netreba.“
- „Python obsahuje pokročilé črty moderných programovacích jazykov, napr. podpora práce s dátovými štruktúrami, objektovo orientovaná tvorba softvéru...“
- „Je to univerzálny programovací jazyk, ktorý poskytuje prostriedky na tvorbu moderných aplikácií ako analýza dát, spracovanie médií, sieťové aplikácie a pod.“
- Python združuje obrovskú komunitu programátorov a expertov.

V tejto práci budeme používať verziu Pythonu 3.9.

3.1.3 Popis prostredia

V tejto kapitole sa budeme venovať popisu prostredia, v ktorom boli vyvíjané riešenia úloh teórie hier. Čo sa týka vývojových prostredí, pythonovské programy môžeme vyvíjať v rôznych prostrediach. Samotný Python poskytuje vlastné vývojové prostredie, avšak je menej intuitívne ako rôzne iné. Ide o vývojové prostredie s názvom IDLE. Existujú aj mnohé ďalšie prostredia, ktoré sú veľmi populárne a obsahujú rôzne automaticky nainštalované funkcionality. Patrí k nim napríklad PyCharm, Visual studio Code, Sublime Text a rôzne ďalšie. Existujú aj prieskumy obľúbenosti jednotlivých vývojových prostredí a na jeden z nich sa pozrieme. Výsledok prieskumu nemusí byť úplne zhodný s realitou, keďže vychádza z obmedzeného množstva hlasujúcich. Avšak aj tento výsledok je zaujímavý a môže poskytnúť ďalší dôležitý pohľad na túto oblasť. Prieskum robila spoločnosť STX Next a vychádza z názorov vývojárov Pythonu z tejto spoločnosti. Ich počet presiahol 200. Súčasťou prieskumu je aj index popularity vývojového prostredia podľa stránky StackOverflow. Dáta z tohto prieskumu sú všeobecné a obsahujú názory na všetky vývojové prostredia, pre akýkoľvek programovací jazyk. Posledným zdrojom údajov je spoločnosť JetBrains. Pri dátach od JetBrains sa zohľadňujú iba programátori pracujúci v Pythone [20].

Na grafe z obrázka 4.1 môžeme vidieť tri stĺpce. Oranžový zahŕňa dáta zo stránky StackOverflow, modrý dáta z JetBrains a posledný zelený stĺpec vyobrazuje dáta spoločnosti STX Next. Na základe týchto dát môžeme zhodnotiť, že prostredia PyCharm a Visual Studio Code sú pomerne používané všetkými tromi typmi ľudí z prieskumu. Rozdiel medzi Visual Studio Code a PyCharmom u užívateľov zo StackOverflow je spôsobený faktom, že Visual Studio Code je univerzálny aj pre iné jazyky, a tým pádom ho využíva väčšie spektrum užívateľov ako niektoré iné vývojové prostredia. Avšak u dvoch skupín ľudí, ktoré sa zameriavajú len na programovací jazyk Python, vyhráva v obľúbenosti PyCharm. Tento prieskum dokazuje, že PyCharm je pomerne obľúbený a využívaný časťou programátorov, ktorí sa venujú Pythonu.



Obr. 3.1 Graf obľúbenosti jednotlivých vývojových prostredí [20]

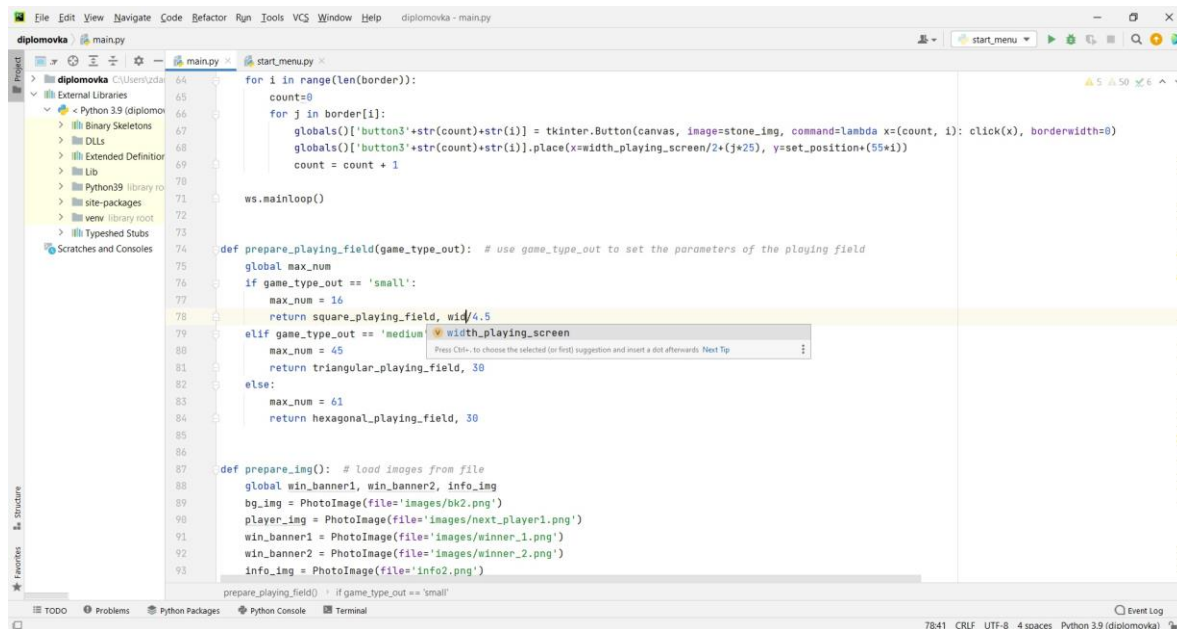
3.1.4 PyCharm

PyCharm je vývojové prostredie vyvinuté pre programovací jazyk Python. Je multiplatformové, čiže je dostupné pre Linux, Windows a aj macOS. Vytvorila ho firma JetBrains, ktorá má hlavné sídlo v Českej republike [21]. PyCharm kombinuje všetky nástroje Pythonu na jednom mieste. Ponúka:

- inteligentný editor kódu
- navigačné a refaktorové nástroje
- debugger
- testovací beh programu

- inšpekciu kódu
- riadenie verzií (VCS)

Okrem týchto funkcionalít obsahuje aj ďalšie, ktoré sú poskytované len v platenej verzii. Z ďalších funkcionalít sú to napríklad databázové nástroje, pomoc pri vývoji webovej stránky, vedecké nástroje a rôzne ďalšie [20].



Obr. 3.2 PyCharm prostredie

Zo spomenutých funkcií je veľkým prínosom inšpekcia kódu. Tá rieši kvalitu kódu a pomocou varovaní a chybových hlásení upozorňuje programátora na nezrovnalosti v kóde, ako aj na to, že jeho kód nespĺňa bežne zabehnuté štandardy písania kódu. Ďalšou prednosťou je dopĺňanie kódu. Táto funkcionalita zrýchľuje písanie kódu a zároveň poskytuje pomoc pri hľadaní jednotlivých možností v kóde.

3.2 Definícia a príklady hier v rozvinutom a normálnom tvare

Definícia hier v rozvinutom tvare:

Hru v rozvinutom tvare pre n hráčov môžeme definovať aj n -ticou so siedmimi prvkami.

Táto časť vychádza zo zdroja [23]:

$$\Gamma := (X, E, P, W, C, p, U)$$

(X, E) - definuje strom hry.

P - Hráčovú časť

W - Informačná časť

C - Časť voľby

p - Priradenie pravdepodobnosti

U - Úžitok, zisk

Definícia hier v normálnom tvare:

Q je konečná neprázdna množina o N prvkoch. Jej prvky očísľujeme od 1 po N a nazveme ich hráči. Ďalej je daných N množín X_1, \dots, X_N a N funkcií $M_1(x_1, \dots, x_N), \dots,$

$M_N(x_1, \dots, x_N)$ definovaných na kartézskom súčine $X_1 \times \dots \times X_N$. Hrou N hráčov v normálnom tvare nazveme množinu $\{Q; X_1, \dots, X_N; M_1(x_1, \dots, x_N), \dots, M_N(x_1, \dots, x_N)\}$. Množinu Q nazveme množinou hráčov. Hráčov budeme často označovať iba ich číslami a písať

$Q = \{1, 2, \dots, N\}$. Množinu X_i nazveme priestorom stratégie i -teho hráča a funkciu

$M_i(x_1, \dots, x_N)$ nazveme výplatnou funkciou i -teho hráča. Prvky $x_i \in X_i$ sa potom volajú stratégie i -teho hráča [9].

Príklad:

Hry v normálnom tvare si vysvetlíme na základe nasledujúceho príkladu. Jedná sa o hru dvoch hráčov.

$$\text{Nech } Q = \{1, 2\}, X_1 = \langle 0, 1 \rangle, X_2 = \langle 0, 1 \rangle \text{ a } M_1(x_1, x_2) = x_1 + x_2, M_2(x_1, x_2) = -(x_1 + x_2)$$

Táto hra sa hrá tak, že hráč 1 zvolí číslo z intervalu $\langle 0, 1 \rangle$. Nezávisle od neho hráč 2 zvolí číslo z rovnakého intervalu a po zverejnení svojich výberov zaplatí hráč 2 hráčovi 1 čiastku rovnú súčtu nimi zvolených čísel. Ak sú obaja hráči inteligentní, najlepšou voľbou pre hráča 2 bude číslo 0 a pre hráča 1 číslo 1 [9].

3.3 Analýza softvérových možností riešenia hier

Slovo analýza podľa definície výkladového slovníka znamená rozkladanie, rozčlenenie predmetu alebo javu na jeho jednotlivé časti. Táto definícia dokonale opisuje to, čo budeme vykonávať v našom príklade. Pôjde o rozbor hier, ich možných postupov a stratégií. Obe hry majú obmedzené množstvo možných stratégií a vďaka nim vieme detailne analyzovať rôzne postupy a možnosti hráčov. Hráči aj v prípade striedania ťahov vedia pomocou analýzy zistiť, kde spravili v hre chybu a ako by sa hra vyvíjala v prípade inak zahranej stratégie. Táto analýza, samozrejme, neprináša hráčovi vždy víťaznú stratégiu. Môže sa javiť ako víťazná, ale je veľa krát závislá od zvolených stratégií druhého hráča. Veľmi dobrým príkladom je hra Bulo, kde v prípade, že sa vrátíme na konci hry o tri ťahy späť, tak hráč, ktorý prehral môže zahrať inú stratégiu ale jeho výhra je stále závislá od toho, aký ťah zahrá druhý hráč. Bližšie sa na túto oblasť pozrieme v časti analýza hry Bulo. Nám však pomôže pochopiť detaily hry, jej možné realizácie z pohľadu hráčov a taktiež nám pomôže pri skúmaní možností analýzy aj z pohľadu programátora. Na analýzu sa používajú rôzne postupy a možnosti zápisu. Pri nasledujúcich hrách sa pozrieme na niektoré z nich.

Na tejto hre si ukážeme strom hry a vysvetlíme si jej pravidlá. Táto hra súvisí aj s hrami, ktoré budú v tejto práci následne softvérovo riešené. Je to jednoduchá hra, ktorá podnietila vznik ďalších hier ako *Bulo* alebo *Jeek*. Jej názov je pravdepodobne odvodený z nemeckého slova *nehmen*, čo v preklade znamená brať [17]. Hru *NIM* môže hrať ľubovoľný počet hráčov. V našom príklade budeme predpokladať, že to budú dvaja hráči. Na začiatku predpokladajme, že máme dve kôpky a v každej z nich sú dva kamienky. Tento počet sme zvolili pre jednoduchosť príkladu. V realite máme podmienku, že máme konečný počet kamienkov rozdelených na kôpky. Hráč si môže zobrať ľubovoľný počet kamienkov. Avšak podmienkou je, aby vždy zobral aspoň jeden kamienok a vždy bral iba z jednej kôpky. Hra končí, keď niektorý z hráčov zoberie posledný kamienok. Ten, kto odoberie posledný kamienok, prehral. Teraz si pravidlá zhrnieme a rozdelíme podľa špecifikácie [16].

Cieľ hry: Cieľom je donútiť svojho súpera, aby zobral posledný kamienok z kôpky.

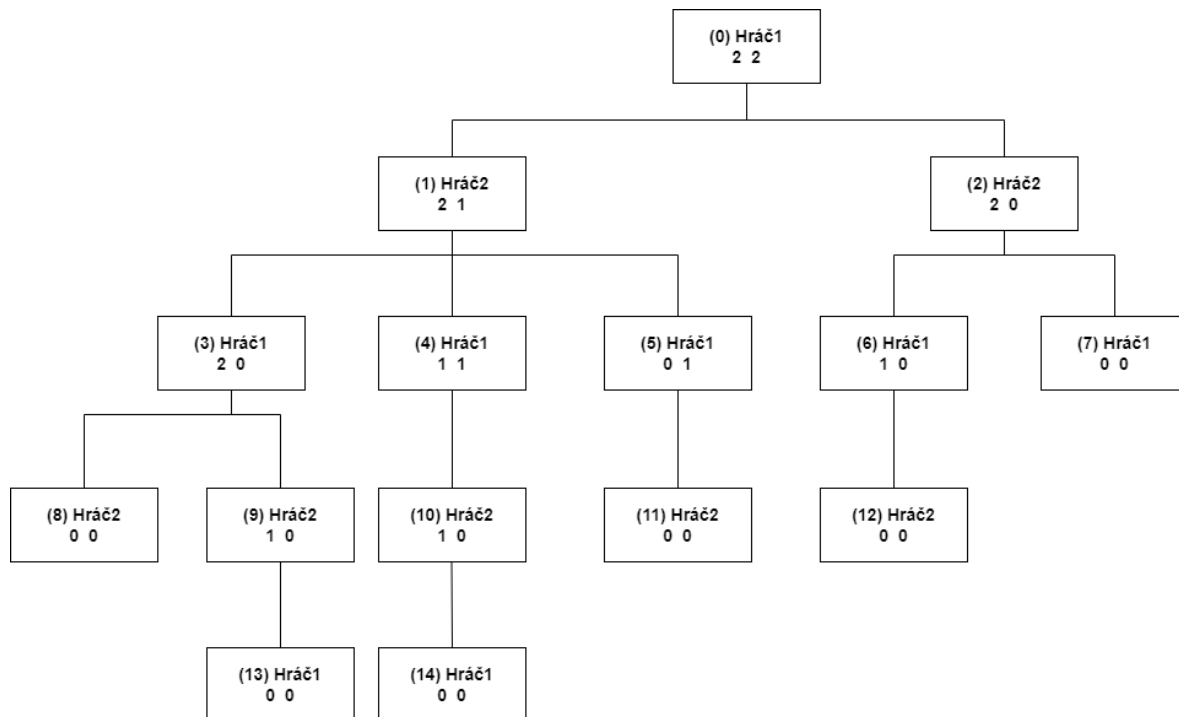
Východisková situácia: Pred hráčmi sú všetky kamene rozdelené do kôpok.

Hra: Hráči sa striedajú v ťahoch pravidelne. Každý hráč má právo zobrať kamienky podľa vyššie stanovených pravidiel.

Koniec hry: Hráč, ktorý zoberal posledný kameň, prehral hru [17].

3.3.1 Strom hry a NIM

Stromy môžu predstavovať rôzne hry. Ak strom predstavuje hru, tak koreň hry zodpovedá počiatkovej pozícii hry. Potomkami každého vrcholu sú zodpovedajúce herné pozície, ku ktorým sa dá dostať jednou z možných akcií [22]. Ako už bolo spomenuté, tak strom hry využívame na znázorňovanie hier v rozvinutom tvare. Ide o súvislý graf. Začiatkový uzol sa zvykne pomenovávať aj koreň. Spôsob, ktorým dostaneme riešenie zo stromu hry, sa volá spätná indukcia [16]. Príklad stromu si ukážeme na hre *NIM*.



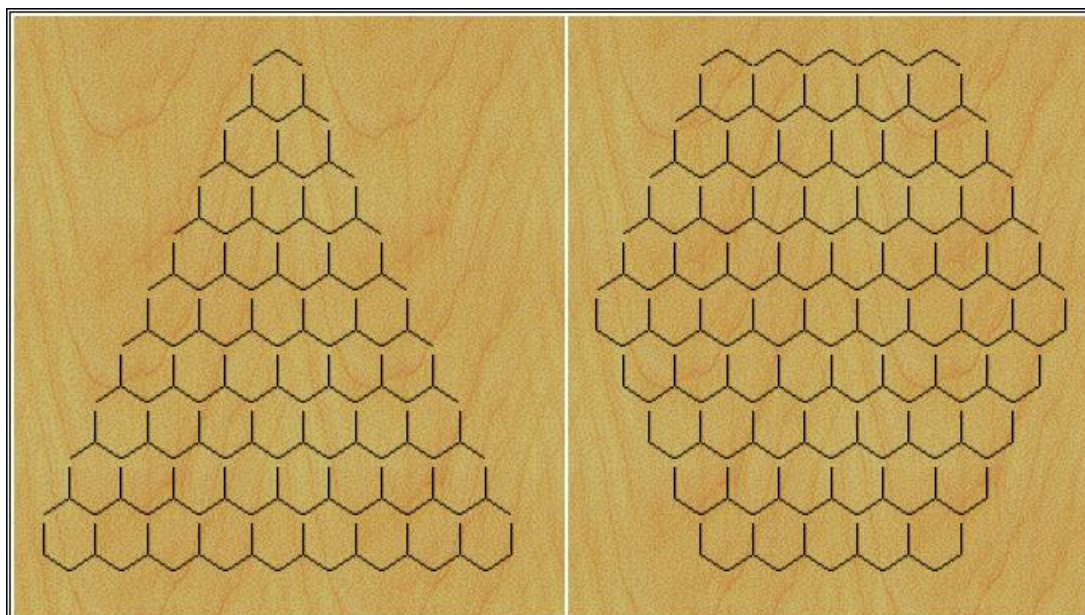
Obr. 3.3: Strom hry Nim [16]

Teraz sa pozrieme na konkrétny príklad, ktorý bude vychádzať z hry *NIM*. Pretože na poradí kôpok nezáleží, netreba rozlišovať napríklad medzi kôpkami (1 2) a (2 1). V každom uzle je zapísané, kto sa rozhoduje. V prvom uzle je to hráč 1, ktorý začína hru. Uzly (0), (1), (2), (3) sú rozhodovacie uzly. Rozhoduje sa hráč, ktorý je v nich vpísaný. Uzly (4), (5), (6), (9) a (10) nie sú typickými rozhodovacími uzlami, lebo hráči majú len jednu voľbu. U koncových uzlov grafu (7), (8), (11), (12), (13) a (14) môžeme podľa mena

vpísaného hráča zistiť, kto vyhral. Vyhral ten, koho meno je napísané v koncovom uzle, keďže už nie sú žiadne možnosti na ťahanie kameňov a súper zobral posledný [16]. Na nasledujúcom obrázku je zobrazený celý strom hry, ktorý graficky znázorňuje jednotlivé ťahy hráčov.

3.3.2 Hra Bulo

V tejto časti sa pozrieme na hru s názvom *Bulo*. Túto hru budeme neskôr analyzovať a pozrieme sa aj na jej softvérové riešenie. Hru môžeme zaradiť do kategórie hier v rozvinutom tvare. *Bulo* je vylepšená obdoba hry *NIM*. Taktiež je známa aj pod názvom *Tac Tix*. Jej tvorcom je Dán Piet Hein. Pravidlá hry sú nasledovné: Hra je navrhnutá pre dvoch hráčov. Hracia plocha obsahuje kamene. Ako základ sa používa hracia doska o veľkosti 4 x 4. Obsahuje teda 16 kameňov. Avšak plocha môže byť aj väčšia, s väčším počtom kameňov a taktiež s rozličným tvarom [18].



Obr. 3.4 Rozloženia hry Bulo [18]

Cieľ hry: Rovnako ako pri hre *NIM*, cieľom je prinútiť súpera, aby zobral posledný kameň z hracej plochy.

Východisková situácia: Všetky polia na hracej ploche sú obsadené kameňmi.

Hra: Prebieha tak, že sa hráči striedajú v ťahoch a v každom ťahu si hráč vyberie ľubovoľný počet kameňov, minimálne však jeden. Pri výbere kameňov počas ťahu platí

podmienka, že všetky kamene musia byť v jednom rade alebo stĺpci a zároveň musia spolu susediť. Čiže hráč nemôže odobrať prvý a tretí kameň v rade, ak už bol druhý kameň odobraný.

Koniec hry: Nastáva, keď jeden z hráčov zoberie posledný kameň. Hráč, ktorý zobral kameň, prehral [18].

3.3.3 *Hra piráta a Sicíľčana*

Túto hru môžeme zaradiť do kategórie hier v normálnom tvare. Ide o hru, kde hráči zahrajú svoje stratégie a hra končí. Je určená pre dvoch hráčov. Jeden z nich má úlohu piráta a druhý úlohu Sicíľčana [19].

Cieľ hry: Úlohou piráta je otráviť druhého hráča a Sicíľčanova úloha je vypiť pohár, v ktorom nie je jed.

Východisková situácia: Na stole sú dva poháre, z ktorých si obaja hráči vyberajú. Jeden je pirátov a druhý Sicíľčana. Predpokladom hry je to, že hráči svoje stratégie volia súčasne.

Hra: Pirát dá do jedného z pohárov jed. Buď teda do svojho, alebo do pohára Sicíľčana. Súper nevie, kde sa jed nachádza, a vyberá si súčasne pohár, z ktorého bude piť. Môže si zobrať buď svoj pohár alebo pohár druhého hráča. Druhý pohár zrejme vypije pirát.

Koniec hry: Hra končí tým, že jeden z hráčov vypije pohár s jedom. Hráč, ktorý vypije jed, prehral [19].

Na túto hru sa bližšie pozrieme v časti analýzy a softvérového riešenia.

3.4 **Analýza hry pirát a Sicíľčan**

V tejto časti zrealizujeme analýzu hry pirát a Sicíľčan a pomenujeme si dôležité fakty a vedomosti potrebné k tejto hre. Pravidlá hry sú popísané v časti 3.3.3.

3.4.1 *Maticová hra*

Hra piráta a Sicíľčana je maticová hra alebo aj hra v normálnom tvare. Keďže ide o maticovú hru, pozrieme sa na jej maticovú reprezentáciu.

Riadky zodpovedajú stratégiám piráta a stĺpce zase stratégiám Sicíľčana. Prvky matice predstavujú hodnoty, ktoré dostane pirát a koľko zaplatí Sicíľčan, alebo všeobecnejšie môžeme povedať, že prvok matice vyjadruje platby. Hráč 1 môže zahrať dve stratégie, ktoré nazveme δ_i . Jedna z nich je naliatie jedu do pohára Sicíľčana. Druhou naliatie jedu do vlastného pohára. Hráč 2 môže tiež zahrať dve stratégie, ktoré nazveme π_j . Pri použití prvej stratégie Sicíľčan vypije nápoj vo svojom pohári. Pri druhej vypije nápoj z pohára piráta. Ak hráč 1 zahrá δ_i a hráč 2 π_j , tak hráč 1 dostane A_{ij} a hráč 2 zaplatí A_{ij} [19]. Môžeme si to ukázať aj na konkrétnom príklade. Hráč 1, teda pirát, zahrá napríklad stratégiu δ_1 a hráč 2, ktorý je Sicíľčan, zahrá π_2 . δ_1 môže predstavovať napríklad stratégiu, kde pirát dá jed do pohára Sicíľčana. π_2 môže predstavovať stratégiu, kde Sicíľčan vypije pohár, ktorý patrí pirátovi. Už podľa popisu použitých stratégií je zjavné, že Sicíľčan vyhral a jed vypil hráč v pozícii piráta. Na výsledok sa môžeme pozrieť aj pomocou matice. Keďže hráč 1 si vybral stratégiu s indexom 1 a hráč 2 s indexom 2, tak hľadáme prvok matice A s indexami 1 a 2. Čiže ide o prvok $A_{1,2}$, ktorý predstavuje hodnotu -1. Podľa definície platieb môžeme povedať, že hráč 1 po zahratí stratégie δ_1 a hráč 2 π_2 , dostane -1 a hráč 2 zaplatí -1. Hráč 1 dostane -1, čo však predstavuje platbu hráčovi 2. Hráč 1 má podľa definície zaplatiť -1, čo ale znamená, keďže ide o záporné číslo, že hráč 2 dostane +1 [19]. Keď si rozoberieme možné použitie stratégie a vplyvy informácií, tak zistíme, že veľa závisí od samotných informácií. Môžeme sa pozrieť, ako by vyzerali stratégie hráčov, ak by bola niektorá informácia známa jednému z hráčov. Ak Sicíľčan nevie, že ho chce pirát otráviť, tak pirát použije stratégiu δ_2 . Sicíľčan použije stratégiu π_2 . Výsledok je, že vyhrá pirát. Ak Sicíľčan vie, že ho chce pirát otráviť, ale pirát nevie, že Sicíľčan to vie, tak pirát použije stratégiu δ_2 . Sicíľčan použije stratégiu π_1 . Výsledok je, že vyhrá Sicíľčan. Sicíľčan vie, že pirát ho chce otráviť, ale aj pirát vie, že Sicíľčan to vie. Tu vzniká otázka: do ktorého pohára dá pirát jed? Tu vidíme, že nastáva problém, a teda že ak nevieme žiadne informácie, ktoré by nám pomohli analyzovať možné ťahy súpera, tak sme odkázaní iba na vlastnú voľbu. Hra má konečný počet ťahov. Hráči zahrajú svoje stratégie a hra končí. Taktiež ju môžeme definovať ako antagonistickú. Jednotliví hráči majú opačné záujmy. Hra má nulový súčet, čiže pre každú dvojicu stratégií δ_i, π_j platí $e_1(\delta_i, \pi_j) + e_2(\delta_i, \pi_j) = 0$ [19]. Hra zároveň nemá jednoznačné riešenie v čistých stratégiách. Práve preto má zmysel túto hru riešiť aj softvérovo.

$$\begin{array}{c} \delta_1 \\ \delta_2 \end{array} \quad \begin{array}{cc} \pi_1 & \pi_2 \end{array} \quad A = \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix}$$

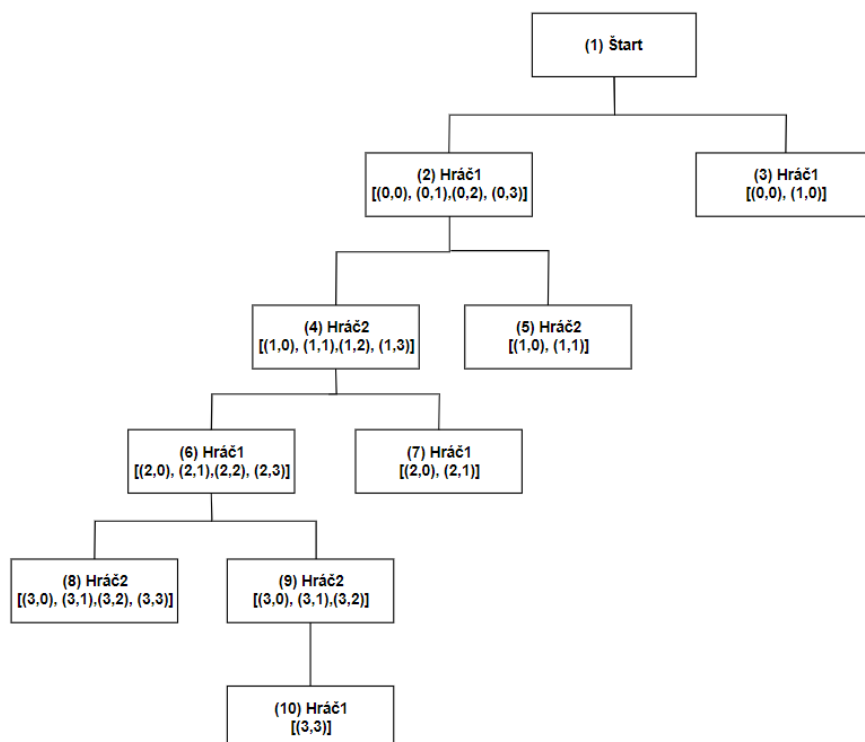
Obr. 3.5 Maticová reprezentácia hry Pirát a sicíľčan

3.5 Analýza hry Bulo

Pravidlá hry *Bulo* sme si vysvetlili v predchádzajúcich častiach. Teraz sa pozrieme na analýzu tejto hry. Pri skúmaní hry použijeme rovnako ako pri hre *NIM* analýzu pomocou stromu.

3.5.1 Strom hry Bulo

Analýzu pomocou stromu hry sme ukázali v predchádzajúcej časti na hre *NIM*. Teraz detailne predstavíme analýzu hry *Bulo* práve pomocou stromu hry. Keďže hra *Bulo* má pomerne veľké množstvo možností, tak sa nepozrieme na celý strom, ale len na jeho časti a zároveň hra bude v rozmeroch štyrikrát štyri. Je to z dôvodu, že je to najmenšie rozloženie hracieho poľa. V strome sa budú nachádzať informácie o tom, ktorý hráč aktuálne skončil ťah. Ďalej bude obsahovať informáciu o vybraných kameňoch. Vybraný kameň bude zapísaný v indexoch pozície v matici. Keďže je matica dvojrozmerná, tak bude obsahovať informáciu o tom, v ktorom riadku a v ktorom stĺpci sa nachádza daný kameň. Indexovanie sa začína nulou. Takže v prípade prvého riadku kameňov bude v zápise uvedené, že ide o riadok nula. Jediný koreň stromu bude označený ako štart, a to z toho dôvodu, že v počiatočnej fáze nie je odstránený žiaden kameň. V kompletnom strome hry môžeme vidieť všetky možné postupy. Vieme tak nájsť konkrétny bod a pozrieť sa, ako inak by sa dalo postupovať tak, aby napríklad druhý hráč vyhral. Nasledujúci príklad nám poslúži na analýzu ťahov hráčov. Túto analýzu neskôr použijeme aj v softvérovom riešení tejto hry. Príklad bude poskladaný z krokov jednotlivých hráčov.



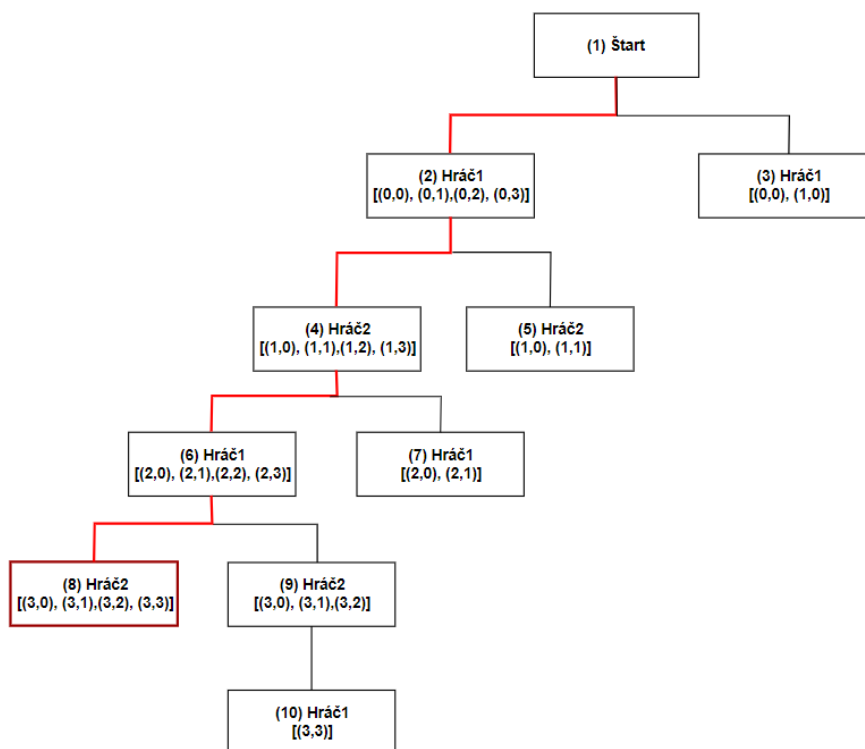
Obr. 3.6 Strom hry Bulo

Začíname koreňom stromu, kde máme napísané slovo štart. Po spustení hry má prvý hráč k dispozícii všetky možné kombinácie kameňov s podmienkou, že musí dodržať pravidlá. Na obrázku 3.2 sa nachádza podhra. V strome tejto hry nie sú vypísané všetky možnosti a vetvenie by bolo, pochopiteľne, mohutnejšie. No ukazuje potenciálne cesty pri použití určitých stratégií v jednotlivých ťahoch. Pri každom ťahu, okrem posledného, ukazujeme dve vetvy. To neznamena, že sú to jediné dve možnosti. Zobrazujú však, že existuje aj iná cesta, resp. cesty.

Hráč 1, ktorý predstavuje prvého hráča, vykonal prvý ťah. V tomto konkrétnom prípade si vybral možnosť zobrať všetky kamene z prvého riadku. Ide o uzol s číslom 2. V ňom sme vybrali kamene s pozíciami [(0,0), (0,1), (0,2), (0,3)]. Uzol číslo 3 je druhá možnosť a je len znázornením, že hráč mal aj iné možnosti ako práve tú, ktorú si vybral. Pre veľké množstvo možností v tejto vetve nebudeme pokračovať. Po odohratí ťahu je na rade hráč 2, ktorý je druhý v poradí. Druhý hráč si vyberal zo všetkých možností okrem tých, ktoré v sebe obsahujú už vybrané kamene. V našom prípade celý prvý riadok. Hráč 2 sa rozhodol zahrať podobnú taktiku ako hráč 1 a vybral celý druhý riadok. V strome to

predstavuje uzol s číslom 4 a hodnotami pozícií kameňov [(1,0), (1,1), (1,2), (1,3)]. Na obrázku 3.3 môžeme vidieť výber daného hráča a aj jednu z alternatívnych vetiev. V tomto momente sa na hracej ploche nachádzalo iba osem kameňov. Z nich si hráč 1 mohol vybrať. V aktuálnej vetve, ktorá je vyznačená červenou farbou, si hráči vybrali vždy rovnaké stratégie, čo sa týka počtu kameňov. Jediný rozdiel je v riadku. A aj v tomto prípade hráč 1 zvolil stratégiu výberu štyroch kameňov v treťom riadku. Ide o uzol s číslom 6 a pozíciami kameňov [(2,0), (2,1), (2,2), (2,3)].

Ďalej nasledoval hráč 2. V hre ostali štyri kamene v poslednom riadku. Hráč 2 sa rozhodol použiť rovnakú stratégiu ako doteraz, a to vybrať celý riadok. V tomto bode končí hra, lebo sa na ploche nenachádzajú žiadne kamene, a hráč 2 prehrá. Táto vetva v strome hry je najkratšia možná. Hráč teda končí v uzle 8 a hodnoty pozície kameňov sú [(3,0), (3,1), (3,2), (3,3)].



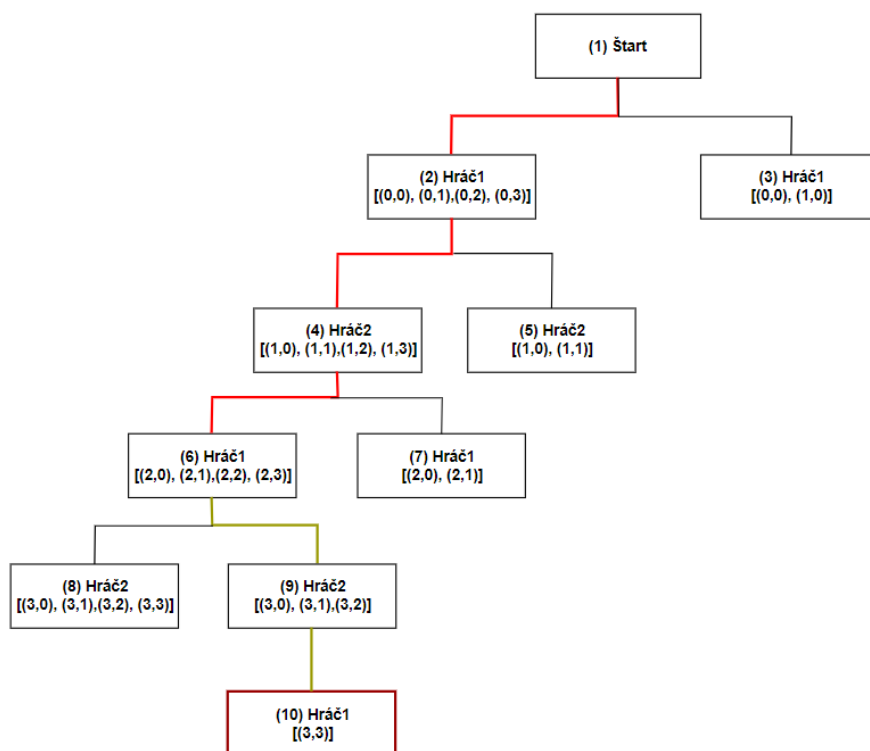
Obr. 3.7 Strom hry Buló s vyznačenou vetvou

3.5.2 Spätná indukcia

Spätná indukcia v teórii hier je iteratívny proces spätného uvažovania v čase, od konca problému alebo situácie, s cieľom vyriešiť konečné hry a odvodiť postupnosť optimálnych akcií. V každej fáze hry spätná indukcia určuje optimálnu stratégiu hráča, ktorý urobí posledný ťah v hre. Potom sa určí optimálna akcia predposledného hráča a zadá sa akcia posledného hráča. Tento proces pokračuje smerom späť, kým sa neurčí najlepšia akcia pre každý bod v čase. V skutočnosti sa určuje Nashovo ekvilibrium každej podhry pôvodnej hry.

Spätnú indukciu využijeme na analýzu našej hry. Keďže sme zistili, že zvolenými stratégiami na konci prehral hráč 2, tak zanalyzujeme možnosti a nájdeme cestu v strome, ktorá by odpovedala víťazstvu hráča 2. Vrátime sa najskôr o jeden ťah späť. Teraz sme sa dostali do uzla s číslom 6 a udialo sa to, že hráč 1 zahral stratégiu, pri ktorej vybral celý tretí riadok kameňov. Vieme, že stratégiu, ktorú vidíme v uzle 8, nie je vhodné použiť, lebo by viedla k rovnakému výsledku. keďže však vieme, že hráč, ktorý zoberie posledný kameň, prehrá, tak našou najlepšou možnosťou je dosiahnuť taký stav, kde hráč 2 dokončí ťah tak, že na hracej ploche ostane len 1 kameň. V tom momente vie, že hráč 1 ho musí zobrať a tým pádom hru prehral. Vieme, že hráčovi 2 ostali štyri kamene a všetky spolu susedia. Hráč 2 teda môže zobrať ľubovoľné množstvo kameňov.

Podľa vyššie napísanej analýzy vieme, že potrebujeme ponechať na konci ťahu jeden kameň. A v tomto prípade je to možné. Hráč 2 zoberie kamene s indexami $[(3,0), (3,1), (3,2)]$ a kameň s indexom $(3,3)$ ostane v hre. Na obrázku č. X vidíme, že sa hráčovi oplatí zvoliť stratégiu z uzla s číslom 9. Touto stratégiou donútil hráča 1, aby zahral jedinou stratégiu, ktorá mu ostala, a to vziať posledný kameň. Na obrázku č. 3.4 ide o uzol 10. Hráč 2 by sa v takomto momente stal víťazom a hráč 1 by danú hru prehral.



Obr. 3.8 Strom hry Bulo s novou vetvou po použití spätnej indukcie

V tomto ukázkovom príklade bola spätná indukcia len o jeden ťah späť. Pri inak zvolených stratégiách by sa to mohlo zmeniť a mohlo by byť potrebné vrátiť sa aj o viacej ťahov späť. Avšak v prípade návratu o viac ako jeden ťah je potrebné vždy si pripraviť a analyzovať aj možnosti, ktoré môže zahrať súper. A tie nevieme veľmi ovplyvniť. Jednoducho povedané, je potrebné pripraviť sa na viacero scenárov a prispôbiť stratégie aj podľa súperových ťahov.

4 Výsledky práce

4.1 Softvérové riešenie hry pirát a Sicíľčan

Ako prvé predstavíme softvérové riešenie maticovej hry pirát a Sicíľčan. V tejto časti práce sa pozrieme na kód, ako aj na samotný postup pri riešení tejto hry. Hra nie je veľmi zložitá a aj to môže byť jeden z dôvodov, prečo táto hra nebola vydaná hráčskym štúdiom alebo programátorom, ktorý by ju spravil verejne prístupnú. Ako jazyk hry sme si vybrali angličtinu, keďže zámerom by bolo aby táto hra nebola vyhradená len pre slovenských hráčov, ale aby bola k dispozícii aj potencionálnym hráčom zo zahraničia.

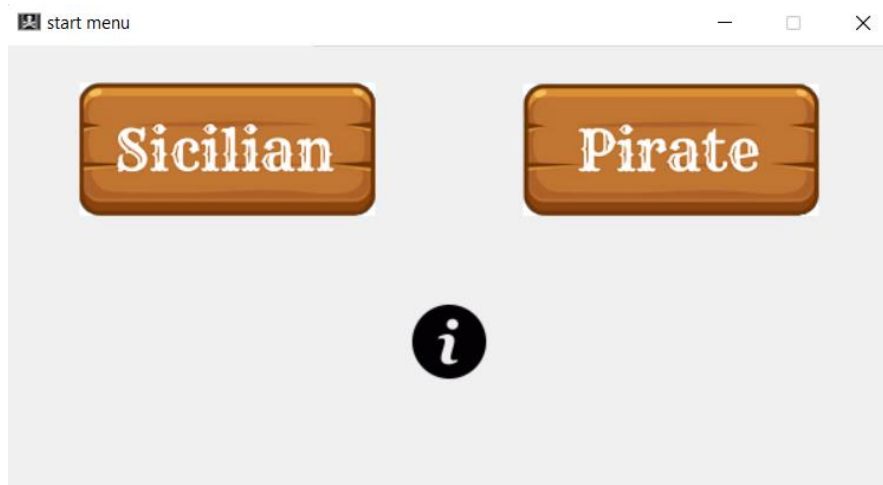
Samotná hra je určená pre dvoch hráčov. Na začiatku prípravy tejto hry sme pôvodne zvažovali, že to bude hra pre dvoch ľudí. Avšak nastal problém v tom, ako dosiahnuť, aby hráči spravili ťah naraz bez toho, aby boli ovplyvnení druhým hráčom. Keďže ide o hru pre jeden počítač, uskutočnenie simultánneho ťahu dvoch hráčov je komplikáciou. Riešením tohto menšieho problému bolo buď spraviť hru cez sieť, alebo vytvoriť hru pre jedného hráča s tým, že druhého hráča bude simulovať počítač. Hra cez sieť by fungovala na princípe, že každý hráč by bol na svojom počítači a pomocou siete by sa posielala informácia o zahratí ťahu a o samotnom vykonanom ťahu. Posielala by sa teda informácia, čo hráč vlastne spravil. Tým by sme odstránili problém s paralelným ťahom hráčov. Toto riešenie je veľmi elegantné, , keď si to však zanalyzujeme, tak pre takúto krátku hru je určite jednoduchšie, ak sa hra nainštaluje na jedno zariadenie a bude možné ju hrať aj bez pripojenia na internet. Preto sme zvolili možnosť vytvorenia hry pre jedného hráča, ktorý ju môže hrať, a súpera simuluje počítač. Výhodou je, že správanie počítača vieme nastaviť tak, aby bolo nezávislé od hráčovej voľby a zároveň ťah počítača nebol viditeľný pre hrajúceho hráča. Týmto sa problém súasných ťahov dvoch hráčov vyriešil.

Ďalšou časťou prípravy tejto hry bol grafický dizajn aplikácie. Zvolili sme si tému podobnú krčmovému stolu v dobe pirátov. Dizajn bol vytvorený pomocou aplikácie Canva a Adobe Photoshop, aby sme zabezpečili jednotnosť a možnosť vlastnej úpravy obrázkov. Hra sa začína úvodnou obrazovkou, kde má hráč k dispozícii tri tlačidlá. V prípade, že hráč nepozná túto hru, tak má k dispozícii tlačidlo s písmenom *i*. Toto tlačidlo otvorí vyskakovacie okno s pravidlami hry a vysvetlí hráčovi detaily hry a aj akým spôsobom môže

pokračovať. Na realizáciu vyskakovacieho okna sme využili modul *messagebox* z *tkinter*. Druhý parameter je dlhší a obsahuje celý text s inštrukciami. Jeho nasledujúci kód je:

```
Messagebox.showinfo("Help", "inštrukcie...")
```

Zvyšné dve tlačidlá slúžia na zvolenie svojej roly v hre, buď teda roly piráta alebo Sicíľčana. Dizajn a aj pozícia všetkých troch tlačidiel sú ukázané na obrázku 4.3. Po stlačení jedného z dvoch hlavných tlačidiel sa hra spustí a otvorí sa nová obrazovka s ďalšími inštrukciami.



Obr. 4.1 Úvodná obrazovka hry pirát a Sicíľčan

Na prvý pohľad nepodstatný prvok, ktorý sme upravili v našej hre, je logo hry. Tento drobný detail nerobí rozdiel v hraní hry ale skôr v celkovom vnímaní a pocitu z hry. Logo sa nachádza v ľavej časti obrazovky. Pre zmenu loga sme použili funkciu *iconbitmap*, ktorá patrí *tkinteru*. Do nej sa vkladá ako parameter obrázok iba typu *ico*.

```
ws.iconbitmap("images/pirate_flag.ico")
```

Nasledujúca obrazovka na obrázku 4.4 obsahuje informáciu o tom, akú rolu si hráč vybral. Táto informácia sa nachádza v lište vľavo hore. Dôležitým prvkom, ktorý je pre hráča informačný, je popis v hornej časti obrazovky. Ten hovorí o tom, čo má daný hráč spraviť. V našom prípade ide o text, ktorý hovorí hráčovi, aby si zvolil jeden pohár a vypil ho. V prípade, že by bola zvolená rola pirát, tak by sa popis zmenil na výzvu pre hráča aby určil, do ktorého pohára chce dať jed. Hra je vytvorená tak, aby hráč vedel, čo konkrétne má spraviť. Na obrazovke sa nachádzajú dve tlačidlá s obrázkom dreveného pohára. Po kliknutí na jeden z nich prebehne proces uloženia voľby a deaktivovania tlačidiel. Tieto

procesy sa vykonajú pomocou jednej funkcie s názvom *click*, do ktorej sa posiela parameter vybraného tlačidla.

```
def click(v):  
    global choose_option  
    choose_option = v  
    cup_sicilian["state"] = "disabled"  
    cup_pirate["state"] = "disabled"  
    drinking_text()
```

Posledným príkazom, ktorý sa vykoná v tejto funkcii, je funkcia s názvom *drinking_text*. Tá zobrazí dve tlačidlá s popisom, že aktuálne hráči pijú svoje nápoje.

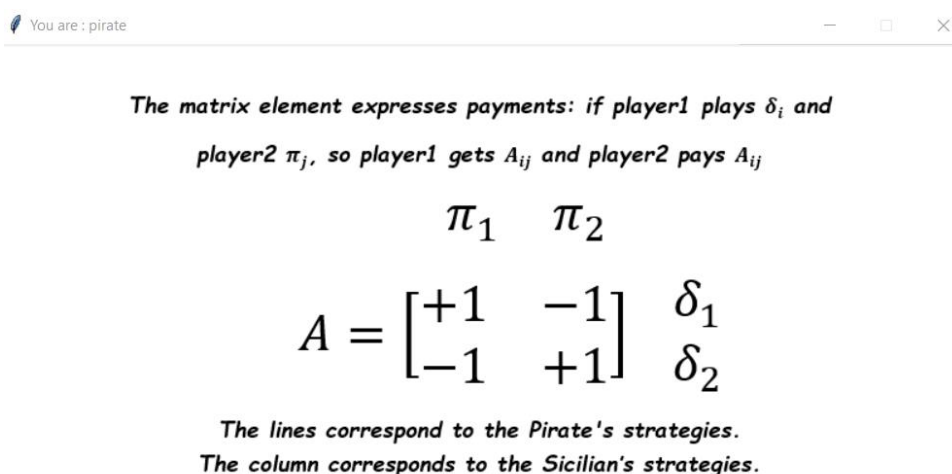


Obr. 4.2 Obrazovka hry pirát a Sicíľčan, kde sa hráč rozhoduje, akú stratégiu použije

Výber počítača je realizovaný náhodným generátorom. Je to preto, lebo chceme, aby hráči robili svoje rozhodnutia nezávisle od seba a lebo ani jedna možnosť nie je nijakým spôsobom výhodnejšia. Preto výber počítača bude úplne náhodný. Pre túto potrebu využijeme modul *random*, ktorý v sebe obsahuje funkcie na generovanie náhodných čísel. Vybrali sme si funkciu *randint*, ktorá poskytuje generovanie celých čísel.

```
random.randint(2)
```

Keďže si počítač môže vybrať len z dvoch možností, a to buď Sicíľčanov alebo pirátov pohár, tak nám stačí generovanie dvoch čísel. Každé číslo bude predstavovať jednu z možností. Ako parameter tejto funkcie sme vložili číslo dva. To nám dáva hodnoty buď 0 alebo 1. Každá z týchto dvoch hodnôt predstavuje jeden z pohárov. Hodnota 0 predstavuje pohár Sicíľčana a hodnota 1 predstavuje pohár piráta. Vygenerovanú hodnotu premeníme na hodnotu odpovedajúcu jednému z pohárov, aby sa na záver dali voľby oboch hráčov porovnať. Po kliknutí na text, ktorý hovorí o tom, že hráči aktuálne pijú, sa ukáže na obrazovke víťazný titulok. Na ňom bude napísané, ktorý z hráčov vyhral a či je to Sicíľčan alebo pirát. Pod týmto titulkom sa bude nachádzať tlačidlo, ktoré po kliknutí otvorí obrazovku s analýzou. Táto analýza bude obsahovať obrázok matice, ktorú sme spomínali vyššie v časti 3.2.1, a taktiež bude obsahovať popis toho, čo znamenajú jednotlivé hodnoty. Analýza slúži na rozoberanie daného problému a vysvetlenie tejto hry a problematiky z pohľadu teórie hier. Analýza ťahu je nepodstatná, keďže hráč môže pri ďalšej hre svoju voľbu zmeniť bez toho, aby sa jeho šance zvýšili alebo znížili. Táto analýza je zobrazená na obrázku 4.5.



Obr. 4.3 Obrazovka hry pirát a Sicíľčan s analýzou hry

4.2 Softvérové riešenie hry Buló

V tejto časti sa pozrieme na zvolené softvérové riešenie stolovej hry *Bulo*. Pravidlá hry sme definovali v časti 1.7.9 a samotné riešenie z nich vychádza. Rozoberieme si časti kódu, ktoré sú podstatné pre pochopenie riešenia, a taktiež sa pozrieme na problémy, ktoré nastali

pri tvorbe programu. Ako jazyk hry sme zvolili angličtinu, ktorá by mala túto hru priblížiť ľuďom z celého sveta.

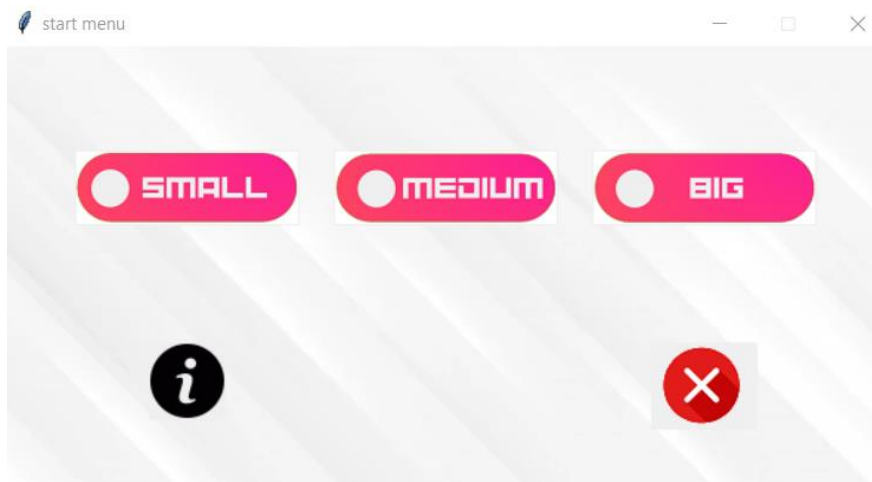
Hra je realizovaná pre dvoch hráčov, hoci existujú alternatívy aj pre väčšie množstvo hráčov. Pred samotným vývojom tejto hry sme riešili dilemu o tom, akým spôsobom bude prebiehať a ako budú hráči hrať. K dispozícii boli tri varianty. Prvé dva boli totožné ako u hry pirát a Sicíľčan. Prvým bolo spraviť hru cez internet. To by znamenalo, že hráči by dokázali hrať jednu hru na dvoch rôznych zariadeniach. Táto alternatíva je zaujímavá pre budúci rozvoj tejto aplikácie a umožnila by ľuďom z celého sveta hrať túto hru medzi sebou. Avšak na vývoj tejto aplikácie by bol určite vhodnejší iný programovací jazyk ako Python. Druhou alternatívou bolo spraviť hru pre jedného hráča. Druhý hráč by bol simulovaný a jednotlivé stratégie by generoval počítač. Tu by bolo základným kameňom vytvorenie logiky počítača. Tá by vychádzala zo stromu hry a počítač by sa pozrel do stromu, aký ťah má zahrať. Keďže však chceme simulovať hru dvoch reálnych hráčov, tak by sme museli pridať omylnosť počítača, aby jednotlivé kroky nerobil len podľa víťazných riešení v strome. Tretím variantom bolo vytvorenie hry pre dvoch hráčov, ktorí budú hrať na jednom počítači. Táto alternatíva najviac simuluje reálnu hru. Pri hre sa hráči striedajú, čiže svoje ťahy môžu vidieť. Zároveň môžu hodnotiť aj správanie hráča a podľa toho prispôbiť stratégiu. Tento variant je taktiež programátorsky jednoduchšia alternatíva ako zvyšné dve. Práve pre tieto dôvody sme sa rozhodli zvoliť túto možnosť.

Pre grafický dizajn aplikácie sme zvolili moderný štýl, a to aj napriek tomu, že hra je stará. Tento štýl nemá hráčovi len ukázať, že nejde len o obyčajnú starú hru, ale zároveň má hráčovi zabezpečiť príjemný zážitok.

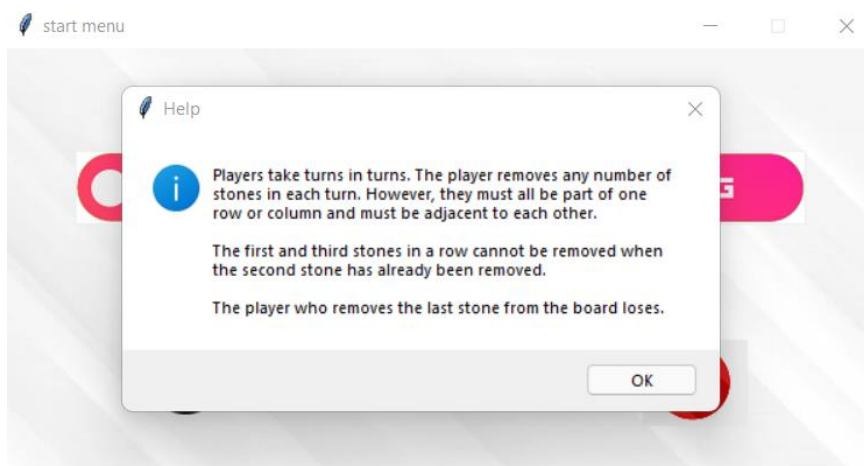
Na úvodnej obrazovke sa nachádza päť tlačidiel. Ružové tlačidlá s názvami *small*, *medium* a *big* slúžia na spustenie hry. Samotné názvy odkazujú na veľkosť hracej plochy. Po kliknutí na tlačidlo *small* sa otvorí plocha o veľkosti štyrikrát štyri alebo teda so 16 kameňmi. V prípade tlačidla *medium* ide o hraciu plochu v tvare trojuholníka so 45 kameňmi. Posledný variant hry je *big*. Ten poskytne hraciu plochu v tvare šesťuholníka so 61 kameňmi. Zároveň veľkosť hry je priamo úmerná náročnosti levelu.

Ďalšími tlačidlami na úvodnej obrazovke sú písmená *i* a *x*. Tlačidlo *i* slúži na otvorenie návodu, ktorý popisuje, ako sa táto hra hrá. Otvorí sa vyskakovacie okno, v ktorom sú detailne spísané pravidlá. Výzor tohto okna je ukázaný na obrázku 4.7. Toto okno je

realizované na rovnakom princípe ako pri hre pirát a Sicíľčan. V samotnom kóde sme použili modul *messagebox*. Tlačidlo s písmenom *x* slúži na zatvorenie obrazovky. Čiže ak sa hráči rozhodnú nehrať túto hru, tak toto tlačidlo im umožňuje ukončiť aplikáciu.



Obr. 4.4 Úvodná obrazovka hry Bulo



Obr. 4.5 Vyskakovacie okno s návodom hry Bulo

Po zvolení jednej z veľkostí hry sa otvorí okno s hrou. V ľavom hornom rohu sa vždy nachádza informácia o type hry. Rozloženie hry je dané poľom vzdialeností na súradnici x od stredu. Týmto sme dosiahli to, že vieme v prípade požiadaviek hráčov pridať aj iné rozloženie hracieho poľa. Nasledujúce pole s názvom *square_playing_field* ukazuje príklad rozloženia kameňov v štvorcovom tvare o veľkosti 4 x 4. Pozícia 0 predstavuje stred hracej plochy s kameňmi. Podmienka, ktorú sme si stanovili pre tieto polia, je, že sa

kamene nachádzajú od seba vzdialené o dve hodnoty. A keďže začíname od stredu, tak sa v poli nachádzajú aj negatívne čísla, ktoré slúžia na ukladanie kameňov po ľavej strane. Rozloženie kameňov podľa poľa *square_playing_field* je ukázané na obrázku 4.8.

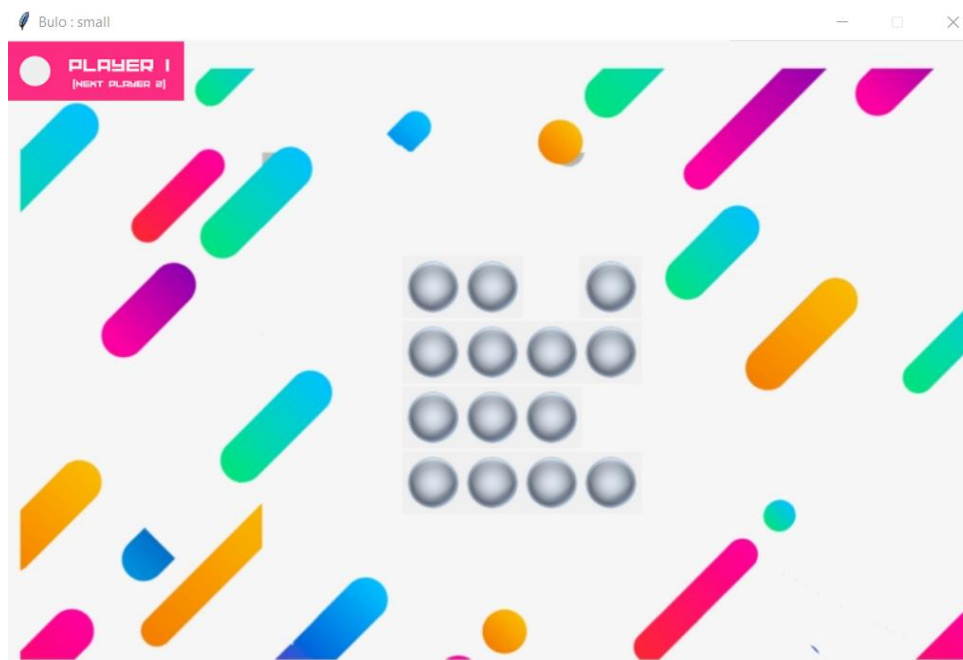
$$\begin{aligned} \text{square_playing_field} = \\ & [[-3, -1, 1, 3], \\ & \quad [-3, -1, 1, 3], \\ & \quad [-3, -1, 1, 3], \\ & \quad [-3, -1, 1, 3]] \end{aligned}$$

Pri rozhodovaní sa o tom, akým spôsobom naprogramovať kamene na hracej ploche, sme zvažovali dve možnosti. Prvou bolo spraviť kamene ako obrázky a potom podľa pozície oboch súradníc zisťovať, či hráč klikol na kameň. Avšak tu by sme kontrolovali každý jeden klik v obrazovke. Druhý riešením je už spomínané použitie tlačidiel, ktoré zabezpečujú to, že kliknutie riešia sami a to len v prípade, že je dané tlačidlo zakliknuté. Zároveň pomocou tlačidla vieme ihneď vyvolať funkciu, ktorá sa má vykonať. Samotné kamene sú teda spravené ako tlačidlá s obrázkom symbolizujúcim kameň. Využili sme funkcionality z modulu *tkinter* s názvom *Button*. Tento vytvorený objekt umiestnime pomocou funkcie *place*, kde nastavíme pozíciu na oboch súradniciach. V samotnom príkaze tlačidla je najdôležitejšia funkcia, ktorá sa vyvolá v prípade kliknutia na dané tlačidlo. Ide o funkciu s názvom *click* s parametrom obsahujúcim pozíciu daného tlačidla v poli. Funkcia *click* skontroluje, či hráčom vybrané tlačidlo môže byť vymazané. Táto kontrola funguje na základe pravidiel, kde hráč, ak berie viac ako jeden kameň, môže zobrať iba kameň, ktorý je buď v rade alebo v stĺpci. Rad alebo stĺpec musí byť rovnaký ako jeho prvý výber, respektíve ak ide o väčšie rozloženie, tak sa môže vymazať aj tlačidlo, ktoré je na diagonále s prvým tlačidlom. Po samotnej kontrole nasleduje vymazanie tlačidla.

```
tkinter.Button(canvas, image = stone_img, command = lambda x
               = (count, i): click(x), borderwidth = 0)
```

V horných rohoch obrazovky sa nachádza tlačidlo s názvom hráča, ktorý je aktuálne na ťahu. V zátvorkách je napísané, čo sa stane po kliknutí na tlačidlo. Teda odovzdá sa ťah druhému hráčovi. Čiže v momente, ak už hráč nechce ťahať ďalšie kamene a chce ukončiť

svoj ťah, musí kliknúť na tlačidlo. Obrázok 4.8 zobrazuje hru, kde už obaja hráči zahrali svoje prvé ťahy a každý z hráčov si zobral práve jeden kameň.



Obr. 4.6 Hracie pole 4x4 hry Bulo s odobratými kameňmi

Kontrola, či si hráč, ktorý je na ťahu, vybral správny kameň, sa deje ihneď po kliknutí naň. Táto kontrola sa líši v závislosti od zvoleného typu hry. V prípade, že ide o hru s názvom *small*, tak sa kontroluje to, či hráč zobral susedný kameň buď z rovnakého riadka, alebo stĺpca. Pri zvyšných dvoch typoch pribúda možnosť kontroly diagonály. Čiže v prípade, že sa kameň nachádza na diagonále s posledným vybraným kameňom a je jeho susedom, tak kontrola povolí odobratie kameňa. Na kontrolu vplýva okrem rozlíšenia veľkosti hry aj to, koľký kameň v poradí vyberáme v jednom ťahu. V prípade úplne prvého kameňa má hráč voľnú ruku a môže si zobrať ktorýkoľvek z kameňov, ktoré sú ešte aktuálne v hre. Ak sa hráč rozhodne odobrať druhý kameň, tak sa po kliknutí spustí prvý typ kontroly. Táto kontrola je realizovaná funkciou s názvom *check_1_move*. Do nej posielame parametre pozície tlačidla. V tejto funkcii kontrolujeme to, či sa vybraný kameň nachádza v rovnakom rade alebo v stĺpci, respektíve na diagonále. Ďalej v nej nastavíme smer. Smer ukladáme do premennej s názvom *direction*. Obsahuje v sebe informácie o posune v rade, stĺpci a súradniciach, ktoré definujú smer. Čiže v prípade, že si hráč vybral kameň z riadku, tak bude premenná obsahovať údaj o súradnici *x*. Premenná *direction* sa bude využívať v prípade, ak sa

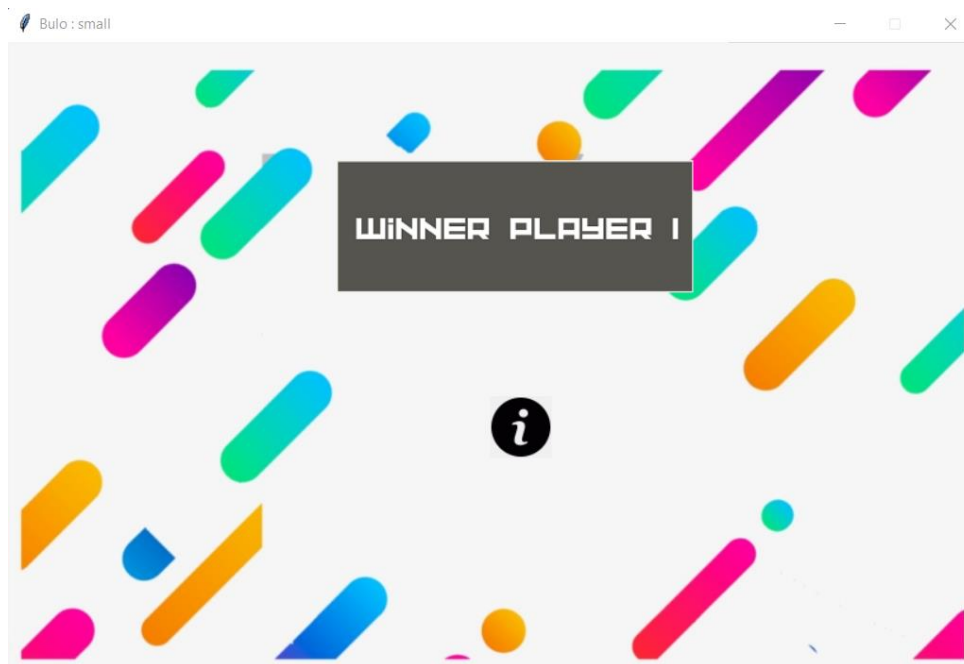
hráč rozhodne ťahať aj ďalšie kamene. Hovorí o tom, v ktorom smere musí hráč zobrať ďalší kameň. Teda buď v rovnakom rade, stĺpci alebo prípadne diagonálne. Pri ťahu druhého kameňa je premenná obsahujúca informáciu smeru zbytočná, pretože až v tomto ťahu sa hráč rozhodne, ktorým smerom bude vyberať nasledujúce kamene. V prípade, že sa hráč rozhodne vybrať aj tretí kameň, tak sa kontrola vykonáva na základe pozície vybraného kameňa a smeru. Ak obe premenné spĺňajú podmienky, tak je hráčovi povolené kameň odobrať. Funkcia, ktorá vykonáva túto nasledovnú kontrolu, má názov *check_2_more_move*. Rovnako ako predchádzajúca funkcia obsahuje parametre pozície kameňa. Nasledujúci kód obsahuje ukážku kódu z funkcie *check_2_more_move*. Ukazuje kontrolu, či sa vybraný kameň nachádza na súradnici *x*, a ak áno, či má správny smer. V prípade, že spĺňa podmienky, tak sa vymaže tlačidlo pomocou funkcie *delete_btn* a nastaví sa pozícia.

```
if coord_axis == 'x':
    if (x - x1) == direction1 and (y - y1) == 0:
        delete_btn(x, y)
        set_position_move(x, y)
```

Po odobratí všetkých kameňov plochy sa zobrazí informácia o tom, kto vyhral hru. Tento titulok je pridaný ako obrázok pomocou príkazu *PhotoImage*. Do tohto príkazu sme vložili jediný parameter, a to cestu v počítači ku konkrétnemu obrázku. Príkaz, ktorý sme použili v našom kóde, vyzerá nasledovne:

```
win_banner1 = PhotoImage(file = 'images/winner_1.png')
```

Ďalším prvkom na tejto ploche je tlačidlo s písmenom *i*. Tentokrát nejde o pravidlá hry, ale o informáciu s analýzou ťahov hráčov. Po kliknutí na toto tlačidlo sa otvorí analýza odohratej hry.



Obr. 4.7 Obrazovka po odohratí hry Bulo, kde sa víťazom stal hráč 1

4.2.1 Analýza

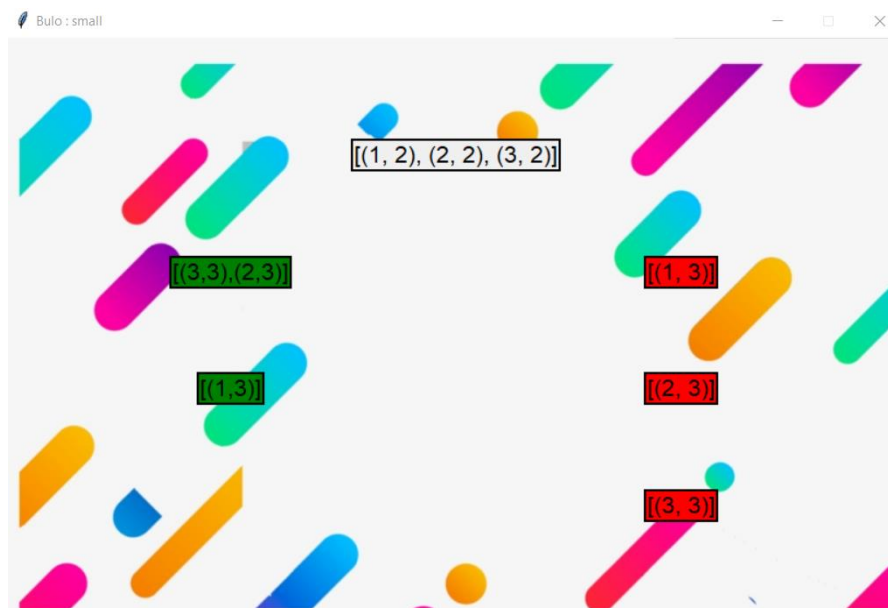
V tejto časti sa pozrieme na softvérové riešenie stromu hry. Všetky teoretické znalosti použité v tejto analýze sú zhrnuté v časti 3.3. Analýza je jedným z dôležitých bodov tejto práce, a preto na ňu kladieme veľký dôraz. Zároveň prináša možnosti rekapitulácie a pomoci s analýzou problému pri samotnej hre. Z týchto, ale aj iných dôvodov je analýza veľmi zaujímavá a je dobrým rozšírením v hre.

Pri príprave riešenia, akým spôsobom vizualizovať ťahy hráčov, sme vychádzali zo známej analýzy pomocou stromu hry. Táto analýza poskytuje vizuálne ľahko pochopiteľný výsledok, ktorý dá každému, kto bude hrať túto hru, čo najviac informácií. Pri rozhodovaní sa, akým spôsobom spraviť analýzu stromu, sme vyberali z viacerých možností. Prvou možnosťou, ktorá je veľmi zaujímavá, je vyhľadávací strom. Pre tento koncept hry je vhodnou možnosťou všeobecný strom. Na rozdiel od binárneho vyhľadávacieho stromu, ktorý je veľmi známy, ponúka možnosť mať viacero synov. Všeobecný strom je teda dátová štruktúra, kde každý vrchol môže mať ľubovoľný počet svojich synov. Pomocou tejto štruktúry by sme sa vedeli vnárať, vynárať a hľadať jednotlivé cesty a alternatívy ciest. Táto štruktúra prináša svoje výhody a jej využitie, či už pri tejto hre alebo aj iných podobných hrách, má svoje opodstatnenie. Dôvod, prečo sme všeobecný strom nevyužili, je, že každý vrchol

má veľké množstvo synov a vytvorenie takéhoto stromu by bolo pomerne komplikované aj pre najjednoduchší typ hry. Ak by sme vybrali hru *small*, tak v prvom ťahu by hráč mal k dispozícii 64 možností. Čiže v dátovej štruktúre by bolo pre prvý vrchol 64 synov. Ďalšou komplikáciou, aj keď nie veľkou, je vytvorenie samotnej dátovej štruktúry, ktorú by sme naplňali dátami.

Už pri analýze tohto riešenia sme sa rozhodli spraviť rozbor len pre typ hry *small*, keďže pre väčšie typy hier by to bolo komplikovanejšie. Samotná analýza aj pre najmenšie rozloženie poskytne dostatočné informácie na pochopenie hry a poskytne aj dostatočné informácie na analýzu ťahov jednotlivých hráčov. Pri väčších rozloženiach hracieho poľa by šlo o podobný postup, avšak rozšírený o ďalšie kroky.

Samotná vizualizácia stromu hry tiež so sebou prináša komplikácie, najmä týkajúce sa veľkosti stromu a prehľadnosti. Riešenie tohto problému, ako aj hľadanie alternatívy k všeobecnému stromu, prinieslo taký výsledok, že budeme analyzovať iba časť tohto stromu a v nej hľadať alternatívne cesty. Tým sme dosiahli, že nebudeme musieť zobrazovať celý strom, ale len časť, ktorá sa bez problémov zmestí do hracej obrazovky.



Obr. 4.8 Analýza - Strom odohratej hry Bulo

Preto sme hľadali spôsob, ako realizovať, aby analýza dávala zmysel hráčom a mala aj pridanú hodnotu,. Hra sa dá analyzovať buď vo všeobecnosti, alebo ako konkrétna jedna hra,

ktorú hráči odohrali. My sme si zvolili druhý spôsob, a to že sa pozrieme na konkrétnu hru, ktorú hráči odohrali. Výhody tohto postupu sú nám ihneď jasné. Prvou je, že je analýza spätá s poslednou odohratou hrou, čo zabezpečuje, že si hráči budú pamätať väčšinu vecí, ktoré sa udiali. Tým pádom poskytne okamžitú spätnú väzbu pre hráča, ktorý môže okamžite analyzovať svoj ťah. Ďalším dôvodom je pochopenie analýzy. Ak si hráči pozrú analýzu ihneď po odohratí hry, tak aj samotné pochopenie analýzy bude hlbšie a budú si ju vedieť prepojiť s nedávno odohratou hrou. A posledným dôvodom je poučenie pre hráča, ktorý prehral. Na základe analýzy sa vie pozrieť, že čo mohol spraviť inak, aby túto hru vyhral. To sú cenné poznatky pre budúce potencionálne hry.

Po vymenovaní niektorých výhod nasleduje samotný opis toho, ako sa analýza realizuje. Základným východiskom je, že nebudeme analyzovať celý strom odohratej hry, ale len jeho časť. Táto časť bude pozostávať z dvoch vetiev. Jedna z nich bude ukazovať vetvu na základe skončenej hry. Bude teda obsahovať vrcholy s hodnotami vybraných kameňov. Druhá vetva bude ukazovať cestu, akou by hráč, ktorý prehral reálnu hru, vyhral. Dôležitým aspektom je aj to, ako ďaleko postupovať naprieč stromom a koľko ťahov spätne analyzovať. Pre našu hru sme sa rozhodli použiť analýzu posledných troch ťahov. Troch preto, aby hráč, ktorý prehral, mohol začať ovplyvňovať hru a nemusel čakať na ťah od druhého hráča, ktorý by mu vyhovoval. Samozrejme, ťah druhého hráča bude musieť prísť, ale čím menší bude ich počet, tým viac to môže mať prvý hráč pod kontrolou. Vrátime sa o tri ťahy späť a z kameňov, ktoré nám tak ostali v hre, vyskladáme novú cestu. Základom tejto analýzy je, že počet ťahov v hre musí byť nepárny. V tom momente dosiahneme to, že vyhrá ten hráč, ktorý pôvodnú hru prehral. Strom začína vrcholom, ktorý obsahuje informáciu o ťahu, ktorý vykonal víťazný hráč. Potom sa vrchol delí na dve vetvy. Červená vetva obsahuje vrcholy obsahujúce informácie o ťahoch, ktoré sa udiali v odohratej hre. V ľavej vetve, ktorá má zelenú farbu, sa nachádzajú hodnoty na základe analýzy a ich počet je teda nepárny. Tie označujú víťaznú vetvu pre druhého hráča.

4.2.2 Kód

V tejto časti sa pozrieme na softvérové riešenie analýzy z pohľadu kódu. Tento kód vychádza z informácií, ktoré sme spomenuli v časti analýzy hry *Bulo*. Kód pozostáva z dvoch kľúčových funkcií, ktoré sú doplnené o pomocné funkcie. Tieto dve funkcie tvoria

základnú kostru analýzy. Prvou z funkcií je funkcia s názvom *tree_path*. Jej úlohou je nájsť víťaznú cestu pre hráča, ktorý prehral aktuálnu hru. Základnou myšlienkou tejto funkcie je pozrieť sa na tento problém z dvoch rôznych pohľadov. Tieto pohľady vychádzajú z vedomosti, že ak chce hráč vyhrať, musí mať páry počet ťahov v strome, mimo koreňa. Tu ide o hráča, ktorý prehral hru. V prípade párneho počtu ťahov oboch hráčov víťazí hráč, ktorý prehral aktuálnu hru. Dostávame výsledok, ktorý ukazuje potenciálnu vetvu v strome s víťazným koncom pre hráča, ktorý prehral pôvodnú hru. Avšak táto problematika je trochu širšia a musíme sa pozrieť na možnú analýzu tohto problému. Jednou z prvých úloh v tejto funkcii je zistiť počet kameňov, ktoré ostali. Vrátime sa o tri ťahy späť. Na základe tohto počtu zistíme, do ktorej z dvoch nasledujúcich podmienok vstúpime. Prvá podmienka ráta s tým, že počet kameňov je páry. Druhá podmienka vychádza z toho, že počet kameňov je nepáry. Toto rozdelenie je dôležité, aby sme vedeli lepšie pracovať s týmito dátami. V prípade nepárneho počtu kameňov vieme povedať, že ak chceme, aby hráč vyhral, potrebujeme nájsť minimálne jeden pár. Je to z dôvodu, ktorý sme spomenuli už vyššie, to že ak hráč chce vyhrať túto hru, tak musí byť v tejto podčasti hry počet ťahov páry. Tým pádom z nepárneho počtu kameňov musíme nájsť minimálne jeden pár, ktorý sa zahrá v jednom kole a vďaka tomu bude možnosť zahrať páry počet ťahov tejto hry. Tým dosiahneme víťazstvo hráča, ktorý pôvodne prehral. Samozrejme, môžeme hľadať aj ďalšie dvojice, ale už musíme myslieť na to, že treba nájsť páry počet dvojíc, ktoré nasledujú. To vo výsledku znamená, že počet párov, ktoré by sme mali nájsť, bude nepáry. Teraz sa pozrieme na druhý prípad, a to keď je počet kameňov páry. Pri analýze vidíme, že ak by každý hráč zahral po jednom kameni v jednom ťahu, tak hráč, ktorý prehral pôvodnú hru, by aktuálnu vyhral. Avšak aj v tomto prípade sa oplatí pozrieť sa na prípadné dvojice kameňov. Tento prípad bude opačný ako predchádzajúca podmienka. Budeme v nej hľadať páry počet dvojíc, lebo len vďaka tomu môže hráč, ktorý pôvodne prehral, hru vyhrať. V prípade nepárneho počtu párov vzniká nepáry počet ťahov, a tým pádom je výsledok rovnaký ako na konci odohranej hry. Okrem týchto podmienok je dôležitou časťou tejto funkcie aj hľadanie dvojíc. V hre je možné nájsť okrem dvojíc jednotlivých prvkov aj trojice a štvorice. Pre jednoduchosť pochopenia a zároveň jednoduchosť kódu sme sa pri analýze rozhodli hľadať iba dvojice a jednotlivé prvky. Vychádza to z toho, že posledné tri ťahy sú väčšinou obmedzené výberom z malého počtu kameňov. Preto nám na dobrý výsledok stačí vyhľadávanie dvojíc. Pravdepodobnosť nájdenia trojice a štvorice je pomerne malá a zároveň ju vieme

nahradiť nájdením niektorého z párov, respektíve jednotlivého kameňa. V nasledujúcom kóde je popísaná funkcia s názvom *tree_path*, ktorú sme popisovali vyššie v texte. Ide o kód, ktorý v sebe obsahuje pseudokód, pričom vnútro podmienok riešiacich, či je počet kameňov párny alebo nie, je napísané v pseudokóde. Funkcia dostáva dva parametre, s ktorými pracuje.

```
def tree_path(tree_btn, win_banner_btn):

    last_3_moves = all_moves[len(all_moves)-1]+all_moves[len(all_moves)-2]
    +all_moves[len(all_moves)-3]

    if len(last_3_moves) % 2 == 0:

        hľadanie cesty pre rôzne počty kameňov

    else:

        hľadanie cesty pre rôzne počty kameňov
```

Teraz sa pozrieme na časť pseudokódu, ktorý sa týka hľadania cesty pre rôzne počty kameňov. Po splnení podmienky, že je počet párny alebo nepárny, sa spúšťa hľadanie jednotlivých ťahov. Tieto ťahy hľadáme z celkového počtu kameňov a podľa ich počtu. Hľadáme primeraný počet dvojíc a samostatných prvkov. Čím je väčší počet kameňov, tým sa snažíme nájsť väčšie množstvo dvojíc. V prípade malého počtu nie je potrebné hľadať veľké množstvo dvojíc. Tým riešime prípady ako napríklad, keď ostane počet kameňov 12. Analyzujeme iba tri ťahy späť, takže viac ako 12 kameňov na ploche ostať nemôže. Je to z dôvodu, že aj maximálny počet kameňov, ktoré môže hráč zobrať v jednom ťahu, je štyri. Na začiatku týchto funkcií hľadáme všetky dvojice, ktoré existujú. Potom na základe počtu týchto dvojíc a počtu kameňov nájdeme riešenie, ktoré nám dá víťaznú cestu hráča, ktorý pôvodne prehral. Táto funkcia nám okrem dvojíc dáva aj samostatné prvky, ktoré ostali bez dvojice, respektíve dvojica nebola vybraná. Druhou kľúčovou funkciou tejto časti programu je funkcia s názvom *tree_visualisation*. Má na starosti vykreslenie dvoch vetiev stromu hry. Táto vizualizácia zahŕňa koreň stromu obsahujúci ťah hráča, ktorý vyhral. Po ňom sa strom rozvetvuje a obsahuje zelenú vetvu a červenú vetvu. Červená vetva zahŕňa prvky, ktoré boli zahrané v predchádzajúcej hre, a zelené prvky, nachádzajúce sa v druhej vetve, ukazujú alternatívnu víťaznú cestu pre druhého hráča. Vizualizácia vyobrazuje

okrem koreňa stromu tieto vetvy nasledovne: Víťazná vetva hráča, ktorý pôvodne prehral, sa vykresľuje takým spôsobom, že sa najskôr použijú existujúce páry, a až potom sa vykreslia samostatné prvky. Červená vetva sa vykreslí v tom poradí, v akom sa jednotlivé kolá udiali.

Funkcia *tree_visualisation* už v sebe neobsahuje žiadnu kontrolu. Vychádza iba z dát, ktoré sme získali pomocou predchádzajúcej funkcie. Do tejto funkcie posielame tri parametre. Prvým sú dvojice kameňov. Druhým sú jednotlivé kamene, ktoré nie sú v žiadnej použitej dvojici, a posledný parameter obsahuje všetky ťahy. Ďalej pracuje s objektami plochy ako *label* a pomocou cyklu prechádzame nájdené výsledky. Tie potom vizualizujeme v jednotlivých *labeloch*. Nasledujúci text obsahuje kód tejto funkcie, ktorý pre rozsiahlu dĺžku nie je kompletný. Obsahuje kľúčové časti a zvyšné časti sú doplnené popisom kódu, ktorý sa v programe nachádza.

```
def tree_visualisation(pairs, items, all_moves):
```

```
    h_item = height/((len(pairs)/2)+len(items)+3)
```

```
    start_node = Label(parametre labelu)
```

```
    for i in range(0, len(pairs), 2):
```

```
        zobrazí dvojice
```

```
    for i in items:
```

```
        zobrazí samostatné kamene
```

```
    for i in range(3):
```

```
        zobrazí prvky vetvy odohratej hry
```

Ďalšou funkciou, ktorú si vysvetlíme v nasledovnom odseku je funkcia s názvom *change_player*. Slúži na zmenu ťahov medzi hráčmi. Jedná sa o funkčnosť tlačidla nachádzajúceho sa buď v ľavej ale pravej hornej časti obrazovky. Táto pozícia tlačidla závisí od toho, ktorý hráč je aktuálne na ťahu. Funkcia má za úlohu meniť hodnoty dvoch premenných. Prvá s názvom *actual_player* hovorí o tom, ktorý hráč je aktuálne na ťahu. Nastavujeme hodnoty 1 a 2. Hodnota 1 hovorí, že aktuálne je na ťahu prvý hráč. Druhá premenná *is_clicked* slúži na to, aby sme skontrolovali, či hráč vykonal aspoň jeden ťah. Keďže podľa pravidiel musí zobrať aspoň jeden kameň. V prípade, že hráč vykoná aspoň jedno odobratie, tak sa hodnota nastaví na *True*. Ak sa hráč rozhodne predať svoj ťah druhému

hráčovi, tak nastavujeme premennú na False. Okrem nastavenia premenných táto funkcia zobrazuje hráčovi tlačidlá na prenechanie ťahu súperovi zavolaním funkcie `set_change_player_param`.

```
def change_player():  
  
    global actual_player, is_clicked  
    if is_clicked:  
  
        is_clicked = False  
        all_moves.append(player_move.copy())  
        player_move.clear()  
  
    if actual_player == 1:  
  
        actual_player = 2  
        set_change_player_param('images/next_player2.png', 670, 0)  
  
    else:  
  
        actual_player = 1  
        set_change_player_param('images/next_player1.png', 0, 0)
```

Zvyšné časti kódu analýzy sa nachádzajú v zdrojovom kóde programu, kde sú aj komentáre k jednotlivým funkciám. Komentáre obsahujú popis toho, na čo konkrétne slúžia jednotlivé funkcie.

Záver

Cieľom tejto inžinierskej práce bola implementácia zvolených typov hier v oblasti teórie hier. Táto skúsenosť je veľmi pozitívna a priniesla veľmi veľa vedomostí na základe, ktorých vidíme, aký potenciál má táto téma.

V práci sme predstavili teóriu v tejto oblasti. Na základe nej sme si vybudovali dobré základy na dobré pochopenie a spracovanie ďalších úloh. Konkrétnymi príkladmi hier a ich analýzou sme prenikli do hĺbky tejto problematiky, ktorú sme neskôr využili pri implementácii. Okrem softvérového riešenia sme sa venovali aj analýze ťahov hráča. Na základe nami zvolených postupov sme prešli k realizácii úloh v programovacom jazyku.

V tejto práci sme softvérovo riešili dve hry. Prvou bola hra pirát a Sicíľčan a druhou hra Bulo. Tieto hry boli riešené pomocou jazyku python na základe metodiky spísanej v tejto práci. Nami zvolené postupy nám dali ako výsledok hotové hry, v ktorých sa nachádza aj samotná analýza. Samotný výsledok je veľmi prijateľný a poskytuje používateľovi presne to, čo očakáva od hry ale zároveň ponúka aj rozšírenú možnosť analýzy. Hra Bulo prináša ešte veľa priestoru na zlepšenie pre ďalšie práce. Jej potenciál je väčší hlavne, čo sa týka hry po sieti, resp. hra s umelou inteligenciou.

Interaktívne možnosti sú určite dobrým smerom na rozšírenie poznania Teórie hier u väčšieho počtu ľudí a určite poskytujú priestor na ďalšiu tvorbu, a prácu v tejto konkrétnej oblasti. Zároveň to môže byť výzva pokračovať v tejto téme a rozširovať možnosti spoznávania aj pre nás, ale aj ostatných.

Zoznam obrázkov

1.1 Klasifikácia hier.....	11
3.1 Graf oblúbenosti jednotlivých vývojových prostredí.....	18
3.2 PyCharm prostredie.....	19
3.3 Strom hry Nim.....	22
3.4 Rozloženia hry Bulo.....	23
3.5 Maticová reprezentácia hry Pirát a sicíľčan.....	26
3.6 Strom hry Bulo	27
3.7 Strom hry Bulo s vyznačenou vetvou.....	28
3.8 Strom hry Bulo s novou vetvou po použití spätnej indukcie.....	30
4.1 Úvodná obrazovka hry pirát a Sicíľčan.....	32
4.2 Obrazovka hry pirát a Sicíľčan, kde sa hráč rozhoduje, akú stratégiu použije.....	33
4.3 Obrazovka hry pirát a Sicíľčan s analýzou hry.....	34
4.4 Úvodná obrazovka hry Bulo.....	36
4.5 Vyskakovacie okno s návodom hry Bulo.....	36
4.6 Hracie pole 4x4 hry Bulo s odobratými kameňmi.....	38
4.7 Obrazovka po odohratí hry Bulo, kde sa víťazom stal hráč 1.....	40
4.8 Analýza – Strom odohratej hry Bulo.....	41

Zoznam použitej literatúry

- [1] GOGA, Marián. Teória Hier, Wolters Kluwer (Iura Edition), 9788080786137.
- [2] MYERSON, R.B.: Game Theory: Analysis of Conflict, Harvard University Press, (1991).
- [3] DÉMUTH, Andrej. 2013. Teória hier a problém rozhodovania, Filozofická fakulta Trnavskej univerzity v Trnave, 978-80-8082-580-5.
<http://ff.truni.sk/sites/default/files/publikacie/demuth_teoria_hier_a_problem_rozhodovania.pdf>."
- [4] POUNDSTONE, William. 2021. John von Neumann : American mathematician, Britannica, <<https://www.britannica.com/biography/John-von-Neumann>>.
- [5] HEMŽAL, Bohdan. 2021. späť na Matematika a fyzika Teória hier „začala“ pokrom a dilemou dvoch väzňov, science.dennikn.sk, <<http://science.dennikn.sk/clanky-a-rozhovory/neziva-priroda/matematika-a-fyzika/5769-teoria-hier-zacala-pokrom-a-dilemou-dvoch-vaeznov>>.
- [6] NOWAK, Martin A. .2015. nature.com: <<https://www.nature.com/articles/522420a>>.
- [7] MÄYRÄ, F.: An Introduction to Game Studies. Games in Culture. London: SAGE 2008, 32 — 51.
- [8] DIXIT, A., K., NALEBUF, B., J.: Thinking Strategically: The Competitive Edge in Business, Politics, and Everyday Life. New York: W. W. Norton & Company 1993, 1 – 6.
- [9] MAŇAS, Miroslav. Teorie her a optimální rozhodování. Praha : SNTL, 1974. IČ 04-012- 74.
- [10] PEŠKO, Štefan. 2019. Základné pojmy teórie hier, <http://frca-tel.fri.utc.sk/~pesko/TH/sTeoriaHier_1.pdf>.
- [11] Andrej Blaho. Programovanie v pythone, 3.vydanie. <<http://input.sk/pdf/python.pdf> >. 2018.
- [12] Python official. Comparing python to other languages.
<<https://www.python.org/doc/essays/comparisons/>> .
- [13] POUNDSTONE, W.: Prisoner's Dilemma. New York: Anchor Books, 1992, 101 – 119
- [14] KOPČOVÁ, Ivica. Statické hry a Nashova rovnováha,
<https://is.muni.cz/el/1431/jaro2012/MF006/um/Strategicke_hry_a_NE_Kopcokova.pdf>.

- [15] PEŠKO, Štefan. 2018. Maticové hry, <http://frcatel.fri.utc.sk/users/pesko/TH/sTeoriaHier_3.pdf>.
- [16] ČIČKOVÁ, Zuzana. Hry v rozvinutom tvare : prednáška,
- [17] DESKOVÉHRY.INFO. Nim, [webová stránka]. <<http://www.deskovehry.info/pravidla/nim.htm>>.
- [18] DESKOVÉHRY.INFO. Bulo, [webová stránka]. <<http://www.deskovehry.info/pravidla/bulo.htm>>.
- [19] CECHLÁROVÁ, Katarína. Maticové Hry, [webová stránka]. <<https://umv.science.upjs.sk/cechlarova/TH/Tyzden04/TH4.pdf>>.
- [20] SZYNDLAR, MARTA. The Best Python IDEs and Code Editors (According to Our Developers and the Python Community), STX Next, <<https://www.stxnext.com/blog/best-python-ides-code-editors/>>.
- [21] JETBRAINS. Corporate Overview, STX Next, STX Next, <https://resources.jetbrains.com/storage/products/jetbrains/docs/jetbrains_corporate_overview.pdf>.
- [22] "MASCHLER, MICHAEL – SOLAN, EILON , ZAMIR, SHMUEL . 2013. Game Theory, Cambridge university press, 2013. 978-1-107-00548-8."
- [23] "GONZÁLEZ-DÍAZ, Julio – GARCÍA-JURADO, Ignacio , FIESTRAS-JANEIRO, M. Gloria . 2010. An Introductory Course on Mathematical Game Theory, 2010. 978-0-8218-5151-7."

Príloha A: Obsah elektronickej prílohy

V elektronickej prílohe priloženej k záverečnej práci sa nachádzajú zdrojové súbory k programu spolu s obrázkami použitými v programe. Zdrojový kód je taktiež zverejnený na stránke https://zdarilek2.github.io/inzynierska_praca/

Príloha B: Používateľská príručka

V tejto časti uvádzame používateľskú príručku ku vytvoreným hrám. Na spustenie programu odporúčame Python 3.9 alebo vyššiu verziu. Je to z dôvodu, že programy boli vyvíjaný práve na verzii 3.9.

V prípade, že chceme spustiť hru pirát a Sicíľčan, tak je potrebné otvoriť priečinok s názvom `pirate_Sicilian` a spustiť súbor `main.py`.

Ak chceme otvoriť hru Bulo, tak je potrebné otvoriť priečinok s názvom `Bulo` a v ňom súbor `start_menu.py`. Po otvorení sa priamo spustí počiatočná obrazovka hry.