

Férové miny

Dokumentace k programu

Lucie Vomelová

Obsah

1	Zadání	2
2	Program	2
2.1	Ovládání	2
2.1.1	Menu	2
2.1.2	Odkrytí políčka	2
2.1.3	Označení miny	2
2.2	Generování herního pole	2
2.2.1	První generování	2
2.2.2	Přegenerování	3
2.3	Otevírání políček	3
2.4	Hledání jistých políček	3
2.5	Kód programu	3
2.5.1	Cell	3
2.5.2	Values	4
2.5.3	MainWindow	4
2.5.4	PlaceMines	4
2.5.5	Timer	5
2.5.6	Neighbours	5
2.5.7	Model	5
2.5.8	Open	5
2.5.9	Solver	5
2.5.10	Options	5

1 Zadání

Implementace klasické hry Hledání min s malou úpravou - pokud uživatel, soudě dle stavu hrací plochy, nemohl mít ani u jednoho políčka jistotu, že na něm mina není (např. vždy na začátku hry), mina po odkrytí libovolného políčka na tomto políčku není. Naopak pokud uživatel mohl mít jistotu, že na některém políčku mina není, a přesto odkryl políčko, u nějž tuto jistotu mít nemohl, mina na něm je.

2 Program

Program je psán v jazyce C# s využitím knihovny WPF. Použila jsem vývojové prostředí JetBrains Rider.

2.1 Ovládání

Ovládání hry je stejné jako u klasického Hledání min. Nová hra se spustí tlačítkem usmívajícího se smajlíka v horní části okna. Čas je měřen od prvního kliknutí do herního pole.

2.1.1 Menu

V levém horním rohu se nachází menu, které umožňuje přejít do okna *Options*. Tam uživatel může nastavit obtížnost hry a zvolit mód hry.

V Debug módu program označí neotevřená jistá políčka růžovou barvou. To se hodí, aby hráč věděl, kam může kliknout (pokud tam není mina). Normal mode vypadá stejně jako běžný Minesweeper, ale stále platí podmínka férovosti.

2.1.2 Odkrytí políčka

Pokud hráč klikne na políčko levým tlačítkem myši, políčko bude odkryto. Pokud na něm není mina, objeví se na jeho místě obrázek buď prázdného políčka nebo číslice 1-8. Odkryté číslo udává počet min, které sousedí s daným políčkem (prázdné pole reprezentuje číslo 0). Pokud ale byla na tomto místě mina, dojde k jejímu "výbuchu" a hra končí.

2.1.3 Označení miny

Pravým tlačítkem myši se pokládá vlaječka (případně otazník). Pravý klik na neotevřené políčko umístí vlaječku, která značí, že na tomto políčku je podle hráče mina. Pravý klik na vlaječku způsobí její odstranění a na jejím místě se objeví otazník. Ten většinou značí, že si hráč není jistý, jestli je na tomto poli mina. Pravý klik na otazník ho odstraní a políčko bude prázdné.

2.2 Generování herního pole

Při každé hře je potřeba vygenerovat herní pole (a případně i přegenerovat znovu), k tomu slouží třída *Model*.

2.2.1 První generování

Herní pole je generováno vždy při prvním kliknutí do hry: jakmile hráč klikne na políčko, tomuto políčku se přiřadí hodnota 0. Nyní se rozmístí všechny miny náhodně do herního pole (jediná podmínka je, že žádná nesmí sousedit s již otevřeným políčkem). Následně se celé herní pole vyplní čísly 0-8, kde číslo políčka je určeno počtem min, které s ním sousedí.

2.2.2 Přegenerování

Jakmile neexistuje neotevřené políčko, na kterém je jisté číslo, může uživatel kliknout kamkoliv (kromě jistých min) a tam musí být číslo. Pokud se treťí na políčko, které obsahuje číslo z předchozího vygenerování, stačí číslo odkrýt. Ale pokud se hráč treťí do miny, musí se herní pole přegenerovat.

Přegenerování probíhá následovně: smažou se všechny miny, které nejsou jisté. Dále se smažou všechna neotevřená čísla. Potom se rozmístí miny kolem otevřených čísel tak, aby mělo každé číslo okolo sebe správný počet min (to zajišťuje třída *PlaceMines*). Nakonec se náhodně rozmístí zbývající miny a zbývající políčka vyplní čísla (stejně jako na začátku).

2.3 Otvírání políček

O otvírání políček se stará třída *Open*. Na začátku hry se otevře první políčko. Po vygenerování hry se otevře oblast okolo něj. Protože první políčko má vždy hodnotu 0, nebude okolo něj žádná mina a rovnou se mohou otevřít i všichni jeho sousedi. Na ně se uplatní stejná podmínka (pokud mají hodnotu 0, otevrou se jejich sousedi atd.).

Nyní program spočítá, která políčka jsou jistá čísla nebo jisté miny (to se děje po každém otevření). Dokud existují jistá čísla, uživatel musí otvírat jen je, jiné otevřené políčko bude vždy obsahovat minu a uživatel prohraje. Pokud uživatel otevře políčko s hodnotou 0, dojde k otevření oblasti stejně jako na začátku.

Pokud existuje buňka taková, že všichni její sousedé jsou jistí a je okolo ní označen správný počet min, pak je možné na ni kliknout a otevřít tak všechny neotevřené sousedy. Zde je možné se splést a prohrát hru - program nedovolí otevřít sousedy buňky, která nemá všechny sousedy jisté. Ale pokud je má jisté a hráč označí vlaječkou špatnou buňku, pak se sousedé otevrou a neoznačená mina vybuchne.

2.4 Hledání jistých políček

Jistá políčka jsou taková, o kterých víme, že jsou to miny nebo čísla. Patří mezi ně již otevřená políčka a některá políčka sousedící s těmi otevřenými. Třída *Solver* hledá jistá neotevřená políčka. Používá dva způsoby:

1. Po každém otevření se zkontroluje u políček, která byla otevřením ovlivněna, počet sousedních jistých min a čísel. Pokud je okolo políčka správný počet min (*jeho_hodnota*) nebo čísel (*8 - jeho_hodnota*), jsou jistá i zbývající sousední políčka. Např. pokud jsou okolo políčka s hodnotou dva dvě jisté miny, zbývající sousedé musí být čísla.
2. Zkusí se všechna korektní rozstavení min okolo otevřených čísel. Pokud je ve všech možnostech rozstavení na nějakém políčku vždy číslo nebo mina, bylo tím nalezeno další jisté políčko. Toto zajišťuje třída *PlaceMines*, jejíž funkci *Solver* volá.

2.5 Kód programu

2.5.1 Cell

Buňka je základní jednotka v programu. Jedna buňka reprezentuje políčko v herním poli, herní pole je tvořeno dvojrozměrným polem buněk. Třída má následující atributy:

1. **row, column** - souřadnice buňky v herním poli.
2. **value** - hodnota buňky. Bud' číslo 0-8 nebo konstanta *Values.MINE*. Pokud ale není buňka jistá, tak když na ni uživatel klikne, bude se vždy chovat jako mina. Jen pokud je jistá, uplatní se její pravá hodnota.
3. **minesLeft** - kolik min okolo buňky ještě není jistých.

4. **unknownLeft** - kolik buněk okolo buňky ještě není jistých.
5. **isOpened, isKnown, isFlag, isQuestionMark** - booleany udávající jestli je buňka otevřená, jistá, označená vlajkou nebo otazníkem.
6. **btn** - tlačítko v herním poli.

Třída dále obsahuje několik metod - např. na nastavení obrázku tlačítka.

2.5.2 Values

Zde jsou uloženy důležité hodnoty, jako je počet min ve hře nebo počet řádků a sloupců. Také se zde nacházejí konstanty, pomocí kterých se označují buňky, na nichž je mina, číslo apod.

2.5.3 MainWindow

Hlavní okno programu. Nachází se tu funkce na přípravu a vykreslení herního pole, dále funkce řešící události stisknutí tlačítek v poli a události tlačítek myši.

Po každém kliknutí do herního pole se zavolá *CheckWin*, která kontroluje, jestli hráč už vyhrál. Výhra nastává ve dvou případech - buď jsou všechny miny označené vlajčkou nebo je počet neotevřených polí roven počtu hledaných min. Pokud je jedna z těchto podmínek splněna, zavolá se *Win* a hra končí.

2.5.4 PlaceMines

Nejkomplikovanější třída, která zajišťuje rozmisťování min při přegenerování pole a při hledání jistých políček. Pro větší přehlednost je rozdělena do několika souborů. Obsahuje strukturu *helpCell*, která kopíruje důležité vlastnosti z *Cell*.

Dvojměrná pole *helpArray* a *parents* mají stejnou velikost jako herní pole. *helpArray* kopíruje herní pole - otevřená políčka mají stejnou hodnotu, jistá buď *Values.MINE* nebo *Values.NUMBER* podle toho, jestli jsou číslo nebo mina, ostatní políčka mají hodnotu *Values.UNKNOWN*. Toto pole slouží ke zkoušení různých možností umístění min. Podle potřeby se části přemazávají, dokud není nalezeno buď jedno korektní řešení (přegenerování pole) nebo průnik všech korektních řešení (hledání jistých políček).

BorderArea - čísla, která jsou otevřená a nemají kolem sebe správný počet min, budou součástí oblasti, okolo níž se budou umisťovat miny. Funkce pro nalezení a práci s touto oblastí jsou v souboru *BorderArea.cs*.

Sekce - Třída používá bruteforce algoritmus na nalezení řešení. Buňky okolo *BorderArea* rozdělí na nepřekrývající se sekce (*SplitIntoSections*). Každá sekce je určena rodičem, což je otevřená buňka, která sousedí se všemi buňkami v sekci. K tomu slouží pole *parents*. Na začátku je každá buňka svým vlastním rodičem. Po zavolání *SplitIntoSections* se herní pole rozdělí na sekce - vezmou se všichni neotevření sousedé buňky a pokud ještě nejsou součástí jiné sekce, přiřadí se jim jako rodič aktuální otevřená buňka. To může vypadat následovně:



Nyní se budou umisťovat miny do sekcí - *FirstCombinationOnArea*. Pokud vrátí funkce true, bylo správné umístění nalezeno (a může dojít k přegenerování hracího pole, pokud se funkce volala z

tohoto důvodu). Pokud se hledají jistá políčka, bude se volat *NextCombinationOnArea*, dokud program nenajde všechny možnosti. To zajišťuje funkce *TryAll*. Protože by tato funkce mohla běžet opravdu dlouho, po 500 nalezených možnostech se ukončí, aby program nebyl příliš pomalý. Tím je způsobeno, že se občas nenajdou všechny jisté miny. To se děje pouze tehdy, když je otevřeno příliš mnoho nedořešených buněk. Proto není dobré otevírat náhodné buňky v hracím poli, ale odkrývat buňky sousedící s již odkrytými buňkami. Bylo by možné nechat program běžet a vyzkoušet všechny možnosti, potom by vždy našel všechny jisté miny. Ale usoudila jsem, že lepší hratelnost hry je přínosnější než nalezení všech jistých buněk za každých okolností.

2.5.5 Timer

Od prvního kliknutí do ukončení hry je hráči počítán čas, který se zobrazuje v pravém horním okénku.

2.5.6 Neighbours

Obsahuje funkce pracující se sousedy buňky.

2.5.7 Model

Třída na generování a přegenerování herního pole. Funkce *ReGenerate* zajišťuje přegenerování - vrací true, pokud bylo přegenerování úspěšné, v opačném případě vrací false a hra končí, protože na posledně otevřeném políčku musí být mina.

2.5.8 Open

Zajišťuje otevírání buněk - obsahuje funkce na otevření samotné buňky (čísla nebo miny) nebo oblasti okolo buňky s hodnotou 0. Dále obsahuje funkci na otevření sousedů buňky v případě, že má okolo sebe označené všechny jisté miny.

2.5.9 Solver

Hledá neotevřené jisté buňky. K tomu slouží funkce *Update*, která nejdříve opakovaně volá *UpdateCell* na všechny buňky, jejichž sousedé se nějak změnili od předchozího kliknutí. Následně volá *placeMines.TryAll* pro nalezení dalších jistých buněk.

2.5.10 Options

Okno nastavení, kde se volí obtížnost a mód hry.