

**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: 20.04.2022

.....

(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 53.

Inžinierska práca sa zameriava na softvérové riešenie vybraných typov hier a zároveň rieši analýzu zvolených hier. K týmto bodom je spracovaná teória, z ktorej vychádza samotná analýza a softvérové riešenie. Softvérové riešenie je zamerané na tvorbu funkčnej hry a taktiež na zobrazenie analýzy hry. V prvej kapitole je opísaný úvod k práci, opísanie problematiky a teórie. V druhej kapitole sa nachádza cieľ inžinierskej práce. Tretia kapitola sa venuje analýze. V nej rozvádzame dôležité body v tejto oblasti pre jednotlivé nami zvolené hry. V poslednej kapitole sa nachádza postup pri tvorbe softvérového riešenie spolu s ukázkami dôležitých funkcií v kóde.

### **Kľúčové slová:**

Bulo, hra pirát a Sicíľčan, analýza hry



## **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 53.

Engineering thesis focuses on software selected types of games and also solves the analysis of selected games. A theory is elaborated on this point, from which only the analysis and software solution are based. The software solution is focused on creating a functional game and also on displaying game analysis. The first chapter describes an introduction to the work, a description of the issue and theory. In the second chapter is the goal of Engineering thesis. The third chapter deals with analysis. In it, we explain the important body in this area for the individual games we have chosen. The last chapter contains the procedure for creating a software solution, along with examples of important functions in the code.

### **Keywords:**

Bulo, game pirate and Sicilian, game analysis





# Obsah

## Úvod 2

<b>1</b>	<b>Súčasný stav problematiky .....</b>	<b>4</b>
1.1	Teória hier a jej definície .....	4
1.2	Vznik Teórie hier .....	4
1.3	John von Neumann.....	5
1.4	John Forbes Nash .....	6
1.4.1	Nashovo ekvilibrium .....	6
1.5	Koncept a základné pojmy v Teórii hier .....	7
1.6	Konfliktné a nekonfliktné situácie .....	10
1.6.1	Hlavné typy konfliktných situácií.....	10
1.7	Typy hier .....	12
1.7.1	Hry v normálnom tvare.....	12
1.7.2	Maticové hry.....	13
1.7.3	Väznova dilema .....	13
1.7.4	Tabuľka možných riešení väznovej dilemy.....	14
1.7.5	Hry v rozvinutom tvare.....	15
1.7.6	Strom hry .....	16
1.7.7	Hra NIM .....	16
1.7.8	Strom hry NIM .....	17
1.7.9	Hra Buló .....	17
1.7.10	Hra piráta a Sicíľčana.....	19
<b>2</b>	<b>Cieľ práce .....</b>	<b>20</b>
<b>3</b>	<b>Metodika práce a metódy skúmania .....</b>	<b>21</b>
3.1	Analýza softvérových možností riešenia hier .....	21
3.2	Analýza hry pirát a Sicíľčan .....	21
3.2.1	Maticová hra .....	21
3.3	Analýza hry Buló .....	23
3.3.1	Strom hry Buló .....	23
3.3.2	Spätná indukcia.....	25
<b>4</b>	<b>Výsledky práce.....</b>	<b>28</b>
4.1	Technológie.....	28
4.1.1	Hp pavilion 14-dv0003nc .....	28
4.1.2	Python.....	28



4.1.3	Popis prostredia .....	29
4.1.4	PyCharm .....	31
4.2	Softvérové riešenie hry pirát a Sicíľčan.....	32
4.3	Softvérové riešenie hry Bulo.....	35
4.3.1	Analýza.....	41
4.3.2	Kód .....	44
<b>Záver</b>	<b>.....</b>	<b>47</b>
<b>Zoznam obrázkov</b>	<b>.....</b>	<b>48</b>
<b>Zoznam tabuliek</b>	<b>.....</b>	<b>49</b>
<b>Zoznam použitej literatúry</b>	<b>.....</b>	<b>50</b>
<b>Príloha A: Obsah elektronickej prílohy.....</b>		<b>52</b>
<b>Príloha B: Používateľská príručka .....</b>		<b>53</b>





# Úvod

Pod pojmom teórie hier si mnohí predstavia kategóriu počítačových hier. Avšak táto téma skrýva oveľa viac ako sa na prvý pohľad môže zdať. Ide o pomerne mladú vedeckú disciplínu ale zároveň jej náznaky boli medzi ľudstvom už 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 ľudí z čisto matematického prostredia. Téma teórie hier zasahuje každého z nás. Nachádza sa v mnohých momentoch nášho života bez toho, že by sme o tom vedeli. Jedná sa napríklad o politické vyjednávanie, policajné výsluchy, ale z úplne bežného života to je aj to, kam ísť na večeru s priateľmi. Rieši teda rozhodovacie problémy vo svete. Ide v princípe o matematický pohľad v oblasti rozhodovania a riešenia rozhodovacích problémov. Oblasť Teórie hier je veľmi obsiahla téma a jej rozvoj, či potenciálne využitie, má obrovský potenciál.

V tejto práci sa dotkneme 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, vďaka čomu zhrnieme všetky podstatné poznatky potrebné k hlbšiemu poznaniu a pochopeniu témy. Okrem definícií a základným údajom sa budeme venovať aj vysvetľovaniu jednotlivých pojmov pre priblíženie témy každému.

Analýzou riešení jednotlivých úloh si otvoríme cestu pre softvérové riešenie jednotlivých úloh a hier. Samotné softvérové riešenie prinesie lepšie pochopenie a zažitie si Teórie hier v praxi. Kľúčovými časťami tejto práce sú analýza a softvérové riešenie. Analýza ponúkne pohľad do teoretickej hĺbky zvolených hier a implementácia ukáže výsledok vo forme kódu.

V diplomovej práci sa konkrétne venujeme hrám, ktoré nie sú príliš známe vo verejnosti ale prinášajú so sebou možnosti bližšieho pochopenia. Zároveň touto prácou máme možnosť ukázať tieto hry ostatným a spopularizovať ich.

Význam tejto práce je teda aj v oblasti školstva, kde môže priblížiť túto tému širšiemu množstvu ľudí, pre ľudí zaoberajúcimi sa Teóriou hier ale zároveň je prínosom aj pre svet informačných technológií svojim spracovaním v programovacom jazyku Python.

# 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 kedy americký matematik John von Neumann dokázal takzvanú základnú vetu teórie hier a to vetu o min-maxe. [1] Následne 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]

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 algoritmus krokov, ktorý ich k tomu privedie je odlišný. [3] Prvé filozoficko-teoretické analyzovanie teórie hier však nachádzame podľa Dona Rossa v dialógu Laches a Symposion 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 je stále veľmi dôležitá.

### 1.3 John von Neumann

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 *klaman* 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. 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.

#### 1.4 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] Nashovmu ekvilibriu sa detailnejšie budeme venovať v kapitole 1.4.1. 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]

##### 1.4.1 Nashovo ekvilibrium

Optimálnu stratégiu hráčov v konfliktnej situácii môžeme nájsť pomocou Nashovho 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 odchyľ 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  $s_i \in S$  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 prinesie nižší zisk ako niektorý iný výsledok hry. Zároveň, Nashova rovnováha nemusí byť v hre jedinečná. Môže ich byť aj viac. [14]

## 1.5 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 cieľenú 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 existenciu hráčov, zákonitosti 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. Cieľené 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 stratégiách. [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 spätosť 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** – Na začiatok je dobré definovať pojem 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 pozriem na ich hodnotenie, tak zistím, 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 časťovým. [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 neurčitou. Príkladom je napríklad výber neznámeho jedla v reštaurácii. Pri rozhodovaní s rizikom a aj s neurčitou si vyžadujú od aktéra použitie rôznych stratégií, ktoré by mali minimalizovať nežiaduce 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]

## 1.6 Konfliktné a nekonfliktné situácie

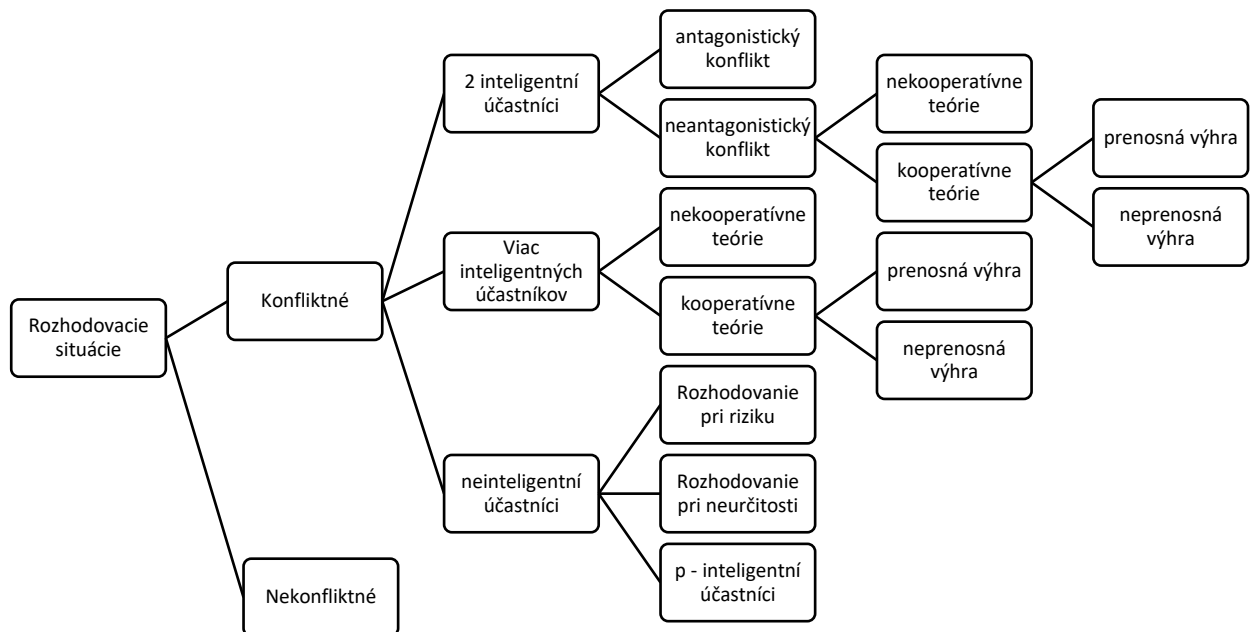
V tejto časti sa pozrieme na konfliktné a nekonfliktné situácie v Teórii hier. Následne ich detailne zanalyzujeme v nasledujúcej časti. Teória hier sa zaoberá riešením nekonfliktných a konfliktných situácií. Nekonfliktnou situáciou nazývame takú, ktorej sa zúčastňuje len jeden účastník. Zároveň pozná 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]

### 1.6.1 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 najjednoduchší typ konfliktu, kde definícia optimálneho rozhodnutia 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 teórie. Kooperatívna teória je situácia, keď sa uzavrie dohoda s druhým účastníkom konfliktu pred rozhodnutím, akú voľbu spraví. Ak to možné nie je, hovoríme o nekooperatívnej teórii. V kooperatívnej teórii je dôležitý prípad, keď sa hráči môžu, napríklad zmluvne dohodnúť, že účastník, ktorému výsledok prináša podstatne lepší výsledok, dá druhému účastníkovi nejakú náhradu respektíve kompenzáciu za to, že mu k lepšiemu výsledku dopomohol vhodnou voľbou rozhodnutí. Vtedy hovoríme o prípade prenosných výhier. Ak to nie je možné, tak hovoríme o neprenosnej výhre. Rozdelenie na

kooperatívne a nekooperatívne teórie je rovnako aj vtedy, ak je počet účastníkov konfliktu väčší ako dva. 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.



Obr. 1.1: Klasifikácia hier [9]

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]

## 1.7 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].

### 1.7.1 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] Konfliktne rozhodovacie situácie vznikajú napríklad pri hraní hier ako karty, šach a podobne. Sú to hry z kategórie salónových. 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]

Príklad. Hra dvoch hráčov v normálnom tvare.

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 ich výberov, zaplatí hráč 2, hráčovi 1 čiastku rovnú súčtu nimi zvolených čísel. Ak sú obaja hráči inteligentní, bude najlepšie ak hráč 2 zvolí číslo 0 a hráč 1 číslo 1. [9]

### 1.7.2 Maticové hry

Jedná sa o konečnú hru v normálnom tvare dvoch hráčov s nulovým súčtom. Matica obsahuje platby hráčov, pri použití jednotlivých stratégií. [23]

#### Definícia

Konečná hra s nulovým súčtom, ktorá je určená reálnou maticou  $A = a_{ij}$ , kde  $i \in X, j \in Y$ , je nazývaná ako maticová hra. Maticu  $A$  nazývame matica hry. [15] Nasledovné definície vychádzajú zo zdroja [15].

$$H_A = (Q = \{1, 2\}, X = (1, \dots, m), Y = \{1, \dots, n\}, M(i, j) = a_{ij})$$

#### Definícia

Nech  $A$  je matica hry. Stratégiu  $(i^*, j^*) \in X \times Y$  maticovej hry  $H_A$  nazveme rovnovážnou, ak pre  $\forall i \in X$  a  $\forall j \in Y$  je

$$a_{ij^*} \leq a_{i^*j^*} \leq a_{i^*j}$$

Rovnovážna stratégia  $(i^*, j^*)$  musí vyhovovať vzťahom ktoré umožňujú efektívne hľadanie rovnovážnej stratégie.

$$a_{i^*j^*} = \max_{i \in X} a_{ij^*}; i \in X$$

$$a_{i^*j^*} = \min_{j \in Y} a_{i^*j}; j \in Y$$

Maticovým hrám sa budeme venovať v časti analýz softvérových možností riešenia hier.

### 1.7.3 Väžňova dilemma

Väžňova dilemma označuje vzorový príklad hry s neurčitou a s nenulovou sumou 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 väžňovej 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 prírodu, 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]

#### *1.7.4 Tabuľka možných riešení väzňovej dilemy*

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 spolupáchatelovi 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 spolupáchatľa, 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 spolupáchatľa, 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 spolupáchatľa neudá a odmietne spolupracovať, výsledkom bude beztrestnosť prvého hráča a jeho sloboda.



Prvý hráč teda pri nespolupráci so spolupáchatelom môže získať 2 roky väzenia alebo slobodu. [3]

	<b>Spolupráca</b>	<b>Zrada</b>
<b>Spolupráca</b>	zisk / zisk	veľká strata / veľký zisk
<b>Zrada</b>	veľký zisk / veľká strata	strata / strata

Tabuľka 1.1: Stratégie hráčov a ich zisky

#### 1.7.5 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, sú to mnohé salónové hry. Napríklad šach, dáma a mnohé ďalšie. V teórii hier ich najčastejšie znázorňujeme špeciálnym grafom. Tento graf má tvar stromu. Ide teda o strom hry. O ňom si povieme v nasledujúcej časti. [16]

Hru 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

### 1.7.6 Strom hry

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

### 1.7.7 Hra NIM

Na tejto hre si ukážeme strom hry a vysvetlíme si jej pravidlá. Táto hra bude súvisieť aj so samotnými hrami, ktoré budú softvérovo riešené v tejto práci. Je to jednoduchá hra, ktorá dala podnet pre vznik ďalších hier ako Bulo alebo Jeek. Jej názov je pravdepodobne odvodený z nemeckého slova nehmen, čo v preklade znamená brať. [17] Jedná sa o hru NIM a môže ju hrať ľubovoľný počet hráčov. V našom príklade budeme predpokladať, že sa jedná o dvoch hráčov. Na začiatku predpokladajme, že máme dve kôpky a v každej z nich sú dva kamienky. Tento výber sme spravili 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 rozdělíme podľa špecifikácie. [16]

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

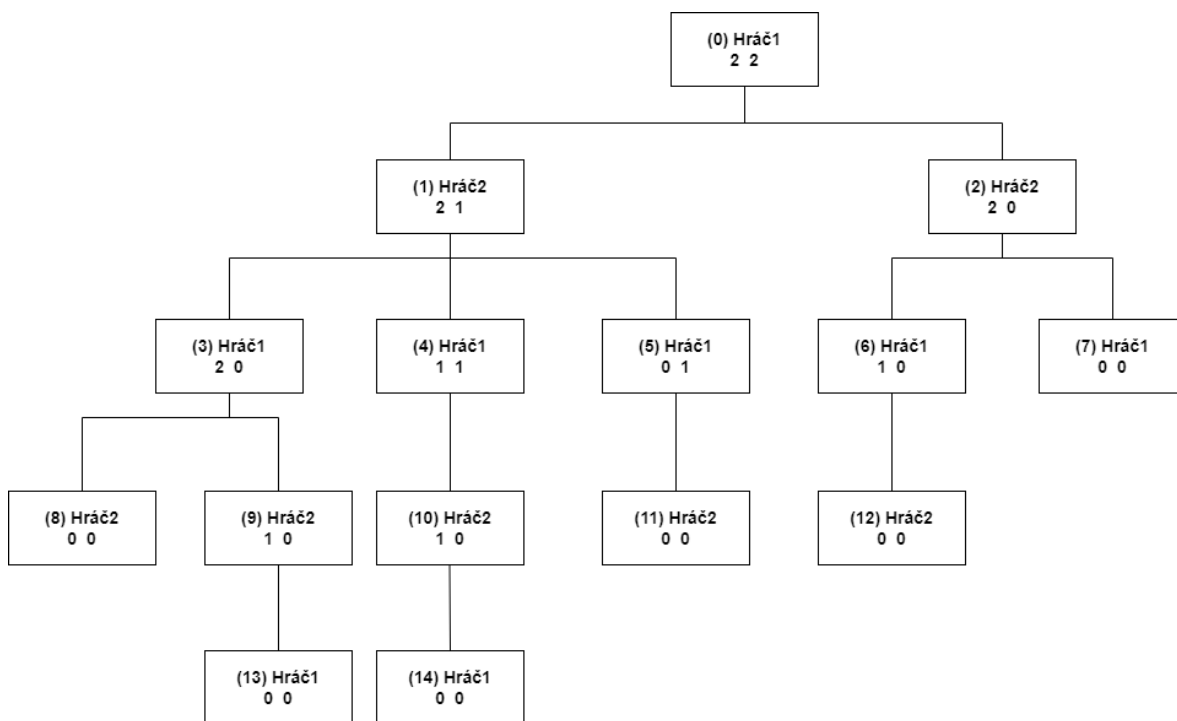
**Východisková situácia:** je taká, že pred hráčmi sú všetky kamene rozdelené do kôpok.

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

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

### 1.7.8 Strom hry NIM

Definíciu stromu hry už máme definovanú. 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 Hráč1, 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.



Obr. 1.2: Strom hry Nim [16]

### 1.7.9 Hra Buló

V tejto časti sa pozrieme na hru s názvom Buló. Túto hru budeme neskôr analyzovať a pozrieme sa aj na jej softvérové riešenie. Hru môžeme zaradiť do kategórie hier

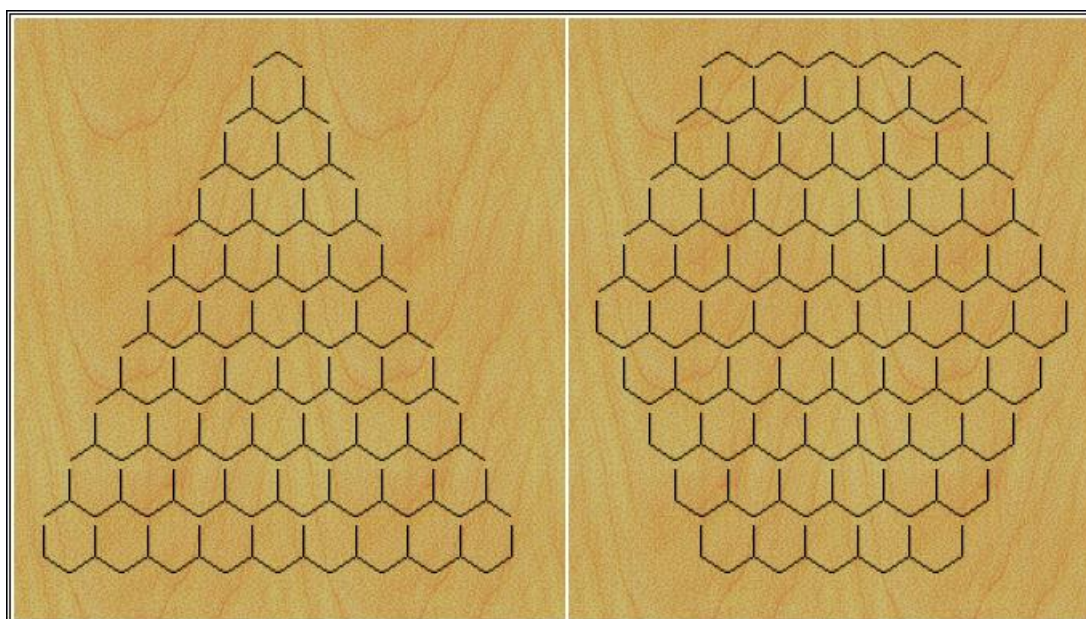
v rozvinutom tvare. Bulo je vylepšená obdoba hry NIM. Taktiež je známa aj pod názvom Tac Tix. Jej autorom je Dán Piet Hein. Pravidlá hry sú nasledovné: Počet hráčov, pre ktorých je táto hra spravená sú dvaja. Hracia plocha obsahuje kamene. Ako základ sa používa hracia doska o veľkosti 4x4. Čiže obsahuje 16 kameňov. Avšak plocha môže byť aj väčšia, s väčším počtom kameňov a taktiež s rozličným tvarom. [18]

**Cieľ hry** je, rovnako ako pri NIM, prinútiť súpera, aby zobral posledný kameň z hracej plochy. [18]

**Východiskovou situáciou** je, že všetky polia na hracej ploche sú obsadené kameňmi. [18]

**Hra** prebieha tak, že sa hráči striedajú v ťahoch a v každom ťahu si hráč vyberie ľubovoľný počet kameňov, ale minimálne jeden. Pri výbere kameňov v 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. [18]

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



Obr. 1.3 Rozloženia hry Bulo [18]

### 1.7.10 Hra piráta a Sicíľčana

Túto hru môžeme zaradiť do kategórie hier v normálnom tvare. Jedná sa o hru, kde hráči zahrajú svoje stratégie a hra končí. Hra je určená pre 2 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 kde nie je jed. [19]

**Východisková situácia:** Na stole sú dva poháre z ktorých si môže vybrať pirát a aj Sicíľčan. Jeden je pirátov a druhý Sicíľčana. Predpokladom hry je to, že hráči svoje stratégie volia súčasne. [19]

**Hra:** Pirát dá do jedného z pohárov jed. Buď teda do svojho alebo do pohára Sicíľčana. Súper nevie o tom 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. [19]

**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.

## 2 Cieľ práce

Cieľom tejto inžinierskej práce je bližšie sa pozrieť na analýzu v Teórii hier a rozobrať si jednotlivé možnosti analýzy. Analýzu je najlepšie ukázať na konkrétnych príkladoch, preto je našim cieľom, aby sa v práci nachádzali dve softvérové riešenia, na ktoré bude aplikovaná analýza, ako aj jej detailný popis. Čiže zámerom je naprogramovať dve hry v oblasti Teórie hier, ich samotná analýza, ktorá môže pomôcť hráčovi, ktorý túto hru hrá alebo sa môže obohatiť o vedomosti z analýzy v Teórii hier. Malo by to priniesť aj obohatenie sveta počítačových hier o dve menej známe hry a zároveň otvoriť možnosť aj pre možné použitie týchto hier, ako aj samotnej práce na vzdelávacie účely.

Úlohou spracovanej analýzy by malo byť poskytnúť matematický pohľad na túto oblasť a jej vysvetlenie čitateľovi. Funkciou programovanej časti je poskytnúť vizuálny zážitok pre používateľa s možnosťou vyskúšania jednotlivých hier. Zároveň dáva možnosť ľuďom zaujímavých sa o informačné technológie, respektíve programovanie, nazrieť do kódu a možnosti riešenia podobných alebo aj rovnakých hier.

V jednoduchosti teda ide o analýzu dvoch konkrétnych hier, ktoré si používateľ môže zahrať a pozrieť si na konci hry samotnú analýzu hry.

## 3 Metodika práce a metódy skúmania

### 3.1 Analýza softvérových možností riešenia hier

Slovo analýza podľa definície výkladového slovníku znamená rozkladanie, rozčlenenie predmetu alebo javu na jeho jednotlivé časti. Tieto definície dokonale opisujú to, čo budeme vykonávať v našom príklade. Pôjde o rozbor hry, jej 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 inej zahraničnej stratégie. Táto analýza samozrejme neprináša vždy víťaznú stratégiu pre hráča. V analýze sa javí ako víťazná, ale je závislá od zvolených stratégií druhým hráčom. Avšak nám pomôže pochopiť detaily hry, jej možné realizácie s 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.

### 3.2 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 1.7.10.

#### 3.2.1 Maticová hra

Hra Pirát a Sicíľčan je maticová hra alebo aj hra v normálnom tvare. Keďže sa jedná o maticovú hru, tak sa pozrieme 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. Ak hráč1 zahrá  $\delta_i$  a hráč2  $\pi_j$ , tak hráč1 dostane  $A_{ij}$  a hráč2 zaplatí  $A_{ij}$ . [19] Môžeme si to ukázať aj na konkrétnom príklade.

Hráč1, teda pirát, zahrá napríklad stratégiu  $\delta_1$  a hráč2, ktorý je Sicíľčan, zahrá  $\pi_2$ .  $\delta_1$  môže

predstavovať napríklad stratégiu, že pirát dá jed do pohára Sicíľčana.  $\pi_2$  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 hráč1 si vybral stratégiu s indexom 1 a hráč2 s indexom 2, tak hľadáme prvok matice A s indexami 1 a 2. Čiže jedná sa o prvok  $A_{1,2}$ , ktorý predstavuje hodnotu -1. Podľa definície platieb môžeme povedať, že hráč1 po zahratí stratégie  $\delta_1$  a hráč2  $\pi_2$ , dostane -1 a hráč2 zaplatí -1. Hráč1 dostane -1, čo však predstavuje platbu hráčovi2. Hráč1 má podľa definície zaplatiť -1, čo ale znamená, keďže sa jedná o záporné číslo, že hráč2 dostane +1. [19]

Keď si rozoberieme možné použitie stratégie a vplyvy informácií, tak zistíme, že veľa závisí na samotných informáciách. Môžeme sa pozrieť, ako by vyzerali stratégie hráčov, ak by bola niektorá informácia známa pre jedného z hráčov. Ak Sicíľčan nevie, že ho chce pirát otráviť, tak pirát použije stratégiu  $\delta_2$ . Sicíľčan použije stratégiu  $\pi_2$ . Výsledok je, že vyhrá pirát. Ak Sicíľčan vie, že ho chce pirát otráviť, ale pirát nevie, že Sicíľčan to vie, tak pirát použije stratégiu  $\delta_2$ . Sicíľčan použije stratégiu  $\pi_1$ . Výsledok je, že vyhrá Sicíľčan. Sicíľčan vie, že pirát ho chce otráviť, ale aj Pirát vie, že Sicíľčan to vie. Tu vzniká otázka, že do ktorého pohára dá Pirát jed? Tu vidíme, že nastáva problém, že ak nevieme žiadne informácie, ktoré by nám pomohli analyzovať možné ťahy súpera, tak sme odkázaný iba na našu 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  $\delta_i, \pi_j$  platí  $e_1(\delta_i, \pi_j) + e_2(\delta_i, \pi_j) = 0$ . [19]

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

Obr. 3.1 Maticová reprezentácia hry Pirát a sicíľčan



### 3.3 Analýza hry Bulo

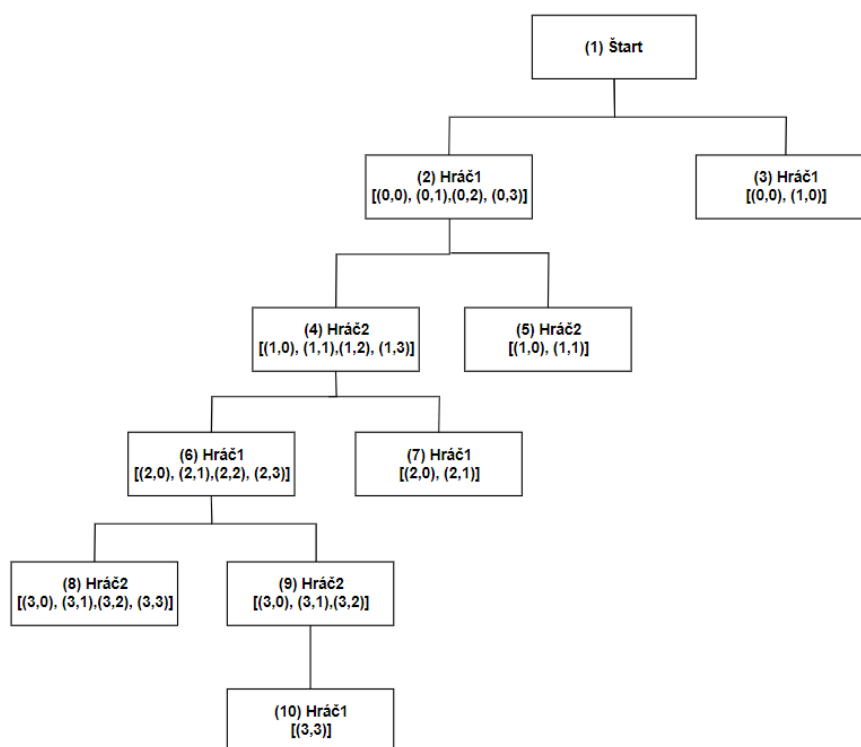
Pravidlá hry Bulo sme si vysvetlili v predchádzajúcich častiach. Teraz sa pozrieme na analýzu tejto hry. Pri skúmaní hry použijeme analýzu pomocou stromu rovnako ako pri hre NIM.

#### 3.3.1 Strom hry Bulo

Analýzu pomocou stromu hry sme ukázali v predchádzajúcej časti na hre NIM. Teraz si detailne rozeanalyzujeme analýzu hry Bulo a 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 štyri krát štyri. Je to z dôvodu, že je to najmenšie rozloženie hracieho poľa. V strome sa budú nachádzať informácie, že ktorý hráč aktuálne skončil ťah. Ďalej tam bude informácia o vybraných kameňoch. Vybraný kameň bude zapísaný v indexoch pozície v matici. Keďže je matica dvojrozmerná, tak bude obsahovať informáciu, 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 sa jedná o riadok nula. Jediný koreň stromu bude označený ako štart a to z dôvodu, že v počiatočnej fáze nie je odstránený žiaden kameň. V kompletnom strome hry môžeme vidieť všetky možné postupy. Takže vieme nájsť konkrétny bod a pozrieť sa, ako inak by sa dalo postupovať, tak aby napríklad druhý hráč vyhral. Analýzu si ukážeme na konkrétnom ukážkovom príklade. Na ňom si taktiež ukážeme, čo vieme dosiahnuť a zistiť pomocou analýzy, a ako ju potencionálne využijeme v kóde. Príklad bude poskladaný z krokov jednotlivých hráčov.

Začíname koreňom stromu kde máme napísané slovo štart. Po spustení hry má prvý hráč k dispozícii všetky možné kombinácie kameňov s obmedzením, že musí dodržať pravidlá. Na obrázku 3.2 sa nachádza podhra. V strome tejto hry nie sú vypísané všetky možnosti a vetvenie by bolo samozrejme mohutnejšie. Ale ukazuje to potencionálne 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 to, že existuje určite aj nejaká iná cesta resp. cesty.



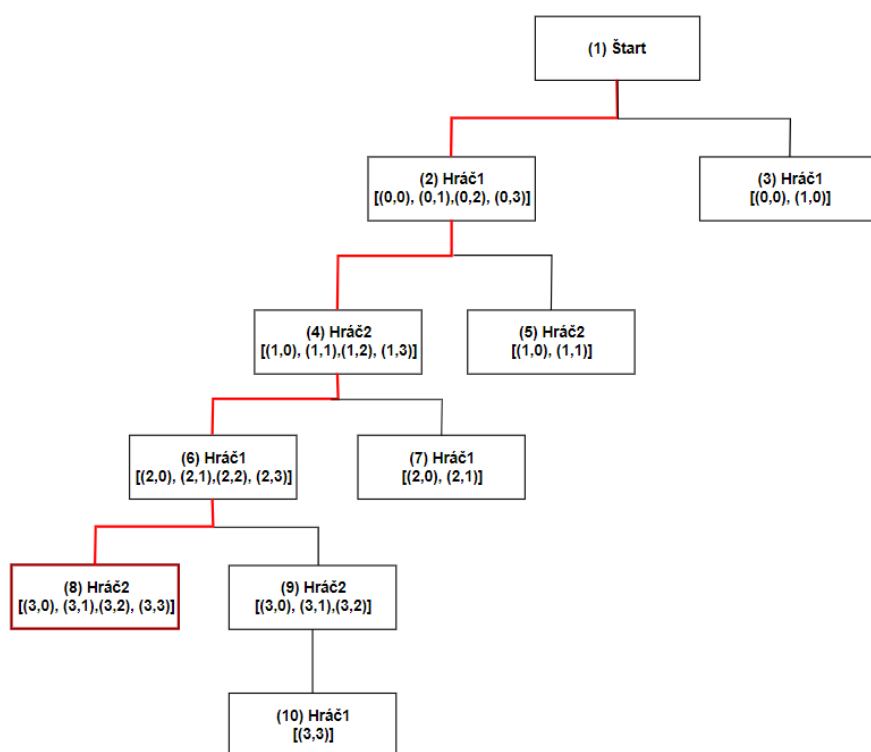
Obr. 3.2 Strom hry Buló

Hráč1, ktorý predstavuje prvého hráča, vykonal prvý ťah. V tomto konkrétnom príklade si vybral možnosť zobrať všetky kamene z prvého riadku. Jedná sa 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. Kvôli veľkému množstvu možností v tejto vetve nebudeme pokračovať. Po odohratí ťahu je na rade Hráč2. Hráč2 je hráčom, ktorý je druhý v poradí. Druhý hráč si vyberal zo všetkých možností okrem tých, ktoré obsahujú v sebe už vybrané kamene. V našom prípade celý prvý riadok. Hráč2 sa rozhodol zahrať podobnú taktiku ako hráč1 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 hráč1 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 Hráč1 zvolil stratégiu výberu štyroch kameňov v treťom riadku. Jedná sa o uzol s číslom 6 a pozíciami kameňov  $[(2,0), (2,1), (2,2), (2,3)]$ .

Ďalej nasledoval hráč2. V hre ostali štyri kamene v poslednom riadku. Hráč2 sa rozhodol použiť rovnakú stratégiu ako doteraz a to, že vyberie celý riadok. V tomto bode končí hra lebo sa na ploche nenachádzajú žiadne kamene a hráč2 prehral. Táto vetva v strome hry je najkratšia možná. Hráč teda končí v uzle 8 a hodnoty pozície kameňov sú  $[(3,0), (3,1), (3,2), (3,3)]$ .



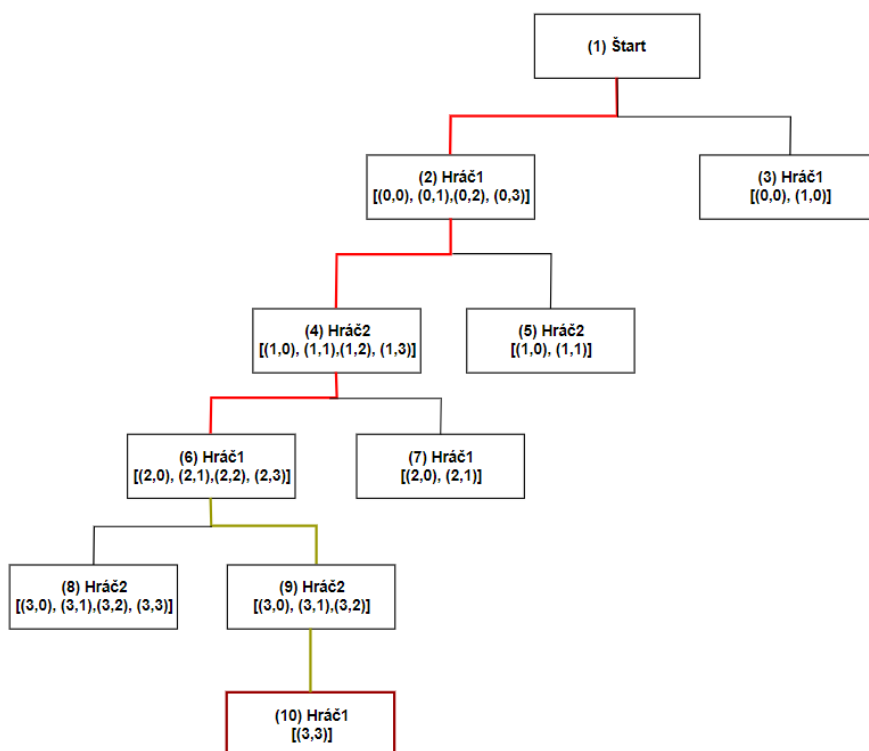
Obr. 3.3 Strom hry Buló s vyznačenou vetvou

### 3.3.2 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 určuje indukcia 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 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 prehrál hráč2, tak zanalyzujeme možnosti a nájdeme cestu v strome, ktorá by odpovedala víťazstvu hráča2. Vrátime sa najskôr o jeden ťah späť. Teraz sme sa dostali do uzla s číslom 6 a udialo sa to, že hráč1 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. Avšak, keďže vieme, že hráč ktorý zoberie posledný kameň prehrá, tak našou najlepšou možnosťou je dosiahnuť taký stav, kde hráč2 dokončí ťah tak, že na hracej ploche ostane len 1 kameň. V tom momente vie, že hráč1 ho musí zobrať a tým pádom hru prehrál. Vieme, že hráčovi2 ostali štyri kamene a všetky susedia. Hráč2 teda môže zobrať ľubovoľné množstvo kameňov.



Obr. 3.4 Strom hry Bulo s novou vetvou po použití spätnej indukcie

Podľa vyššie napísanej analýzy vieme, že potrebujeme ponechať na konci ťahu jeden kameň. A v tomto prípade je to možné. Hráč2 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 uzlu s číslom 9. Touto stratégiou donútil hráča1 aby zahral jedinou stratégiu, ktorá mu ostala a to vziať posledný kameň. Na obrázku č.3.4 sa jedná o uzol 10. Hráč2 by sa v takomto momente stal víťazom a hráč1 by danú hru prehral.

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

## 4 Výsledky práce

### 4.1 Technológie

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

#### 4.1.1 *Hp pavilion 14-dv0003nc*

Keďže sa jedná o riešenie, ktoré bude bežať na počítači, tak je potrebné si predstaviť aj prístroj, na ktorom je dané riešenie vytvárané a testované. Jedná sa teda o notebook značky Hp. Ako jeho parametre by sme uviedli procesor od firmy Intel. Ide o procesor Intel Core i5 jedenástej generácie. Jeho rýchlosť frekvencie je 2.4 GHz. Ďalšími parametrami sú RAM pamäť. Toto zariadenie má 16 GB operačnej pamäte. Čo sa týka grafiky, tak obsahuje grafiku Intel iris xe graphics. Jedná sa o vstavanú grafiku. Posledným dôležitým parametrom, ktorý je vhodné spomenúť je operačný systém, na ktorom sú dané riešenia vytvárané. Windows 11 je najnovšia verzia tohto operačného systému na trhu od firmy Microsoft.

#### 4.1.2 *Python*

Python je moderný a veľmi populárny programovací jazyk. Autorom je Guido van Rossum. Používajú ho firmy ako napríklad Uber, PayPal a Google. Jedná sa o jazyk, ktorý je interpretér a je teda rozdielny od kompilačných jazykov ako sú napríklad Pascal, C a C++.[11] Zaujímavé je porovnanie veľmi 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. Ale kvôli tomu sa predlžuje čas spustenia. Môžeme si zobrať 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 hneď vykonať danú operáciu. Avšak výhodou Pythonu je rýchlosť vývoja aplikácií. Už len to, že nie je nutné definovať dátové typy uľahčuje a zrýchľuje proces vývoja.[12]

Hlavné vlastnosti jazyka Python sú:

- „veľmi jednoduchá a dobre čitateľná syntax a keďže Python je aj vysoko interaktívny, je vhodný aj pre vyučovanie programovania“ [11]
- „na rozdiel od staticky typovaných jazykov, pri ktorých je treba dopredu deklarovať typy všetkých dát, je Python dynamicky typovaný, čo znamená, že neexistujú žiadne deklarácie.“ [11]
- „Python obsahuje pokročilé črty moderných programovacích jazykov, napr. podpora práce s dátovými štruktúrami, objektovo-orientovaná tvorba softvéru...“ [11]
- „Je to univerzálny programovací jazyk, ktorý poskytuje prostriedky na tvorbu moderných aplikácií, takých ako analýza dát, spracovanie médií, sieťové aplikácie a pod.“ [11]
- Python má obrovskú komunitu programátorov a expertov.[11]

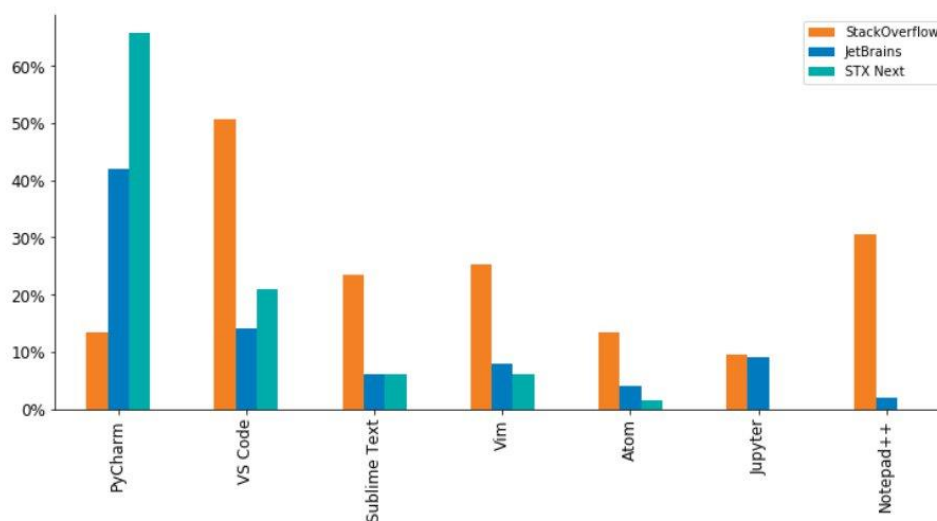
V tejto práci budem používať verziu pythonu 3.9.

#### 4.1.3 *Popis prostredia*

V tejto časti sa budeme venovať popisu prostredia, v ktorom boli vyvíjané riešenia úloh Teórie hier. Čo sa týka vývojových prostredí, tak pythonovské programy môžeme vyvíjať v rôznych prostrediach. Samotný python poskytuje vlastné vývojové prostredie, avšak je menej intuitívne ako rôzne iné. Jedná sa o vývojové prostredie s názvom IDLE. Existujú teda aj viaceré ďalšie, ktoré sú veľmi populárne a majú v sebe automaticky nainštalované funkcionality. Medzi nimi je 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 firma STX Next a je založený na názoroch Python vývojárov z tejto firmy. Ich počet presiahol 200. Ďalším parametrom tohto prieskumu je index popularity vývojového prostredia podľa stránky StackOverflow. Dáta z tohto prieskumu sú všeobecné a obsahujú názory na všetky vývojové prostredia, pre akýkoľvek programovací jazyk. A posledným zdrojom údajov je JetBrains. Pri dátach od JetBrains sa zohľadňujú iba programátori pracujúci v Pythone. [20]

Na grafe z obrázku 4.1 môžeme vidieť tri stĺpce. Oranžový obsahuje dáta zo stránky StackOverflow, modrý obsahuje dáta z JetBrains a posledný zelený stĺpec obsahuje dáta z firmy STX Next. Na základe týchto dát môžeme povedať, že PyCharm a Visual Studio Code sú pomerne používané u všetkých troch tipov ľudí z prieskumu. Dôvod prečo je rozdiel medzi Visual Studio Code a PyCharmom u užívateľov zo StackOverflow je, že Visual Studio Code je univerzálny aj pre iné jazyky a tým pádom ho využíva väčšie spektrum užívateľov ako niektoré iné vývojové prostredia. Avšak v dvoch skupinách ľudí, ktoré sú zamerané len na programovací jazyk Python, vyhráva v obľúbenosti PyCharm. Tento prieskum naznačuje to, že PyCharm je pomerne obľúbený a využívaný časťou programátorov, ktorí sa venujú Pythonu.



Obr. 4.1 Graf obľúbenosti jednotlivých vývojových prostredí [20]

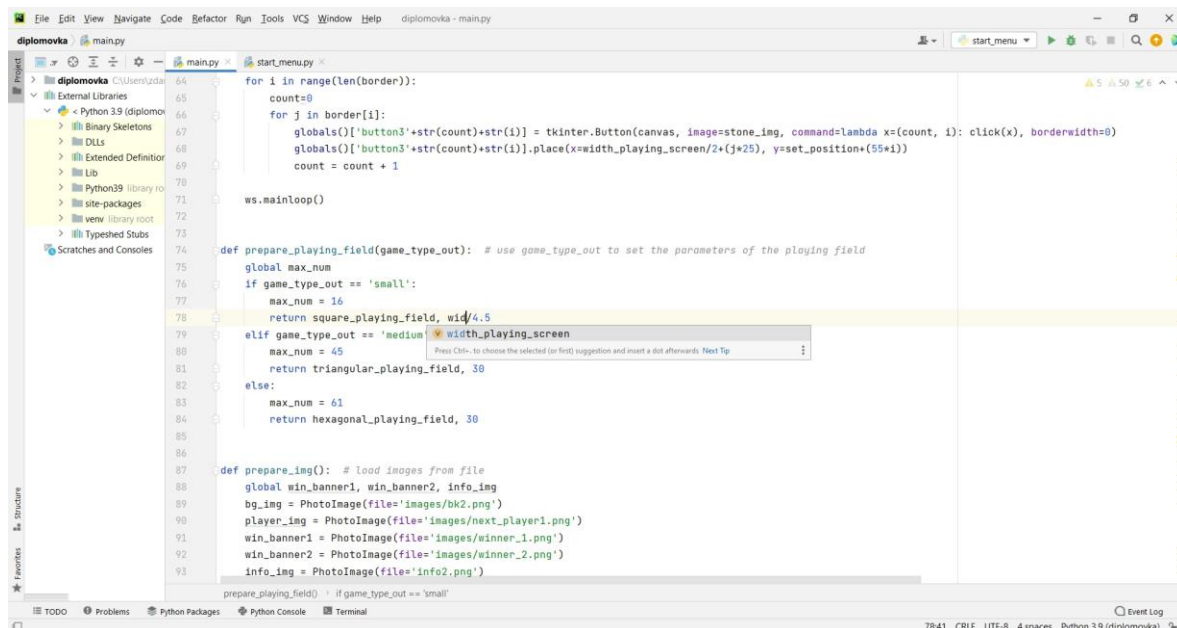


#### 4.1.4 PyCharm

PyCharm je vývojové prostredie spravené pre programovací jazyk Python. Je multiplatformové, čiže je poskytované pre Linux, Windows a aj macOS.[21] 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 prípade platenej verzie. Z ďalších funkcionalít sú to napríklad databázové nástroje, pomoc pri vývoji web stránky, vedecké nástroje a rôzne ďalšie. [20]



Obr. 4.2 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, keď jeho kód nespĺňa bežne zabehnuté štandardy písania kódu. Ďalšou prednosťou je dopĺňanie kódu. Táto funkcionálna zrychlí písanie kódu a zároveň poskytuje pomoc pri hľadaní jednotlivých možností v kóde.

## 4.2 Softvérové riešenie hry pirát a Sicíľčan

Prvým softvérovým riešením je maticová hra pirát a Sicíľčan. V tejto časti sa pozrieme na samotný 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 by bolo cieľom aby táto hra nebola uzatvorená len pre slovenských hráčov ale aby bola k dispozícii aj potencionálnym hráčom zo zahraničia.

Samotná hra je určená pre dvoch hráčov. A na začiatku prípravy tejto hry sme pôvodne zvažovali, že to bude hra pre dvoch ľudí. Avšak nastal problém pri momente ako dosiahnuť to, aby hráči spravili ťah naraz bez toho aby boli ovplyvnení druhým hráčom. Keďže sa jedná o hru na jednom počítači, tak je spravenie súčasného ťahu dvoch hráčov komplikáciou. Dve riešenia tohto menšieho problému boli buď spraviť hru po sieti alebo vytvoriť hru pre jedného hráča, a druhého hráča bude simulovať počítač. Hra po sieti by fungovala na princípe, že každý hráč by bol na svojom počítači a pomocou siete posielal informáciu 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 avšak, keď si to roznalýzujeme, tak pre takto krátku hru je určite jednoduchšie, ak stačí hru nainštalovať na jedno zariadenie a navyše je to možné hrať aj bez pripojenia na internet. Preto sme zvolili možnosť spraviť hru 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 súčasných ťahov dvoch hráčov vyriešil.

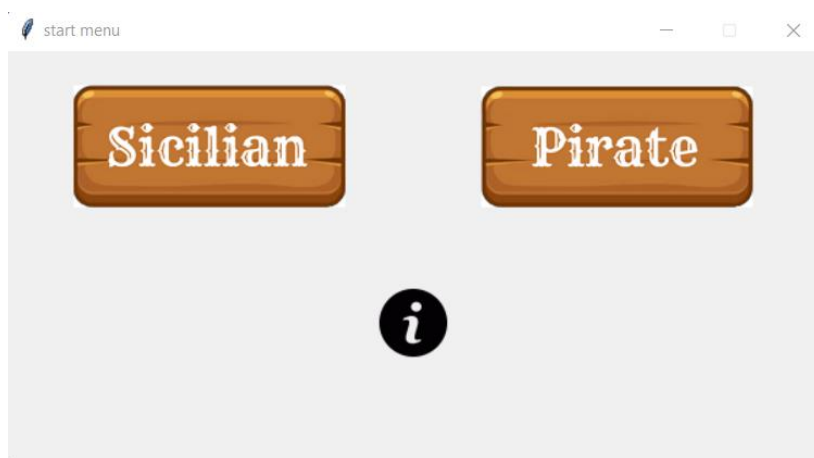
Ďalšou časťou prípravy tejto hry bol grafický dizajn aplikácie. Zvolili sme si tému podobnú krčmovému stolu v dobe pirátov. Dizajn bol tvorený pomocou aplikácie Canva a Adobe photoshop kvôli jednodotnosti a možnosti vlastnej úpravy obrázkov.

Hra sa začína úvodnou obrazovkou, kde má hráč k dispozícii tri tlačidlá. V prípade, že hráč nepozná túto hru, tak hráč 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 vybranie svojej role v hre, buď teda rolu 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. 4.3 Úvodná obrazovka hry pirát a Sicíľčan

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 informačný pre hráča, je popis v hornej časti obrazovky. Ten hovorí, čo má daný hráč spraviť. V našom prípade sa jedná 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 vykonávajú 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. 4.4 Obrazovka hry pirát a Sicíľčan, kde sa hráč rozhoduje, akú stratégiu použije

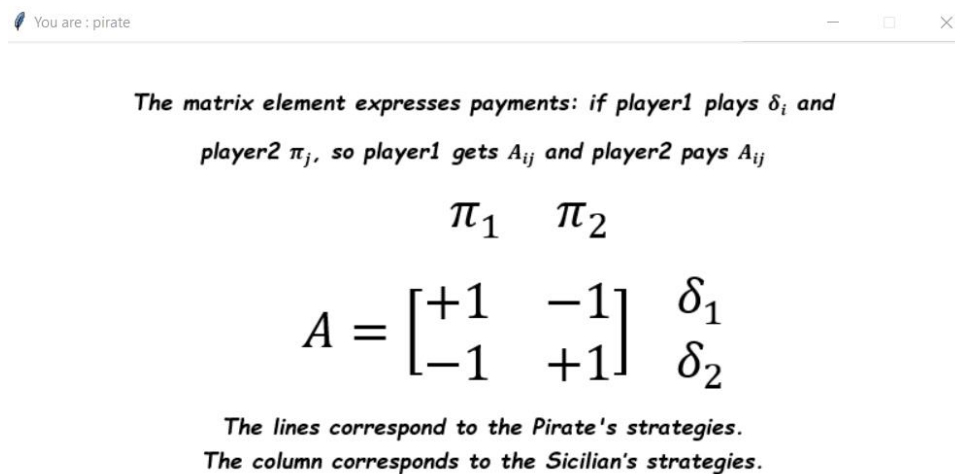
Výber počítača je realizovaný náhodným generátorom. Je to z dôvodu, lebo chceme aby hráči robili svoje rozhodnutia nezávisle od seba a ani jedna možnosť nie je nijakým spôsobom výhodnejšia. Preto výber počítača bude úplne náhodný. Pre túto potrebu využijeme modul `random`, ktorý v sebe obsahuje funkcie 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, 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. 4.5 Obrazovka hry pirát a Sicíľčan s analýzou hry

### 4.3 Softvérové riešenie hry Bulo

V tejto časti sa pozrieme na zvolené softvérové riešenie stolovej hry Bulo. Pravidlá hry sme definovali v časti 1.7.9 a aj 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. Jazyk hry sme zvolili angličtinu, ktorá by mala túto hru priblížiť ľuďom z celého sveta.

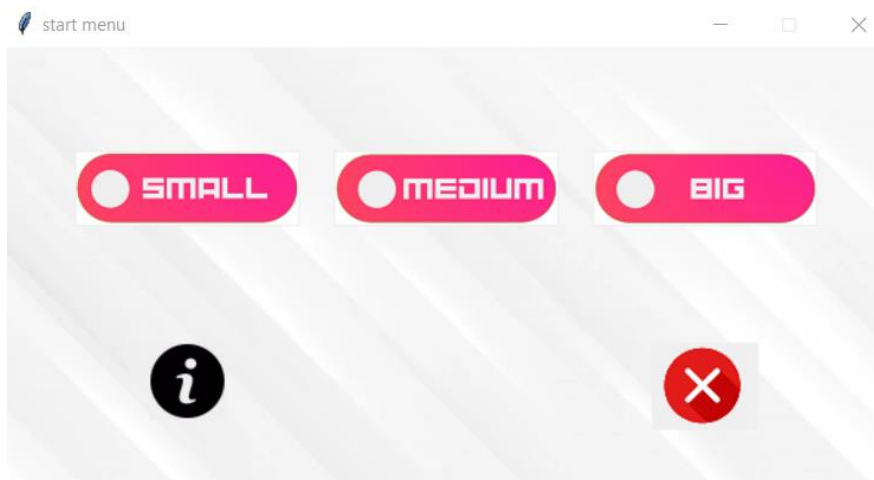
Hra je realizovaná pre dvoch hráčov aj keď existujú alternatívy, aj pre väčšie množstvo hráčov. Pred samotným vývojom tejto hry sme na začiatku riešili dilemu o tom, akým spôsobom bude prebiehať a ako budú hráči hrať. K dispozícii boli tri varianty. Prvé dve

boli totožné ako u hry pirát a Sicíľčan. Prvou 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. Avšak z dôvodu, že chceme simulovať hru dvoch reálnych hráčov, tak by sme museli pridať omylnosť počítača aby jednotlivé kroky nerobil len podľa víťazných riešení v strome. A tretou variantou 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. Táto varianta je taktiež programátorsky jednoduchšia alternatíva ako zvyšné dve. Práve pre tieto dôvody sme sa rozhodli zvoliť túto možnosť.

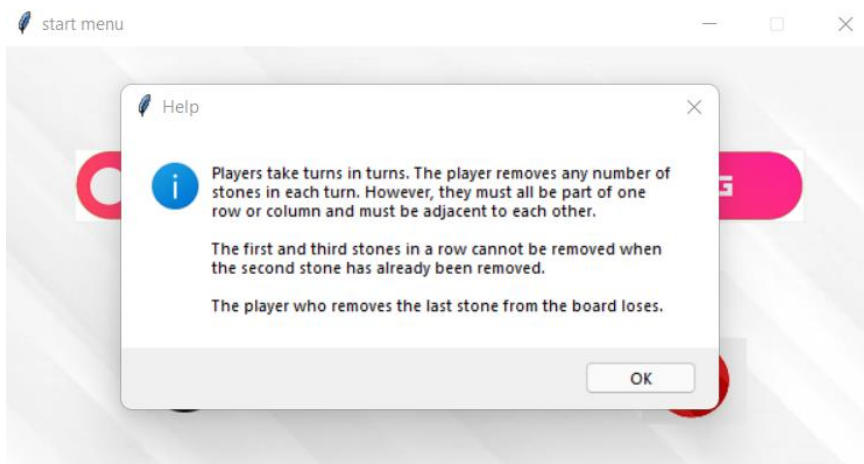
Pre grafický dizajn aplikácie sme zvolili moderný štýl a to aj napriek tomu, že hra je veľmi stará. Tento štýl má hráčovi ukázať nie len to, že hra nie je len nejaká stará hra, ale zároveň aby mal hráč príjemný zážitok.

Na úvodnej obrazovke sa nachádza päť tlačidiel. Ružové tlačidlá s názvami small, medium a big slúžia na spustenie hry. Samotné názvy hovoria o veľkosti hracej plochy. V prípade kliknutia na tlačidlo small sa otvorí plocha o veľkosti štyri krát štyri alebo teda so 16 kameňmi. V prípade medium tlačidla sa jedná 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 s písmenom i a x. Tlačidlo i slúži na otvorenie návodu, 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 v prípade, že sa hráči rozhodnú nehrať túto hru, tak toto tlačidlo im umožňuje ukončiť aplikáciu.



Obr. 4.6 Úvodná obrazovka hry Bulo



Obr. 4.7 Vyskakovacie okno s návodom hry Bulo

Po zvolení jednej z veľkosti hry sa otvorí okno s hrou. V ľavom hornom rohu sa vždy nachádza informácia o type hry. Rozloženie hry je dané poľom vzdialeností na x súradnici od stredu. Týmto sme dosiahli to, že si 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 krát 4. Pozícia 0 je stred hracej plochy s kameňmi. Podmienka, ktorú sme si stanovili pre tieto polia je, že sa kamene nachádzajú od seba vzdialené 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.

*square\_playing\_field* =

[ [-3, -1, 1, 3],

[-3, -1, 1, 3],

[-3, -1, 1, 3],

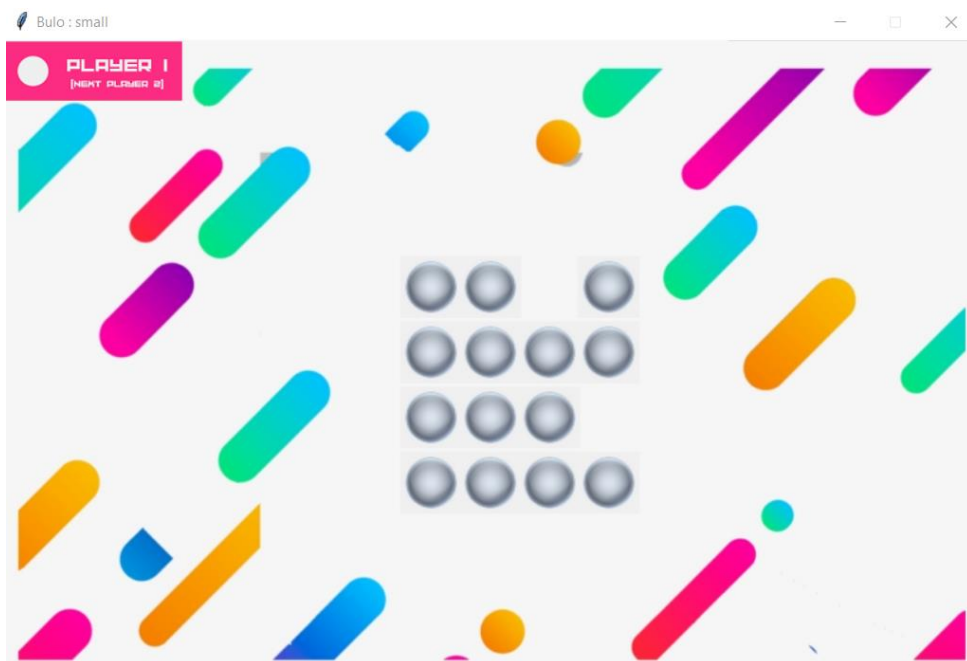
[-3, -1, 1, 3] ]

Pri rozhodovaní o tom, akým spôsobom naprogramovať kamene v hracej ploche sme zvažovali dve možnosti. Prvou bolo spraviť kamene ako obrázok 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 poskytujú to, že kliknutie riešia sami a to len v prípade, že je dané tlačidlo zakliknuté. Zároveň pomocou tlačidla vieme ihneď zavolať funkciu, ktorá sa má vykonať. Samotné kamene sú teda spravené ako tlačidlá s obrázkom symbolizujúci 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 volá v prípade kliknutia na dané tlačidlo. Jedná sa 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 môže zobrať, v prípade, že berie viac ako jeden kameň, iba kameň, ktorý je buď v rade alebo v stĺpci. Rad alebo stĺpec musí byť rovnaký ako jeho prvý výber, respektíve ak sa jedná 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, že 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, tak musí kliknúť na tlačidlo. Obrázok 4.8 zobrazuje hru, kde už obaja hráči zahráli svoje prvé ťahy a každý z hráčov si zobral práve jeden kameň.





Obr. 4.8 Hracie pole 4x4 hry Bulo s odobratými kameňmi

Kontrola, či si hráč, ktorý je na ťahu vybral správny kameň sa deje ihneď po kliknutí na neho. Táto kontrola sa líši od zvoleného typu hry. V prípade, že sa jedná 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, ak 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 x súradnici. Premenná `direction` sa bude využívať

v prípade, ak sa hráč rozhodne ťahať aj ďalšie kamene. Hovorí o tom, že v ktorom smere musí hráč zobrať ďalší kameň. Teda buď v rovnakom rade, stĺpci alebo poprípade diagonálne. Pri ťahu druhého kameňa je premenná držiaca 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`. Ukaže kontrolu, či sa vybraný kameň nachádza na x súradnici a ak hej, 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:`*

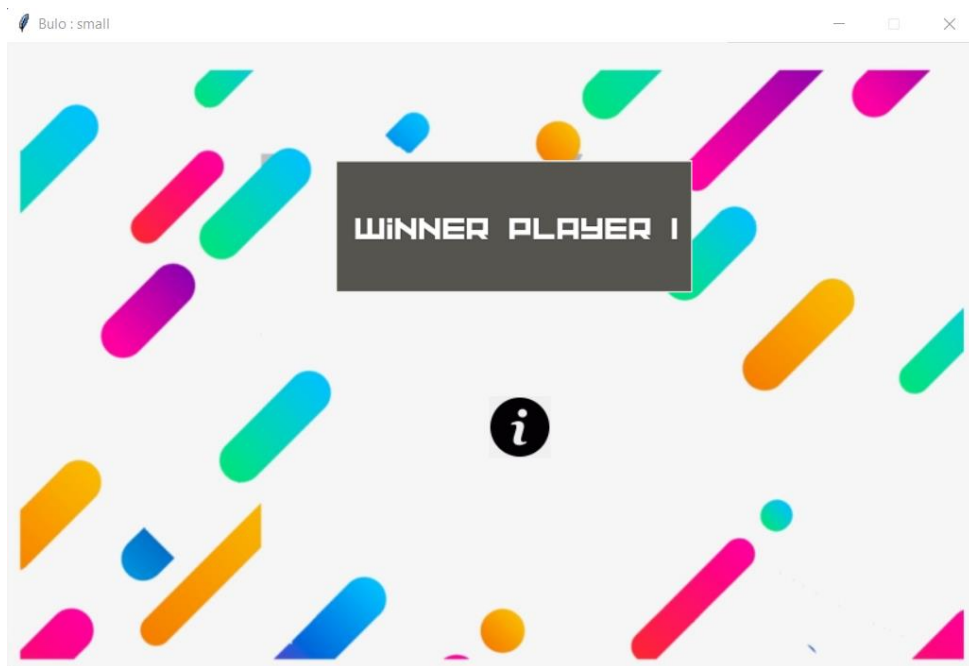
*`delete_btn(x, y)`*

*`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 sa nejedná o pravidlá hry ale ide o informáciu s analýzou ťahov hráčov. Po kliknutí na toto tlačidlo sa otvorí analýza odohratej hry.



Obr. 4.9 Obrazovka po odohratí hry Bulo, kde sa víťazom stal hráč 1

#### 4.3.1 Analýza

V tejto časti sa pozrieme na softvérové riešenie analýzy 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 pomoc s analýzou problému pri samotnej hre. Z týchto ale aj iných dôvodov je analýza veľmi zaujímavá a je dobrým rozšírením v hre.

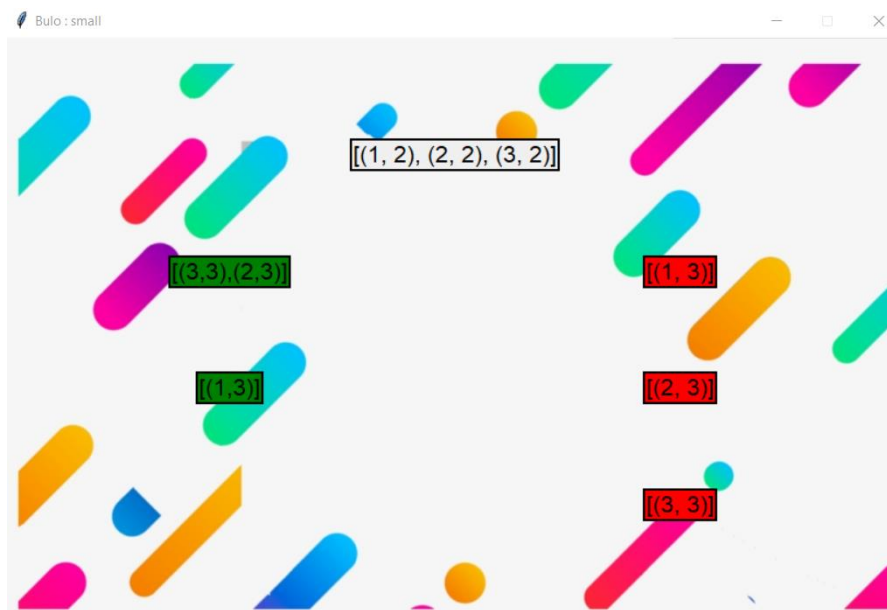
Pri príprave riešenia, akým spôsobom analyzovať túto hru sme vychádzali zo známej analýzy pomocou stromu hry. Táto analýza poskytuje vizuálne ľahko pochopiteľný výsledok, ktorý dá každému, kto bude hrať túto hru, čo najviac informácií. Pri rozhodovaní, akým spôsobom spraviť analýzu stromu, sme vyberali 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 už pre najjednoduchší typ hry. Ak by sme vybrali hru small, tak v prvom ťahu má hráč 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 analýzu len pre typ hry small, keďže pre väčšie typy hier by to bolo komplikovanejšie. Samotná analýza aj pre najmenšie rozloženie poskytne dostatočné informácie pre pochopenie hry a poskytne aj dostatočné informácie na analýzu ťahov jednotlivých hráčov. Pri väčších rozloženiach hracieho poľa by sa jednalo o podobný postup avšak rozšírený o ďalšie kroky.

Samotná vizualizácia stromu hry tiež so sebou prináša komplikácie. Hlavne, čo sa týka veľkosti stromu a zároveň prehľadnosti. Riešenie tohto problému ako aj hľadanie alternatív k všeobecnému stromu prinieslo 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. 4.10 Analýza - Strom odohratej hry Bulo

Preto, aby analýza dávala zmysel hráčom a mala aj pridanú hodnotu, sme hľadali spôsob, ako to realizovať. Hra sa dá analyzovať buď vo všeobecnosti alebo ako konkrétna jedna hra, ktorú hráči zahrali. 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 poskytuje to, že si hráči budú pamätať väčšinu vecí, ktoré sa udiali. Tým pádom poskytne okamžitú spätnú väzbu pre hráča a 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 väčšie a budú si to vedieť prepojiť s nedávno odohratou hrou. A posledným dôvodom, ktorý spomeniem, 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 bodom je to, 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. Dôležitou časťou je aj to, že ako hlboko sa vnárať do stromu 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 dúfať 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 týmto pádom ostali v hre vyskladáme novú cestu. Základom tejto analýzy je, že počet ťahov v hre musí byť nepárny. V tom momente dosiahneme to, že vyhrať 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.

#### 4.3.2 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 robia 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 párny počet ťahov v strome, mimo koreňa. Tu sa jedná o hráča, ktorý prehral hru. V prípade párneho počtu ťahov oboch hráčov, víťazí hráč, ktorý prehral aktuálnu hru. Dostávame výsledok, ktorý ukazuje potencionálnu 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, že do ktorej z dvoch nasledujúcich podmienok vstúpime. Prvá podmienka ráta s tým, že počet kameňov je párny. Druhá podmienka vychádza z toho, že počet kameňov je nepárny. Toto rozdelenie je dôležité preto, lebo na to aby sme vedeli lepšie pracovať s týmito dátami je potrebné vydedukovať, že ako sa k nemu správať. V prípade nepárneho počtu kameňov vieme povedať, že ak chceme aby hráč vyhral potrebujeme nájsť minimálne jeden pár. Je to z dôvodu, ktorý sme si spomenuli už vyššie. A to, že ak hráč chce vyhrať túto hru, tak musí byť v tejto podčasti hry počet ťahov párny. Tým pádom z nepárneho počtu kameňov musíme nájsť minimálne jeden pár, ktorý sa zahrá v jednom kole a vďaka tomu je tam možnosť zahráť párny počet ťahov tejto hry. Tým dostávame víťazstvo hráča ktorý pôvodne prehral. Samozrejme môžeme hľadať aj ďalšie dvojice ale už musíme myslieť na to, že musíme nájsť párny počet dvojíc, ktoré nasledujú. To vo výsledku znamená, že počet párov, ktoré by sme mali nájsť bude nepárny. Teraz sa pozrieme na druhý prípad a to keď je počet kameňov párny. Pri analýze vidíme, že ak by každý hráč zahrál po jednom kameni v jednom ťahu, tak hráč, ktorý prehral pôvodnú hru by aktuálnu vyhral. Avšak aj v tomto prípade sa oplatí pozrieť na prípadné dvojice kameňov. Tento prípad bude opačný ako predchádzajúca podmienka. Budeme v nej hľadať párny počet dvojíc lebo len vďaka tomu môže hráč, ktorý pôvodne prehral hru vyhrať. V prípade nepárneho počtu párov vzniká nepárny počet ťahov a tým pádom je výsledok zhodný 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 jednoduchosti kódu sme 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. Jedná sa o kód, ktorý v sebe obsahuje pseudokód a vnútro podmienok riešiacich, či je počet kameňov párný alebo nie, je napísaný 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 hovorí o hľadaní cesty pre rôzne počty kameňov. Po splnení podmienky, že je počet párný 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 nemôže ostať. Je to z dôvodu, že aj maximálny počet kameňov, ktoré môže hráč zobrať v jednom ťahu sú 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 obsahuje koreň stromu obsahujúci ťah hráča, ktorý vyhral. Po ňom sa strom rozvetvuje a obsahuje v sebe zelenú vetvu, a červenú vetvu. Červená vetva obsahuje prvky ktoré boli zahrané v predchádzajúcej hre a zelené prvky, nachádzajúce sa v druhej vetve, ukazujú alternatívnu víťaznú cestu pre druhého hráča. Vizualizácia vizualizuje okrem koreňa stromu tieto vetvy nasledovne. Víťazná vetva hráča, ktorý pôvodne prehral sa vykresľuje spôsobom, že sa najskôr použijú existujúce páry a až potom sa vykreslia samostatné prvky. Červená vetva sa vykreslí v tom poradí ako 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 nie je kompletný kvôli svojej dĺžke. Obsahuje kľúčové časti a zvyšné časti sú doplnené popisom kódu, ktorý sa v programe nachádza.

```
def tree_vizualisation(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
```

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



## Záver

Cieľom tejto inžinierskej práce bola analýza zvolených typov hier v oblasti Teórie hier a ich implementácia spolu s analýzou. 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 analýzy ťahov hráča sme sa venovali aj možnému softvérovému riešeniu a na základe nami zvolených postupov sme prešli k realizácii úloh v programovacom jazyku.

Nami zvolené postupy nám dali na konci výsledky hotových hier, 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 a jej analýza prináša ešte veľa priestoru na zlepšenie pre ďalšie práce. V našej práci riešime analýzu iba jednoduchého prípadu rozloženia plochy. Určite by bolo dobré, v budúcnosti spraviť analýzu aj pre väčšie hracie plochy. Tieto analýzy môžu poskytnúť ďalší zo zaujímavých pohľadov pre každého z nás.

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. Pre nás to zároveň môže byť výzva pokračovať v tejto téme a rozširovať možnosti spoznávania aj pre nás, ale aj ostatných.

## Zoznam obrázkov

1.1 Klasifikácia hier.....	11
1.2 Strom hry Nim.....	17
1.3 Rozloženia hry Bulo.....	18
3.1 Maticová reprezentácia hry Pirát a sicílčan.....	22
3.2 Strom hry Bulo .....	24
3.3 Strom hry Bulo s vyznačenou vetvou.....	25
3.4 Strom hry Bulo s novou vetvou po použití spätnej indukcie.....	26
4.1 Graf oblíbenosti jednotlivých vývojových prostředí.....	30
4.2 PyCharm prostredie.....	31
4.3 Úvodná obrazovka hry pirát a Sicílčan.....	33
4.4 Obrazovka hry pirát a Sicílčan, kde sa hráč rozhoduje, akú stratégiu použije.....	34
4.5 Obrazovka hry pirát a Sicílčan s analýzou hry.....	35
4.6 Úvodná obrazovka hry Bulo.....	37
4.7 Vyskakovacie okno s návodom hry Bulo.....	37
4.8 Hracie pole 4x4 hry Bulo s odobratými kameňmi.....	39
4.9 Obrazovka po odohratí hry Bulo, kde sa víťazom stal hráč 1.....	41
4.10 Analýza – Strom odohratej hry Bulo.....	42

## **Zoznam tabuliek**

Tabuľka 1.1: Stratégie hráčov a ich zisky.....	15
--	----

## 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](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://frcatel.fri.utc.sk/~pesko/TH/sTeoriaHier\\_1.pdf](http://frcatel.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](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](http://frcatel.fri.utc.sk/users/pesko/TH/sTeoriaHier_3.pdf)>.
- [16] ČÍČKOVÁ, 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](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/inzinierska\\_praca/\*](https://zdarilek2.github.io/inzinierska_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 pirát\_Sicilcan 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.