

# Game AI – praxe (r. 2002)

- Game AI is behavioral, not scientific. ⇒ Nejde o to vytvořit inteligenci, ale zlepšit hratelnost skrze iluzi intelligence.

## **Základní problém:** Plánování a rozhodování

### Nejčastěji řešené problémy

- volba cílů a akcí (decision making) ⇒ SDK
- hledání optimální cesty (pathfinding) ⇒ 3D

### Drobnější problémy:

- realistické rozhovory (přirozený jazyk)
- sportovní komentáře
- generování map
- automatická analýza map
- a řada dalších viz prof. Laird

### Jak se řídí chování jednotek, botů a jiné havěti:

- hard-coded (maximálně lze mít 90 vojáků, po 45 kusech rolníci nosí jen poloviční náklad surovin)
- konečné automaty
- expertní systémy
- skripty

Dobrá zpráva: CG už bylo dost, AI je na řadě.

## **NPC Conversations**

- **Text Parsing** – Player needs to think about problems, rather than just walk through all the possibilities.

ASK THE MAN ABOUT THE WHALE

"Oh, the whale. He's a real killer alright. Nary a man sails the sea without keeping one eye out for the whale."

ASK THE MAN ABOUT A BOAT

"A boat? Surely you're not thinking of sailing? You're crazier than I thought!"

TAKE THE WINE

"I don't know what a WINE is"

TAKE THE FLASK

"I don't know what a FLASK is"

TAKE THE DAMN BOTTLE

"I don't know what a DAMN is"

TAKE THE BOTTLE

"You have taken the bottle."

## **- Menu-Driven Conversations**

Angry Orc: "Hey! Nobody gets in here without the password."

A. "Yes. The password. I know it."

B. "The password? Oh, I forgot it. It's downstairs. I'll be right back."

C. "Password? I don't need no stinkin' password!"

## **- Hybrid Conversations**

## Pathfinding

- Dijkstrův algoritmus (nepřiliš častý)
- A\* (v naprosté většině RTS včetně Warcraft II)
- Floodfill algorithm – 'Ant Races'

Velmi pěkná demo na A\* algoritmus lze najít mezi odkazy na stránce UIR.

A\* nelze spouštět při každém pohybu jednotek/botů ⇒ uchovávat v tabulce ⇒ problém s paměťovou složitostí ⇒ hierarchical pathfinding (room region, floor region, building region, street region, city region)

## Waypoints<sup>1</sup>

- Boti ve FPS hrách umí „opisovat“ od lidského hráče (jeden z mnoha: RealBot)
- Podobně lze použít i buňky nebo grid.
- Bounding box (vyhnutí se rohům)

**The local environment.** Waypoints describe, for example, how dark the location is, which types of movement are required (crouching, swimming, using a ladder), whether a door or button entity is present.

**The membership(s) of higher-level terrain concepts.** Waypoints may be grouped to represent terrain concepts such as a room, a lake, or the blue team's base.

**The relations with other waypoints.** Waypoints have relations with other waypoints: is there a valid line-of-sight, how long does it take to get from one waypoint to the other.

**The focus** (in the relations with other waypoints). A waypoint whose relations, such as line-of-sight or easy access, are clustered primarily in a single narrow direction has focus in that direction.

---

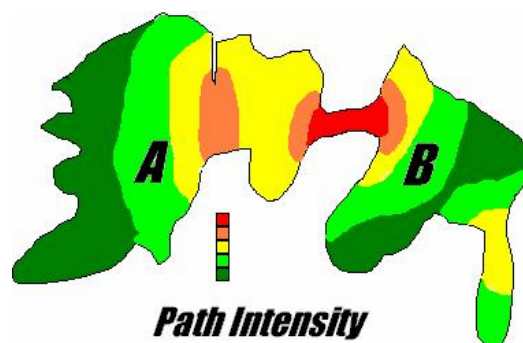
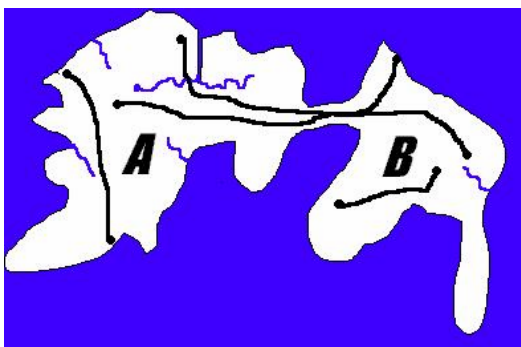
<sup>1</sup> Uživatel může vyznačit cestu svých jednotek na víc kliknutí (dvě výhody: rychlejší výpočet, lze se vyhnout nebezpečí).

## Terrain/map understanding (důležitost Panamské šíje)

### Zkušenosti:

- Vyhrává hráč, který uhlídá důležitý power-up popř. zbraň ⇒ viz profesionální hráči.
- Např. v CTF módu je třeba identifikovat bottle-neck – strategické místo přes které je třeba projít (lze počítat počet cest do báze, které bot z daného waypointu uhlídá... pozn.: line-of-sight = line-of-fire)

1. Determine all contiguous map regions using a floodfill algorithm
2. For each contiguous map region calculate:
  1. its size in squares
  2. its "center of gravity" (this need not actually be in the feature)
  3. other regions that abut it (using edges recorded above)
3. Sub-divide each large contiguous feature into regions. This is difficult to do in a "human" way but I think it can be done reasonably. The key idea is to use a shortest
4. Now do for regions what we did for contiguous areas, calculate:
  1. its size in squares, figure edge squares for each region
  2. its "center of gravity" (this will now most likely be in the feature)
  3. other regions that abut it including necks (using edges recorded above)



**Spojování (grouping) jednotek** – lze mírně modifikovat cest, aby si jednotky nepřekážely

**Stavba silnic** – která místa na mapě se často vyskytují na cestách

**Jak si poradit s pohybujícími se překážkami** – přepočítávat cestu (během pohybu jednotky) nebo zajistit, aby změna na mapě resp. cestě spustila trigger a přepočítání cesty.

**Predikce pohybu překážky** – příliš obtížné, nepoužívá se. (a sám to neumím)

## **Line of Sight (LoS) problem (triky)**

- Reflexivní LoS
- LoS drahý výpočet pro terén bez překážek  
(  $N * (N-1) * \text{AverageRange}$  )
- Lze rozdělit atribut "can see" a atribut "has LoS"  
(tmavé místo nebo nehlučné jednotky  $\Rightarrow$  ani jeden nemá "can see", ale oba mají LoS)
- Pro spřátelené jednotky netřeba počítat LoS nebo "can see" (většinou).
- Netřeba počítat dál, než je dosah „útočníka“ a „napadaného“
- Využívat vlastností terénu (místnosti oddělené dveřmi).
- Uchovávat výsledky jako bitové vektory
- Někdy se vyplatí ukládat předpočítané hodnoty společně s waypoints

# Bots

Hard coded vs. použití stand-alone SDK pro expertní analýzu (Decision Making System – DMS)

SOAR (viz SOAR tutorial 1–5 ... AI pro hru PacMan a AI pro Quakebot)

Tactics outline (založeno na Wants & Needs)

- Collect-Powerups
  - Pick up visible weapons that are useful to the bot
  - Get health/armor if low on health/armor
  - Pickup up other good weapons/ammo if they are close by to deny them to the enemy
- Attack
  - Use circle-strafe
  - Move to best distance for current weapon
  - Jump during an attack to avoid getting hit
- Chase
  - Turn toward an enemy based on sound of running
- Wander
  - Move around a room and if a door is detected, go through it.

RealBot – model chování hráče

- **Emoce** (strach, agresivita, sebedůvěra, radost/šťěstí a frustrace)
- **Touhy** (splnění úkolu, týmová práce, camping)
- **Zkušenosti** (místnosti kde je mnoho mrtvých jsou označeny jako „nebezpečné“, ale také „důležité“ kde hráč kempuje, kde byl bot zabit, kudy hráč chodí nejčastěji, kudy nechodí vůbec)

## Emaptie

Q: Co dělá hráč, který právě zaběhl za roh?

A: Dělá to co bych dělal já (bot) se stejným vybavením, stejným power-upem...

## Operation Flashpoint

– waypoints, sensors, triggers

## Jak začít ?

Botman (HL & CS guru)

<http://www.planethalflife.com/botman/>

<http://www.valvesoftware.com/hlSDK.htm> (HalfLife SDK 2.2)

## Script vs. program

```
Amit [to Steve]: Hello, friend!  
Steve [nods to Bryan]: Welcome to CGDC.  
[Amit exits left.]
```

Note that it's very high level: it doesn't specify exactly how Amit speaks or how Steve nods or even the timing. For example, the conversation might look like this:

```
Amit.turns_towards(Steve);  
Amit.walks_within(3);  
Amit.says_to(Steve, "Hello, friend!");  
Amit.waits(1);  
Steve.turns_towards(Bryan);  
Steve.walks_within(5);  
Steve.nods_to(Bryan);  
Steve.waits(1);  
Steve.says_to(Bryan, "Welcome to CGDC.");  
Amit.waits(3);  
Amit.face_direction(DIR_LEFT);  
Amit.exits();
```

## Wargames

"DMS" – "decision-making-system".

DMS může být

- jednoduchá neuronová síť
- hard-coded pravidla
- fuzzy logika
- vyhledávací tabulka
- nebo cokoliv jiného

Je třeba aby DMS byl jednoduchý (málo vstupů málo výstupů) a „trackable“. Nelze tedy kontrolovat a vyhodnocovat pozici každého vojáka. Řešení: **hierarchický DMS**.

- Lokální DMS přijímá strategické informace od globálního DMS a interpretuje je na taktické úrovni.
- Tedy: Jak se informace „posílají nahoru“ zobecňují se, když rozkazy postupují „dolů“ stávají se konkrétnějšími.
- Dobrý příměr je řetěz rozkazů v armádě
- The Clash of Civilizations
- Taktická část AI se aktivuje mnohem častěji.

---

Strategie a taktika jsou synonyma, jen stojí v jiných úrovních.

Strategie = politická rozhodnutí v závislosti na okolních státech, aliancích, osobnosti nepřátelských vůdců, etice a morálce (použít/nepoužít nějaký typ zbraně), timing, kvalita logistiky, pozice v boji, peníze, stav bitvy...



## Grouping

- jednodušší handskake při udílení rozkazů
- při pohybu jednotky se lze lépe vyhýbat překážkám
- flexibilní tým, odolný vůči různým typům odporu
- snazší koordinace útoku

## Level of Detail

### Influence Mapping

### Loyalty tracking

### Morálka, módy a vzorce chování

OOP – Vysílače (hladový simík, lednice vysílá, že má jídlo, jídlo vysílá, že má být ohřáto)

## Další techniky

- Šablony útoků oponenta
- Databáze posledních úspěchů a neúspěchů
- Bag of tricks

## Best of the Web

<http://www.gameai.com/>

<http://www.gamasutra.com/>

<http://www.gamedev.net/gamedev.asp>

<http://www.cris.com/~swoodcoc/ai.html>

<http://www-cs-students.stanford.edu/~amitp/gameprog.html>

<http://xenon.stanford.edu/~amitp/gameprog.html>

<http://yoda.cis.temple.edu:8080/UGAIWWW/lectures95/search/alpha-beta.html>

<http://news.alife.org/>

## Best of the Newsgroups

[rec.games.programmer](mailto:rec.games.programmer)

[rec.games.chess](mailto:rec.games.chess)

[rec.games.go](mailto:rec.games.go)

[rec.games.\\*](mailto:rec.games.*)

[comp.ai.alife](mailto:comp.ai.alife)

[comp.ai.games](mailto:comp.ai.games)