Project Assignment

Mikael Olofsson

Blekinge Institute of Technology

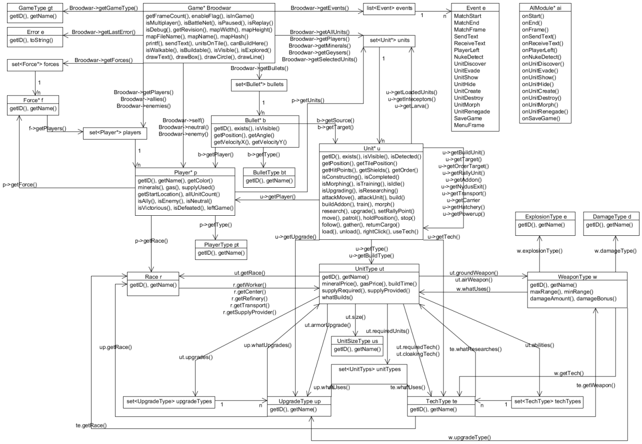
2017

Based on a lecture by Diego Navarro

BWAPI Overview

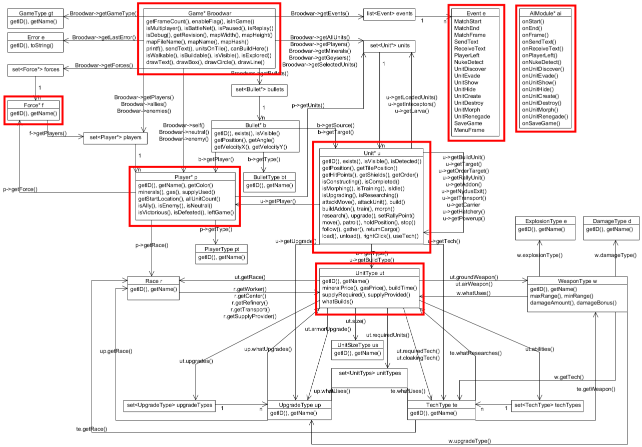
To begin with, lets have an overview of the BWAPI structure...

BWAPI Overview



Taken from:`http://i.imgur.com/lppnE.png`

BWAPI Overview



Taken from:`http://i.imgur.com/lppnE.png`

BWAPI Overview

You will need to review it more carefully
There are 6 main classes in the BWAPI:

- Game
- Player
- Unit
- Bullet
- AIModule and Event

BWAPI Overview

Game Class

- The Game class in main control interface of the BWAPI.
- Allows to retrieve values from other class elements (Player, Units, etc.)
- Allows to manipulate general features of the game itself (Pause, move screen, player selection, player cursor position, etc.)

BWAPI Overview

Game Class

- getAllUnits: Returns all visible units
- getEvents: Return the list of event for current frame, corresponding from AIModule callbacks.
- pingMinimap
- isWalkable
- isBuildable
- Also draw functions

Game class: `https://code.google.com/p/bwapi/wiki/Game`

BWAPI Overview

Player Class

- Retrieves information about the current player(s) in the current game
- Player basic info (name, units, type, race, etc.)
- Win states, amount of resources, units (alive and dead), upgrades, weapon range and CD.

Game class: `https://code.google.com/p/bwapi/wiki/Player`

BWAPI Overview

Unit Class

- Unit class gather information from a particular unit and issue orders (tactics)
- Each unit will have a unique unit object that remains until the end of the game
- Divided into two sections: Information retrieval and Command methods
- Unit information will depend on unit visibility (accessibility tier)

BWAPI Overview

Unit Class

Information retrieval methods

- isVisible
- isPatrolling // isHoldingPosition
- isGatheringMinerals // isCarryingMinerals
- isDetected // isAttacking
- getOrder
- hasPath

Game class: `https://code.google.com/p/bwapi/wiki/Unit`

BWAPI Overview

Unit Class

Command methods

- attack
- gather // build // upgrade
- move // patrol // follow // holdPosition
- issueCommand

Game class: `https://code.google.com/p/bwapi/wiki/Unit`

BWAPI Overview

Bullet Class

- Bullet class retrieves information from non-melee attacks (bullets, spells , missiles, etc.) as long as they are visible
- Bullet units are not deleted but reused with a new ID once they are destroyed
- getSource // getType
- getTarget // getTargetPosition
- isVisible

Game class: `https://code.google.com/p/bwapi/wiki/Bullet`

BWAPI Overview

AIModule and Event Classes

- AIModule is the class used to develop a custom AI
- Event refers to the callbacks for AIModule class and are obtained by calling Game::getEvents
- onStart // onFrame
- onUnitCreate // onUnitDiscover
- onUnitMorph // onUnitDestroy

Game class: `https://code.google.com/p/bwapi/wiki/AIModule`

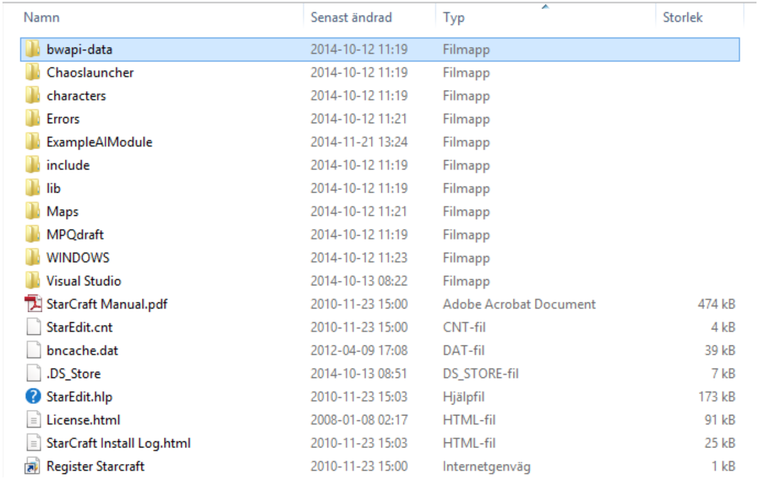
The Project Files

To implement your assignment you will need 3 main elements:

- StarCraft Game
- Chaos Launcher
- Visual studio 2013 build tools

Once you install you project files, you might have something like this:

The Project Files



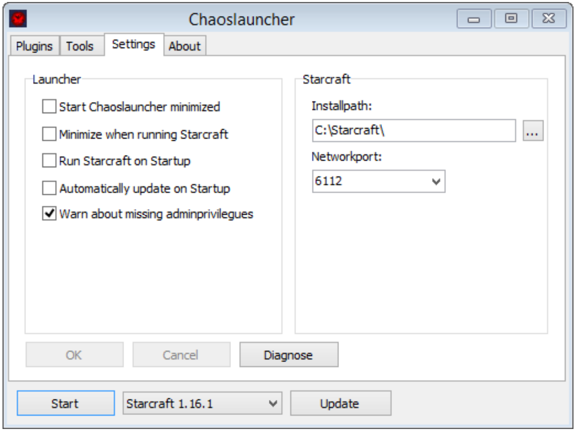
Namn	Senast ändrad	Typ	Storlek
bwapi-data	2014-10-12 11:19	Filmapp	
Chaoslauncher	2014-10-12 11:19	Filmapp	
characters	2014-10-12 11:19	Filmapp	
Errors	2014-10-12 11:21	Filmapp	
ExampleAIModule	2014-11-21 13:24	Filmapp	
include	2014-10-12 11:19	Filmapp	
lib	2014-10-12 11:19	Filmapp	
Maps	2014-10-12 11:21	Filmapp	
MPQdraft	2014-10-12 11:19	Filmapp	
WINDOWS	2014-10-12 11:23	Filmapp	
Visual Studio	2014-10-13 08:22	Filmapp	
StarCraft Manual.pdf	2010-11-23 15:00	Adobe Acrobat Document	474 kB
StarEdit.cnt	2010-11-23 15:00	CNT-fil	4 kB
bncache.dat	2012-04-09 17:08	DAT-fil	39 kB
.DS_Store	2014-10-13 08:51	DS_STORE-fil	7 kB
StarEdit.hlp	2010-11-23 15:03	Hjälpfil	173 kB
License.html	2008-01-08 02:17	HTML-fil	91 kB
StarCraft Install Log.html	2010-11-23 15:03	HTML-fil	25 kB
Register Starcraft	2010-11-23 15:00	Internetgenväg	1 kB

The Project Files

- Visual Studio: Installer of VS2013
- include: BWAPI and BWTA
- ExampleAIModule: Reference files for bot developing
- Chaoslauncher: Injector for AI DLL
- bwapi-data: This is were your compiled DLL need to be placed (inside AI folder) This is automatically done at compile time.

The Project Files

Remember to set the installation path of the Starcraft in the ChaosLauncher



Launcher

- [ ] Start Chaoslauncher minimized
- [ ] Minimize when running Starcraft
- [ ] Run Starcraft on Startup
- [ ] Automatically update on Startup
- [x] Warn about missing adminprivilegues

Starcraft

Installpath: C:\Starcraft\

Networkport: 6112

Starcraft 1.16.1

Additional Tips and References

The full documentation for the BWAPI (v.3.7.4) can be found here:
`https://code.google.com/p/bwapi/wiki/OldDocumentation`

There is also a Starcraft guide with advanced information for bot development:
`https://code.google.com/p/bwapi/wiki/StarcraftGuide`

Additional Tips and References

For general information about map, position and coordinates, you could use:

- Region class: `https://code.google.com/p/bwapi/wiki/Region`
- Position and TilePosition class: `https://code.google.com/p/bwapi/wiki/Misc`

Additional Tips and References

When in doubt.. Debug!

- Broodwar::printf()
- Broodwar::draw...

Tiles: 64 x 64 tiles == 2048 x 2048 pixels
TilePosition a(1,2) == Position b(32,64)
(Position)a == Position(32,64)
(TilePosition)b == TilePosition(1,2)

Additional Tips and References

Take time to test every improvement done on your AI (even on early stages of development)

- if Flag::CompleteMapInformation is enable, you will be able to access all the units on the map.
- if Flag::UserInput is enable, you will be able to retrieve information from a determined user (selected units, messages, etc.)

Questions?