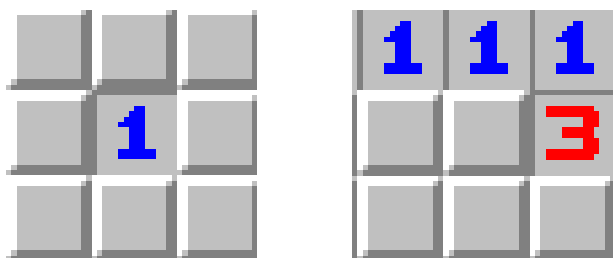


# Bot pro hru Hledání Min

Vypracoval: Jan Filip R21281

## 1. Pravidla hry

Hlavním cílem hry je odkrýt všechny miny na poli. Kdybychom je odkrývali náhodně, tak by to asi nebyla moc zábavná hra, a proto po odkrytí každého políčka se ukáže, kolik je kolem tohoto políčka, v okruhu jednoho políčka, min.



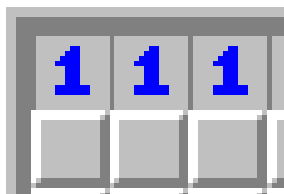
Z prvního obrázku můžeme vyčíst, že jedno z osmi políček bude obsahovat bombu, nevíme ale stoprocentně které z políček to bude. Z druhého obrázku ale můžeme vyčíst, že bomba bude na prostředním políčku, protože pravé horní políčko má stejný počet neprozkoumaných políček jako bomb, které okolo něj jsou.

To je v podstatě vše co potřebujete vědět, abyste mohli začít hru hrát.

## 2. Pokročilé taktiky

Hra je sice jednoduchá, má však pár taktik, které lze implementovat, aby jste museli co nejméně hádat a mohli tak vyhrávat více her.

### 2.1. Logická dedukce

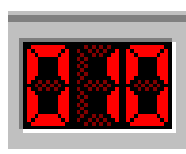


Tento obrázek by podle základních vědomostí nešel vyřešit, avšak můžete si uvědomit, že nám jednička v levém horním rohu říká, že v políčkách okolo je jedna bomba.



Pokud se poté podíváme na prostřední jedničku, můžeme vydedukovat, že pokud je v políčkách označenými černými X sto procentně jedna bomba, značí to, že na políčku označeným zelenou fajfkou bomba být nemůže.

## 2.2. Počet bomb



Hra po celou dobu hraní zobrazuje počet bomb, které ještě můžete najít. To sice může vypadat jako pouhý ukazatel pokroku, ale ve finální části hry se může počet bomb využít i takticky.

Pokud máte například jednu neodkrytou minou, a víte že je v okolí nějakého políčka, tak si můžete být jistí, že zbytek neodkrytých políček je stoprocentně bezpečný.

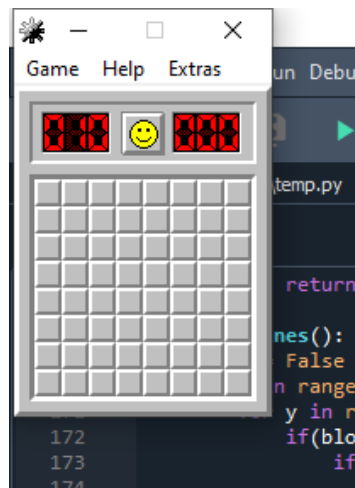


Na tomto obrázku můžeme vidět, že ač se koukáme z jedniček či z dvojky, tak máme šanci 50 na 50, že trefíme bombu. Protože nám ale chybí pouze jedna bomba, tak nám může dojít, že v pravém dolním rohu bomba stoprocentně nebude, tudíž políčko můžeme odkrýt.

## 3. Fungování bota

### 3.1. Příprava

Na začátku programu se hra zapne a přesune na souřadnice  $x=0$  a  $y=0$ , což odpovídá levému hornímu rohu. Zde je proto, aby se program mohl řídit podle jednotlivých pixelů (jednotlivé políčka mají rozměry 16x16 pixelů). Deklarují se zde také všechny knihovny, proměnné a obtížnost hry.



### 3.2. Průběh hry

#### 3.2.1. Začátek

Bot začíná políčkem v pravém horním rohu. Po kliku zjistí (podle barvy pixelu) číslo políčka, na které právě klikl a číslo se zapíše do dvourozměrného pole, pro napodobení hrací plochy v počítači. Pokud na políčku číslo není, znamená to, že se otevřelo více políček najednou, tudíž se prohledá okolí všech prázdných políček, které se nově otevřely.



```
[[9 9 1 0 0 0 0 0]
 [9 9 1 0 0 0 0 0]
 [9 9 1 1 1 1 0 0]
 [9 9 9 9 9 1 0 0]
 [9 1 1 2 1 3 0 0]
 [9 1 0 0 0 0 0 0]
 [9 1 1 2 1 2 2 2]
 [9 9 9 9 9 9 9 9]]
```

### 3.2.2. Výpočet šance

Po zjištění všech čísel se projede dvourozměrné pole a ke každému číslu se vypočítá pravděpodobnost, že je v okolí bomba pomocí tohoto vzorečku.

$$(((\text{Číslo-počet bomb v okolí}) / \text{Číslo}) / \text{prázdné políčka v okolí}) * \text{Číslo}$$

Tato pravděpodobnost se poté přiřadí, v jiném dvourozměrném poli, prázdným políčkům v okolí čísla. Pokud je však šance 1, znamená to, že víme, že na políčku je bomba. Naopak pokud je šance 0, tak na políčku stoprocentně bomba není a hned na něj klikne.

```
[ [0.  0.5  0.  0.  0.  1.  0.25 0. ]  
[0.33 0.5  0.  0.  0.  0.  0.33 0. ]  
[0.  0.33 0.  0.  0.  0.  0.33 0. ]  
[0.29 0.33 0.  0.  0.  0.  0.33 0. ]  
[0.29 0.  0.33 0.33 0.33 0.33 0.  0.14]  
[0.29 0.29 0.29 0.  0.  0.14 0.14 0.14]  
[0.  0.  0.  0.  0.  0.  0.  0. ]  
[0.  0.  0.  0.  0.  0.  0.  0. ] ]
```

### 3.2.3. Nejmenší šance

Pokud nemáme na výběr a žádné políčko neumíme stoprocentně určit, nestačí nám udělat nic jiného, než kliknout na políčko, které má nejmenší pravděpodobnost toho, že je pod ním schovaná bomba. Políčko zjistíme hledáním nejmenší pravděpodobnosti v dvourozměrném poli.

Lze ale zjistit pravděpodobnost všech zbývajících políček, které jsme ještě ani nepotkali tímto vzorečkem.

$$(\text{celkový počet bomb-počet najitých bomb}) / \text{počet volných políček}$$

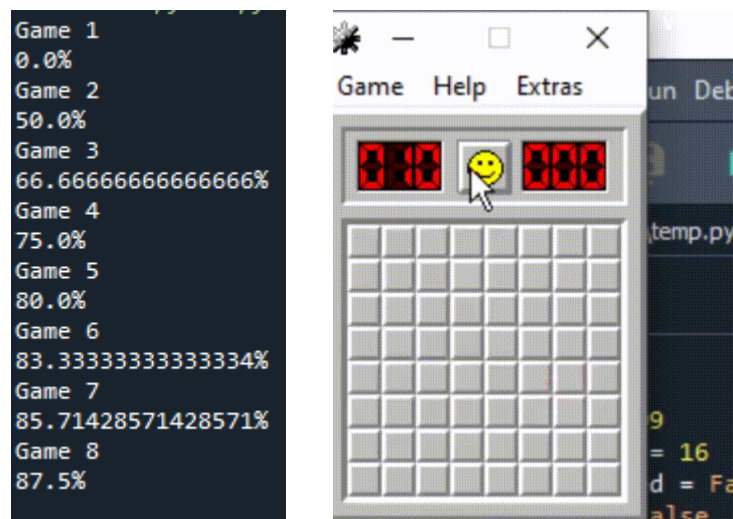
Pokud tedy zjistíme, že je pravděpodobnější, že náhodným políčkem máme větší šanci na přežití, tak klikneme na první ještě neobjevené políčko.

### 3.2.4. Kontrola smrti

Jelikož má číslo 7 stejnou rgb hodnotu jako bomba, program zjišťuje jestli je naživu podle emoji nad hrací plochou, které mění výraz podle toho co se děje na hrací ploše. Pokud se zjistí, že hra byla prohraná nebo vyhraná, tak přestane hrát.

### 3.2.5. Opakování hry

Program je v loopu s základní hodnotou počtu opakování na 50 her. Po každém opakování se píšou statistiky do pythonové konzole.



## 3.3. Statistika a rekordy

Rychlostně je program dostačující, v každé obtížnosti je třikrát rychlejší než rekordy dobrých hráčů hledání min. Světové rekordy jsou však 1, 8 a 29 sekund, kde můj program porazil pouze expertní rekord.

Fastest Mine Sweepers		
Beginner:	1,99 seconds	AI
Intermediate:	12,50 seconds	AI
Expert:	28,85 seconds	AI

Kvůli jednoduché logice, rekordů nedosahuje ani v poměru výher. Beginner vyhrává přibližně 55% času proti 81% vyřešitelných her. Intermediate má 46% ku 60% a v expertovi program vyhrává v pouhých 3,8% her ku normálním 34%.

## 4. Možné zlepšení

### 4.1. Implementace pokročilých taktik

Program obsahuje pouze základní znalosti hry, tudíž protože složitější problémy neumí vyřešit, tak častěji hádá, což se také odráží na jeho podprůměrnému poměru výher.

### 4.2. Přepsání v jiném programovacím jazyce

Celý program je napsaný v Pythonu, což mu dodává špetku jednoduchosti kódu, zapisuje se to ale na jeho rychlosti, protože je pomalejší oproti jiným programovacím jazykům.

### 4.3. Optimalizace

V programu jsem až tak nedbal na optimalizaci. Kdybych měl více času a méně jiných věcí na práci, tak bych program vylepšil, aby fungoval rychleji a efektivněji.