

Semestrální práce z ISW

Úprava bota pro počítačovou hru Counter Strike

Petr Povolný

20. 6. 2003

<http://povolny.wz.cz/skola/isw>

Obsah

1.	ZADÁNÍ	3
2.	OBECNÉ INFORMACE	3
3.	JAK PODBOT FUNGUJE?	4
4.	ZMĚNY Z HLEDISKA UŽIVATELE – UŽIVATELSKÁ DOKUMENTACE	6
5.	ZMĚNY Z HLEDISKA KÓDU – PROGRAMÁTORSKÁ DOKUMENTACE	7
6.	ZÁVĚR.....	10
7.	PRAMENY.....	11

1. Zadání

Vytvořte zcela nového nebo doplňte již existujícího bota, do některé z následujících her:

- Quake2
- Quake3 nebo mod Urban Terror
- Counter Strike = mod pro Halflife

Po předběžném prozkoumání možností a seznámení s jednotlivými hrami jsem chtěl pracovat na doplnění a vylepšení bota pro Urban Terror, protože se mi z nabízených her jevil jako nejhratelnější. Bohužel se nepodařilo získat jeho zdrojové soubory, a proto jsem nakonec pracoval na botu pro Counter Strike.

Mým hlavním snažením bylo "naučit" tohoto bota pohybovat se ve dvojicích.

2. Obecné informace

Boti pro Counter Strike existují jako dynamicky linkované (DLL) knihovny. Vývojář bota tak nemusí mít k dispozici kompletní zdrojový kód hry (modu), ale stačí zdrojové kódy DLL knihovny. Ta musí exportovat funkce, které potom volá hra.

Pokud by chtěl někdo napsat kompletního bota od základu, musel by vědět, jak se tyto funkce musí jmenovat, aby komunikace mezi enginem hry a DLL knihovnou fungovala správně. Já jsem se však rozhodl pouze obohatit již existujícího bota, a proto jsem se nemusel těmito věcmi zabývat.

Po prozkoumání situace na internetu jsem se rozhodl pro "rozšíření" bota pojmenovaného PODBot (Ping of Death Bot), konkrétně verzi PODBot26. Tento bot je volně dostupný na stránkách www.planethalflife.com/botman/. Na stejné adrese je k dispozici nepřeberné množství informací o tvorbě botů a to nejen pro Counter Strike.

Archiv podbot26_src.zip

Aresář Botman_Readme

Obsahuje pouze soubor readme.txt, který obsahuje základní informace o tom jak bot funguje, jak zjišťovat informace se kterými chceme při vývoji pracovat apod.

Adresář Podbot

Obsahuje vlastní zdrojové soubory. Ty jsou uloženy v dalších několika adresářích: Common, Dlls, Docs, Engine, PM_SHARED. Z těchto adresářů jsem využil pouze adresář Dlls, který obsahuje Visual C++ projekt DLL knihovny.

Soubor Readme_first.TXT

3. Jak PODBot funguje?

Obecně

Na začátku hry (každého kola) je provedena funkce `BotSpawnInit()`, která inicializuje všechny potřebné struktury, popřípadě bota teprve vytvoří (pokud začíná úplně první kolo).

Bot má na svoje akce každou vteřinu k dispozici 30 framů (ve smyslu časového úseku). Na začátku každého framu je spuštěna funkce `startFrame()`, která volá pro každého bota funkci `BotThink()`, což je hlavní funkce inteligence bota.

Právě funkce `BotThink()` se stará o zjišťování stavu bota (živý/mrtvý/zraněný) a provádění na něm závislých reakcí. V případě, že má bot zadány nějaké úkoly (tasks), stará se o jejich provedení atd. S trochou nadsázky se dá říci, že všechny ostatní klíčové funkce se volají v těle této rutiny nebo v tělech jí volaných rutin.

Na konci kola je provedena funkce `updateGlobalExperienceData()`, která vyhodnotí úspěšnost bota a nastaví podle toho příslušné dovednosti, např. zaznamená nebezpečná místa, kde byl bot zabit.

Jak boti komunikují?

Základní prostředkem pro komunikaci jsou fronty zpráv. Existují funkce pro vložení zprávy do fronty a pro její vyjmutí. Zprávy mohou být adresné (pro konkrétního bota) nebo obecné a pracují asynchronně.

Používané funkce:

```
BotGetMessageQueue(bot_t *pBot)
BotPushMessageQueue(bot_t *pBot, int iMessage)
BotCheckMessageQueue(bot_t *pBot)
```

Úkoly

Boti mohou mít zadány úkoly, které se snaží vyplnit. Například zadání (ve hře) požadavku "Cover me" použitím vysílačky vyústí v uložení zprávy do fronty pro všechny boty v "doslechu". Pokud některý z kolemstojících botů tuto zprávu "pozitivně" vyhodnotí, přidělí se mu úkol `TASK_COVERME`. Ten pak plní dokud není splněn nebo pokud mu není zadán další úkol. Všechny zadané úkoly se ukládají do zásobníku, což znamená, že bot vždy plní poslední zadaný úkol. Po jeho dokončení začne plnit předposlední úkol atd.

Splnění úkolu může vypadat různě. Například u úkolu `TASK_FOLLOWUSER` (provádí radio-příkaz "Follow me") je vyhodnocováno, zda bot může svého kolegu stále následovat, tzn. jestli je naživu, jestli je v dohledu apod.

Používané funkce

```
BotGetSafeTask(bot_t *pBot)
BotPushTask(bot_t *pBot, bottask_t *pTask)
BotRemoveCertainTask(bot_t *pBot, int iTaskNum)
BotRemoveCertainTask(bot_t *pBot, int iTaskNum)
BotTaskComplete(bot_t *pBot)
```

Pohyb bota

Bot má několik základních parametrů, které určují jakým směrem a jak rychle se pohybuje. Tyto parametry se v DLL vlastně jen nastavují jejich zpracování a "provedení" si pak zajišťuje přímo engine hry.

Příklad

float f_max_speed;	maximální rychlost bota se zbraní, kterou drží
float f_move_speed;	aktuální rychlost pohybu vpřed (vzad – záporná hodnota)
float f_sidemove_speed;	aktuální rychlost do strany (úkroky)

Obecně se boti pohybují po předem známých "waypointech". Mají předdefinované útočné waypointy, obrané waypointy atd. pro každou mapu. Pokud tyto waypointy pro danou mapu neexistují, není ani možné boty do takové mapy přidat.

Používané funkce

```
BotFindWaypoint( bot_t *pBot )
GetValidWaypoint(bot_t *pBot)
BotFindDefendWaypoint(bot_t *pBot, Vector vecPosition)
BotFindCoverWaypoint(bot_t *pBot, float maxdistance)
GetBestNextWaypoint(bot_t *pBot)
```

Vlastnosti

Bot má velké množství vlastností a flagů. Ty jsou sdruženy ve struktuře bota bot_t.

Příklad

float fBaseAgressionLevel;	základní úroveň agresivity bota
float fBaseFearLevel;	základní úroveň strachu bota
float fAgressionLevel;	aktuální úroveň agresivity bota
float fFearLevel;	aktuální úroveň strachu bota
boolean bDead;	určuje, zda je bot ještě na živu

Podle hodnot těchto atributů bot reaguje na většinu podnětů ze svého okolí, provádí rozhodnutí o svém pohybu, střelbě nebo vyhledává nejbezpečnější úkryt.

4. Změny z hlediska uživatele – uživatelská dokumentace

PODBot byl obohacen o týmový pohyb po dvojicích. Nová vlastnost se nazývá multibot. Pokud je tento mód aktivní (viz níže) boti se chovají následujícím způsobem:

- 1) Na začátku kola je vždy provedeno zpárování botů v každém z týmů.
- 2) Jeden z dvojice je **leader** – ten se vlastně pohybuje normálně. Druhý z dvojice, tzv. **follower** se snaží svého leadra následovat. Pokud jej vidí pohybuje se po jeho boku nebo těsně za ním. Pokud leadra nevidí je jeho primárním úkolem najít jej.
- 3) Pokud je jeden z dvojice zabit, zbylý člen týmu pokračuje sám do doby než se mu naskytne možnost vytvořit nový pár. To je možné pokud zbude někdo z jiného páru nebo pokud je počet botů v týmu lichý.

Instalace

V adresáři hry **HalfLife** by měl existovat adresář **cstrike**. Do něj nainstalujte některého z podbotů, např. soubor **pod25ins.exe**. Tím se v adresáři **cstrike** vytvoří adresář **PODBot**. Do něj ručně nakopírujte soubory **PODBot.dll** a **podbot.cfg** z archivu **multibot.zip** (není podmínkou - v něm je pouze multibot přednastaven, aby se aktivoval ihned po startu).

Aktivace vlastnosti multibot

1) konzolový příkaz

Vepište do konzole (de/aktivuje se klávesou ~) příkaz "multibot on", pokud chcete vlastnost aktivovat. Pro její vypnutí použijte "multibot off".

2) soubor podbot.cfg

V souboru **podbot.cfg** v adresáři **podbot** nastavte "multibot on/off" podle stejné logiky jako v konzole. Pokud použijete soubor **podbot.cfg** z archivu **multibot.zip**, stačí najít místo, kde je vlastnost multibot uvedena a nastavit ji na požadovanou hodnotu.

Známé problémy

- Někdy se stane, že jeden pár pobíhá v jedné uzavřené oblasti, aniž by leader dvojice měl nějaký určitý směr. Jedná se o chybu vlastního bota a ne vlastnosti multibot, neboť toto chování se občas vyskytuje i při běžné hře v souvislosti s příkazem "Follow me".
- Při aktivaci vlastnosti multibot v průběhu rozehraného kola se teoreticky může stát, že se nespárují všichni boti, přestože jich je sudý počet. V takovém případě vepište do konzole příkaz "killbots" nebo "newround", což způsobí start nového kola a správnou aktivaci párů.

5. Změny z hlediska kódu – programátorská dokumentace

Práci jsem vyvíjel a testoval s následujícími verzemi softwaru:

- o Halflife 1.1.1.0
- o Counter Strike 1.5 Full
- o instalace PODBoťa - **pod25ins.exe**
- o zdrojové soubory PODBoťa – **podbot26_src.zip**

Obecně

Všechny úpravy, které jsem ve zdrojových souborech DLL knihovny provedl jsou uvozeny komentářem:

```
// P.P. YYYY-MM-DD
```

kde	P.P.	= Petr Povolný
	YYYY	= rok (2003)
	MM	= měsíc (06)
	DD	= den (04-19)

Více řádkové úpravy jsou ohraničeny zakomentovanými hvězdičkami, jednořádkové pouze výše uvedeným označením.

Je tedy možné nechat si vyhledat všechny výskyty řetězce "P.P." a poté zkoumat pouze mnou provedené úpravy.

Pro svou práci jsem do projektu knihovny přidal dva soubory: **multibot.cpp** a **multibot.h**. Upraveny byly následující soubory: **bot.cpp**, **bot.h**, **bot_client.cpp**, **dll.cpp**.

Základním stavebním kamenem mého obohacení je doplněná funkce **multibotSpawnInit()**, která provádí párování botů. Detailní popis viz níže.

Byl přidán úkol **TASK_MULTIBOT**, který dostane zadán bot, který následuje svého kolegu. Tento úkol používá stejně jako např. úkol **TASK_FOLLOWME** funkci **BotFollowUser()**, která však byla lehce upravena.

Párové chování botů lze aktivovat/deaktivovat konzolovým příkazem "multibot on/off". Většina přidaného kódu pak začíná ifem, který zjišťuje, zda je multibot aktivován či nikoliv.

Párové chování lze také nastavit v konfiguračním souboru **podbot.cfg**.

Úpravy

multibot.h

- 1) deklarace struktury multibot

```
typedef struct multibot_t {  
  
    bool leader;                // is the bot LEADER ?  
    bool searchingFollower;     // is LEADER searching for FOLLOWER ?  
    int twin;                   // index of the multibot's TWIN  
  
} MULTIBOT;
```

- 2) deklarace rutin pro práci s botem
- 3) deklarace externích proměnných
- 4) `#ifdef MULTIBOT_LOG` – pokud není zakomentováno, provádí se logování multibotí funkcionality do souboru **multibot.log**

multibot.cpp

- 1) rutina pro inicializaci bota
- 2) rutina pro reinicializaci bota

bot.h

- 1) deklarace dvou přidáných konzolových příkazů
- 2) deklarace dvou konstant pro označení úkolu MULTIBOT
- 3) obohacení struktury `bot_t` o atribut `multibot`
- 4) deklarace rutin pro párování botů a případné logování

bot.cpp

- 1) přidání úkolu do pole úkolů `taskFilters[]`
- 2) funkce `BotSpawnInit()` (úprava)
 - a. inicializace bota (alokace paměti)
 - b. provedení párovací rutiny `multibotSpawnInit`
- 3) úprava (ve funkci `Botcreate()`) označení před jménem bota z [P0D] na [M-B] pokud je multibot aktivní již při tvorbě jmen (načten z **podbot.cfg**)
- 4) `BotFollowUser()` (úprava)
 - a. pokud je multibot aktivní, vyhodnocuje se pouze zda je leader ještě naživu
 - b. nekontroluje se, jestli je leader v dohledu apod. jako při FOLLOWUSER
- 5) oddělení jednotlivých kol hry v případném logovacím souboru (funkce `UpdateGlobalExpierienceData()`)
- 6) `BotCheckRadioCommands()` (úprava)
 - a. pokud je multibot aktivní, je nežádoucí používání příkazu "Follow me"
 - b. tento radiový příkaz je sice volán, ale žádný z případných vykonavatelů jej nepřijme
 - c. obvykle bot tento příkaz přijímá s 50% pravděpodobností (náhodné číslo mezi 0 a 100 musí být menší než 50); v případě

aktivního multibota je místo prahu 50 nastaveno -1, čili podmínka nemůže být nikdy splněna

7) BotThink() (úprava)

- a. aktivace multibota za běhu - pokud dochází k aktivaci až v průběhu kola provádí se alokace paměti a inicializace na místě
- b. párování v průběhu kola – pokud jednomu z dvojce byl zabit kolega, zbylý člen páru se snaží najít do dvojice jiného spolubojovníka
- c. task TASK_MULTIBOT – provádí se pouze u botů, kteří následují, ne u leaderů;
 - i. pokud není leader mrtev, pokračuje se v plnění úkolu
 - ii. pokud leader není v dohledu, dostanou k vykonání úkol TASK_MOVE_TO_POSITION, který zajistí "doběhnutí" příslušného leadra
- d. střelba botů – kvůli testování byl přidán konzolový příkaz "mbshoot on/off"; pokud je off, boti mají zablokovanou střelbu (granáty a dýmovnice používají); střelbu lze deaktivovat pouze v případě aktivního multibota

8) multibotSpawnInit() (nová) - provádí párování botů do týmů leader-follower podle následující logiky:

- a. Každý bot vstoupivší do této funkce předpokládá, že bude follower a prochází pole všech botů a hledá leadra z jeho týmu, který ještě nemá žádného followera.
- b. Pokud jej nenalezne, stane se sám leadrem a nastaví flag searchingFollower na true, na znamení toho že hledá followera. Další bot z jeho týmu, který do rutiny vstoupí jej pak nalezne.
- c. Pokud je tedy leader nalezen, je provedeno spojení obou botů. Do atributu twin obou botů je uložen index jeho kolegy z tabulky botů. Poté je followerovi zadán úkol TASK_MULTIBOT, který zařídí následování příslušného leadra.

9) logTwinConnections() (nová) – pokud je definován přepínač MULTIBOT_LOG, zapisuje do logu tabulku botů, ve které je zaznamenáno jejich propojení.

bot_client.cpp

- 1) BotClient_CS_DeathMsg() (úprava) – tato funkce je volána pokaždé když je zabit člen nějakého týmu. V takovém případě je zrušeno existující spojení páru a přeživší člen je připraven pro začlenění do páru nového.

d11.cpp

- 1) inicializace externích proměnných multibotON a mbshootON
- 2) nastavení proměnné multibotON přes konzolový příkaz

Poznámka

V případě zadání příkazu "multibot ?" do konzoly je vypsána informace o tom, zda je multibot aktivní a pokud je, je vypsána i tabulka multibotích spojení.

- 3) nastavení proměnné `mbshootON` přes nedokumentovaný konzolový příkaz – pouze pro testování
- 4) nastavení proměnné `multibotON` při načtení ze souboru **podbot.cfg**

6. Závěr

Tato práce byla pro mne osobně velice zajímavá, protože programování her mě lákalo od jakživa. Přestože se jednalo pouze o úpravu části existujícího kódu, umožnilo mi to proniknout hlouběji do dané problematiky a poopravit některé z mých "naivních" představ o jejích principech.

Doufám a věřím, že použitím "mého vylepšeného" bota hra neztratila nic ze své hratelnosti a přitažlivosti, možná právě naopak.

7. Prameny

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

- nejlepší stránka o tvorbě botů do Halflifu a několika dalších her
- na <http://www.planethalflife.com/botman/links.shtml> jsou odkazy na spousty dalších stránek o botech

<http://www.counter-strike.net/>

- stránka modu Counter Strike na enginu Halflifu

<http://ai-depot.com>

- základní informace o tvorbě počítačových her
- tutoriály

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

- abeceda herního vývojáře :-)

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

- informace o umělé inteligenci ve hrách

<http://www.gamers.org/dEngine/quake/>

- stránka vývojářů hry Quake

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

- stránky her Quake, Quake2, Quake3 a jejich modů

<http://www.urbanterror.net/>

- stránky modu Urban Terror na enginu Q3A