

# Zápočtový program – hledání min

Vojtěch Hudeček  
Zimní semestr ak. rok 2012/2013

## Dokumentace k programu – programátorská část

### 1. Úvod

Program je napsaný v jazyce Python a používá standartní moduly. K zobrazení GUI je použit modul wxPython. Použité ikony jsem vytvořil v programu GIMP. Od klasického hledání min se liší především možností kroku zpět a kontrolou hratelnosti. Byl vyvíjen a testován na OS Ubuntu Linux.

### 2. Reprezentace dat

#### Vnitřní uložení dat

Pro každé pole existuje vlastní instance třídy **Field**, která obsahuje údaje o okolním počtu min, označených polí, odkrytých polí, počtu sousedů a dále údaje o příslušném poli. Pole jsou zobrazována pomocí komponenty wx.GenBitmapToggleButton, tedy dvoustavového tlačítka s bitmapou. Pole jsou číslována od 0 do n-1, nejsou uloženy údaje o souřadnicích. Tento způsob reprezentace je poměrně paměťově náročný, nicméně usnadňuje manipulaci s daty a odstraňuje nutnost při každé operaci s polem zjišťovat informace procházením sousedních polí. Použití bitmapových tlačítek zřejmě není optimální, protože poměrně dlouho trvá změna většího počtu polí.

Ostatní pomocné proměnné se uchovávají jako statické proměnné tříd Game a Mines.

#### Vstup a výstup

Program kromě interaktivního vstupu a výstupu pomocí obrazovky a myši využívá pouze souborů s uloženými výsledky. Pro každý možný počet min existuje vlastní soubor. Data jsou uložena pomocí modulu Pickle. Ten převádí objekty na speciálně formátované řetězce a zpět.

### 3. Datové struktury

#### 3.1. Grafické rozhraní

Zobrazení oken umožňují třídy Choice a Mines, obě dědí od wx.Frame.

Třída **Choice** zajišťuje zobrazení okna s volbou parametrů hry, obsahuje pouze komponenty pro navolení parametrů a pomocné proměnné uchovávající zvolené parametry. Obsahuje metody **SetS**, pro změnu rozměrů, a **SetC**, pro změnu počtu min. Dále metodu **Set** pro uložení parametrů a metodu **Create**, která vytvoří instanci třídy Game a Mines.

Třída **Mines** slouží k zobrazení hracího plánu a nástrojové lišty, dále obsahuje instanci třídy Game reprezentující stav hry a řadu metod zajišťujících funkčnost.

#### 3.2. Data

Třída **Game** obsahuje informace o stavu hry. To znamená, že obsahuje všechny instance třídy Field, uložené v poli, dále obsahuje seznam pozic min, spojový seznam označených polí, spojový seznam tahů a počet špatně označených polí. Dále také pomocné proměnné, a instanci třídy results s výsledky.

Třída **Results** obsahuje pouze seřazený seznam výsledků pro daný počet min.

Třídy **LinkedList** a **ListNode** slouží pro vytváření obousměrného spojového seznamu. LinkedList obsahuje metody Insert, Remove a RemoveBy, sloužící pro vkládání a mazání prvků ze

seznamu. Spojový seznam je třeba k reprezentaci možných tahů a označených polí, protože ty v průběhu přibývají i mizí.

### 3.3. Funkčnost

Funkčnost programu zajišťují metody třídy Mines. Hlavní z nich jsou:

**Back** – Provede akci krok zpět. Zmenší počet tahů, skryje poslední odkryté pole, vymaže ho ze spojového seznamu a upraví informace sousedů. V případě, že hra byla skončena, znovu spustí časomíru a umožní pokračovat ve hře změnou příznaku konce hry. Pokud je hra v situaci po prvním tahu, není krok zpět umožněn.

**Click** – Volá se po kliknutí levým tlačítkem na pole herního plánu. Pokud pole není označeno, provede se tah (jeden nebo více) pomocí metody Move. Provedené tahy se uloží do spojového seznamu, některé se případně odeberou. Pokud je tah první, spustí se časomíra a rozprostřou miny. Na konci provede kontrolu hratelnosti

**Step** – Provedení tahu nápovědy. Nápověda předá pole, kterému v okolí zbývá jen odkryt nebo označit pole a Step označí/odkryje jedno z těchto polí.

**Hint** – Volá se dvěma způsoby – buď má provést tah nebo kontrolu hratelnosti. V obou případech provede kontrolu hratelnosti v závislosti na zvolené úrovni nápovědy. Po nález provede možný tah nebo nastaví příznak hratelnosti. Pokud nenalezne žádné vhodné pole, odkryje jedno či více polí v okolí odkrytých tak, aby bylo možné hrát. Kontrola hratelnosti probíhá procházením spojového seznamu odkrytých polí. Na první úrovni se kontroluje, zda pole již nemá označeno v okolí dostatečný počet min nebo naopak, jestli již nemá odkryta potřebná okolní pole. Na druhé úrovni navíc zkusí provést označení neodkrytých okolních polí a zkontroluje, zda označení tohoto pole nemá za následek znemožnění dalších tahů.

**ShowPrev, ShowNext** – 2 téměř totožné metody volané při rozkrývání většího počtu polí (kliknutím na pole nesousedící s žádnou minou). Provedou ověření všech sousedů, pokud soused nemá v okolí žádnou minou, provede se rekurzivní volání, opět všemi směry.

**Mark** – Provede (od)označení daného pole. Změní bitmapu na tlačítku, přidá nebo odebere pole ze spojového seznamu, informuje sousedy, případně změní počet špatně označených polí. Po provedení tahu také pozmění seznam možných tahů a provede kontrolu hratelnosti. V případě, že je označeno tolik polí jako min, zkontroluje správnost a ukončí hru.

**Move** – Provede odkrytí daného pole. Pozmění seznam, informuje sousedy, zvýší počet tahů a zobrazí příslušnou bitmapu. V případě, že hráč šlápl na minu pomocí pomocné procedury zobrazí všechny miny a ukončí hru.

**Reset** – Znovu začne hru se stejnými parametry. Vynuluje časomíru, vytvoří novou instanci třídy Game.

## 4. Průběh prací, zhodnocení

Nejdříve jsem vytvořil grafické rozhraní a hru podle klasických pravidel. Poté jsem přidal zaznamenávání do spojových seznamů, což umožnilo přidat krok zpět a nápovědu. Při implementaci nápovědy jsem odhalil a opravil některé nepřesnosti v zaznamenávání stavu hry a poupravil některé metody.

Celkově se myslím podařilo splnit cíl, nicméně s některými věcmi nejsem úplně spokojen. Nejzásadnější je asi použití tlačítek, které značně zpomaluje načtení herního plánu.