AI Script Tutorial

How it Works

The AI file should be seen as having a tree structure. The AI system will go through all the tasks at the highest level and execute the one with the highest priority. A task can either perform an action or be parent to a number of other tasks. When a parent task is executed, the process is repeated on its children; that is, the highest priority one is selected to be executed. It's important to remember that the AI goes down this tree repeatedly; it's always making a decision. Each time it does so is called a cycle.

AI scripts are .txt files and can be found in the *ai* folder.

Definitions

Concurrent

Tasks can be marked as concurrent. After a concurrent task has finished executing, the next highest priority task at that level will execute (if there is one).

Postfix expression

A postfix expression is just a mathematical expression using postfix notation. If you're unfamiliar with it, just do a Google search. It basically just puts operators after both operands rather than between, eliminating the need for parentheses. For example, *(5 + 2) * 7* in postfix is *5 2 + 7 **. All operators and operands must be separated by spaces. GS supports a wide variety of operands, listed in the reference section of this document. Because constant priorities are of little use, operands can be data sources from the game.

Data Source

Your postfix expressions can contain variables from the game. Each data source has the form ***$<#>***. For example, *$24* represents your player's shields on a scale of 0 to 1. Data sources are listed in the reference section. Their values are only fetched from the game once per cycle even if requested by multiple expressions.

FID

Function IDs are used to refer to executable actions that GS can perform as a result of the script. FIDs are just numbers. There are FIDs for walking, controlling the aimbot, and weapon management among others. In the script, they follow an exclamation mark and are followed by a parameter postfix expression within parentheses, terminated by a semicolon.

Syntax

The AI scripting language has a recursive structure:

task:

```
<*>[<postfix expression>] <name> {
      <task(s)>
}
```

or

task:

```
<*>[<postfix expression>] <name> !<FID>(<postfix expression>);
```

Anything following a *#* on a line is ignored as a comment. The syntax for a task is a postfix expression in brackets, followed by a name, and either an action (*!**<FID>(<postfix expression>)**;*) or a brace (*{*) to start a new block of child tasks. Putting an asterisk (*) in front of the task marks it as concurrent. A line of the form *>**<filename>*** redirects the parser to another file temporarily, allowing you to build scripts out of other scripts. The filename must be relative to the *ai\* folder.

Use the *$* symbol to refer to a data source in the postfix expression. To perform actions, use the "*!*" symbol followed by an FID #, then a parameter between parentheses. End the line with a semicolon. Some FIDs don't use parameters, so just put any number between the parentheses. Lists of data sources and FIDs are available in the reference section.

Notes

Modes and some data sources will have initial values when the AI is started:

- All storage values are 0
- Arc mode is off
- Gravity scale is 1
- Projectile velocity is 0
- Look-ahead mode is off
- Strafe mode is off

Things to keep in mind:

- If you prefer opening braces to be on their own line, that's fine. Follow whitespace conventions.
- If your script won't load, check "Show Debug Output" and try again to see where the problem is.
- If your cycle is taking longer to execute than the AI refresh rate, you'll get warnings in the debug output. If the previous cycle is still executing when a new one is triggered, the new one will be ignored.
- Some parameters, logical operators, and the return values of some comparison operators are interpreted as 0 = false, anything else = true.
- A priority of 0 doesn't mean the task won't be executed. Priorities can be negative as well.

Syntax highlighting for shell or python works well for GuiltySpark scripts. Included in this release is a gss.lang file for full syntax highlighting in gedit.

Example 1

Let’s take a look at the included *wander.txt* script:

```
1:  #!10000
2:
3:  *[1] GO_TO_RANDOM_NODE !4($35);
4:  *[1] SET_LOOK_AHEAD_MODE !14(1);
```

Line 1 is just a comment, but it’s a special kind of comment. Any comment of the form *!<#>* will automatically set the AI refresh rate to that number when loaded. The number is in ms and can be as high as 60,000 (1 minute).

Lines 3 and 4 are both tasks performing actions. Note that both tasks are concurrent (*) and have the same priority expression: *[1]*. Their priorities are a constant 1, and since it’s unclear which one will be executed first, I made both tasks concurrent to make sure both of them are executed. If you want tasks to execute in a certain order, order their priorities and make them concurrent so they all execute.

The next part of a task is its name. Names cannot contain spaces. The name is not important for the execution of the task; it’s just there to help you remember what the task does. Following the name you’ll see FIDs 4 and 14. FID 4 tells the bot to walk to a node, and I gave it $35 (random node number) as a parameter. FID 14 toggles look-ahead mode. A parameter of 1 turns it on. In summary, this script walks to a random node and enables look-ahead mode every cycle, waiting 10 seconds between cycles. Look-ahead only needs to be enabled once, but for simplicity’s sake it’s done every cycle.

Example 2

Open up *new001.txt* next:

```
1:  #!100
2:
3:  [0 $15 =] INIT {
4:        *[1] SET_VALUE1 !9(1);
5:        *[1] START_AIMBOT !11(1);
6:  }
7:
8:  [0.5] RUN {
9:        *[1] GOTO_CLOSEST_ENEMY !3($9);
10:       [0.5] SET_TARGET !1($9);
11:
12:       *[$49] SHOOT_AT_THEM {
13:             [1] MOUSE1 !6(30);
14:       }
15: }
```

Here you can see the other type of task which is parent to a block of child tasks.

Look at the priority expression for *INIT*: *[0 $15 =]*. Data source *$15* is a storage value you can set in your script to remember numbers between cycles. It initially equals 0 upon starting the AI. The = operator compares 0 and $15 for equality and returns 1 if equal, 0 if not. Since they are indeed equal, the priority of *INIT* is 1. The priority of *RUN* is only 0.5 and will not be executed because *INIT* is not concurrent.

The AI now needs to decide what to execute within the *INIT* task. Like in Example 1, both tasks here are concurrent and so both get executed. FID 11 enables or disables the aimbot depending on the parameter (1 means enable). FID 9 sets storage value #1's value to the parameter, which is also 1. Execution of the cycle ends here and *$15* is now 1 instead of 0.

At the start of the second cycle, *INIT* has a priority of 0 because *$15* is not equal to 0. Thus the higher priority *RUN* task is executed. Inside *RUN*, you can see it constantly tries to walk to the nearest enemy. It will also always set the aimbot target to the closest enemy. However, it may or may not execute *SHOOT_AT_THEM*. The data source *$49* has value 1 when the target is visible and 0 when not. When *SHOOT_AT_THEM* has priority 1, it is executed before *SET_TARGET* but *SET_TARGET* is still executed due to the fact that the other two tasks are concurrent. However, when *$13* is 0, *SHOOT_AT_THEM* is never executed because the higher priority *SET_TARGET* is not concurrent.

All this script does is run to the closest enemy and shoot at them when they're visible. Now you know why it's called *new001.txt*.

Reference

Operators

Arithmetic

- `+` **sum**
- `-` **difference**
- `*` **product**
- `/` **quotient**
- `^` **power**
- `%` **modulo**

Comparison/Relational

- `=` **equal**
- `<` **less than**
- `>` **greater than**
- `` ` `` **max**
- `~` **min**

Logical

- | **or**
- & **and**

Data Sources

- PLAYER_X = 0,
- PLAYER_Y = 1,
- PLAYER_Z = 2,
- CAN_WALK = 3, **has value 0 if console, text chat, or menu open. Equal to 1 otherwise. The bot will check this automatically before pressing keys.**
- VIEW_ANGLE_H = 4, **gives your player's horizontal view angle in radians.**
- VIEW_ANGLE_V = 5,
- CAMERA_X = 6,
- CAMERA_Y = 7,
- CAMERA_Z = 8,
- CLOSEST_ENEMY = 9, **gives the player index of the enemy closest to you.**
- LOCAL_TEAM = 10,
- PLAYER_COUNT = 11,
- LOCAL_PING = 12,
- CLEAR_SHOT = 13, **has value 1 if a name is being displayed on the screen for who you're aiming at, 0 otherwise.**
- SHIFT_KEY = 14, **has value 1 when the shift key is pressed, 0 otherwise.**
- VALUE1 = 15, **is one of 5 storage values, used for memory between cycles. All values equal 0 at the start of the AI.**
- VALUE2 = 16,
- CLOSEST_ALLY = 17, **is like CLOSEST_ENEMY, just for allies.**
- CLOSEST_ANYONE = 18,
- CLOSEST_DIST = 19, **gives the distance of the closest enemy by default, but if CLOSEST_ANYONE or CLOSEST_ALLY is requested before this during the current cycle then the distance to those players will be the value.**
- TARGET_HEALTH = 20, **gives the health of the aimbot's target on a scale of 0 to 1. You need to set the aimbot's target first.**
- LOCAL_HEALTH = 21,
- TARGET_DIST = 22,
- TARGET_SHIELD = 23,
- LOCAL_SHIELD = 24,
- TARGET_CAMO = 25, **has value 1 if the aimbot's target has active camouflage, 0 otherwise.**
- LOCAL_CAMO = 26,
- TARGET_STANCE = 27, **has value 1 if the aimbot's target is crouching, 2 if they're jumping, 0 otherwise.**
- LOCAL_STANCE = 28,
- ENEMY_NEAREST_VIEW = 29, **gives the player index of the enemy nearest to where you're looking.**
- ALLY_NEAREST_VIEW = 30,
- ANYONE_NEAREST_VIEW = 31,
- VALUE3 = 32,
- VALUE4 = 33,
- VALUE5 = 34,
- RANDOM_NODE = 35, **gives the number of a randomly selected node from the loaded graph.**
- ZOOM_LEVEL = 36, **has value 0 when not zoomed in, 1 when 2x, 2 when 8x.**

- FLASHLIGHT = 37, **has value 1 when your flashlight is on, 0 when not.**
- PRIMARY_WEAPON = 38, **players can hold 2 weapons: weapon 0, and weapon 1. This data source tells you which one of your weapons is readied.**
- WEAPON0_TYPE = 39, **gives a number to indicate the weapon's type. See the table of weapon types in the reference section.**
- WEAPON1_TYPE = 40,
- WEAPON0_CLIP = 41, **for magazine weapons, gives the number of bullets left in the magazine on a scale of 1 (full) to 0 (empty). For battery-powered weapons, this represents the heat on a scale of 1 (cool) to 0 (hot).**
- WEAPON1_CLIP = 42,
- WEAPON0_RESERVE = 43, **gives the total percentage of rounds or battery charge left in the weapon, based on maximum carrying capacity for magazine weapons.**
- WEAPON1_RESERVE = 44,
- GRENADE_TYPE = 45, **has value 0 when frag grenades are selected and 1 for plasma grenades.**
- FRAG_GRENADE_COUNT = 46,
- PLASMA_GRENADE_COUNT = 47,
- RANDOM = 48, **gives a random number between 0 and 1**
- TARGET_VISIBLE_ADVANCED = 49 **has value 0 when the line of sight to the target is occluded by BSP surfaces, 1 otherwise.**

Weapon Types

- FUEL_ROD_GUN = 0,
- NEEDLER = 1,
- ASSAULT_RIFLE = 2,
- FLAMETHROWER = 3,
- GRAVITY_RIFLE = 4,
- NEEDLER2 = 5,
- PISTOL = 6,
- PLASMA_PISTOL = 7,
- PLASMA_RIFLE = 8,
- ROCKET_LAUNCHER = 9,
- SHOTGUN = 10,
- SNIPER_RIFLE = 11

FIDs

- PRINT = 0, **prints the parameter to the AI output window. This is handy for debugging your files.**
- SET_TARGET_PLAYER = 1, **sets the current target of the aimbot. The parameter is a player index.**
- GOTO_NODE = 2, **instructs the bot to walk to the specified node**
- GOTO_PLAYER = 3, **instructs the bot to walk to the given player (by player index).**
- GOTO_NODE_ALT = 4, **is the same as GOTO_NODE, but tries to take an alternate path than one previously selected.**
- GOTO_PLAYER_ALT = 5,
- MOUSE1 = 6, **simulates an event of mouse1. A parameter of -1 means button down, 0 means up, and a value >0 holds the button for that many milliseconds.**
- MOUSE2 = 7,

- SLEEP = 8, **pauses the AI for the given number of milliseconds.**
- SET_VALUE1 = 9, **sets the storage value #1 to the parameter.**
- SET_VALUE2 = 10,
- TOGGLE_AIMBOT = 11, **given parameter 0, stops the aimbot. Otherwise, starts it.**
- *(FID 12 reserved)*
- SET_TARGET_NODE = 13, **sets the aimbot's target to the specified graph node rather than a player.**
- SET_LOOK_AHEAD_MODE = 14, **a parameter of 1 forces look-ahead mode in the path following. Look-ahead overrides any other aiming the aimbot should be doing (players or nodes). Pass a parameter of 0 to disable look-ahead mode.**
- *(FID 15 reserved)*
- SET_VALUE3 = 16,
- SET_VALUE4 = 17,
- SET_VALUE5 = 18,
- AIMBOT_ENABLE_ARC_MODE = 19, **enables ballistic aiming where the projectile is affected by gravity. The parameter is the gravity scale used by the projectile (grenades use 1, fuel rods 1.1, and rockets 0). Enabling this mode also makes the aimbot aim try to land the projectile at the target's feet.**
- AIMBOT_DISABLE_ARC_MODE = 20,
- SET_PROJECTILE_VELOCITY = 21, **sets the velocity of the projectile being shot (grenades use 10, fuel rods 16, rockets 12). This affects how much the aimbot leads the target. Set it to 0 to ignore travel time completely. Note that it already leads player targets for network latency.**
- GOTO_OBJECTIVE = 22, **instructs the bot to walk to the objective with objective index as parameter.**
- GOTO_OBJECTIVE_ALT = 23,
- SET_STRAFE_MODE = 24, **with a parameter of 1, causes the bot to walk erratically when following a path, making it a harder target to hit. Pass a parameter of 0 to turn off strafe mode.**
- CROUCH = 25, **simulates pressing CTRL to crouch. A parameter of -1 holds the key down, 0 releases, and a value >0 hold for that many milliseconds**
- JUMP = 26, **works as above with CROUCH.**
- SWITCH_WEAPONS = 27,
- MELEE = 28,
- RELOAD = 29,
- ZOOM = 30, **a parameter of 0 zooms out completely, while a parameter of 1 means 2x. Zooming to 8x is currently unsupported.**
- FLASHLIGHT = 31, **sets the flashlight off with parameter 0 and on otherwise**
- SET_PRIMARY_WEAPON = 32, **switches weapons if the readied weapon is not the desired primary (weapon 0 or 1).**
- BACKPACK_RELOAD = 33, **switches to the secondary weapon while performing a reloading glitch, causing the original primary weapon to be reloaded already when switched back to (done by pressing R twice quickly, then TAB).**
- EXCHANGE_WEAPON = 34, **picks a weapon up off the ground (if there is one) by pressing X—faster than holding E.**
- ACTION = 35, **presses E, allowing the bot to enter a vehicle if at one (or reload if not).**
- SWITCH_GRENADE_TYPE = 36, **switches to frag with parameter 0, and plasma otherwise.**
- CHAT = 37, **types a chat message using the text files in the chat\ folder. A**

parameter of 1 will type 1.txt, 2 for 2.txt, etc.

- NOP = 99 **Any FID not listed here will be interpreted as “do nothing”. Use 99 for consistency.**
