

**MATEMATICKO-FYZIKÁLNÍ
FAKULTA**
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Dann Mosný

Interpret společenských her hraných na mapě složené ze čtverečků

Katedra softwaru a výuky informatiky

Vedoucí bakalářské práce: RNDr. Martin Pergel, Ph.D.

Studijní program: Informatika

Praha 2024

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V Hvar, Chorvátsko dne 18.7.2024

Podpis autora

Chcel by som sa chcel poďakovať všetkým svojim učiteľom a cvičiacim, ktorí mi dali znalosti potrebné na vypracovanie tejto práce. Osobitne by som sa chcel poďakovať svojmu vedúcemu RNDr. Martinovi Pergelovi, Ph.D za jeho pozitívny prístup, ochotu a cenné rady. Nakoniec by som sa chcel poďakovať svojej rodine a svojej priateľke za ich stálu podporu.

Název práce: Interpret společenských her hraných na mapě složené ze čtverečků

Autor: Dann Mosný

Katedra: Katedra softwaru a výuky informatiky

Vedoucí bakalářské práce: RNDr. Martin Pergel, Ph.D., Katedra softwaru a výuky informatiky

Abstrakt: Cílem této bakalářské práce bylo využít společné vlastnosti některých společenských hier na vytvoření univerzální aplikace, v které je možné tyto hry hrát. Konkrétně jsme se zaměřili na hry, které jsou hrané na mapě složené z samostatných štvorcových dílků. Hráči mohou dílky otáčet, přemísťovat a přidávat, a tým hrací plochu měnit. Strany dílků spájají cesty, po kterých se pohybují hráči. Použivatelé by měli mít možnost této univerzální aplikaci zadat pravidla libovolné hry tohoto typu a následně v ní tuto hru hrát. Zadávání pravidel by mělo být jednoduché i pro uživatele bez znalosti programování. Rozhodli jsme se, že pravidla budou uživatelé zadávat do osobitných souborů. Pro ně jsme navrhli strukturu, která popisuje, z akých dílků se hra skládá a aké jsou její pravidla. Soubory s pravidly jsou ve formátu JSON, který je snadno čitelný pro lidi a zároveň je vhodný na zpracování počítačem. Vytvorili jsme program, který tato pravidla načítá a na základě nich vytvoří grafické uživatelské rozhraní, v kterém je možné danou hru hrát. Program kontroluje dodržování pravidel, zdůrazňuje možné tahy a vyhodnocuje, kdo vyhrál.

Klíčová slova: společenská hra, interpreter, mapa složená z dílků, zadávání pravidel

Title: Interpreter of Board Games Played on a Map Consisting of Possibly Movable Square Tiles

Author: Dann Mosný

Department: Department of software and computer science education

Supervisor: RNDr. Martin Pergel, Ph.D., Department of software and computer science education

Abstract: Goal of this bachelor thesis was to take advantage of common features of some board games to create a universal application, in which these games can be played. We focused on games played on a map consisting of individual square tiles. Players can rotate, move and add these tiles and therefore change the playing board. Players move on paths, which connect sides of tiles. Users should be able to input rules of any game of this type into this universal application and consequently play this game in the application. Inputting the rules should be easy even for users not capable of programming. We decided that users will input the rules into separate files. For them, we designed a structure which describes the rules of the game and describes tiles that make up the board. Files with rules are in the JSON file format, which is easily readable by humans and also suitable for computer processing. We made a program which can load these game rules and based on them create a graphical user interface in which the game can be played. The program enforces following of the rules, highlights possible moves and evaluates who had won.

Keywords: board game, interpreter, map consisting of tiles, inputting rules

Obsah

Úvod	7
1 Predpoklady	9
1.1 Hranie spoločenských hier na počítači	9
1.1.1 Lokálny a online multiplayer	9
1.1.2 Striktné a sandboxové hry	10
1.1.3 Existujúce implementácie	10
1.2 Spoločenské hry hrané na potenciálne pohyblivých štvorcíkoch . .	11
1.2.1 Všeobecný popis	11
1.2.2 Možné akcie	11
1.2.3 Pravidlá vzorových hier	14
2 Výsledok práce a jej limity	16
2.1 Výsledok	16
2.2 Limity	17
3 Používateľská dokumentácia	18
3.1 Spustenie	18
3.2 Ovládanie	18
3.2.1 Uživatelské rozhranie	18
3.2.2 Vykonávanie akcií	20
4 Vytváranie hier	23
4.1 Formát Json	23
4.2 Štruktúra a možnosti pravidiel	23
5 Riešené problémy a urobené rozhodnutia	31
5.1 Multiplayer	31
5.2 Konfiguračné súbory s pravidlami	31
5.3 Reprezentácia ciest na dielikoch	32
5.4 Otáčanie dielikov	32
5.5 Číslovanie ciest	33
5.6 Vyradenie hráčov z hry	33
5.7 Odídenie mimo plochy	33
5.8 Výpočet možných ťahov	34
6 Programátorská dokumentácia	35
6.1 Nástroje použité na vývoj	35
6.2 Objektový návrh	35
6.3 Trieda Board	36
6.3.1 Trieda Tile	37
6.3.2 Trieda Player	37
6.4 Trieda GameController	38
6.5 Trieda GameDrawer	39
Záver	41

Literatúra	43
A Prílohy	44

Úvod

Motivácia

Pri prevádzaní spoločenských hier do digitálnej podoby je nepraktické prevádzať každú spoločenskú hru osobitne, najmä keď sú si mnohé spoločenské hry navzájom podobné. Namiesto naprogramovania každej spoločenskej hry je efektívnejšie urobiť jeden univerzálny program pre viac hier. Výhodou takéhoto programu je, že spoločné vlastnosti hier sú naprogramované iba raz. Odlišujúce sa pravidlá každej hry potom používatelia špecifikujú zvlášť pre každú hru v jej konfiguračnom súbore. Takýto univerzálny program dokáže súbor s pravidlami spracovať a vytvoriť grafické používateľské prostredie, v ktorom je možné danú hru hrať.

Okrem toho, že tento prístup je efektívny, umožňuje používateľom rýchlo dopĺňať do programu ďalšie hry bez znalosti programovania. Na pridanie novej hry stačí len vytvoriť konfiguračný súbor, v ktorom popíšu pravidlá. Týmto spôsobom môžu používatelia okrem pridávania a modifikovania už existujúcich hier vytvárať aj svoje vlastné nové hry.

Cieľ

Zamerali sme sa na skupinu hier, ktorú sme nazvali *hry hrané na mape zloženej zo štvorčekov*, ktoré majú tieto spoločné vlastnosti:

- hracia plocha je zložená z dielikov v tvare štvorca
- každý dielik obsahuje cesty spájajúce rôzne strany dieliku
- hráči ovládajú figúrky, ktoré sa pohybujú po týchto cestách
- hráči sa striedajú po ťahoch
- dieliky je možné otáčať, posúvať a pridávať
- dieliky môžu hráči držať na ruke

Príkladmi takýchto hier sú napríklad: *Amazing Labyrinth*, *Tsuro*, *Snakes and Ladders* a *Mice in a maze*.

Naším cieľom pre túto bakalársku prácu bolo:

- navrhnuť používateľsky prívetivú štruktúru konfiguračných súborov na popisovanie pravidiel hier nami zvoleného typu
- naprogramovať univerzálny program – interpret – schopný tieto súbory spracovať a na základe nich vytvoriť danú hru
- vytvoriť užívateľské rozhranie, v ktorom je možné dané hry hrať
- vytvoriť ukázkové konfiguračné súbory s pravidlami pre hry *Amazing Labyrinth*, *Tsuro*, *Snakes and Ladders* a *Mice in a maze*

Štruktúra práce

V prvej kapitole rozoberieme predpoklady práce — podrobne popíšeme typ hier, ktorý sme si zvolili a popíšeme už existujúce možnosti hrania týchto spoločenských hier na počítači. V druhej kapitole popíšeme, čo je výsledkom našej práce a aké má limity. Tretia kapitola obsahuje opis toho, ako sa ovládajú hry v našej aplikácii. V štvrtej kapitole je návod pre používateľov, ako vytvárať vlastné hry pomocou vytváranie konfiguračných súborov s pravidlami. V piatej kapitole uvedieme, aké rozhodnutia sme pri implementácii programu urobili a aké problémy sme riešili. Posledná šiesta kapitola popisuje, ako je štruktúrovaný samotný program a ako sme implementovali jeho najdôležitejšie časti.

1 Predpoklady

V tejto kapitole najprv všeobecne popíšeme, aké sú možnosti pri prevádzaní hier do digitálnej podoby, a ktoré z týchto možností boli zvolené pri už existujúcich programoch na hranie spoločenských hier. Následne podrobne popíšeme typ hier, na ktorý sme sa zamerali a uvedieme pravidlá konkrétnych hier, na ktorých budeme demonštrovať schopnosti nášho programu.

1.1 Hranie spoločenských hier na počítači

Digitálne verzie spoločenských hier umožňujú používateľom hrať hry bez vlastníctva fyzickej verzie týchto hier. Tiež môžu umožňovať hrať spolu hru ľuďom, ktorí sa nenachádzajú na jednom mieste. Ďalšou výhodou digitálnych spoločenských hier je, že automatizujú niektoré úkony, ktoré by inak museli hráči robiť manuálne, ako je napríklad príprava hry a počítanie skóre. Zároveň tieto verzie môžu dávať pozor na dodržiavanie pravidiel a nedovoliť hráčom robiť neplatné ťahy. Táto funkcia zabraňuje podvádaniu, ale aj neúmyselným porušeniam pravidiel.

Pre popis toho, ako je digitálna spoločenská hra implementovaná sme si vybrali dve kritériá. Prvým kritériom je, či všetci hráči na hru využívajú to isté zariadenie – *lokálny multiplayer*, alebo hrá každý na svojom zariadení – *online multiplayer*. Druhým kritériom je, či hra kontroluje dodržiavanie pravidiel, tie sme nazvali *striktné*, alebo dovoľujú hráčom robiť akékoľvek ťahy, tie sme nazvali *sandboxové*.

1.1.1 Lokálny a online multiplayer

Lokálny multiplayer

Digitálne spoločenské hry, ktoré majú lokálny multiplayer sa hrajú na jednom zariadení. Toto je vhodné pre ťahové spoločenské hry — hráči sa striedajú pri zariadení, podľa toho, kto je práve na ťahu. Lokálny multiplayer tiež funguje dobre, keď je väčšina dôležitých prvkov hry zdieľaných medzi hráčmi, napríklad ak sú všetky figúrky na jednej spoločnej ploche. To dovoľuje všetkým hráčom sledovať stav hry a premýšľať nad svojím ďalším ťahom, aj keď nie sú práve na ťahu. Ak je v hre veľa prvkov, ktoré sú relevantné len pre jedného hráča, je zobrazovanie hry nepraktické. V takom prípade je jedna možnosť zobrazovať len informácie, ktoré sa týkajú hráča na ťahu. To však znamená, že hráči, ktorí nie sú práve na ťahu nevidia pre nich relevantné časti hry. Druhou možnosťou je zobrazovať informácie týkajúce sa všetkých hráčov zároveň, čo je však neprehľadné, najmä na zariadeniach s menším displejom. Výhodou lokálneho multiplayeru oproti online multiplayeru je, že hra nepotrebuje pripojenie k sieti.

Online multiplayer

Digitálnu spoločenskú hru s online multiplayerom hrá každý hráč na svojom vlastnom zariadení. Zariadenia všetkých hráčov sú pripojené k sieti a komunikujú spolu. To znamená, že hráči nemusia byť fyzicky spolu, čo je veľká výhoda oproti lokálnemu multiplayeru. Vlastné zariadenie zároveň dovoľuje hre zobrazovať

každému hráčovi iba informácie, ktorého sa ho týkajú. Zároveň to umožňuje, aby hráči mali tajné informácie, ktoré ostatní hráči nevidia. Príkladom sú napríklad karty na ruke. Nevýhodou online multiplayeru je, že problémy so sieťou môžu hru spraviť nehrateľnou.

1.1.2 Striktné a sandboxové hry

Striktné hry

Striktné implementácie hier dovoľujú hráčovi vykonať iba akcie, ktoré sú v súlade s pravidlami. Tieto „legálne“ ťahy sú väčšinou zvýraznené, čo hráčom zjednodušuje hranie, najmä pri hrách so zložitými pravidlami. Tiež to predchádza chybám, kedy hráč poruší pravidlá nevedome. Takéto hry tiež zvyknú pomáhať hráčom pri vykonávaní zložitejších akcií, ktoré by napríklad vyžadovali počítanie alebo zohľadňovanie množstva informácií. Typickými príkladmi sú automatické počítanie skóre, automatické aktualizovanie stavu hry a miešanie kariet, čo by inak hráči museli robiť manuálne.

Sandboxové hry

Hry so sandboxovou implementáciou poskytujú používateľom prostredie, v ktorom môžu hýbať figúrkami, hádzať kockami, vykladať karty a podobne bez akýchkoľvek obmedzení. Ide o prevedenie blízke fyzickým verziám spoločenských hier. Program poskytuje všetko potrebné na hranie, ale do hrania nijako nezasahuje. V takejto implementácii musia hráči všetko robiť manuálne, ale zároveň majú možnosť nehrať striktné podľa pravidiel a upraviť si ich podľa svojich preferencií.

1.1.3 Existujúce implementácie

Medzi najznámejšie platformy umožňujúce hranie rôznych digitálnych spoločenských hier patria *Tabletop Simulator*, *Board Game Arena* a *Tabletopia*.

Tabletop Simulator

Tabletop Simulator je aplikácia, ktorá spadá do nami definovaných kategórií online multiplayer a sandbox. Ide vlastne o fyzikálnu simuláciu stola so spoločenskou hrou, s ktorou môžu hráči ľubovoľne interagovať. Môžu zdvíhať a posúvať figúrky, vykladať karty na stôl, hádzať kockami a podobne. Používatelia majú možnosť do hry pridávať nimi vytvorené modely hracích dielikov, čím môžu vytvárať vlastné hry. Taktiež môžu do simulácie pridávať nové funkcie pomocou skriptov v jazyku Lua. [1]

Board Game Arena

Board Game Arena je podľa nášho delenia striktná online multiplayer platforma dostupná z internetového prehliadaču. Hry sú vykresľované v 2D, platforma kontroluje platnosť ťahov, zvýrazňuje možné ťahy a automatizuje zdĺhavé manuálne úkony. Používatelia nemôžu vytvárať svoje vlastné hry, ale môžu sa pridať do tímu developerov a implementovať spoločenskú hru, na ktorú získala platforma licenciu. Hry sú vyvíjané pomocou PHP, SQL, HTML, CSS a Javascriptu. [2]

Tabletopia

Tabletopia je podobne ako Tabletop Simulator sandboxová online multiplayer platforma. Podobne ako Tabletop Simulator funguje ako fyzikálna simulácia. Na rozdiel od Tabletop Simulatoru sa Tabletopia dá používať priamo z internetového prehliadača. Hry je možné vytvárať pomocou vyrábania nových dielikov. Tie sa vytvárajú pomocou nahrávania textúr na poskytnuté objekty, ako sú karty a kocky. [3]

1.2 Spoločenské hry hrané na potenciálne pohyblivých štvorcových

1.2.1 Všeobecný popis

Typ hier, ktoré budeme implementovať sme nazvali *Spoločenské hry hrané na mape zloženej z potenciálne pohyblivých štvorcových dielikoch*. Ide o ťahové strategické spoločenské hry, ktoré sú špeciálne tým, že hrací plán nie je stále rovnaký, ale je zložený z viacerých dielikov, ktoré môžu hráči otáčať, premiestňovať a pridávať. Menenie hracieho plánu dáva hráčom mnoho možností ako dosiahnuť svoj cieľ. Takéto hry teda hráčom môžu prinášať pocity riešenia hlavolamu, kde sa snažia nájsť optimálnu zmenu hracej plochy. Premiestňovanie dielikov môžu hráči využiť nielen vo svoj prospech, ale aj ako spôsob prekazenia plánov protihráčov. Hráč teda na svojom ťahu musí rozmýšľať, ako jedným ťahom čo najviac pomôcť sebe a zároveň uškodiť ostatným. Pri plánovaní musí hráč rátať s menením plochy ostatnými hráčmi v nasledujúcich kolách. Neustále menenie sa hracej plochy núti hráčov improvizovať a adaptovať svoje plány na aktuálny stav mapy. Nepredvídateľnosť a vysoká miera interakcie medzi hráčmi robí tieto hry zaujímavými a zábavnými.

V týchto hrách každý hráč ovláda práve jednu figúrku. Figúrky sa pohybujú medzi dielikmi po cestičkách – prepojeniach niektorých strán štvorcového dieliku. V niektorých hrách je rozlišované okrem toho, ktoré strany cesta spája aj kde presne na strane je spojenie. V takom prípade môže napríklad okrem dielika ktorý má navzájom prepojené tri strany, existovať aj dielik, ktorý má tiež prepojené tri strany, ale na každej strane sú dve cesty, ktoré nie sú navzájom spojené (viď obrázok 1.1).

Dieliky hracej plochy môžu mať špeciálne efekty, ako je napríklad presun hráčov alebo dokonca ich elimináciu z hry. V takej hre hráč vyhrá vtedy, ak zostane posledným hráčom. V iných hrách sa na výhru hráč potrebuje dostať na nejaký konkrétny dielik alebo pozbierať zadané predmety z hracej plochy.

1.2.2 Možné akcie

Hráč na ťahu môže podľa pravidiel hry vykonať niektoré z akcií: *Pohyb*, *Pohyb všetkých*, *Posunutie stĺpca/riadku dielikov*, *Pridanie dielika na plochu* alebo *Vymenenie dielika z plochy*.

Pohyb

Hráč posunie svoju figúrku z dielika, na ktorom sa práve nachádza na iný dielik s ním prepojeným cestičkami. V niektorých hrách je tento pohyb neobmedzený a je možné sa presunúť na ľubovoľný dielik, na ktorý vedie cesta. V iných hrách je vzdialenosť, o ktorú sa hráč môže posunúť obmedzená alebo určená hodom kockou. Taktiež môže byť v pravidlách dané, že hráči sa môžu pohybovať len smerom vpred po cestičkách a nemôžu sa vracieť.

Pohyb všetkých

Všetci hráči sa musia posunúť smerom dopredu až na koniec cestičky, na ktorej sa práve nachádzajú.

Posunutie stĺpca/riadku dielikov

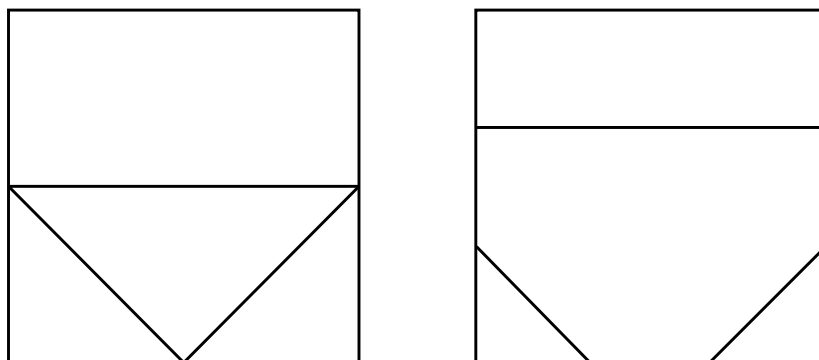
Pokiaľ je celá štvorcová mapa zaplnená dielikmi, niektoré hry dovoľujú hráčom posunúť celý riadok alebo stĺpec dielikov. Napríklad pokiaľ hráč posunie riadok dielikov doprava, dielik z prvého stĺpca sa posunie do druhého stĺpca, dielik z druhého stĺpca sa posunie do tretieho stĺpca a tak ďalej. Dielik v poslednom stĺpci sa vysunie von z hracej plochy a v prvom stĺpci zostane prázdne miesto (viď obrázok 1.2). Analogicky to funguje pre posúvanie stĺpcov. Prázdne miesto, ktoré vznikne po posunutí je podľa pravidiel zaplnené dielikom, ktorý bol buď vysunutý z hracej plochy, alebo dielikom z vrchu kopy. Pokiaľ sa na dieliku, ktorý bol vysunutý nachádzala figúrka, figúrka sa presunie na dielik, ktorý bol doplnený na prázdne miesto na druhej strane riadka alebo stĺpca. V prípade posunutia riadku doprava sa teda figúrka z posledného stĺpca posúvaného riadka presunie do prvého stĺpca toho istého riadka. Pravidlá hry môžu špecifikovať, ktoré riadky a stĺpce môžu a nemôžu byť posúvané.

Pridanie dieliku na plochu

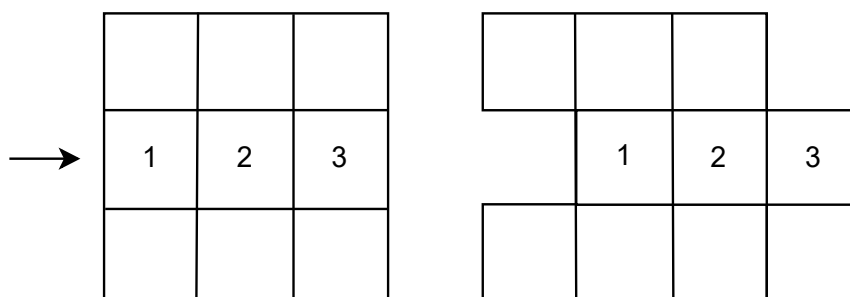
Hráč môže na prázdne miesto na ploche (bez dieliku) pridať nový dielik. Ten môže byť buď z vrchu kopy dielikov, alebo z ruky hráča. Pokiaľ je z ruky, hráč si namiesto pridaného dielika potiahne z kopy nový dielik. Hráč môže pred pridaním dielik ľubovoľne natočiť. Pravidlá niektorých hier vyžadujú, aby bol nový dielik pridaný tak, aby jednou zo svojich cestičiek predĺžil cestu, na ktorej sa hráč práve nachádza (viď obrázok 1.3).

Vymenenie dielika

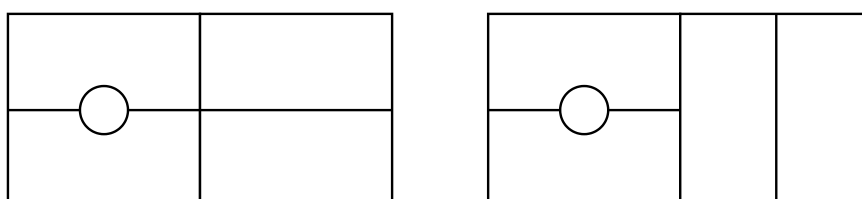
Hráč odoberie dielik z hracej plochy a vymení ho s dielikom z ruky alebo z kopy. Pravidlá zakazujú vymeniť dielik, na ktorom sa nachádza figúrka nejakého hráča. Taktiež môžu špecifikovať konkrétne dieliky, ktoré nemôžu byť vymenené. Táto akcia môže prevrátiť dieliky, ktoré sú skryté — otočené zadnou stranou bez cestičiek hore. Skrytý dielik je odobratý z hracej plochy a nahradený dielikom otočeným stranou s cestičkami hore.



Obrázek 1.1 Dielik s jednou (vľavo) a s dvomi cestami (vpravo) na každej strane.



Obrázek 1.2 Pred (vľavo) a po (vpravo) posunutí druhého riadku smerom doprava.



Obrázek 1.3 Správne (vľavo) a nesprávne (vpravo) pridanie dielika tak, aby predĺžil cestu s hráčom (reprezentovaný kruhom).

1.2.3 Pravidlá vzorových hier

Uvedieme stručné zhrnutie pravidiel pre ukázkové hry, ktoré spadajú do nami definovanej skupiny hier hraných na mape zloženej zo štvorcíkov.

Amazing labyrinth

V tejto hre má každý hráč kôpku kartičiek s rôznymi objektami, ktoré musí nájsť. Tieto objekty sa nachádzajú na dielikoch na hracej ploche. Zvíťazí hráč, ktorý ako prvý navštívi všetky objekty na jeho kartičkách a následne sa vráti na dielik, na ktorom začína. Vždy na začiatku ťahu musí hráč zmeniť hraciu plochu posunutím jedného riadka alebo stĺpca dielikov. Hracia plocha je celá zaplnená dielikmi a jeden dielik je vždy navyše, mimo hracieho plánu. Týmto dielikom sa zaplní prázdne miesto po posúvaní a vysunutý dielik zostáva mimo hracej plochy. V ďalšom ťahu on zaplní prázdne miesto vzniknuté pri posúvaní. Posúvaním si hráč snaží vytvoriť cestu k predmetu, ktorý práve hľadá. Po posúvaní môže hráč posunúť svoju figúrku po vytvorených cestách ako ďaleko chce. Ak sa dostane na dielik s predmetom, ktorý hľadá, zoberie si so svojej kôpky ďalšiu kartičku s hľadaným predmetom. Ak už na kôpke nemá ďalšiu kartu, našiel všetky svoje predmety a k víťazstvu mu už zostáva sa len dostať na dielik, na ktorom začína.

Tsuro

Cieľom hry je zostať posledným hráčom v hre. Hráči začínajú na okraji prázdnej hracej plochy. Každý hráč má na ruke tri dieliky. Vždy keď je hráč na ťahu, musí priložením nového dielika predĺžiť cestu, na ktorej sa nachádza jeho figúrka. Následne si na ruku potiahne nový dielik z kopy. Po pridaní dielika musia všetci hráči musia posunúť svoje figúrky až na koniec ciest, na ktorých sa nachádzajú. Pokiaľ sa figúrka nejakého hráča týmto dostane mimo hracej plochy, je daný hráč vyradený z hry. Hráči sa teda pridávaním dielikov na plochu snažia vytvoriť také cesty, ktoré ich udržia na hracej ploche a ostatných hráčov dovedú mimo plochy.

Mice in a maze

Hráči v tejto hre predstavujú myši, ktoré sa snažia zo stredu hracej plochy doniesť syr do svojho hniezda. Na začiatku každého ťahu musí hráč posunúť svoju figúrku po ceste, na ktorej sa nachádza až po jej koniec – slepú uličku, stred plochy so syrom, alebo hniezdo na kraji plochy. Pokiaľ sa posunie na stredný dielik so syrom, jeden položí na svoju figúrku a snaží sa ho doniesť do svojho hniezda. Nezávisle od toho, do ktorého hniezda ho cesta dovedie, odovzdá syr vlastníčkovi hniezda. Hracia plocha tiež obsahuje „spoločné hniezda“, ktoré patria dvom hráčom, medzi ktorých hniezdami sa spoločné hniezdo nachádza. Hra začína so všetkými dielikmi otočenými rubom (zadnou stranou) hore. Na takto otočené dieliky hráči nemôžu vstupovať. V druhej fáze ťahu, po pohybe, môže hráč vymeniť dielik pred ním alebo za ním na ceste za dielik na jeho ruke. Po výmene je nový dielik otočený lícom (prednou stranou) hore. Hra končí, keď bol všetok syr odnesený do hniezd. Vyhráva hráč, ktorý má v svojich hniezdach najviac syra.

Snakes and ladders

Ide o jednoduchú hru, v ktorej sa hracia plocha nemení. Napriek tomu spĺňa podmienky na to, aby sme ju mohli implementovať naším programom. Hracia plocha je tvorená jedinou cestou od štartu po cieľ. Každý hráč na svojom ťahu hodí štandardnou hracou kockou a posunie sa (ak môže) o hodené počet dielikov smerom k cieľu. Pokiaľ hráč skončí na jednom zo špeciálnych dielikov so začiatkom rebríka, posunie sa po rebríku na dielik bližšie k cieľu, na ktorom rebrík končí. Ak sa dostane na dielik s hlavou hadom, posunie sa po ňom na dielik ďalej od cieľa, na ktorom má had svoj chvost. Vyhráva hráč, ktorý sa ako prvý dostane na cieľový dielik na konci cesty.

2 Výsledok práce a jej limity

V tejto kapitole popíšeme, čo všetko sme v našej práci implementovali – aké sú schopnosti nášho programu a kde má svoje limity. Taktiež našu aplikáciu porovnáme s už existujúcimi platformami na hranie spoločenských hier popísaných v sekcii 1.1.3.

2.1 Výsledok

Podľa nášho delenia zo sekcie 1.1 sme vytvorili hru, ktorá je lokálna a striktná – hrá sa na jednom počítači a dovoľuje len ťahy v súlade s pravidlami. Náš program umožňuje hrať spoločenské hry z rodiny hier hraných na potenciálne pohyblivých štvorcokoch popísanej v sekcii 1.2.1. Pravidlá aktuálne hranej hry program načítava zo súborov vo formáte JSON. Ide o formát, ktorý ma pevne stanovenú štruktúru (popísanú v sekcii 4.1), čiže je jednoduché pre program ho spracovať. Zároveň je však formát JSON čitateľný pre ľudí, čo znamená, že takýto súbor s pravidlami dokáže vytvoriť aj bežný používateľ.

Na ukážku funkčnosti programu sme vytvorili súbory s pravidlami hier *Amazing labyrinth*, *Tsuro*, *Mice in a maze* a *Snakes and ladders*. Tieto hry nie sú našim výtvorom a nevlastníme ich autorské práva. V tejto práci sú používané len na účely demonštrácie schopností nášho programu.

Používatelia môžu v súbore s pravidlami nastaviť, ktoré akcie a v akom poradí hráči vykonávajú. Ďalej môžu pre každú akciu špecifikovať jej pravidlá – vzdialenosť pohybu, ako sa zapĺňa vzniknutá medzera pri posunoch riadkov a stĺpcov, či sa dieliky pridávajú z ruky alebo z kôpky, ako otočené dieliky je možné vymieňať a podobne.

Pri hracej ploche môžu určovať jej rozmery, ktoré riadky a stĺpce nemožno posúvať a ktoré dieliky nie je možné vymeniť.

Používatelia taktiež určujú aké štvorcové dieliky budú v hre – koľko a akých cestičiek na sebe budú mať a či sa po vkročení na ne aktivujú nejaké efekty. Taktiež môžu nastaviť začiatkové pozície štvorcových dielikov a prípadne nechať nezadané prázdne miesta na ploche vyplniť náhodnými dielikmi. Dieliky môžu obsahovať predmety, pri ktorých je možné určiť, či ich hráči môžu zobrať a v akom množstve.

Určiť sa dá aj koľko hráčov hra bude mať, kde začínajú a aké sú ich ciele. Ciele môžu byť taktiež náhodne priradené z používateľom zadaného zoznamu cieľov v hre. Hráči môžu taktiež na ruke držať nejaký počet dielikov, ktorý je tiež možné špecifikovať.

Tiež je možné špecifikovať, kedy hra skončí – či je to vtedy, keď niektorý hráč splnil všetky svoje ciele, zostal posledným hráčom v hre alebo vtedy, keď je naplnená nejaká podmienka. Ak hra skončí naplnením nejakej podmienky, víťazom je hráč s najväčším počtom bodov. Čím hráči získavajú body používateľ tiež určuje v pravidlách.

Prehľad všetkých možností pri vytváraní hier je v sekcii 4.2.

V porovnaní s platformami *Tabletop Simulator* a *Tabletopia* je naša hra striktná

(vid. sekciu 1.1.2) – vynucuje dodržiavanie pravidiel a zvyrazňuje možné ťahy. Platforma *Board Game Arena* je taktiež striktná, ale na rozdiel od našej aplikácie neumožňuje vytváranie ľubovoľných vlastných hier.

2.2 Limity

Používatelia pri vytváraní pravidiel majú len obmedzený počet možností pre nastavenia, ktoré nie sú číselné — napríklad nie je možné mať v svojej hre akciu, ktorá nie je jedna z akcií *Pohyb*, *Pohyb všetkých*, *Posunutie stĺpca/riadku dielikov*, *Pridanie dielika na plochu* a *Vymenenie dielika z plochy*. Kombináciu rôznych nastavení môžu používatelia vytvoriť množstvo rôznych hier, ale nie je možné pridávať nové druhy akcií, efektov dielikov, cieľov hráčov, podmienok koncov hry a podobne cez súbory s pravidlami. To je možné dosiahnuť len upravením kódu programu.

Spôsob, akým fungujú pravidlá tiež znemožňuje mať rôzne pravidlá pre rôznych hráčov. Hráči môžu začínať na rôznych miestach a mať rôzne ciele, ale všetky ostatné pravidlá sú aplikované rovnako na všetkých hráčov.

Nakoľko hranie prebieha lokálne na jednom počítači, nie je možné aby hráči mali nejaké skryté informácie. V tejto rodine hier sa to týka len dielikov, ktoré hráči v niektorých hrách držia na ruke. Tie nie je možné na jednom monitore ukázať hráčovi na ťahu bez toho, aby ich nevideli aj zvyšní hráči. Skryté informácie však nehrajú v hrách tohto typu veľkú rolu.

Nakoľko príprava hry prebieha presne podľa pravidiel načítaných zo súboru, hra až na možnosť náhodného rozmiestnenia dielikov začína vždy rovnako. Nie je teda možné, aby si napríklad na začiatku hry hráči zvolili, kde chcú začínať – začiatková pozícia hráčov je pevne daná v pravidlách.

Užívateľské rozhranie je tiež pevne dané a nie je ho možné meniť. Zmeniť, ako sa vykresľujú hráči a dieliky, alebo kde a aké informácie sa zobrazujú nie je možné bez upravenia kódu.

Existujúce implementácie platforiem na hranie spoločenských hier *Tabletop Simulator*, *Tabletopia* a *Board Game Arena* umožňujú hrať väčšie množstvo spoločenských hier, nakoľko nie sú obmedzené len na jeden ich typ. Zároveň na rozdiel od nášeho programu umožňujú hranie online.

3 Používateľská dokumentácia

Táto kapitola obsahuje návod, ako našu aplikáciu spustiť a ako ju ovládať.

3.1 Spustenie

Program je určený pre operačný systém Windows. Po stiahnutí a rozbalení príloženého priečinku `BoardgameIntepreter.zip` otvorte podpriečinok `App` a v ňom spustíte súbor `BoardgameIntepreter.exe`.

3.2 Ovládanie

3.2.1 Uživatelské rozhranie

Hlavné menu

Po spustení programu sa zobrazí hlavné menu, z ktorého je možné spustiť pripravené ukážky hier alebo spustiť vlastnú hru používateľa (viď obrázok 3.1). Konkrétna hra sa spúšťa kliknutím ľavým tlačidlom myši na biely obdĺžnik s textom „Play“ nasledovaný názvom pripravenej hry alebo s textom „Play Your Own Game“ pre spustenie používateľom vytvorenej hry. Návod ako vytvárať hry vlastnými pravidlami je v kapitole 4. Pokiaľ program narazí pri načítaní pravidiel na zle zadané pravidlá, v spodnej časti obrazovky sa zobrazí chybové hlásenie popisujúca chybu v súbore s pravidlami. Pokiaľ prebehne načítanie pravidiel bez problémov, spustí sa samotná hra.

Stav hry

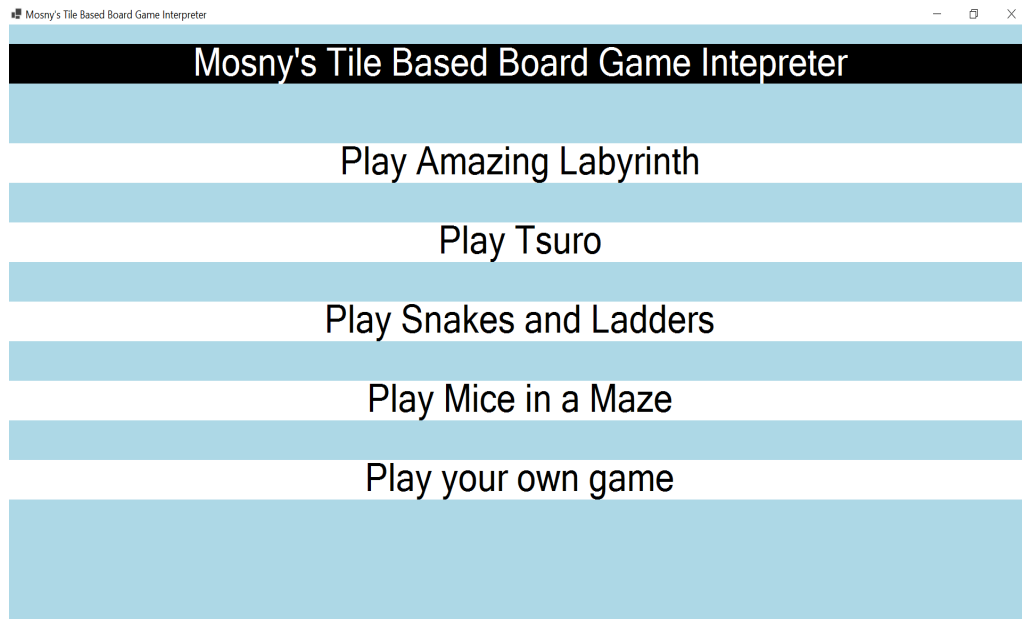
Každá hra je vykreslená v štvorcovej mriežke. Vonkajšie štvorce, teda prvé a posledné stĺpce a riadky sú používané na zobrazovanie informácií o stave hry. Zvyšok – vnútorné štvorčeky – predstavuje samotnú hraciu dosku (viď obrázok 3.2).

V ľavom hornom rohu je informácia o tom, ktorý hráč je práve na ťahu. Pod textom *Current player*: je farebný kruh vo farbe aktuálneho hráča.

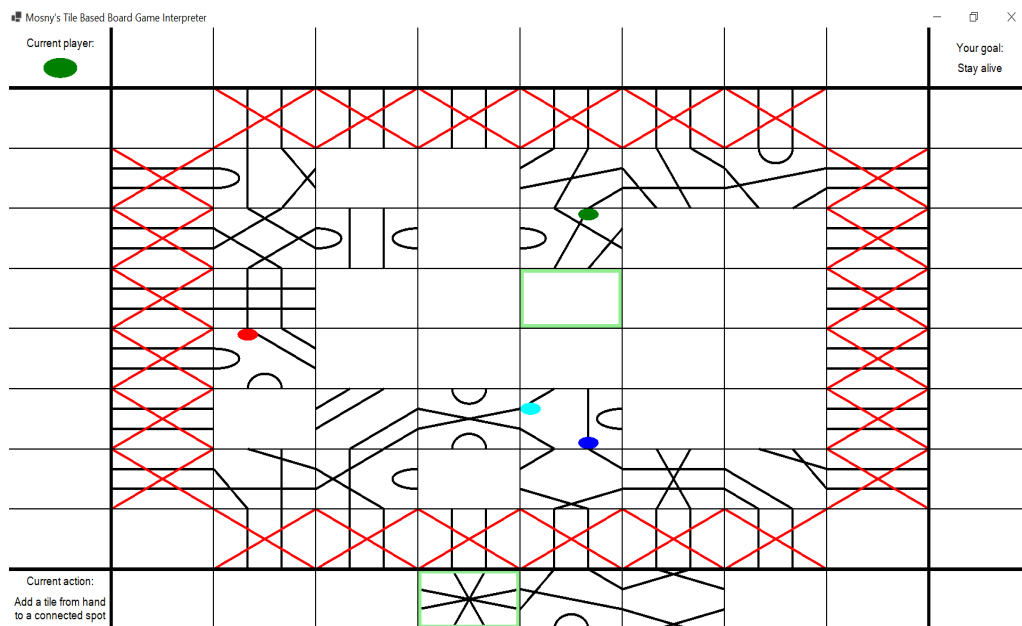
V ľavom dolnom rohu pod textom *Current action*: je informácia, ktorú akciu hráč práve vykonáva. Pokiaľ sa na danú akciu vzťahujú nejaké pravidlá (napríklad obmedzená vzdialenosť pohybu), je tu taktiež uvedená.

V pravom hornom rohu je vypísaný aktuálny cieľ hráča, pokiaľ hráč nejaký má. Cieľ hráča môže byť dostať sa na dielik na daných súradniciach (stĺpce a riadky sú číslované od nuly odhora a zľava), pozbierať určitý počet predmetov alebo sa dostať na dielik s nejakým konkrétnym predmetom. V prípade, že hráč nemá zadané špecifické ciele, je uvedený cieľ byť posledným hráčom v hre, alebo získať čo najviac bodov. Body rozhodujú o víťazovi, pokiaľ hra má nejakú podmienku na ukončenie hry.

Pokiaľ nejaká hra využíva kôpku nepoužitých dielikov (napríklad sa z nej pridávajú dieliky na plochu), vrchný dielik z kôpky je zobrazený v pravom dolnom rohu.



Obrázek 3.1 Hlavné menu.



Obrázek 3.2 Grafické používateľské rozhranie hry.

Ak má hráč nejaké dieliky na ruke a aktuálna akcia ich využíva, sú zobrazené v spodnom riadku.

Hracia doska

Cesty na dielikoch sú zobrazované ako jednoduché čiary spájajúce rôzne strany dielika. Pokiaľ sa 2 cesty križujú, neznamená to, že sú prepojené. Ak má teda dielik jednou čiarou (cestou) spojenú hornú a spodnú stranu a druhou cestou spojenú ľavú a pravú stranu, nie je možné sa dostať napríklad z ľavej strany na hornú. Dielik, ktorý dovoľuje prechádzať medzi ľubovoľnými stranami má priame čiary spájajúce každé 2 strany (viď obrázok 3.3).

Pokiaľ je dielik otočený zadnou stranou hore, je vykreslený ako šedý štvorec. Pokiaľ je miesto na ploche prázdne, je vykreslené ako prázdny biely štvorec.

Na dielikoch, ktoré na sebe majú nejaké predmety sú vypísané ich názvy a prípadne ich množstvo, pokiaľ je na dieliku viac predmetov toho istého druhu.

Efekty, ktoré majú dieliky sú vykresľované rôzne, podľa druhu efektu. Pokiaľ dielik posielá hráčov na iný dielik, program vykreslí šípku od tohto dielika po cieľový dielik. Dieliky, ktoré eliminujú hráča po vstupe na ne sú prečiarknuté červeným krížom. Dieliky, na ktoré hráči nosia predmety (väčšinou za to získavajú body) majú na sebe text *Collected*. Po prinesení predmetov na takýto dielik na ňom pribudne text hovoriaci, aké predmety a v akom počte boli prinesené. Efekty, ktoré posielajú hráča na jeho počiatočnú pozíciu, spôsobujú vynechanie kola a poslanie jeho predmetov preč sú všetky vykreslené ako modrý kríž cez dielik, nakoľko sú používané spolu na jednom dieliku.

Hráči sú vykreslení ako farebné kruhy na dielikoch. Nie sú vykresľovaní na strede cesty, ale vždy sa nachádzajú na jednej jej strane. Toto má význam, pokiaľ hráči nemajú dovolené sa vracat a musia ísť po ceste stále dopredu — značí to, ktorým smerom hráč ide. Hráč sa nachádza vždy na tej strane cesty a dielika, z ktorej prišiel. Pohybom dopredu prejde celú cestu na dieliku a vyjde z neho na opačnej strane ako začal svoj pohyb. Pokiaľ sa teda hráč pohybuje po ceste cez dieliky zľava doprava, svoj pohyb vždy ukončí na ľavej strane dielika.

Pokiaľ hráč má nejaký predmet, je zobrazený na jeho farebnom kruhu ako menší čierny kruh v jeho strede.

3.2.2 Vykonávanie akcií

Pohyb

Pokiaľ je aktuálna akcia *Pohyb* (*Move*), sú zeleným okrajom zvýraznené dieliky, na ktoré sa hráč môže pohnúť. Kliknutím ľavého tlačidla myši na takýto dielik sa hráč naň posunie. Pokiaľ žiaden dielik nie je zvýraznený, hráč nemá žiadne možnosti na pohyb a kliknutie ľavým tlačidlom myši na ľubovoľné miesto spustí ďalšiu akciu v poradí.

Pohyb všetkých

Akcia *Pohyb všetkých* (*Move all*) je automaticky vykonaná po kliknutí ľavým tlačidlom myši kamkoľvek. Všetci hráči budú posunutí smerom dopredu na koniec cesty, na ktorej sa nachádzajú.

Posunutie riadka alebo stĺpca

Posunutie riadka alebo stĺpca (Push row or column) hráč aktivuje kliknutím ľavého tlačidla myši na políčko so šípkou vedľa riadku alebo stĺpca, ktorý chce posunúť. Šípka ukazuje smer, ktorým sa stĺpec alebo riadok posunie. Pokiaľ pri stĺpci alebo riadku šípka nie je, nie je ho možné posunúť. Ak je v pravom dolnom rohu (mimo hracej plochy) zobrazený dielik, ide o dielik na vrchu kopy, ktorý bude použitý na zaplnenie prázdneho miesta vzniknutého po posúvaní. Tento dielik je pred posúvaním možné otáčať kliknutím naň. Vysunutý dielik sa stane novým dielikom na vrchu kopy.

Ak vrch kopy nie je zobrazený, prázdne miesto bude zaplnené práve vysunutým dielikom.

Pridanie dielika

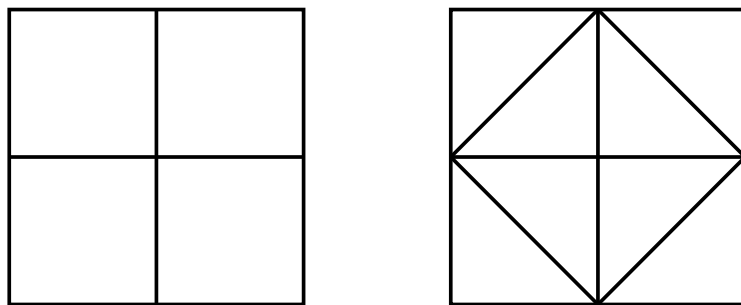
Pridanie dielika (Add tile) umožňuje hráčovi pridať dielik na prázdne miesto na hracej doske. Pokiaľ je v pravom dolnom rohu obrazovky zobrazený dielik z vrchu kopy, bude hráč pridávať tento dielik. Kliknutí ľavým tlačidlom myši naň je možné ho ešte pred pridaním otočiť do požadovanej orientácie.

Pokiaľ sa nepridávajú dieliky z kopy, ale z ruky hráča, dieliky ktoré hráč drží by sa zobrazia v spodnom riadku. Dielik na ruke, ktorý je aktuálne vybraný na pridávanie je zvýraznený zeleným okrajom. Kliknutie ľavého tlačidla myši na iný dielik na ruke ho vyberie. Po opätovnom kliknutí na už vybraný dielik z ruky je ho možné otáčať.

Miesta, na ktoré je možné dielik z kôpky alebo z ruky pridať, sú zvýraznené zeleným okrajom. Po kliknutí na takéto miesto ľavým tlačidlom myši sa toto miesto vybraný dielik pridá.

Výmena dielika

Pre vykonanie akcie *Výmeny dielika (Exchange tile)* musí hráč vymeniť dielik z vrchu kôpky alebo z ruky za dielik na ploche. Tak ako pri pridávaní, ak je zobrazený dielik vpravo dole, vymieňa sa za dielik z plochy za dielik z kopy. Ak je zobrazená v spodnom riadku ruka hráča, vymení sa dielik z plochy za dielik z ruky. Dieliky z ruky sa vyberajú kliknutím na ne (vybraný je zvýraznený). Otáčať dielik z kôpky alebo vybraný dielik z ruky je možné kliknutím naň ľavým tlačidlom myši. Kliknutie na jeden z dielikov na hracej ploche, ktorý je na okrajoch zvýraznený zelenou farbou ho vymení. Pokiaľ bol dielik na ploche otočený zadnou stranou hore, po vymenení sa odhalí a je otočený prednou stranou hore.



Obrázek 3.3 Dílek s prepojenými protilahlými stranami (vľavo) a dílek s prepojenými všetkými stranami (vpravo).

4 Vytváranie hier

V tejto kapitole vysvetlíme, ako vytvárať vlastné hry. Popíšeme formát JSON, v ktorom sú súbory, do ktorých sa pravidlá zadávajú. Tiež uvedieme, akú majú pravidlá štruktúru a aké možnosti pri ich vytváraní používateľa majú.

Vlastnú hru môžu používatelia vytvoriť špecifikovaním jej pravidiel v súbore `rules.json` v podpriechniku `App` stiahnutej zložky `BoardgameInterpreter.zip`. Meno súboru nie je možné meniť.

4.1 Formát Json

Formát `Json` [4] je kolekciou párov zložených z *názvu* a *hodnoty*. Súbor začína symbolom `{` a je ukončený symbolom `}`. *Názov* páru je textový reťazec uvedený v úvodzovkách. Za ním nasleduje dvojbodka a *hodnota* priradená k tomuto názvu. Páry sú od seba oddelené čiarkou (posledný pár teda za sebou čiarku už nemá). Hodnota môže byť rôznych typov: textový reťazec, číslo, boolean (pravdivostná hodnota), zoznam alebo objekt.

Hodnota typu textového reťazca je rovnako ako názov text v úvodzovkách, napríklad `"name": "Dann"` .

Čísla sa uvádzajú priamočiara bez špeciálnych symbolov, napríklad `"age": 21` .

Boolean (pravdivostná) hodnota sa uvádza ako text *true* alebo *false* bez úvodzoviek, napríklad `"alive": true` .

Zoznam hodnôt je viacero hodnôt (reťazec, číslo, boolean, ...) oddelených čiarkami v hranatých zátvorkách, napr. `"cities": ["Dunajská Lužná", "Praha"]` .

Objekt je kolekcia párov *názvu* a *hodnoty* oddelených čiarkami v zložených zátvorkách, napríklad `"author": {"name": "Dann", "age": 21}` .

4.2 Štruktúra a možnosti pravidiel

Pokiaľ je niekde v pravidlách nutné špecifikovať súradnice nejakého dielika, teda číslo stĺpca a riadku, v ktorom sa nachádza, začínajú tieto čísla od nuly. Stĺpce sa číslujú zľava doprava, riadky odhora dole. Stĺpec najviac vľavo má teda číslo 0, takisto ako riadok úplne hore.

Pravidlá špecifikujú stranu dielika jedným veľkým písmenom. Písmeno `U` označuje hornú stranu dielika, písmeno `D` dolnú stranu dielika, písmeno `L` ľavú a písmeno `R` pravú stranu dielika. Cesty na strane dielika sa označujú číslami, ktoré rovnako ako čísla stĺpcov a riadkov začínajú od nuly. Cesty na stranách sa číslujú v smere hodinových ručičiek. Napríklad, ak sú na stranách dielikov dve cesty, tak cesta na hornej strane vľavo má číslo 0 a cesta vpravo má číslo 1. Na spodnej strane má cesta vľavo číslo 1 a cesta vpravo má číslo 0.

Pravidlá sa určujú nasledujúcimi poľami. Niektoré je potrebné vždy v pravidlách špecifikovať, iné je možné vynechať – tie pravidlá potom budú automaticky nastavené na ich predvolenú hodnotu.

Pole `turn_structure` je zoznamom textových reťazcov, ktorý špecifikuje, aké akcie a v akom poradí hráč počas svojho ťahu vykoná. Akcie sú vykonávané v takom poradí, v akom sú zadané v zozname. Akcia pohybu sa zadáva ako reťazec `move`, akcia pohyb všetkých ako `moveAll`, akcia posunutia stĺpca alebo riadka ako `push`, akcia pridania dieliku ako `add` a akcia vymenenia dieliku sa zadáva ako `exchange`. Pokiaľ sa teda hráč v svojom ťahu najprv posunie a potom pridá na plochu dielik, v pravidlách to bude zadané ako `"turn_structure": ["move", "add"]`. Toto pole je nutné špecifikovať.

Pole `movement_type` má typ textového reťazca a určuje, ako sa môžu hráči pohybovať. Možnosť `free` dovoľuje hráčom sa posunúť na ľubovoľný dielik, ktorý je s hráčom prepojený cestou. Možnosť `forced` núti hráča sa posunúť až na koniec (prípadne jeden z koncov) cesty, na ktorej sa nachádza. Posledná možnosť `dice` znamená, že sa pohyb hráčov bude riadiť hodom kockou – hráči sa musia posunúť presne o taký počet dielikov aký hodili na kocke. Najmenšie a najväčšie číslo, ktoré v tomto prípade môže na kocke padnúť určujú polia `min_move_distance` a `max_move_distance`. Pokiaľ užívateľ toto pole vynechá, automaticky sa nastaví na možnosť `free`.

Číselné pole `min_move_distance` špecifikuje, o akú minimálnu vzdialenosť (počet dielikov) sa hráč musí posunúť. Hodnota 1 napríklad znamená, že hráči pri pohybe nemôžu zostať stáť na mieste a musia sa posunúť aspoň o 1 dielik. Predvolená hodnota je 0. Pri nastavení `movement_type` na možnosť `dice` toto pole určuje najmenšie číslo, ktoré môže na kocke padnúť.

Číselné pole `max_move_distance` dovoľuje obmedziť maximálnu vzdialenosť (počet dielikov), o ktorú sa môžu hráči posúvať. Pohyb, ktorý nemá maximálnu vzdialenosť sa špecifikuje nulou alebo ľubovoľným záporným číslom. Neobmedzený pohyb je predvolená možnosť, ktorá je nastavená, ak používateľ toto pole neuvedie. Pri nastavení `movement_type` na možnosť `dice` toto pole určuje najväčšie číslo, ktoré môže na kocke padnúť.

Pole `only_forward_movement` je typu boolean. Hodnota `true` zapne, že hráči musia pri pohybe pokračovať v smere, v ktorom išli doteraz – nemôžu sa otočiť a vrátiť sa. Predvolená hodnota je `false`.

Pole `cycle_pushed_out_tiles` určuje, ktorým dielikom sa zaplní medzera vzniknutá pri posúvaní riadkov a stĺpcov. Predvolená hodnota `true` znamená, že na zaplnenie je použitý práve vysunutý dielik. Hodnotou `false` je určené, že prázdne miesto je zaplnené dielikom z vrchu kopy (a vysunutý dielik sa dá namiesto neho na vrch kopy).

Pole `adding_from` je textové a má dve možnosti: `pile` a `hand`. Hodnota `pile`, ktorá je predvolená, znamená, že pri akcii pridávania dieliku na hraciu dosku sa pridáva dielik z vrchu kopy. Hodnota `hand` znamená, že hráči pridávajú na plochu dieliky z ruky.

Pole `adding_mode` je tiež textové s dvomi možnosťami. Predvolená mož-

nosť `anywhere` dovoľuje pri akcii pridávania hráčovi pridať dielik na ľubovoľné prázdne miesto na doske. Možnosť `connected` dovoľuje hráčovi pridať dielik len na prázdne miesta, ktoré sú spojené s hráčom – končí na nich cesta, na ktorej sa hráč nachádza.

Textové pole `exchange_with` je analogické fungovaním k polu `adding_from`, ale popisuje akciu vymieňania dieliku. Má rovnaké dve možnosti: predvolenú `pile` a `hand`. Pri akcii vymieňania dielika možnosť `pile` znamená, že sa dielik z plochy zamieňa za dielik z vrchu kopy. Pri možnosti `hand` hráč zamieňa dielik z plochy za dielik zo svojej ruky.

Textové pole `exchanging_mode` podobne ako pole `adding_mode` určuje, ktoré dieliky z plochy pri akcii vymieňania môže hráč vymeniť. Ak je zvolená možnosť `anywhere`, je možné vymeniť ľubovoľný dielik na ploche (okrem dielikov zakázaných v poli `not_exchangeable_tiles`). Možnosť `connected` dovoľuje vymeniť iba dieliky, ktoré ležia na ceste, na ktorej sa hráč nachádza. Predvolená možnosť je `anywhere`.

Boolean pole `exchange_only_flipped` dovoľuje možnosťou `true` zapnúť, že hráči môžu vymeniť z hracej dosky iba dieliky, ktoré sú otočené zadnou stranou hore. Predvolená možnosť `false` dovoľuje vymieňať ľubovoľne otočené dieliky.

Číselné pole `max_hand_size` špecifikuje, koľko najviac dielikov môže hráč držať na ruke. Pokiaľ v hre hráči nemajú dieliky na ruke, je to špecifikované hodnotou 0 alebo záporným číslom. Hodnota 0 je predvolená.

Číselným polom `max_player_items` sa určuje, aký je najväčší počet predmetov, ktoré hráči môžu mať. Predvolená hodnota je nula, znamenajúca, že hráč predmety z dielikov nemôže brať.

Pokiaľ v hre hráči zbierajú predmety z dielikov, pole `items_picked_up` určuje, koľko predmetov jedného druhu môže hráč z dielika naraz zobrať. Ak hráč môže zobrať všetky, je to nastavené záporným číslom. Táto možnosť je predvolená.

Pole `random_goal_pool` je zoznamom textových reťazcov, ktoré popisujú ciele, ktoré sú náhodne priradené hráčom. Predvolená hodnota je prázdny zoznam.

Cieľ dostať sa na nejaký dielik sa zadáva kľúčovým slovom `reach`, za ktorým sú dve čísla oddelené medzerou – tie určujú súradnice stĺpca a riadka, na ktorý sa má hráč dostať. Cieľ dostať sa do napríklad ľavého horného rohu by sa teda zadal ako `"reach 0 0"`.

Cieľ získať nejaký počet jedného predmetu je zadaný kľúčovým slovom `collect` nasledovaným číselne zadaným množstvom a názvom predmetu, ktorý má hráč získať. Množstvo aj názov sú oddelené medzerami. Cieľ získať päť predmetov s názvom `coin` (minca) by sa zadal ako `"collect 5 coin"`.

Cieľ dostať sa na dielik s nejakým predmetom je popísaný kľúčovým slovom `find` a medzerou oddeleným názvom predmetu.

Zoznam textových reťazcov `end_of_game conditions` dovoľuje určiť podm-

enky konca hry. Hra skončí akonáhle je aspoň jedna z nich naplnená. Jediným typom podmienky je minútie sa nejakého predmetu, teda keď sa nejaký predmet už nenachádza voľne na dielikoch ani ho nemajú žiadni hráči. Daný predmet sa v takom prípade nachádza už len na špeciálnych dielikoch s efektom na zbieranie predmetov. Táto podmienka sa zadáva kľúčovým slovom `no` nasledovaným medzerou a názvom daného predmetu.

Boolean pole `fill_unspecified_tiles` určuje, či používateľom nešpecifikované prázdne miesta na ploche majú zostať prázdne alebo majú byť zaplnené náhodnými nepoužitými dielikmi. Predvolená hodnota `false` ich nechá prázdne, hodnota `true` ich náhodne zaplní.

Pole `board` je objekt obsahujúci dvojicu popisujúce samotnú hraciu plochu.

Číselné polia `width` a `height` objektu `board` určujú rozmery hracej dosky – jej šírku a výšku v počte dielikov. Tieto polia je nutné zahrnúť v pravidlách.

Polia `unpushable_columns` a `unpushable_rows` objektu `board` sú zoznamy čísiel, ktoré určujú, ktoré stĺpce a riadky hráči nemôžu posúvať. Zakázanie posúvania prvého a tretieho riadku by sa teda zadalo ako `"unpushable_rows": [0,2]`. Predvolená hodnota oboch polí je prázdny zoznam, teda dovolenie posúvania všetkých riadkov a stĺpcov.

Pole `not_exchangeable_tiles` objektu `board` je zoznam textových reťazcov, ktorý určuje, ktoré dieliky nie je možné hráčmi vymieňať. Dieliky sú špecifikované ich súradnicami – číslom stĺpca a riadku, v ktorom sa nachádzajú. Číslo stĺpca a riadku každého dielika sú spolu v jednej textovej položke zoznamu a sú od seba oddelené čiarkou. Ak teda nie je dovolené vymeniť dielik v ľavom hornom rohu plochy a dielik napravo od neho, je to v pravidlách zapísané ako `["0,0", "1,0"]`. Ak je toto pole vynechané, je dovolené vymeniť každý dielik.

Číselným polom `paths_on_tile_side` objektu `board` sa určuje, koľko rôznych neprepojených ciest môže vychádzať z jednej strany dielika. Predvolená hodnota je 1, teda iba jedna cesta na každej strane.

Pole objektov `tiles` popisuje, aké všetky dieliky sa v hre budú používať. Každý jednotlivý objekt v zozname popisuje jeden dielik. Toto pole je nutné zahrnúť v pravidlách.

Pole `connections` objektu v zozname `tiles` je zoznam textových reťazcov popisujúcich ktoré časti ktorých strán dieliku sú spojené cestičkami. Body na stranách dielikov spájané cestičkami sú označené písmenom zodpovedajúcemu strane a číslom označujúcim miesto na danej strane. Napríklad, ak sú na stranách dielikov dve cesty, tak cesta na hornej strane vľavo je označená `U0` a cesta vpravo je označená `U1`. Na spodnej strane je cesta vľavo označená `D1` a cesta vpravo je označená `D0`. Každá položka v zozname popisuje jednu cestu na dieliku – body na stranách, ktoré spája. Označenia bodov sú oddelené čiarkami. Ak sú napríklad na dieliku dve cesty, jedna, ktorá spája ľavú, hornú a dolnú stranu a druhá cesta,

ktorá (na iných miestach) spája hornú a dolnú stranu, pole `connections` bude vyzeráť takto `["L1,U0,D1", "U10,D0"]` (viď obrázok 4.1). Predvolená hodnota je prázdny zoznam – dielik bez cestičiek.

Pole `location` objektu v zozname `tiles` určuje pozíciu na hracej ploche, na ktorej sa na začiatku hry dielik nachádza. Je to číselný zoznam, ktorý by mal obsahovať dve čísla – číslo stĺpcu a číslo riadku, v tomto poradí. Napríklad pre dielik, ktorý začína v druhom stĺpci a treťom riadku vyzerá pole `location` takto: `[1,2]`. Pokiaľ používateľ nešpecifikuje pozíciu, daný dielik bude v kôpke alebo bude priradený na náhodné prázdne mieste na ploche, ak má pole `fill_unspecified_files` hodnotu `true`.

Boolean pole `face_down` objektu v zozname `tiles` umožňuje špecifikovať, ktorou stranou má byť dielik na začiatku hry otočený hore. Predvolená hodnota `false` otočí dielik prednou stranou hore, hodnota `true` ho otočí zadnou stranou hore.

Boolean pole `movement_ending` objektu v zozname `tiles` dovoľuje možnosťou `true` z dielika spraviť dielik, ktorý zastavuje pohyb hráčov. To znamená, že pokiaľ hráč prechádza týmto dielikom, musí na ňom zastaviť a ukončiť pohyb – a to aj v prípade, že by iné pravidlá hráčovi dovoľovali alebo prikazovali ísť ďalej. Predvolená hodnota `false` dieliku tento efekt nepridá.

Boolean pole `can_turn_around` objektu v zozname `tiles` umožňuje možnosťou `true` hráčom sa na danom dieliku otočiť, aj keď je otáčanie v poli `only_forward_movement` zakázané. Teda ak hráč začína svoj pohyb na takomto dieliku, môže z neho odísť ľubovoľným smerom, aj keby ho pravidlo `only_forward_movement` nútilo pokračovať v pohybe doterajším smerom. Predvolená hodnota je `false`.

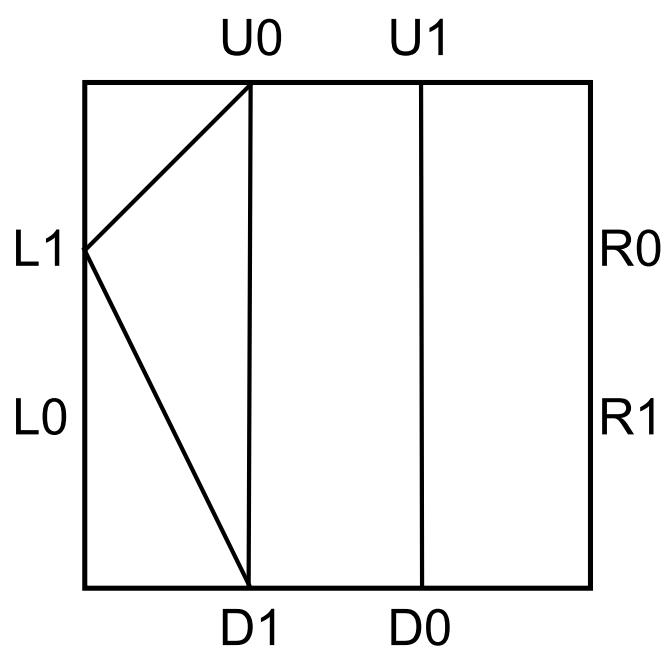
Zoznam textových reťazcov `items` objektu v zozname `tiles` určuje, aké predmety sa na dieliku budú nachádzať. Položky zoznamu sú názvy predmetov. Pokiaľ sa na dieliku nachádza viac rovnakých predmetov, za názvom predmetu je možné medzerou oddeleným číslom tento počet špecifikovať, napríklad `"items": ["coin 3", "diamond"]`. Predvolená hodnota je prázdny zoznam – žiadne predmety na dieliku.

Pole `effects` objektu v zozname `tiles` je zoznam textových reťazcov udávajúcich, aké špeciálne efekty dielik má – veci, ktoré sa stanú, keď na nich ukončí pohyb nejaký hráč. Predvolená hodnota je prázdny zoznam – dielik bez špeciálnych efektov.

Efekt poslania hráča na iný dielik je zadaný kľúčovým slovom `move`. Nasledujú medzerami oddelené čísla stĺpca a riadku, do ktorých bude hráč poslaný. Za nimi je opäť medzerami oddelené písmeno a číslo určujúce, na ktorú cestu na ktorej strane dielika bude hráč poslaný. Napríklad poslanie hráča na dielik v ľavom hornom rohu, konkrétne na jeho dolnú cestu na pravej strane je zadané reťazcom `"move 0 0 D 1"`.

Efekt vyradenia hráča z hry je zadaný reťazcom `eliminate`.

Dielik, ktorý zbiera a uchováva predmety od hráčov má efekt zadaný



Obrázek 4.1 Cesty zodpovedajúce poľu connections: ["L1,U0,D1", "U10,D0"].

reťazcom `collect`.

Efekt, ktorý hráčovi zoberie predmety a vráti ich na nejaký dielik je zadaný kľúčovým slovom `send` nasledovaný medzerami oddelenými číslami stĺpca a riadka dieliku, na ktorý budú predmety vrátené.

Efekt vrátenia hráča na jeho počiatočnú pozíciu je zadaný reťazcom `return`.

Efekt vynechania nasledujúcich ťahov hráča je zadaný kľúčovým slovom `skip` nasledovaným medzerou oddeleným číslom určujúcim, koľko nasledujúcich kôl daný hráč vynechá a nemôže v nich hrať.

Číselné pole `amount` objektu v zozname `tiles` umožňuje určiť koľko dielikov tohto typu – s rovnakými cestami, predmetmi a efektami v hre bude. Predvolená hodnota je 1. Pokiaľ je dielikov jedného typu viac, nie je im možné v poli `location` priradiť pozíciu, keďže viacero dielikov nemôže byť na rovnakom mieste.

V pravidlách je nutné zahrnúť pole `players`, ktoré je zoznamom objektov predstavujúcich hráčov.

Číselné pole `starting_x` objektu v zozname `players` určuje číslo stĺpca, v ktorom je dielik, na ktorom hráč začína. Toto pole nie je možné vynechať.

Číselné pole `starting_y` objektu v zozname `players` určuje číslo riadku, v ktorom je dielik, na ktorom hráč začína. Pole nie je možné vynechať.

Textové pole `starting_side` objektu v zozname `players` určuje jedným veľkým písmenom, na ktorej strane dielika hráč začína. Toto pole je potrebné zahrnúť.

Číselné pole `starting_path_index` objektu v zozname `players` určuje číslo cesty na už špecifikovanej strane dieliku, na ktorej hráč začína.. Toto pole tiež nie je možné vynechať.

Pole `goals` objektu v zozname `players` je textový zoznam cieľov, ktoré sú unikátne pre daného hráča. Ciele hráč plní v takom poradí, v akom sú zadané v tomto zozname. Ciele sa zadávajú tak ako je popísané pri poli `random_goal_pools`. Ako prvok zoznamu je možné použiť aj kľúčové slovo `random` nasledované medzerou oddeleným číslom. Takýto textový prvok bude nahradený príslušným počtom náhodne vybraných cieľov zo zoznamu `random_goal_pools`. Predvolená hodnota je prázdny zoznam – hráč nemá žiadne unikátne ciele.

Pole `end_of_game_scoring` objektu v zozname `players` je zoznam textových reťazcov, ktoré popisujú, za ktoré predmety hráč získava body, pokiaľ je hra ukončená jednou z podmienok v poli `end_of_game_conditions`. Každý prvok v zozname musí začínať kľúčovým slovom `items` a byť nasledovaný medzerou oddeleným názvom predmetu. Takýto reťazec hovorí, že hráč získava body za každý predmet daného typu, ktorý na konci má. Je možné špecifikovať získavanie bodov aj za predmety na dielikoch slúžiacich na odkladanie a uchovávanie predmetov (dieliky s efektom `collect`). To sa zadáva tým, že za kľúčovým slovom `items` a názvom sú pridané dve medzerou oddelené čísla – súradnice dieliku, ktorého predmety sú zarátané do skóre hráča. Predvolená hodnota tohto poľa je prázdny

zoznam – hráči nezískavajú body a vyhrávajú splnením svojich cieľom alebo eliminovaním všetkých protihráčov.

5 Riešené problémy a urobené rozhodnutia

V tejto kapitole uvedieme, aké rozhodnutia sme pri vypracovávaní práce urobili – aké možnosti sme zvažovali a aké boli dôvody pre naše rozhodnutia. Tiež popíšeme, ako sme vyriešili problémy, na ktoré sme pri práci narazili.

5.1 Multiplayer

Hry, ktoré sa dajú hrať *online* majú oproti hrám hraným lokálne na jednom počítači mnoho výhod, tak ako sme to popísali v sekcii 1.1.1. Implementácia hrania *online* je však značne náročnejšia oproti implementácii hrania lokálne. Nakoľko hlavným zameraním tejto bakalárskej práce je vytváranie vlastných hier, ktoré je možné aj lokálne, implementovanie hrania *online* bolo nad rámec tejto práce. Hry v našej aplikácii sa teda hrajú lokálne.

5.2 Konfiguračné súbory s pravidlami

Pri voľbe toho, ako budú vyzerat a fungovat súbory s pravidlami sme mali viacero možností. Základným rozhodnutím bolo, či sa pravidlá budú zadavat vyplňaním vopred definovaných polí v nejakom dátovom formáte, alebo sa budú zadavat v nejakom nami vymyslenom pseudo-programovacom jazyku. Pseudo-programovací jazyk by dával používateľom viac možností pri vytváraní pravidiel – neboli by obmedzení vopred danou štruktúrou nejakého dátového formátu. Tento spôsob zadávania pravidiel je však pre používateľov komplikovanejší. Nakoľko naším cieľom je poskytnúť možnosť vytvárat vlastné hry bežným používateľom bez akýchkoľvek programátorských znalostí, zvolili sme spôsob zadávania pravidiel vyplňaním dátového súboru, čo je jednoduchšie a zrozumiteľnejšie. V takomto súbore si používateľ pri každom z pravidiel vyberie jednu z jeho možností. Keďže sme obmedzili náš program len na jednu kategóriu hier, neprichádzajú používatelia oproti zadávaniu pseudo-programovacím jazykom o veľa možností. Kombináciou rôznych možností pri každom z pravidiel je stále možné vytvorit širokú škálu hier tohto typu.

Pri vyberaní vhodného dátového formátu sme brali do úvahy najmä jeho jednoduchosť a čitateľnosť pre používateľov. Používateľsky prívetivé formáty, ktoré sme zvažovali boli **CSV**, **JSON** a **XML**. Zároveň sme požadovali, aby zvolený formát mal štruktúru, v ktorej by sa jednoducho písali pravidlá, ktorých formu sme si predstavovali ako názov pravidla nasledovaný jednou z jeho možností. Zároveň sme chceli, aby formát podporoval vytvorenie nejakých objektov alebo štruktúr, čo by nám uľahčilo popisovanie dielikov, z ktorých je zložená hracia plocha.

Formát **CSV** [5] je veľmi jednoduchý, nakoľko ide len o hodnoty oddelené čiarkami. Tento formát je však primárne určený pre tabulkové dáta. To ho nerobí úplne vhodným pre naše účely, nakoľko naše pravidlá sú skôr definované ako

dvojice. Taktiež nie je v tomto formáte možné priamočiaro definovať objekty alebo štruktúry.

Formát JSON [6] je oproti formátu CSV zložitejší, ale je stále jednoducho čitateľný. Hodnoty sú v tomto formáte vo forme dvojíc, čo zodpovedalo našej predstave o štruktúre pravidiel. Zároveň podporuje vytváranie objektov, ktoré by mohli byť použité na jednoduchšie zadávanie informácií o jednotlivých dielikoch.

Formát XML [7], je podobne ako formát JSON založený na dvojiciach a tiež obsahuje podštruktúry, takže tiež zodpovedal naším požiadavkám. Oproti formátu JSON nám však prišiel ako menej čitateľný pre používateľov, pretože každá hodnota musí okrem svojho mena obsahovať aj koncovú značku, v ktorej je meno hodnoty znova zopakované. Pre porovnanie, jednoduché pole s menom v JSON súbore vyzerá ako "name": "Dann" a to isté pole v XML súbore vyzerá ako <name>Dann</name>. Rozhodli sme sa teda pre pravidlá používať súbory vo formáte JSON.

5.3 Reprezentácia ciest na dielikoch

Každý dielik obsahuje objekty reprezentujúce jednotlivé cesty. Jeden takýto objekt obsahuje informácie o tom, ktoré strany a na ktorých miestach cesta spája. Priamočiary spôsob ako všetky cesty z jedného dieliku uložiť je v jednom zozname za sebou. Avšak tento spôsob ukladania ciest komplikuje rozhodovanie o tom, či sú dva dieliky vedľa seba prepojené cestou. Ak sa chce hráč posunúť po ceste smerujúcej na susedný dielik, musí sa manuálne prehľadať celý zoznam ciest susedného dieliku, či obsahuje cestu, ktorá nadväzuje na cestu s hráčom.

Aby sme toto vyhľadávanie nadväzujúcich ciest uľahčili, rozhodli sme sa jednotlivé cesty uložiť do štruktúry podľa toho, ktoré strany spájajú. Dielik obsahuje štyri zoznamy ciest, každý zodpovedajúci cestám spájajúcich jednu stranu. Cesty sú uložené za sebou podľa poradia bodov na strane, ktoré prepájajú. Pri použití takéhoto ukladania ciest je rozhodnutie o prepojení susedných dielikov jednoduché – vieme presne, kde by mala byť uložená cesta susedného dieliku, ktorá by nadväzovala na cestu na dieliku s hráčom. Následne stačí len skontrolovať, či cesta na danom mieste existuje. Keďže však cesty spájajú viaceré strany dieliku, nie je zrejmé, v zozname ktorej strany by sa mala cesta nachádzať. Riešením je uložiť kópiu objektu cesty na príslušné miesta do zoznamu každej strany, ktoré spája. Oproti implementácii jedným zoznamom je teda táto implementácia pamätovo náročnejšia, ale značne uľahčuje hľadanie spojení medzi dielikmi, čo náš program často potrebuje vyhodnotiť.

5.4 Otáčanie dielikov

Pri otáčaní dielika sme zvažovali dve možnosti implementácie. Prvou možnosťou bolo nemeniť samotné dáta o cestách na dieliku a len si udržiavať záznam o tom, ako je dielik práve otočený. Táto možnosť robí samotné otáčanie triviálnym, ale značne komplikuje kód hľadania ciest medzi dielikmi. Pri tejto implementácii by sa pri každom rozhodovaní o prepojení dielikov muselo vypočítať, ktorými stranami sa dva susedné dieliky vlastne dotýkajú. Druhá možnosť implementá-

cie, ktorú sme zvolili my, je pri otáčaní navzájom vymeniť záznamy o cestách pre jednotlivé strany. To znamená, že každý zo štyroch zoznamov zodpovedá vždy rovnakej strane. Toto síce robí otáčanie zložitejšie, ale nakoľko sa vyhľadáva cesta oveľa častejšie ako sa otáča dielikmi, považujeme tento prístup za lepší. Zároveň to zjednodušuje kód a robí ho viac čitateľným, pretože je na prvý pohľad jasné, ktorý zoznam ciest zodpovedá ktorej strane.

5.5 Číslovanie ciest

Pri ukladaní ciest na určité miesta v štruktúrach dieliku bolo potrebné určiť, v akom poradí sa budú cesty na jednej strane číslovať. Zvolili sme číslovanie v smere hodinových ručičiek. Na hornej strane sa teda čísluje od cesty vľavo, na pravej strane od hornej cesty, na spodnej strane sa čísluje od cesty napravo a na ľavej strane od cesty dole. Druhá alternatíva číslovania bola na hornej a dolnej (respektíve ľavej a pravej) strane určiť rovnaký smer, v ktorom sa bude číslovať. Napríklad by sa aj na hornej aj na dolnej strane číslovalo zľava doprava. To by malo výhodu, že priame cesty, napríklad z „prava“ hornej strany do „prava“ dolnej strany by boli na oboch koncoch označené rovnakým číslom, čo by bolo pre používateľov viac intuitívne. Rozhodli sme sa pre číslovanie v smere hodinových ručičiek najmä preto, že značne uľahčuje našu implementáciu otáčania dielikov – po vymenení ciest na rôznych stranách nie je potrebné meniť ich čísla, smer v ktorom sa čísluje zostáva zachovaný.

5.6 Vyradenie hráčov z hry

V našom programe sú objekty reprezentujúce hráčov uložené spolu v jednom zozname v poradí, v akom sa striedajú na ťahoch. Pokiaľ je hráč eliminovaný z hry, priamočiare riešenie ako to reprezentovať je tohto hráča zo zoznamu hráčov odstrániť. Po odstránení nejakého hráča zo zoznamu však akákoľvek informácia obsahujúca čísla hráčov prestáva byť aktuálna, pretože po odstránení hráča zo zoznamu sa hráčom, ktorý išli po ňom zmení ich číselné označenie. Napríklad po odstránení prvého hráča sa z druhého hráča stane nový prvý hráč. Rozhodli sme sa preto eliminovanie hráča reprezentovať tak, že si o každom hráčovi explicitne pamätáme, či je ešte v hre. Pri eliminácii hráča sa teda len táto informácia o ňom zmení, ale samotný hráč v zozname zostáva. To zároveň poskytuje priestor pre budúce rozšírenia, ktoré by mohli pracovať s vyradenými hráčmi, napríklad ich za určitých podmienok vracieť späť do hry.

5.7 Odídenie mimo plochy

V niektorých hrách je možné sa zámerne alebo z donútenia dostať mimo hraciu plochu. To sa stane v situácii, kedy hráč ide po ceste, ktorej koniec je na strane dielika na okraji hracej plochy. Po vyjdení mimo plochy hráč bežne dostane nejaký trest, napríklad to býva vrátenie sa na začiatočnú pozíciu a vynechanie

nasledujúceho ťahu alebo eliminácia hráča. Naša implementácia pohybu nedovoľuje hráčom opustiť hraciu plochu, zároveň sme však chceli, aby v našom programe bolo možné hrať aj hry, v ktorých táto možnosť existuje. Tento problém sa dá vyriešiť tým, že používateľ okrem dielikov potrebných na hranie samotnej hry definuje v pravidlách dieliky navyše, ktoré budú reprezentovať priestor mimo plochy. Používateľ rozmery plochy definuje o dva stĺpca a riadky väčšie ako hra vyžaduje a po okraji hracej plochy manuálne rozmiestni tieto dieliky. Efekty, ktoré sa aktivujú po vyjdení mimo plochy používateľ dá týmto okrajovým dielikom.

5.8 Výpočet možných ťahov

Pôvodne sme to, či je ťah hráča v súlade s pravidlami kontrolovali v momente, kedy hráč zadal ťah, ktorý chce urobiť. Napríklad vždy, keď hráč určil, na ktorý dielik sa chce posunúť, náš program zistil, či existuje nejaká cesta spájajúca hráča s týmto dielikom. Tento prístup mal nevýhodu, že ak sa hráč snažil urobiť nedovolený ťah a kontrola teda zlyhala, tak pri ďalšom hráčovom pokuse urobiť ťah sa táto kontrola musela zopakovať. Rozhodli sme sa zmeniť náš prístup a nevyhodnocovať jednotlivé ťahy hráča, ale ešte pred tým, ako hráč začne svoj ťah, zistiť, ktoré všetky ťahy sú v súlade s pravidlami. Napríklad predtým, ako hráč bude zadávať svoj pohyb, náš program nájde všetky dieliky, ktoré sú spojené s hráčom a hráč sa na ne môže posunúť. Vždy keď hráč zadá svoj ťah, stačí len jednoducho skontrolovať, či sa jeho ťah nachádza v predpočítanom zozname možných ťahov. Tento prístup nám zároveň umožnil hráčovi zobrazit všetky jeho možné ťahy, čo mu uľahčuje hranie.

6 Programátorská dokumentácia

V tejto kapitole popíšeme štruktúru nášho programu. Podrobnejšie rozoberieme najdôležitejšie triedy, ich úlohu, metódy a ako sú implementované.

6.1 Nástroje použité na vývoj

Aplikáciu sme vyvinuli v objektovo orientovanom programovacom jazyku C#. Konkrétne ide o formulárovú aplikáciu vytvorenú pomocou .NET frameworku `Windows Forms` – v ňom sme vytvorili používateľské rozhranie. Na načítanie JSON súborov s pravidlami využívame systémovú knižnicu `System.Text.Json`.

6.2 Objektový návrh

Po zvolení konkrétnej hry z hlavného menu načítame informácie o nej z jej JSON súboru s pravidlami statickou triedou `RulesDeserializer`. Tá ich načíta do objektu triedy `JsonRules`, ktorej polia zodpovedajú poliam v súbore s pravidlami. Má v sebe objekt triedy `JsonBoard` a zoznamy objektov tried `JsonTile` a `JsonPlayer`, ktoré zodpovedajú JSON objektom v súbore.

Následne objekty s načítanými dátami spracujeme statickou triedou `JsonRulesProcessor`. Tá kontroluje, či sú dáta zadané v správnom formáte a či polia v nich obsahujú povolené hodnoty. Na základe načítaných dát vytvoríme triedy `Board`, `GameController` a `GameDrawer`, ktoré už sú používané pri samotnom hraní.

Hlavné triedy `Board`, `GameController`, `GameDrawer` majú každá svoju špecifickú úlohu (viď obrázok 6.1). Takéto rozdelenie kódu na jasne oddelené časti prispieva k jeho čitateľnosti, udržateľnosti a jeho jednoduchej budúcej rozšíriteľnosti.

V triede `Board` je uložený stav samotnej hry – rozmery hracej plochy, aké dieliky hra obsahuje, ich rozloženie na hracej ploche, polohu hráčov a podobne. Zároveň poskytuje metódy na manipuláciu s hrou – hľadanie dostupných možných ťahov, hýbanie s hráčmi a pridávanie, otáčanie a pohyb s dielikmi.

Trieda `GameController` slúži na ovládanie hry. Jej hlavnou úlohou je spracovávať vstupy od používateľov a kontrolovať ich. Pokiaľ vyhodnotí, že akcia, ktorú chce hráč vykonať je v súlade s pravidlami hry, zavolá príslušnú metódu triedy `Board` aby zmenila stav hry.

Úlohou triedy `GameDrawer` je vykresľovať hru pre používateľa. Hracia plocha s dielikmi a hráčmi sa vykresľuje podľa dát o stave hry získaných z triedy `Board`. Informácie o tom, ktorý hráč je na rade, a ktorá akcia a za akých podmienok práve prebieha sú zobrazované na základe dát z triedy `GameController`.

V nasledujúcich sekciách popíšeme najdôležitejšie polia a metódy týchto tried.

6.3 Trieda Board

Pole `Tiles`, reprezentuje hraciu plochu. Ide o dvojrozmerné pole obsahujúce objekty triedy `Tile`, kde každý prvok reprezentuje jeden hrací dielik, prípadne jeho neprítomnosť.

Pole `UnusedTiles` je zásobník obsahujúci objekty triedy `Tile`, ktorý predstavuje kôpku hracích dielikov. Dieliky z kôpky sa v niektorých hrách pridávajú na plochu, zamieňajú sa za dieliky z plochy alebo slúžia na dopĺňanie dielikov na ruke hráča. Pole je typu zásobník, pretože sa dieliky vždy pridávajú alebo odoberajú len z vrchu kopy.

Pole `Players`, obsahuje všetkých hráčov. Hráči sú reprezentovaní objektami triedy `Player`, ktoré obsahujú ich polohu, dieliky na ich ruke, pozbierané predmety a ich ciele.

Metóda `GetAllPossibleMoves` nájde všetky dieliky, ktoré sú cestami spojené so zadaným dielikom. Argumentami metódy je možné určiť obmedzenia pri hľadaní, napríklad rozsah vzdialeností, v ktorej sa hľadajú spojené dieliky, alebo smer, v ktorom sa má hľadať. Taktiež je možné určiť, aké typy dielikov má táto metóda nájsť – metóda môže hľadať aj napojené miesta, na ktorých sú skryté dieliky, alebo na ktorých dieliky vôbec nie sú. Špecifikovanie hľadaného typu dieliku umožňuje túto metódu používať nie len na hľadanie možných posunutí, ale aj hľadanie možných miest na pridávanie dielikov, alebo hľadanie dielikov, ktoré je možné vymeniť. Možnosť špecifikovať spôsob hľadania umožňuje túto metódu používať aj pre akcie, ktoré sú obmedzené nejakými špeciálnymi pravidlami. Metóda používa prehľadávanie do šírky od zadaného začiatočného dieliku. Metóda využíva prehľadávanie do šírky namiesto prehľadávania do hĺbky, pretože argumentami môže byť určená maximálna vzdialenosť, do ktorej sa má vyhľadávať. To umožňuje prehľadávanie do šírky predčasne ukončovať po dosiahnutí tejto vzdialenosti, čo zefektívňuje vyhľadávanie. Vyhľadávanie využíva pomocnú metódu `ExistsPathToAdjacentTile`, ktorá rozhoduje, či sú dva dieliky vedľa seba prepojené cestou.

Metóda `MovePlayer`, presunie hráča na zadaný dielik. Po posunutí aktivuje všetky efekty dieliku, na ktorý sa hráč presunul. Taktiež vyhodnotí zbieranie predmetov z tohto dieliku. Ak hráč nedosiahol svoj maximálny počet dielikov a v hre je dovolené predmety zbierať, metóda dá dovolený počet predmetov z dielika hráčovi.

Metódy `PushRowRight`, `PushRowLeft`, `PushColumnDown` a `PushColumnUp` slúžia na posúvanie stĺpcov a riadkov dielikov na hracej ploche. Ktorý riadok alebo stĺpec má byť posunutý je zadané jeho indexom, pričom číslovanie začína od nuly a riadky sú číslované zhora dole a stĺpce zľava doprava. Posúvanie dielikov je implementované analogickým presúvaním ich objektov v dvojrozmernom poli `Tiles`. Ak sa teda napríklad posúva druhý riadok dielikov o jeden dielik smerom doľava, objekty v druhom riadku zoznamu `Tiles` sú posunuté do stĺpca s o jedno menším indexom. Dielik, ktorý bol pôvodne v prvom stĺpci je vybraný von zo zoznamu. Do posledného stĺpca v danom riadku je vložený podľa pravidiel buď vybraný prvý

dielik alebo dielik z vrchu zásobníku `UnusedTiles` reprezentujúci kôpku.

Metóda `AddTile` na zadané prázdne miesto v zozname `Tiles` pridá nový dielik. Metóda `ExchangeTile` vymieňa zadaný dielik z kopy za iný dielik. Obe metódy majú argument špecifikujúci, či dielik pridávaný na plochu je z vrchu kôpky (zásobníku `UnusedTiles`) alebo je z ruky hráča. Ak je použitý dielik z ruky hráča, metóda dá na ruku hráča nový dielik z vrchu kôpky.

6.3.1 Trieda `Tile`

Trieda `Tile` reprezentuje jeden štvorcový hrací dielik s cestami. Cesty spájajú strany dieliku, pričom podľa pravidiel práve hranej hry môže každú stranu spájať len obmedzený počet ciest. Cesty môžu prepájať rôzne strany, ale môžu existovať aj „cykly“, teda cesty, ktoré prepájajú stranu dieliku samu so sebou. Trieda `Tile` obsahuje pre každú stranu zoznam jej ciest. Cesty na každej strane sú indexované v smere hodinových ručičiek. Každá samostatná cesta je reprezentovaná triedou `Path`, ktorá popisuje, ktoré strany a na ktorom mieste cesta prepája. Popisuje to tak, že pre každú stranu má pravdivostnú hodnotu určujúcu, či cesta spája túto stranu a ak áno, má uložené, na ktorom indexe.

Pole `Items` obsahuje predmety, ktoré sa na dieliku nachádzajú. Pole je implementované ako slovník, aby sa dalo ľahko vyhľadať, či sa nejaký predmet na dieliku nachádza alebo nie. Kľúčmi slovníka sú názvy predmetov a hodnotami sú množstvá, v akom sa na dieliku nachádzajú.

Pole `Effects` obsahuje zoznam efektov, ktoré sa aktivujú, keď na dielik vstúpi hráč. Efekty sú reprezentované potomkami abstraktnej triedy `TileEffect`. Každá takáto trieda obsahuje metódu `ApplyEffect`, ktorá efekt aktivuje. Trieda `Board` túto metódu zavolá pri vstupe hráča na tento dielik.

Metódy `RotateClockwise` a `RotateCounterClockwise` otáčajú dielik v smere alebo proti smeru hodinových ručičiek. Argumentom oboch metód je číslo udávajúce, koľko otočení o 90° sa má vykonať. Napríklad otočenie o 180° sa teda zadáva číslom 2. Otáčanie je implementované výmenou zoznamov ciest pre jednotlivé strany. Nakoľko sme zvolili na každej strane indexovanie ciest v smere hodinových ručičiek, nie je potrebné po výmene zoznamov meniť poradie ciest v nich – indexy zostávajú zachované. Je však potrebné pre každú cestu aktualizovať, ktoré strany teraz spája – na to sú určené metódy `RotateClockwise` a `RotateCounterClockwise` triedy `Path`.

6.3.2 Trieda `Player`

Trieda `Player` reprezentuje jedného hráča. Sú v nej informácie o jeho polohe – na ktorom dieliku sa práve nachádza a na ktorej konkrétnej ceste je, o predmetoch, ktoré pozbieral a o jeho cieľoch.

Pole Items obsahuje predmety, ktoré hráč pozbiera. Rovnako ako pri predmetoch na dieľiku ide o slovník, aby sa zefektívnilo rozhodovanie, či hráč má alebo nemá daný predmet. Klúče sú názvy predmetov, hodnoty počet, ktorý hráč má.

Pole Hand je zoznam objektov triedy `Tile`, ktorý reprezentuje dieľiky, ktoré má hráč na ruke. Pole SelectedIdIndex obsahuje index dieľika v tomto zozname, ktorý je práve zvolený. Práve zvolený dieľik z ruky sa používa na pridávanie alebo vymieňanie dieľikov, keď sú volané metódy `AddTile` a `ExchangeTile` triedy `Board`.

Pole Goals reprezentuje ciele, ktoré hráč musí splniť aby vyhral. Ciele sú plnené v takom poradí, v akom sú zadané v tomto zozname. Poradové číslo cieľa, ktorý hráč práve plní je v poli CurrentGoalIndex. Každý cieľ je reprezentovaný potomkom abstraktnej triedy `PlayerGoal` a obsahuje metódu `IsCompletedBy`, ktorá rozhoduje, či je daný cieľ hráčom splnený.

Pole EndOfGameScorings slúži na vypočítanie skóre hráča na konci hry. Pole je zoznamom objektov typu `EndOfGameScoring`, ktoré reprezentujú spôsoby získavania bodov. Bodový zisk každého spôsobu sa vypočíta jeho metódou `CalculateScore`.

Metóda `Win` vracia pravdivostnú hodnotu, ktorá hovorí, či hráč vyhral. Najprv sa skontroluje, či nie je hráč posledným nevyradeným hráčom z hry, čo by znamenalo jeho automatické víťazstvo. Ak nie je posledným hráčom, metóda skontroluje, či hráč nevyhral splnením všetkých svojich cieľov v poli Goals. Index v poli CurrentGoalIndex určuje, ktorý cieľ je práve plnený. Ciele v zozname Goals pred týmto indexom sú už splnené. Ak je aktuálny cieľ splnený, metóda skontroluje ďalšie ciele v poradí pre prípad, že hráč splnil viacero cieľov naraz a posunie index CurrentGoalIndex na najbližší nesplnený cieľ. Ak taký už nie je, hráč má splnené všetky svoje ciele a vyhral. Splnenie cieľa je kontrolované volaním metódy `IsCompletedBy` na objekte reprezentujúcom cieľ.

6.4 Trieda `GameController`

Pole InputSpecifications je štruktúra, ktorá obsahuje informácie o tom, ktoré akcie hráči na svojom ťahu vykonávajú a aké pravidlá sa na tieto akcie vzťahujú. Obsahuje tiež zoznam podmienok konca hry. Objekty reprezentujúce podmienky sú potomkami abstraktnej triedy `EndOfGameCondition`. Tie obsahujú metódu `IsEndOfGame`, ktorá vyhodnocuje splnenie podmienky.

Pole TurnStructureIndex obsahuje index špecifikujúci, ktorú akciu hráč práve vykonáva. Ide o index do zoznamu `TurnStructure` uloženého v štruktúre InputSpecifications, ktorý obsahuje poradie akcií, ktoré hráči vykonávajú.

Pole CurrentPlayerIndex obsahuje číslo hráča, ktorý je práve na ťahu.

Pole CurrentPossibleLocations obsahuje zoznam možných akcií hráča na ťahu. Možné ťahy sú vypočítané a uložené do tohto zoznamu vždy keď je ukončená akcia a hráč ide vykonávať ďalšiu akciu v poradí.

Pole `End` obsahuje pravdivostnú hodnotu hovoriacu, či už hra skončila. Ak áno, tak do poľa `WinnersIndexes` sa uložia čísla hráčov, ktorí vyhrali.

Hlavnou metódou triedy `GameController` je metóda `Input`. Jej argumentami sú súradnice štvorca v zobrazenej mriežke (zloženej z hracej dosky a okolia s informáciami), na ktoré hráč klikol. Podľa aktuálnej akcie zavolá jednu z pomocných metód `ResolveMove`, `ResolveMoveAll`, `ResolvePush`, `ResolveAdd` alebo `ResolveExchange`, ktorá vyhodnotí hráčov ťah. Každá z nich vracia pravdivostnú hodnotu, ktorá hovorí, či zadaný ťah bol v súlade s pravidlami a bol vykonaný. Ak áno, metóda skontroluje, či týmto ťahom hra neskončila. To sa mohlo stať, ak nejaký hráč zvíťazil, alebo bola naplnená niektorá z podmienok konca hry. Víťazstvo hráčov je kontrolované metódou `Win` na objektoch triedy `Player`, ktoré reprezentujú hráčov. Podmienky konca hry sú kontrolované pomocou metódy `IsEndOfGame` objektov triedy `EndOfGameCondition` reprezentujúcich tieto podmienky. Ak nastal koniec hry splnením niektorej z nich, určí sa víťaz zrátaním bodov jednotlivých hráčov. Ak koniec hry nenastal, metóda `Input` zistí, či bola aktuálna akcia dokončená, a ak áno, nastaví index `TurnStructureIndex` na ďalšiu akciu v poradí. Prípadne tiež nastaví index `CurrentPlayerIndex` na ďalšieho hráča, ak hráč na ťahu dokončil svoju poslednú akciu.

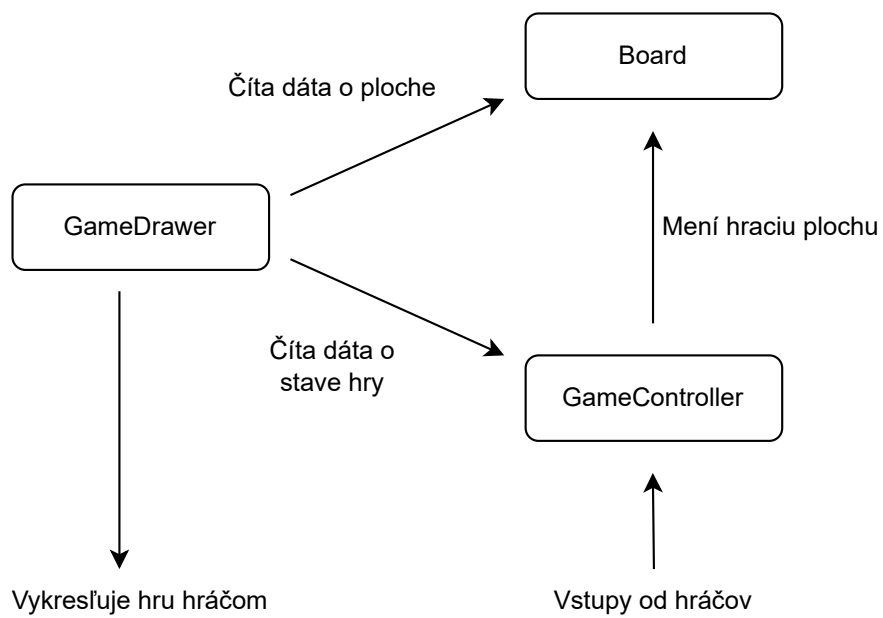
Metóda `GetPossibleMoves` vracia všetky miesta, na ktoré sa hráč môže v súlade s pravidlami posunúť. Miesta získava pomocou zavolania metódy `GetAllPossibleMoves` triedy `Board` s parametrami z poľa `InputSpecifications` popisujúcimi pravidlá pohybu. Metódy `GetPossibleConnectedAddLocations` a `GetPossibleConnectedExchangeLocations` vracajú s hráčom prepojené miesta, na ktoré je možné pridať, alebo s ktorými je možné vymeniť dielik. Tie tiež získavajú volaním metódy `GetAllPossibleMoves` triedy `Board` s parametrami popisujúcimi pravidlá danej akcie.

Metóda `FindScoringWinners` vyhodnotí skóre každého hráča. Skóre hráča je súčtom bodov získaných rôznymi spôsobmi, reprezentovanými objektami triedy `EndOfGameScoring` v poli `EndOfGameScorings` v objekte hráča. Každý spôsob vráti počet získaných bodov metódou `CalculateScore`. Indexy hráčov s najväčším počtom bodov sa uložia do poľa `WinnersIndexes`.

6.5 Trieda `GameDrawer`

Hlavnou metódou triedy je metóda `DrawBoard`, ktorá vykresľuje celú hru. Používa pomocné metódy na vykresľovanie jednotlivých dielikov, ciest, hráčov a informácií o aktuálnej akcii. Taktiež obsahuje metódu `HighlightTile`, ktorá zvýrazní zadaný dielik zeleným okrajom. Tou sa zvýrazňujú miesta na hracej ploche uložené v poli `CurrentPossibleLocations` objektu triedy `GameController`, ktoré predstavujú možné ťahy hráča na rade.

Vykresľovanie je implementované dynamicky, teda nie sú dopredu určené rozmery vykresľovaných prvkov, ale sú vyrátané podľa veľkosti okna, v ktorom sú vykresľované. Výsledkom je, že aplikácia je korektne vykresľovaná nezávisle od toho, aké veľké má používateľ okno aplikácie.



Obrázek 6.1 Objektový návrh programu.

Záver

Splnenie zadania

V úvode práce sme si vytýčili cieľ vytvoriť univerzálny program, v ktorom je možné hrať spoločenské hry nami zvoleného typu – hry hrané na ploche zloženej z potenciálne pohyblivých štvorcových dielikov.

Najdôležitejšou časťou práce bolo navrhnuť formát a štruktúru súborov, v ktorých používatelia zadávajú pravidlá hier. Naším cieľom bolo, aby zadávanie bolo jednoduché aj pre používateľov bez akejkoľvek znalosti programovania. Pre súbory sme zvolili formát JSON, ktorý je čitateľný pre ľudí a zároveň jednoducho spracovateľný počítačom. Štruktúru sme navrhli tak, že si používatelia pri každom pravidle vyberú jednu z možností. Výber z obmedzeného počtu možností je jednoduchý spôsob zadávania pravidiel. Aj napriek obmedzeným možnostiam je možné vytvoriť veľké množstvo hier kombináciou rôznych možností pri každom z pravidiel.

Vytvorili sme program, ktorý vie tieto súbory s pravidlami hier spracovať a umožňuje používateľom tieto hry hrať. Program sme rozdelili do troch hlavných častí s vlastnými úlohami – uchovávanie dát o stave hry, manipulovanie nimi podľa vstupov od hráčov a zobrazovanie hry. Rozdelenie programu zlepšuje jeho čitateľnosť a uľahčuje jeho budúce rozširovanie.

Možnosti vytvárať a hrať hry v našom programe sme demonštrovali tým, že sme vytvorili súbory s pravidlami pre už existujúce štyri spoločenské hry *Amazing Labyrinth*, *Tsuro*, *Snakes and Ladders*, *Mice in a maze* a umožnili ich tak používateľom hrať.

Význam práce je najmä pri prevádzaní spoločenských hier tejto rodiny do digitálnej podoby. Hry nie je potrebné znova celé programovať – s použitím nášho programu stačí v jednoduchom súbore s pravidlami popísať, ako sa daná hra hrá. Program je tiež možné využiť pri vymýšľaní nových hier na rýchle vytvorenie ich prototypu a otestovanie hrateľnosti.

Možné rozšírenia

Nakoľko hlavným cieľom tejto práce bolo preskúmať vytváranie hier pomocou zadávania pravidiel v jednoduchom súbore, v porovnaní s už existujúcimi aplikáciami na hranie spoločenských hier naša aplikácia v ostatných aspektoch zaostáva.

Najväčším nedostatkom oproti existujúcim aplikáciám je absencia možnosti hrať hry na viacerých zariadeniach *online*. Naša aplikácia umožňuje hranie len lokálne na jednom zariadení, čo je pre používateľov obmedzujúce. Pridanie možnosti hrania *online* by hranie spravilo praktickejším, nakoľko by hráči nemuseli byť fyzicky spolu pri jednom zariadení.

Grafické rozhranie je oproti existujúcim aplikáciám veľmi jednoduché. Hracia plocha a informácie o stave hry sú v našej aplikácii vykresľované len pomocou jednoduchých geometrických tvarov a textu. Zlepšenie grafického rozhrania tak, aby sa viacej podobalo na fyzické verzie spoločenských hier by našu aplikáciu spravilo pre používateľov viac atraktívnou.

V našej aplikácii je možné hrať len hry jedného typu. Rozšírenie programu tak, aby podporoval hranie hier ďalších typov by dalo používateľom viac možností pri vytváraní vlastných hier.

Literatúra

1. HENRY, J. *About Tabletop Simulator* [online]. [cit. 2024-04-22].
Dostupné z: <https://www.tabletopsimulator.com/about>.
2. ISABELLI, G.; COLIN, E. *Úvod do BGA* [online]. [cit. 2024-04-22].
Dostupné z: <https://cs.boardgamearena.com>.
3. BOKAREV, T. *About Tabletopia* [online]. [cit. 2024-04-22].
Dostupné z: <https://tabletopia.com/about>.
4. CROCKFORD, D. *Introducing JSON* [online]. [cit. 2024-07-05].
Dostupné z: <https://www.json.org>.
5. SHAFRANOVICH, Y. *Common Format and MIME Type for Comma-Separated Values (CSV) Files* [online]. [cit. 2024-07-08].
Dostupné z: <https://datatracker.ietf.org/doc/html/rfc4180>.
6. BRAY, T. (ed.). *The JavaScript Object Notation (JSON) Data Interchange Format* [online]. [cit. 2024-07-08].
Dostupné z: <https://datatracker.ietf.org/doc/html/rfc8259>.
7. THOMPSON, H.; LILLEY, C. *XML Media Types* [online]. [cit. 2024-07-09].
Dostupné z: <https://datatracker.ietf.org/doc/html/rfc7303>.

A Prílohy

Príručník BoardGameInterpreter v elektronickej prílohe obsahuje podpriechinky SourceCodes a App.

Podpriechinok SourceCodes obsahuje súbory so zdrojovými kódmi nášho programu:

- Program.cs
- BoardGameInterpreter.cs
- Board.cs
- GameController.cs
- Tile.cs
- Player.cs
- GameDrawer.cs
- JsonHandling.cs

Podpriechinok App obsahuje samotnú aplikáciu – spustiteľný súbor BoardGameInterpreter.exe a súbory potrebné na jej spustenie. Zároveň obsahuje ukážkové súbory s pravidlami hier:

- amazing_labyrinth_rules.json
- snakes_and_ladders_rules.json
- tsuro_rules.json
- mice_in_a_maze_rules.json

Taktiež obsahuje súbor rules.json, do ktorého používateľia zadávajú pravidlá vlastnej hry.