

Univerzita Karlova v Praze  
Matematicko-fyzikálna fakulta

# Bakalárska práca

Eva Pešková

Codewars, vojna robotov

Katedra softwarového inžinierstva

Vedúci bakalárskej práce: Mgr. Tomáš Poch

Všeobecná informatika

2009

Prehlasujem, že som svoju bakalársku prácu napísala samostatne a výhradne s použitím citovaných prameňov.  
Súhlasím so zapožičiavaním práce.

V Prahe 14. mája 2009

Eva Pešková

# Obsah

# Kapitola 1

## Uvod

## Kapitola 2

# Analýza

### 2.1 Výber knižnice SDL

Knizncia SDL bola vybraná z troch dôvodov : Je pomerne prenositeľná a priamo určená pre vytváranie 2D hier.

TODO Výhody, nevýhody, citácie zo stránky

Kedže súčasťou tejto práce bolo naimplementovať prostredie pre botov, kde sa predpokladá minimálne množstvo (oproti iným hrám) volieb, formulárov apod., ktoré sa v SDL nepodporuje, nebolo treba siahnuť k iným knizniciam, ktoré majú podobné vlastnosti ako SDL.

TODO porovnanie knižníc

## **2.2 Codewars a jeho alternativy**

### **2.2.1 Robowars**

### **2.2.2 Pogamut**

## **2.3 Problem optimalnej strategie viacerych hracov**

## **2.4 Prakticke vyuzitie**

## Kapitola 3

# Implementácia prostredia

### 3.1 Hracia plocha

#### 3.1.1 Generovanie map

#### 3.1.2 Vlastnosti stien

### 3.2 Jazyk hry a priebeh hry

Po načítaní všetkých robotov, ich programov sa hra kladá z trojice (FrontaUdalosti f, Mapa m, ŽiviBoti b). Samotná hra prebieha tak, že pokiaľ existujú živí boti, spracuje sa nasledujúca udalosť z FrontyUdalosti. O zaradenie späť a prípadné vymazanie sa v prípade úmrtia sa objekty starajú sami.

Objekty = steny, boti, strely.

Stav robota vzhľadom na svet charakterizuje šesťica: (InstrukcionStack IS, InstructionPointer IP, Hitpoints HP, CoordX x, CoordY y, Direction direct)

**Značenie 1.** Pod objektami sveta rozumieme všetky prvky, ktoré svet tvoria a zároveň ho z času na čas menia, t.j. v tomto prípade strely, steny, a samotní roboti. Napríklad strela mení svet tým, že sa pohne alebo narazí, istý druh steny sa môže posunúť a pod.

**Značenie 2.** Tick je jednotka virtuálneho času sveta. Timeout je počet tickov, ktoré má robot k dispozícii na premýšľanie a tento čas nesmie prekročiť.

**Značenie 3.** Svet je trojica (FrontaUdalosti, Bojisko, Aktívni roboti), kde FrontaUdalosti je fronta objektov, ktoré sú na rade, aby zmenili časť sveta. Presný spôsob definovania poradia je popísaný neskôr.

Popis jednotlivých inštrukcií a ich vplyv na svet je symbolicky popísaný takto: (bez parametrov, ktoré nemajú vplyv na svet):

prikazy komunikujúce so svetom:  $step \left\{ \begin{array}{l} Svet : < f.pop() + reschedule(objekt), m.reinsert(objekt) > \\ Bot : < IS, IP + 1, HP, x + direct.y, y + direct.y, direct > \end{array} \right.$

$turnLeft \left\{ \begin{array}{l} Svet : < f.pop() + reschedule(objekt), m, b > \\ Bot : < IS, IP + 1, HP, x, y, NewDirect > \end{array} \right.$

$turnRight \left\{ \begin{array}{l} Svet : < f.pop() + reschedule(objekt), m, b > \\ Bot : < IS, IP + 1, HP, x, y, NewDirect > \end{array} \right.$

$turn \left\{ \begin{array}{l} Svet : < f.pop() + reschedule(objekt), m, b > \\ Bot : < IS, IP + 1, Hitpoints, X, Y, NewDirect > \end{array} \right.$

$see \left\{ \begin{array}{l} Svet : < f.pop() + reschedule(objekt), m, b > \\ Bot : < IS, IP + 1, Hitpoints, X, Y, NewDirect > \end{array} \right.$

$shoot \left\{ \begin{array}{l} Svet : < f.pop() + reschedule(objekt, Strela), m, b > \\ Bot : < IS, IP + 1, Hitpoints, X, Y, NewDirect > \end{array} \right.$

Ďalšie inštrukcie vôbec nekomunikujú so svetom, preto sú uvedené oddelene. Tieto inštrukcie robot vykonáva počas svojho premýšľania. Keďže tým robot vôbec nepotrebuje vedieť o objektoch sveta iných než je on sám, po prevedení takýchto inštrukcií sa mení stav robota takto:

### 3.2.1 Priebeh penalizácie za inštrukciu

Obrázok 3.1: Premýšľanie robota

### 3.2.2 Detekcia zacyklenia

## 3.3 Vlastnosti robotov

Výsledok programu, ktorého inštrukcie robot vykonáva, je závislý na konkrétnom type robota. Typ robota je definovaný niekoľkými parametrami, ktoré si bežný užívateľ samostatne na začiatku hry definuje == UholViditelnost, PolomerViditelnosti, Obrana, Hitpoints, TypStrely, VelkostPamate==

Užívateľ si typ strely definuje tak, že dostane k dispozícii určitý počet bodov a tie musí medzi jednotlivými voľbami rozdeliť



Tabuľka 3.1: Vlastnosti robota

Vlastnosť	Vplyv
polomer viditeľnosti	Robot dokáže na určitú vzdialenosť rozpoznať object, táto vlastnosť určuje, koľko políčok dopredu v priamom smere( v smere, akým je bot aktualne otočený) vidí. Tento počet sa pod iným uhlom o niečo zmení,
uhol viditeľnosti	maximálny rozptyl viditeľnosti - robot nevidí celý kruh okolo seba, ale iba určitú výseč
Obrana	Obranne číslo bota. Používa sa pri strete s nejakou prekážkou. Viz kolízie.
Hitpoints	životnosť bota, akonahle klesne na nulu, z fronty akcií sa jeho nasledujúca akcia odstráni a robot mizne do prepadlišta dejin
[1ex] Typ strely	Strela je sa definuje podobne ako samotný bot, ale ma iné parametre

Tabuľka 3.2: Vlastnosti Strely

Vlastnosť	Vplyv
Odrážavosť	definuje, ako moc sa strela od steny odrazí, v podstate je to to iste ako hmotnosť
Rýchlosť	definuje preplanovanie vo fronte akcií
Hitpoints	dolet strely, ten sa znižuje s počtom tikov a súčasne kolíziami
Utok	Utočné číslo bota, používa sa pri utoku, viz kolízie

Bot premýšľa tak, že sa nechá bežať dovtedy, kým z jeho programu nenarazí na inštrukciu, ktorá by ovplyvnila svet, alebo kým nepresiahne timeout. Ak presiahne timeout, preruší sa, preplánuje vo fronte udalostí ďalej (presnejšie o TIMEOUT tikov ďalej)

Velmi špeciálny je prípad, keď užívateľ nenastaví robotovi žiadnu pamäť. Takýto robot nie je schopný si zapamätáť jedinou premennú a ani vráťť návratové hodnoty, teda v kóde programu by nemali byť definované žiadne funkcie, ale iba procedúry. Je nutné povedať, že funkcie a procedúry sa v žiadnom prípade nechovajú ako premenné a teda nezabierajú robotovi pamäť. Teda v tomto prípade dojde k masívnemu využitiu rekurzií. Samotne vstavané príkazy sú však vlastne funkcie, ale keďže sa nikam nemajú priradiť, ich návratová hodnota sa zahodí. Nevýhodou tohoto typu bota je však skutočnosť, že nemá povolené napríklad strieľanie, pretože to je funkcia, ktorá potrebuje za každých okolností parametre.

## Kapitola 4

# Sietova komunikacia

## Kapitola 5

# Výber a popis testovaných algoritmov

5.1 Algoritmus 1

5.2 Algoritmus 2

5.3 Algoritmus 3

**Kapitola 6**

**Testovanie**

**Kapitola 7**

**Vylepšenia**

**Kapitola 8**

**Záver**

# Literatúra