Tool Assisted Speedrunning(TAS) with the Source Pause Tool(SPT)

What is a TAS?

TAS-ing or Tool Assisted speedrunning is sending input for the game by an external program so you can basically play the game perfectly. TAS-ing is really hard so don't expect awesome results without putting a lot of effort into doing it but once you are done the results are simply stunning. (eg.: I spent 6 hours picking up a suitcase).

Prerequisites

So first of all download Half-Life 2 Source Unpack from here. After that's done, get the latest SPT for TASing here. Put spt.dll into the "/hl2/" folder of Half-Life 2 Source Unpack. After this create a file named "autoexec.cfg" in the "/cfg/" folder of the game. Then put this inside that file: "plugin_load spt" alternatively you can type this every time you load the game.

Making your first script

Create a file named "tas.cfg" (you can name it anything) in the "/cfg/" folder of hl2 then launch the game. After this is done make a save where your script will start taking the control. To continue I would recommend binding 2 keys. One to load your save and "untas" and one to start the script. The first key will load the save so you don't have to type to command every time and "untas" (this is just optional) so you don't have to type a million commands every time you retry. And the other key is just "exec tas.cfg" so your tas script starts executing (these binds can be found below).

After this you open your “tas.cfg” with some text editor (I recommend Visual Studio Code) and start making your first script. At the top we usually put a couple commands that set up stuff in the game so you don’t have to type them every time. The most common way is like so:

```
host_framerate 0.015
record d1_canals_02_TAS
_y_spt_afterframes_await_load
```

At the end of the “header” I inserted “**_y_spt_afterframes_await_load**” (this command will wait for the game to load a map/save and continue executing the afterframe commands) because pauses are timed so you have to set SPT’s pause cvar to “**y_spt_pause 0**”. After this is done you will start scripting the game below this code. To schedule a command for the game you use “_y_spt_afterframes X command” where X is the number of frames to wait and execute the command and command is the command to execute. For example “_y_spt_afterframes 10 +jump” <- this will send the engine the jump command after 10 frames.

Running the TAS

Since you already have 2 keys bound. 1 to load your save and ‘untas’ and another to exec the tas script. You hit your first key and while the game is loading hit your second key. Due to the fact that you have “**_y_spt_afterframes_await_load”** on the top of your script it will wait for your game to load so the script runs consistently.

Auto-strafing

Strafing is the most optimal way to move around the map because the player will move with mathematically the most optimal strafes. To enable this use “tas_strafe 1”. After it’s enabled you can set the direction with “tas_strafe_yaw”.
For example to start an abh with auto-strafing use this:

- `_y_spt_afterframes <x> "+duck"`
- `_y_spt_afterframes <x+9> "tas_strafe 1; tas_strafe_yaw <direction>"`
- `_y_spt_afterframes <x+27> "+jump"`

`So in a real script it would look like this:`

- `_y_spt_afterframes 5 "+duck"`
- `_y_spt_afterframes 14 "tas_strafe 1; tas_strafe_yaw 10"`
- `_y_spt_afterframes 32 "+jump"`

Please note that you need a bit of momentum to auto-strafe.

Happy TAS-ing.

Extras

SPT commands:

- **_y_spt_afterframes** - with this command you can execute a command after X amount of frames has passed. Note that X is calculated from the script start so first command should be X second should be X+Y and so on. Example: _y_spt_afterframes “10” “+jump”
- **_y_spt_setyaw** - change the direction the player is looking at (yaw is left and right). Example: _y_spt_setyaw “10”.
- **_y_spt_setpitch** - change the direction the player is looking at(pitch is up and down). Example _y_spt_setpitch “10”
- **_y_spt_getvel** - prints your X Y Z velocity to the console.
- **_y_spt_afterframes_await_load** - waits until a load happens and continues the afterframes que after that you should put this on the beginning of every script.

Normal commands:

- **+forward/+back/+moveright/+moveleft** = movement commands
- **+jump**
- **+duck** = crouch
- **toggle_duck**
- **+attack** = primary fire
- **+attack2** = secondary fire
- **+use**
- **getpos** = outputs your position on the map to console
- Strafing commands:
    - **tas_strafe** - this enable or disables auto strafing. Tas_strafe 1 to enable tas_strafe 0 to disable it.
    - **tas_strafe_yaw** - change the direction you are strafing towards. Example: tas_strafe_yaw “10”.

It’s recommended to make a bind to disable every “TAS command” so you can move around freely.

```
alias untas "-forward;-back;-moveleft;-moveright;-jump;-duck;-attack;-attack2;tas_strafe 0;_y_spt_afterframes_reset;-y_spt_duckspam;-walk;-speed;_y_spt_pitchspeed 0;_y_spt_yawspeed 0;-use;-usespam;norm;y_spt_pause 0"
```

When creating a TAS for easier testing bind a button to load the save and "untas" and another one to start the script.

```
alias start "untas;load startmove;
alias runtas "exec tas;"
bind c start
bind v runtas
```

How to calculate yaw/pitch speed for angle changes

_y_spt_yawspeed = change in angle/ticks/2

Example: You want to go from 150 to 120 in 10 ticks:
Change in angle: -30
_y_spt_yawspeed = -30/10/2 = -1.5

In the script:

```
_y_spt_afterframes x "_y_spt_yawspeed -1.5"
_y_spt_afterframes x+10 "_y_spt_yawspeed -1.5"
```

The only thing here really to note is that _y_spt_yawspeed rotates the view twice the value you have it set to per tick, which is a little bit counterintuitive.

Useful aliases

```
sensitivity 0
alias backstrafe "tas_strafe_buttons 6 2 5 3"
alias frontstrafe "tas_strafe_buttons -1"
alias tuse "+use;_y_spt_afterframes 1 -use"
alias _ST0P "stop;toggleconsole"
alias ST0P "tas_strafe_type 0;tas_strafe 0;tas_maxspeed
0;-speed;-jump;-duck;-walk;-attack;-attack2;-use;_y_spt_pitchspeed 0;_y_spt_yawspeed 0;echo
#SAVE#;_y_spt_afterframes 20 _ST0P"
alias +tspeed "tas_maxspeed 320;+speed"
alias -tspeed "tas_maxspeed 0;-speed"
alias _usespam0 "+use;_y_spt_afterframes 1 -use;_y_spt_afterframes 2 _usespam"
alias +usespam "alias _usespam _usespam0;_usespam"
alias -usespam "alias _usespam"
alias attack "+attack;_y_spt_afterframes 1 -attack"
alias attack2 "+attack2;_y_spt_afterframes 1 -attack2"
alias walk "+walk;y_spt_afterframes 1 -walk"
alias jump "+jump;_y_spt_afterframes 1 -jump"
alias stopspin "_y_spt_pitchspeed 0;_y_spt_yawspeed 0"
alias backstrafe "tas_strafe_buttons 6 2 5 3"
alias frontstrafe "tas_strafe_buttons -1"
alias tuse "+use;_y_spt_afterframes 1 -use"
alias _ST0P "stop"
alias ST0P "save tas_stop;echo #SAVE#;_y_spt_afterframes 20 _ST0P"
alias +tspeed "tas_maxspeed 320;+speed"
alias -tspeed "tas_maxspeed 0;-speed"
alias _usespam0 "+use;_y_spt_afterframes 1 -use;_y_spt_afterframes 2 _usespam"
alias +usespam "alias _usespam _usespam0;_usespam"
alias -usespam "alias _usespam"
alias attack "+attack;_y_spt_afterframes 1 -attack"
alias attack2 "+attack2;_y_spt_afterframes 1 -attack2"
alias walk "+walk;y_spt_afterframes 1 -walk"
alias jump "+jump;_y_spt_afterframes 1 -jump"
alias fast "y_spt_cvar fps_max 300"
alias norm "y_spt_cvar fps_max 66.666666666"
alias slow "y_spt_cvar fps_max 10"
alias stopspin "_y_spt_pitchspeed 0;_y_spt_yawspeed 0;
alias tpause "pause;_y_spt_afterframes 1 unpause"
alias tjump "+jump; _y_spt_afterframes 1 -jump"
alias untas "-forward;-back;-moveleft;-moveright;-jump;-duck;tas_strafe
```

```
0;_y_spt_afterframes_reset;-y_spt_duckspam;-walk;-speed;_y_spt_yawspeed 0"
alias start "untas;load 001-canals01-7035  ;
alias runtas "exec 002-canals01_tas"
bind c start
bind v runtas
bind k "_y_spt_afterframes_reset;untas"
```

An example script

```
Record d1_trainstation_01_tas
_y_spt_afterframes_await_load
_y_spt_afterframes "1" "_y_spt_setyaw 165;_y_spt_setpitch 0"
_y_spt_afterframes "5" "+forward;+moveright;"
_y_spt_afterframes "40" "+duck;+jump"
_y_spt_afterframes "60" "-duck;-moveright;-jump"
_y_spt_afterframes "95" "_y_spt_setpitch 0;_y_spt_setyaw 20;" //Out of train
_y_spt_afterframes "102" "+jump;+duck;"
_y_spt_afterframes "110" "tas_strafe 1;tas_strafe_yaw 14;"
```

Credits

- YaLTeR for spt
- Rama for introducing me into TAS-ing
- Jukspa for that awesome google doc and being awesome
- Centaur1um for creating the beautiful community