Gymnázium Arabská, Praha 6, Arabská 14
Obor programování

Autosweeper

Adam Suchý

Květen, 2021

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V ..... dne ..... Adam Suchý .....

Abstrakt

Cílem tohoto ročníkového projektu je vytvořit hru, ve které je hráči ukázáno minové pole s úkolem ho zneškodnit. Hráč má za úkol najít všechny miny tím, že postupně odkrývá jednotlivá políčka. Při každém odkrytí je hráči poskytnuta informace s množstvím min v okolních osmi políčkách. Když jsou všechna políčka kromě min odhalena, tak hráč zvítězil. Jestliže hráč odhalí minu, je hra prohrána.

Obsah

1 Úvod

1.1 Zadání projektu

Autosweeper je variace na známou hru jménem Minesweeper, ve které se hledají miny. Hráč dostane libovolně velké dvourozměrné pole čtverečků, pod kterými se můžou skrývat miny. Cílem hry je postupně odhalovat čtverečky pod kterými miny nejsou. Při každém odhalení je hráči ukázáno číslo, které značí počet min v okolních osmi políčkách. Z těchto informací musí hráč hádat, jaké políčka může odhalit.

Součástí programu je automat, který hru umí řešit. Hráč si může vybrat mezi třemi herními módy: samostatný, s pomocí a automatický. Když si hráč vybere samostatnou hru, tak se automat do hry nijak nezapojuje. Při hře s pomocí může hráč kliknout na tlačítko "analyzovat", které automat spustí a ukáže hráči jaké políčka jsou vyhodnocena jako nebezpečná. V automatickém módu automat řeší hru celou sám.

Při jakékoliv hře bude také běžet časovač, aby hráč mohl vidět své zlepšení, či zhoršení. Jestliže hru bude řešit jen automat, tak bude časovač běžet jen při jeho "přemýšlení"neboli vyhodnocování nebezpečnosti políček.

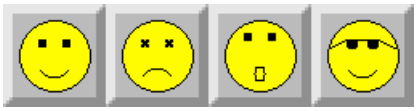
1.2 Hra Minesweeper a její historie

Základní část programu Autosweeper je hra Minesweeper, prvně vydaná společností Microsoft v roce 1990 jako součást balíčku video her pro operační systém Windows 3.0. Avšak originální nápad této hry náleží video herní firmě Quicksilva a její hře Mined-Out (1983) pro pročítače ZX Spectrum napsané v jazyce COBOL. Zatímco Mined-Out byla jediná doložená inspirace pro Minesweeper, první hra s podobnou pointou byla Cube (1973) od Jerimaca Ratliffa. Princip hry Minesweeper byl tedy jeden z nejstarších. Dokáže se vyrovnat například hře Tetris, nebo Pac-Man. [1]

2 Tvorba hry

2.1 Uživatelské rozhraní

Program je postaven na grafické knihovně JavaFX pro programovací jazyk Java. Vzhledem k povaze hry je JavaFX zcela dostačující.



Obrázek 1: Všechny stavy smajlíku. Popisky (zleva doprava): Neutrální, Prohra, Hráč odhaluje minu, Výhra

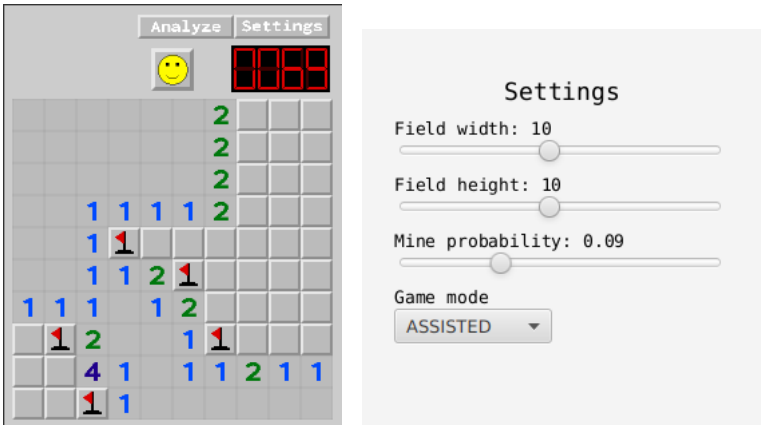
2.1.1 Struktura

Celé uživatelské rozhraní se nachází ve dvou oknech. V prvním okně (obrázek 2a) uživatel najde hru společně s časovačem a dvěma až třemi tlačítky. Autosweeper lze používat jen s tímto jedním oknem, ale kdyby chtěl uživatel změnit nastavení hry, může tak učinit ve druhém okně (obrázek 2b), které otevírá tlačítko "settings". V nastavení lze změnit velikost pole, pravděpodobnost min a herní mód. Druhé okno je možné mít otevřené pořád, nastavení se přenese vždy jen do příští hry, kterou začneme zmáčknutím smajlíku. Smajlík v prvním okně také slouží jako indikátor stavu hry. (obrázek 1)

- Okno 1
  - Tlačítko se smajlíkem na restart hry, indikátor stavu hry
  - Tlačítko na otevření nastavení
  - Tlačítko na analýzu v módu s pomocí
  - Herní pole
  - Časovač
- Okno 2
  - Posuvník upravující šířku pole
  - Posuvník upravující výšku pole
  - Posuvník upravující pravděpodobnost výskytu min
  - Tlačítko s výběrem herních módů

2.1.2 Design

Vzhled programu je pokus o klon originálního designu Windows Minesweeper. Jako vzor byla použita jedna z mnoha webových implementací [2], jsou z ní vybrané zejména barvy čísel a vzhled smajlíku při různých stavech hry. Původě mělo herní pole být tvořeno z několika tlačítek neboli objektů typu



(a) Hlavní okno (b) Okno s nastavením

Obrázek 2: Okna s uživatelským rozhraním

`javafx.scene.control.Button`, to by ale vyžadovalo hodně stylování a pravděpodobně také vytvořit si své vlastní písmo s originálními číslicemi. Místo tlačítek jsou tedy využité `javafx.scene.image.ImageView`, které umožňují zobrazit výstřižek obrázku. Na jednotlivé `ImageView` jsou poté navázány funkce spouštějící se při zmáčknutí, nebo puštění talčítka od myši.

Podobný problém nastal u časovače. Optimálně by měl být jeho sedmi-segmentovaný displej generován, ale moderní počítače mají dost paměti na to, aby mohla každá číslice být uložená jako obrázek. Každá viditelná cifra je stejně jako u všeho ostatního jeden `ImageView`, všechny jsou poté spravovány třídou `SevenSegmentDisplay`.

Všechny obrázky byly tvořeny mnou v programu Aseprite (Aseprite je placený, ale open-source. Je tedy zcela možné si Aseprite zkompilovat zadarmo, což byla metoda, kterou jsem zvolil).

2.2 Herní pole a stav hry

Herní pole je hlavní částí programu. Poté co ho ovladač hlavního okna vytvoří, spravuje celou hru. Při inicializaci nového pole (`PlayingField`) je třeba poskytnout jen herní nastavení. Pole poté vygeneruje všechny políčka, které lze dostat zpátky pomocí metody `getCellAt`. Aby bylo možné pole rychle vykreslit, je implementována metoda `render`, která jako parametr dostane JavaFX `GridPane`, do kterého vloží všechny políčka.

2.2.1 Komunikace mezi herním polem a hlavním ovladačem

Po inicializaci pole a zavolání metody `render` se vše odehrává uvnitř nově vytvořeného objektu. Pro snadnou komunikaci s hlavním ovladačem a zajištění reagování ostatních prvků v uživatelském rozhraní existuje systém událostí. Ovladač může použít metodu `setEventHandler` pro nastavení funkce, která bude naslouchat událostem ve hře. Například pří začátku hry nastane událost typu `EventType.START`. Naslouchající funkce tuto událost zachytí a zapne časovač. Další herní události jsou například při odhalování políčka, odhalení miny (= prohra), nebo při výhře.

2.2.2 Stav hry

V samotném objektu herního pole není uloženo moc informací. Jsou v něm uložené jen odkazy na jednotlivé políčka, celkový počet bomb a počet neodhalených políček. Poslední dvě informace jsou potřeba jen při vyhodnocování, jestli je hra dohraná, či ne. Většina informací o stavu hry jsou v jednotlivých políčkách, což jsou objekty třídy `Cell` (která dědí z již zmíněné třídy `ImageView`). Každé políčko ukládá svou pozici, jestli je bomba, nebo označené. Má také speciální hodnotu `solver_flags`, která slouží automatu k ukládání informací z minulých tahů, a mnoho metod pro ulehčení práce s polem v automatu, nebo při vyhodnocování počtu bomb v okolí políčka.

Jedna z výhod ukládání si informací rovnou v políčkách, které dědí z `ImageView`, je možnost

automatické aktualizace v uživatelském rozhraní. Také se s takovými objekty jako s celky velmi dobře pracuje v automatu.

3 Automat

Automat, který dokáže řešit hru, je hlavní odlišení projektu Autosweeper od původní hry. Algoritmus dostane jako vstupní data herní pole a jeho stav přesně tak, jak je zobrazeno na obrazovce. Neví tedy, kde miny opravdu jsou a kde ne. Cíl automatu je poté navrhnout možné tahy, které si myslí, že jsou bezpečné. Je strukturovaný do tří úrovní, každá se snaží najít možné tahy, pokud žádný nenajde, je spuštěna další úroveň.

3.1 První tah

První tah hry se liší od ostatních tím, že je jisté, že není možné vydedukovat pozici žádné bomby. Nejdříve je nutné vyřešit problém s rozlišením "začínající"hry od ostatních. Hra je pro automat začínající, jestliže se na herním poli nevyskytuje žádná políčka s číslem (počtem bomb v okolí) 0 a počet odehraných tahů je méně než čtyři.

V této situaci automat odkrývá políčka v rozích, protože vzhledem k počtu okolních políček je nejvíce pravděpodobné, že stav políčka nebude možné vydedukovat. [3] Pravděpodobnost výhry se tím sice nezmění, ale algoritmus alespoň neplýtvá čas řešením hry ve které bude muset hádat.

3.2 První úroveň

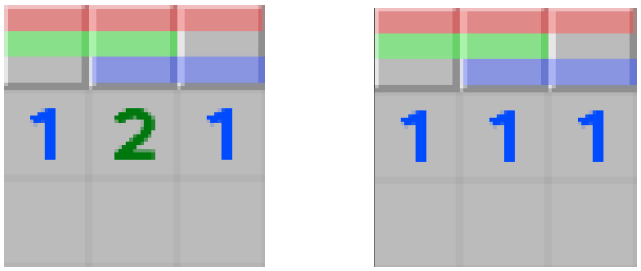
První úroveň automatu je algoritmus, který se dívá na stav každého odhaleného políčka a pokud se jeho číslo schoduje s počtem neodhalených sousedů (8 sousedících políček) tak všechny sousedy označí jako bomby. Po zkontrolování celého pole jde zpátky na začátek a prochází celé pole znovu. Hledá políčka u kterých se jejich číslo schoduje s počtem nalezených bomb, jestliže taková situace nastane, jsou odhaleni všichni sousedi, kteří nebyli označeni jako bomba.

Komplexitu prvního stupně odhaduji na $O(8n)$. V praxi je ale vyhodnocování pole mnohem rychlejší vzhledem k počtu políček s číslem 0, které jsou pro výpočet zanedbatelné.

3.3 Druhá úroveň

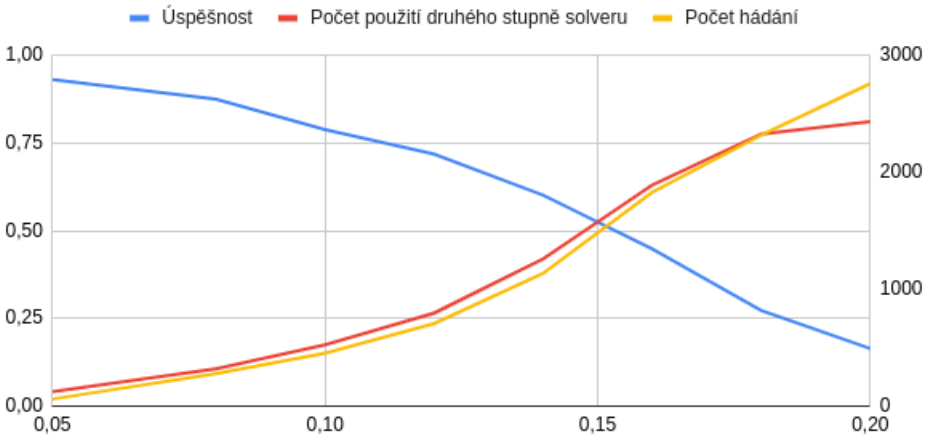
Druhá úroveň je spuštěna v případě, že první úroveň nic nenašla. Pracuje již s označenými bombami a snaží se vyřešit další situace za pomocí spojení více neznámých políček do množin, neboli "kyblíků". Na obrázcích (3) lze vidět jakou strukturu kyblíky mají. Červenou je označen kyblík prostřední číslice, levá číslice má zelený a pravá modrý. Červený kyblík je hlavní, jelikož to je kyblík políčka, které se snažíme vyřešit. Všechny ostatní kyblíky musí být podmnožinou hlavního, abychom je mohli od sebe odčítat. Tato limitace omezuje druhou úroveň na řešení situací 3x3. První úroveň tomuto velmi pomáhá, protože může působení druhé úrovně rozšířit tím, že označí už odhalené bomby a zvýší tak šanci, že nějaký kyblík nebude přesahovat ven z hlavního.

Poté co byli nalezeny všechny kyblíky a bylo zajištěno, že se mezi nimi nenachází žádné duplikáty, je potřeba vyřešit průniky dvou, či více kyblíků. V situaci na obrázcích (3) se nachází jeden průnik a to mezi zeleným a modrým kyblíkem. Algoritmus se nejdřív snaží najít jakékoliv políčko, které je ve více než jednom kyblíku. Pokud nějaké políčko najde, snaží se dohledat



(a) Situace z dvěmi bombami (b) Situace s jednou bombou

Obrázek 3: Situace řešitelné 2. úrovní automatu



Obrázek 4: Úspěšnost automatu na poli 20x20 pro různé obtížnosti hry

celý průnik. Každý průnik poté zkusí vyhodnotit tím, že odečte očekávaný počet bomb v průniku od všech pronikajících kyblíků včetně hlavního kyblíku. Jestliže součet vypočítaných hodnot je roven 0, pak je v průniku bomba a jakékoliv jiné políčko algoritmus považuje za bezpečné. Když součet naopak 0 roven není, je třeba průnik ze všech pod-kyblíků odstranit bez odčítání z celkového počtu bomb. Tedy průnik obsahuje všechny miny v hlavním kyblíku, jestliže následující rovnost platí:

$$\sum_{x \in K} |x - \min K| = 0$$

Kde K je množina počtů bomb všech pronikajících kyblíků a hlavního kyblíku.

Po ošetření průniků algoritmus odčítá všechny pod-kyblíky od hlavního (tzn. odstranit všechny políčka pod-kyblíku z hlavního kyblíku a odečtení počtu bomb). Všechny zbylé políčka v hlavním kyblíku lze považovat za bezpečné za podmínky, že počet bomb se rovná 0. Při nerovnosti algoritmus výpočet zahodí a přechází na další políčko.

3.4 Třetí úroveň

Aby automat vždy navrhl nějaké políčko, je potřeba třetí úroveň. Tato úroveň se nesnaží vydedukovat bezpečné políčka, ale spočítá jednoduchou pravděpodobnost bomby na každém políčku. Vzhledem k tomu, že výpočet pravé pravděpodobnosti výskytu bomby na políčku je stejně těžký úkol jako samotné řešení, není pravděpodobnost z výpočtu pravdivá, ale jaksi orientační. Každé políčko dostane pravděpodobnost jednoduchým výpočtem, který se provede pro každé odhalené políčko: $p = \frac{c}{n}$, kde c je číslo na odhaleném políčku a n je počet neodhalených sousedů. V případě, že na jedno políčko bude víc pravděpodobností, algoritmus vybere nejvyšší. Po výpočtu všech pravděpodobností algoritmus vybere políčko s nejmenší pravděpodobností výskytu bomby.

3.5 Limity

Autosweeper nedokáže vyřešit všechny Minesweeper hry. Řešení těchto her je těžký matematický problém, Richard Kaye ve své studii nazvané "Minesweeper is NP-Complete" argumentuje mnoha způsoby, že Minesweeper problém je NP-úplný [4]. To by znamenalo, že je jeden z nejtěžších matematických problémů a jeho komplexita by byla stejná jako ta známého problému s obchodním cestujícím [5].
Vždy úspěšný automat je hned vyloučen při prvním tahu, kdy je vždy možné narazit na minu. Záleží tedy na implementaci hry, jestli první tah udělá zaručeně dobrým, nebo jestli herní pole bude už přegenerované (způsob používán Autosweeperem).

Aby se dala kvantifikovat úspěšnost Autosweeperu, byl spuštěn vždy na 1000 her pro různé nastavení a z výsledků se vytvořil graf (obrázek 4). Na grafu lze vidět, že automat má při jednoduché hře (pravděpodobnost výskytu min $0,05$) vysokou úspěšnost a zároveň řeší většinu tahů jen pomocí prvního stupně. Při pravděpodobnosti $0,15$ je automat úspěšný jen v polovině her. Tyto hry jsou těžké i pro lidské

hráče. Při pravděpodobnosti $0,20$, tedy každé páté políčko je v průměru mina, automatu selhává i druhý stupeň algoritmizace a je vidět, že automat ve většině případech hádá.

4 Instalace

Instalace Autosweeperu je poměrně jednoduchá. Postup k instalaci:

1. Zkompilujte zdrojový kód, nebo si stáhněte zkompilovaný program z GitHubu
2. Výsledný `.jar` dejte společně s `resources` do jedné složky
3. Spusťte `.jar` program

5 Závěr

Myslím si, že se mi Autosweeper podařil. Uživatelské rozhraní je přesně takové, jaké jsem si představoval. Jediné z čeho jsem lehce zklamaný je samotný automat. Původně jsem zamýšlel implementovat algoritmus popisovaný ve studii Bercerry [3], ale nechtěl jsem implementovat věc, které nerozumím. I přes lehké zklamání, že automat není 100% úspěšný jsem rád, že umí vyřešit i poměrně těžké hry. Na projektu by se dal zlepšit samotný automat, ale také i celková organizace kódu, jako například render políček, který je sice velmi automatický, ale občas dělá neplechu.

Odkazy

[1] *Windows Minesweeper*. URL: http://www.minesweeper.info/wiki/Windows_Minesweeper.

[2] *Play Minesweeper*. URL: http://www.play-minesweeper.com/.

[3] David Bercerra. "Algorithmic Approaches to Playing Minesweeper". In: (2015).

[4] Richard Kaye. "Minesweeper is NP-complete". In: *The Mathematical Intelligencer* 22.2 (březen 2000), s. 9–15. DOI: 10.1007/bf03025367. URL: https://doi.org/10.1007/bf03025367.

[5] *Problém s obchodního cestujícího*. URL: https://cs.wikipedia.org/wiki/Probl%C3%A9m_obchodn%C3%ADho_cestuj%C3%ADc%C3%ADho.

Seznam obrázků