```
/* ----------------------------------------------------------------------------------------

	"Just Capture The Flag" - by Digi (aka Hunter-Digital)

	Thread: http://forums.alliedmods.net/showthread.php?t=132115

	Change log below.

---------------------------------------------------------------------------------------- */

new const MOD_TITLE[] =			"Capture the Flag"	/* Please don't modify. */
new const MOD_AUTHOR[] =		"Digi"			/* If you make major changes, add " & YourName" at the end */
new const MOD_VERSION[] =		"1.32c"			/* If you make major changes, add "custom" at the end but do not modify the actual version number! */

/*
	Below you can enable/disable each individual feature of this plugin
	NOTE: Remember to compile the plugin again after you modify anything in this file!

	----------------------------------

	Description: This uses Orpheu module to make infinite round end and trigger round end on flag captures
	If set to false, ophreu will not be used anymore and rounds could end if all players from one team are dead and round can't end upon flag capture.
*/
#define FEATURE_ORPHEU			true


/*
	Description: This hooks the buy system of the game and changes it, allowing everybody to buy all weapons.
	If set to false, it will disable all buy related stuff like: buy menu, spawn weaopns, special weapons, even C4!
	Disable this if you want to use another plugin that manages buy or doesn't use buy at all (like GunGame)
*/
#define FEATURE_BUY			true


/*
	Description: This allows players to buy and use C4 as a weapon, not an objective, it can be defused tough but the defuser gets a usable C4 back. C4 kills everything in it's radius, including teammates.
	If set to false, C4 usage will be completly disabled so it can't be bought.
```

```
	Requirements: FEATURE_BUY must be true
*/
#define FEATURE_BUYC4			true


/*
	Description: This allows players to have an adrenaline amount and when it reaches 100, they can use combos.
	If set to false, it will disable all adrenaline stuff, including combos, rewards, buying with adrenaline, everything.
*/
#define FEATURE_ADRENALINE		true

/* ---------------------------------------------------------------------------------------
	Skip this, advanced configuration more below
*/
#if FEATURE_BUY == true && FEATURE_BUYC4 == true
#define FEATURE_C4 true
#else
#define FEATURE_C4 false
#endif

#include <amxmodx>
#include <amxmisc>
#include <hamsandwich>

#if FEATURE_ORPHEU == true
#include <orpheu_memory>
#include <orpheu>
#endif

#include <fakemeta>
#include <cstrike>
#include <engine>
#include <fun>


/* ---------------------------------------------------------------------------------------

	CVars for .cfg files:

		ctf_flagreturn (default 120) - flag auto-return time
		ctf_weaponstay (default 30) - how long do weapons and items stay on ground
		ctf_itempercent (default 30) - chance that items spawn when a player is killed, values from 0 to 100
```

ctf_sound_taken (default 1) - toggles if the "flag taken" sounds can be heard
ctf_sound_dropped (default 1) - toggles if the "flag dropped" sounds can be heard
ctf_sound_returned (default 1) - toggles if the "flag returned" sounds can be heard
ctf_sound_score (default 1) - toggles if the "X team scores" sounds can be heard
ctf_respawntime (default 10) - players respawn time (use -1 to disable respawn)
ctf_spawnmoney (default 1000) - money bonus when spawning (unless it's a suicide)
ctf_protection (default 5) - players spawn protection time (use -1 to disable protection)
ctf_dynamiclights (default 1) - set the default dynamic lights setting, players will still be able to toggle individually using /lights
ctf_glows (default 1) - set if entities can glow, like when players have flag or an adrenaline combo, weapons start to fade, etc.
ctf_nospam_flash (default 20) - delay of rebuying two flashbangs in a life
ctf_nospam_he (default 20) - delay of rebuying a HE grenade in a life
ctf_nospam_smoke (default 20) - delay of rebuying a smoke grenade in a life
ctf_spawn_prim (default "m3") - spawning primary weapon, set to "" to disable
ctf_spawn_sec (default "glock") - spawning secondary weapon, set to "" to disable
ctf_spawn_knife (default 1) - toggle if players spawn with knife or not
ctf_sound_taken (default 1) - toggles if the "flag taken" sounds can be heard
ctf_sound_dropped (default 1) - toggles if the "flag dropped" sounds can be heard
ctf_sound_returned (default 1) - toggles if the "flag returned" sounds can be heard
ctf_sound_score (default 1) - toggles if the "X team scores" sounds can be heard

Primary weapons: m3,xm1014,tmp,mac10,mp5,ump45,p90,galil,ak47,famas,m4a1,aug,sg552,awp,scout,sg550,g3sg1,m249,shield
Secondary weapons: glock,usp,p228,deagle,elites,fiveseven

mp_c4timer (recommended 20) - time before the C4 devices explode
mp_winlimit - first team who reaches this number wins
mp_timelimit - time limit for the map (displayed in the round timer)
mp_startmoney (recommended 3000) - for first spawn money and minimum amount of money
mp_forcecamera - (0/1 - spectate enemies or not) mod fades to black if this is on and player is in free look (no teammates alive)
mp_forcechasecam - (0/1/2 - force chase cammera all/team/firstperson) same as above

mp_autoteambalance - enable/disable auto-team balance (checks at every player death)

Map configurations are made with;

ctf_moveflag red/blue at your position (even if dead/spec)
ctf_save to save flag origins in maps/<mapname>.ctf

Reward configuration, 0 on all values disables reward/penalty.

```
	[REWARD FOR]                    [MONEY]     [FRAGS]     [ADRENALINE]
*/
#define REWARD_RETURN               500,        0,          10
#define REWARD_RETURN_ASSIST        500,        0,          10

#define REWARD_CAPTURE              3000,       3,          25
#define REWARD_CAPTURE_ASSIST       3000,       3,          25
#define REWARD_CAPTURE_TEAM         1000,       0,          10

#define REWARD_STEAL                1000,       1,          10
#define REWARD_PICKUP               500,        1,          5
#define PENALTY_DROP                -1500,      -1,         -10

#define REWARD_KILL                 0,          0,          5
#define REWARD_KILLCARRIER          500,        1,          10

#define PENALTY_SUICIDE             0,          0,          -20
#define PENALTY_TEAMKILL            0,          0,          -20

/*
	Advanced configuration
*/

const ADMIN_RETURN =                ADMIN_RCON  // access required for admins to return flags (full list in includes/amxconst.inc)
const ADMIN_RETURNWAIT =            15          // time the flag needs to stay dropped before it can be returned by command

new const bool:CHAT_SHOW_COMMANDS =     true    // show commands (like /buy) in chat, true or false

const ITEM_MEDKIT_GIVE =            25          // medkit award health for picking up
```

```
new const bool:ITEM_DROP_AMMO =                 true            // toggle if killed players
drop ammo items
new const bool:ITEM_DROP_MEDKIT =               true            // toggle if killed players
drop medkit items

#if FEATURE_ADRENALINE == true
new const bool:ITEM_DROP_ADRENALINE =           true            // toggle if killed players
drop adrenaline items
const ITEM_ADRENALINE_GIVE =                    5               // adrenaline reaward for
picking up adrenaline

const Float:SPEED_ADRENALINE =          1.3             // speed while using "speed"
adrenaline combo (this and SPEED_FLAG are cumulative)

const Float:BERSERKER_SPEED1 =                  0.7             // primary weapon
shooting speed percent while in berserk
const Float:BERSERKER_SPEED2 =                  0.3             // secondary weapon
shooting speed percent while in berserk
const Float:BERSERKER_DAMAGE =                  2.0             // weapon damage
percent while in berserk

const INSTANTSPAWN_COST =                       50              // instant spawn (/spawn)
adrenaline cost

#endif // FEATURE_ADRENALINE

const REGENERATE_EXTRAHP =                      50              // extra max HP for
regeneration and flag healing

const Float:SPEED_FLAG =                0.9             // speed while carying the enemy
flag

new const Float:BASE_HEAL_DISTANCE =    96.0            // healing distance for flag

#if FEATURE_C4 == true

new const C4_RADIUS[] =                         "600"           // c4 explosion radius
(must be string!)
new const C4_DEFUSETIME =                       3               // c4 defuse time

#endif // FEATURE_C4

new const FLAG_SAVELOCATION[] =                 "maps/%s.ctf" // you can change where
.ctf files are saved/loaded from
```

#define FLAG_IGNORE_BOTS true // set to true if you don't want bots to pick up flags

/*

Change log:

v1.32c:

* Changed files: jctf.sma

- Added ctf_dynamiclights cvar to enable server operators to disable dynamic lights by default, players can still re-enable them individually using /lights

- Added ctf_glows cvar to enable server operators to disable glows on entities for performance, however it will degrade gameplay due to lack of visual information

v1.32b:

* Changed files: jctf.sma

- Fixed rewards beeing default to 0 for steal and capturing flags (my bad).

v1.32:

* Changes files: jctf.sma

- Removed forcing of CVars - if you depended on that, use amxx.cfg to store default values of cvars.

- Added posibility to disable respawn using ctf_respawntime 0.

- Added posibility to disable spawn protection using ctf_protection 0.

- Added posibility to disable dropped weapons fading out using ctf_weaponstay 0.

- Added posibility to disable flag auto-return using ctf_flagreturn 0.

- Fixed GameName printing CS 1.6 for CZ too, now it detects game name, if's unknown it doesn't write anything.

- Fixed player rewards still printing if all rewards are 0.

- Fixed MP3 files printing WAV error messages due to incorrect precaching method.

v1.31:

* Changed files: jctf.sma, jctf_resources.zip

- Fixed issue where plugin wouldn't trigger "Round_End" on ctf_flagendround 1 that caused issues with other plugins.

- Added new orpheu signatures: functions/EndRoundMessage and functions/CHalfLifeMultiplay/UpdateTeamScores.

- Changed "TeamScore" message to emessage, that way it can be hooked by other plugins.

- Added updating team scores internally too.

- Some other minor changes.

v1.3:

* Changed files: jctf.sma, lang/jctf.txt, jctf_resources.zip

- Added CVars: ctf_flagendround, ctf_flagcaptureslay and ctf_infiniteround (see thread for descriptions)
- Changed /help command, it now prints most mod settings and features (which are enabled/disabled)
- Added new orpheu signatures: functions/InstallGameRules and memory/CGameRulesOffsets (they're in jctf_resources.zip)
- Changed feature define FEATURE_INFROUND to FEATURE_ORPHEU because it also disables flag capture ending round
- Changed contents of lang keys: JOIN_NOFEATURES
- Renamed lang keys: HELP_3_* to HELP_4_* and DEAH_NOFREELOOK to DEATH_NOFREELOOK
- Added lang keys: DEATH_FLAGCAPTURED, STARTING_NEWROUND, and HELP_3_*
- Added so that on new round, flags are returned to base (no matter what the reason of new round)

v1.28c:
* Changed files: jctf.sma
- Fixed if you reload map with the plugin disabled the rounds will end normally.

v1.28b:
* Changed files: jctf.sma
- Fixed a compile error when switching FEATURE_ADRENALINE to false.

v1.28:
* Changed files: jctf.sma, jctf.txt
- Fixed buying VGUI issues, plugin now closes buy VGUI and opens custom buy menu without client changes.
- Removed VGUI related ML keys from jctf.txt.

v1.27:
* Changed files: jctf.sma, jctf.txt
- Added FEATURE disabling, you can now toggle individual features from the sma file.
- Added ctf_flagheal cvar which toggles if flag bases heal teammates.
- Changed HUD message to Director HUD messages, nicer and have no channel limit
- Increased char limit for "Freelook mode blocked" ML text from 32 to 48 chars.
- Rearranged sma configuration variables, you can find them easier on top of the file.
- Added hints and adrenaline menu to the ML support.
- Decreased chance that flag gets stuck in a wall (dramatically decreased collision box and added proximity check for pickup)

- Decreased C4's price from $15000 to $12000 and adrenaline cost from 100 to 80
- Some minor optimizations

v1.26:
* Changed files: jctf.sma, jctf.inc(jctf_api.zip), lang/jctf.txt
- Fixed "ED_Alloc: No free edicts" crashes (hopefully).
- Added Multilingual support, natives support it too.
- Added printing of admin commands in chat that obey amx_show_activity.
- Added a check of players spawning HP and Armor so Booster and flags know how much to heal.
- Added configurable extra HP the booster heals.
- Changed armor regeneration from Booster so it no longer sets armor to kevlar+helm if you have kevlar only.
- Added option to ignore bots from taking flags, default OFF (search sma for // CONFIGURABLE).
- Added checking of player's team range when touching flag in case it's a spawned spectator.
- Fixed bots not buying weapons, but be warned, they buy at default prices and without adrenaline.
- Changed default mp_buytime from 0 to 99999999 so bots can always buy, this doesn't affect players.
- Added API natives: jctf_get_flagcarrier(id) and jctf_get_team(id)
- Added FLAG_ADMINRETURN for the jctf_flag() forward, it triggers when an admin returns the flag using command
- Fixed ctf_moveflag command not showing Y axis in the log

v1.25:
- Changed ctf_moveflag to support red and blue inputs
- Fade to black no longer affects spectators
- Added "ctf_returnflag <red/blue>" admin command that returns the specified flag if it was dropped for at least 15 seconds (configurable)
- Optimized some minor stuff

v1.24:
- Changed ctf_weaponstay's default value from 30 to 15 (seconds)
- Changed ctf_itempercent's default value from 30 to 25 (percent)
- Added Scout and G3SG1 as special weapons.
- Added the "configurable" flag in the source for weapon money and adrenaline costs
- Added configurable weapon and zoom-in running speeds
- Added cvars for controling sounds: ctf_sound_taken/dropped/returned/score
- Added how much money/adrenaline you need in the "Not enough money/adrenaline" message

v1.23:

- plugin no longer disables VGUI menu automatically but it notifies VGUI menu users upon spawn
- added new in-source configuration: CHAT_SHOW_COMMANDS

v1.22:
- fixed a bug where you'd get twice the adrenaline when using the jctf_add_adrenaline() native while using a reason
- fixed adrenaline HUD not updating when using jctf_add_adrenaline()

v1.21:
- added FLAG_MANUALDROP for jctf_flag() events
- fixed some example issues in jctf_addon_example.sma
- updated version check to 1.21 since jctf.inc was altered
- added a console print upon plugin load with plugin name and version
- forced to reset jctf_version to the current version if mod is updated while server is running

v1.2:
- added forwards and natives that other plugins could use
- added configurable spawning weapons using CVars, they're also forced to reset to prevent unwanted usage between maps
- minor adjustments in the code to fit the latest modifications

v1.13:
- fixed some issues with the autoteambalance
- fixed spawn protection countdown not stop when prematurely disabling protection
- fixed game name displaying version as number instead of string
- added mod version in quick help and help console message
- added checking of mp_forcechasecam too for the fade to black feature
- fixed the possiblity of not beeing blinded while in freelook after a respawn

v1.12:
- fixed a bug where transfered people (autoteambalance) would actually block the team they left from beeing joined, the "there are too many CTs/Ts" thing.
- added fade to black when mp_fadetoblack is disabled and mp_forcecamera is enabled while player is on free view (no teammates alive)

v1.11:
- fixed a rare bug where players won't respawn after beeing killed
- some optimizations in code

v1.1:
- removed previous round-end blocking and added new round blocking method via Orpheu module

- fixed various possible crashes caused by previous round-end blocking methods

v1.06:
- changed buy menu text so that items you can't afford are in red text
- fixed various speed issues with player zooming, shield and flag taking
- re-done player rendering system and altered some colors and values
- unforced most mod cvars, only two cvars remain forced

v1.05:
- fixed round-blocking player deaths's animations
- changed the sounds of medkit and adrenaline item pickup
- and of course, other small adjustments

v1.04:
- other small adjustments here and there
- removed the spectating bots, now players' deaths are blocked if they'll trigger round end, they wouldn't even notice
- altered C4 plant radio message to a be compatible with client modifications and other hooks

v1.03:
- added /help
- minor adjustments and code moved around

v1.02:
- fixed items not fading out
- fixed the simple auto-team balance
- fixed various possible errors

v1.01:
- minor fixes of typos and checks

---------- */

```
new const INFO_TARGET[] =                   "info_target"
new const ITEM_CLASSNAME[] =                "ctf_item"
new const WEAPONBOX[] =                     "weaponbox"

#if FEATURE_C4 == true

new const GRENADE[] =                       "grenade"

#endif // FEATURE_C4

new const Float:ITEM_HULL_MIN[3] =          {-1.0, -1.0, 0.0}
```

```
new const Float:ITEM_HULL_MAX[3] =		{1.0, 1.0, 10.0}

const ITEM_AMMO =				0
const ITEM_MEDKIT =				1

#if FEATURE_ADRENALINE == true

const ITEM_ADRENALINE =				2

#endif // FEATURE_ADRENALINE

new const ITEM_MODEL_AMMO[] =			"models/w_chainammo.mdl"
new const ITEM_MODEL_MEDKIT[] =			"models/w_medkit.mdl"

#if FEATURE_ADRENALINE == true

new const ITEM_MODEL_ADRENALINE[] =		"models/can.mdl"

#endif // FEATURE_ADRENALINE

new const BASE_CLASSNAME[] =			"ctf_flagbase"
new const Float:BASE_THINK =			0.25

new const FLAG_CLASSNAME[] =			"ctf_flag"
new const FLAG_MODEL[] =		"models/th_jctf.mdl"

new const Float:FLAG_THINK =			0.1
const FLAG_SKIPTHINK =				20 /* FLAG_THINK * FLAG_SKIPTHINK
= 2.0 seconds ! */

new const Float:FLAG_HULL_MIN[3] =		{-2.0, -2.0, 0.0}
new const Float:FLAG_HULL_MAX[3] =		{2.0, 2.0, 16.0}

new const Float:FLAG_SPAWN_VELOCITY[3] =	{0.0, 0.0, -500.0}
new const Float:FLAG_SPAWN_ANGLES[3] =		{0.0, 0.0, 0.0}

new const Float:FLAG_DROP_VELOCITY[3] =		{0.0, 0.0, 50.0}

new const Float:FLAG_PICKUPDISTANCE =		80.0

const FLAG_LIGHT_RANGE =			12
const FLAG_LIGHT_LIFE =				5
const FLAG_LIGHT_DECAY =			1

const FLAG_ANI_DROPPED =			0
const FLAG_ANI_STAND =				1
```

```
const FLAG_ANI_BASE =                   2

const FLAG_HOLD_BASE =                  33
const FLAG_HOLD_DROPPED =               34

#if FEATURE_ADRENALINE == true

const ADRENALINE_SPEED =                1
const ADRENALINE_BERSERK =              2
const ADRENALINE_REGENERATE =                   3
const ADRENALINE_INVISIBILITY =         4

new const MENU_ADRENALINE[] =           "menu_adrenaline"
new const MENU_KEYS_ADRENALINE =        (1<<0)|(1<<1)|(1<<2)|(1<<3)|(1<<9)

#endif // FEATURE_ADRENALINE

#if FEATURE_BUY == true

new const WHITESPACE[] =                " "
new const MENU_BUY[] =                  "menu_buy"
new const MENU_KEYS_BUY =
(1<<0)|(1<<1)|(1<<2)|(1<<3)|(1<<4)|(1<<5)|(1<<6)|(1<<7)|(1<<8)|(1<<9)

new const BUY_ITEM_DISABLED[] =         "r"
new const BUY_ITEM_AVAILABLE[] =        "w"

#if FEATURE_ADRENALINE == true

new const BUY_ITEM_AVAILABLE2[] =       "y"

#endif // FEATURE_ADRENALINE

#endif // FEATURE_BUY

new const SND_GETAMMO[] =               "items/9mmclip1.wav"
new const SND_GETMEDKIT[] =             "items/smallmedkit1.wav"

#if FEATURE_ADRENALINE == true

new const SND_GETADRENALINE[] =         "items/medshot4.wav"
new const SND_ADRENALINE[] =            "ambience/des_wind3.wav"

#endif // FEATURE_ADRENALINE

#if FEATURE_C4 == true
```

```
new const SND_C4DISARMED[] =                "weapons/c4_disarmed.wav"

#endif // FEATURE_C4

new const CHAT_PREFIX[] =           "^x03[^x04 CTF^x03 ]^x01 "
new const CONSOLE_PREFIX[] =                "[ CTF ] "

const FADE_OUT =                            0x0000
const FADE_IN =                             0x0001
const FADE_MODULATE =                       0x0002
const FADE_STAY =                           0x0004

const m_iUserPrefs =                510
const m_flNextPrimaryAttack =               46
const m_flNextSecondaryAttack =     47

new const PLAYER[] =                        "player"
new const SEPARATOR[] =                     " - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - -"
#define NULL                        ""

#define HUD_HINT                            255, 255, 255, 0.15, -0.3, 0, 0.0, 10.0,
2.0, 10.0, 4
#define HUD_HELP                            255, 255, 0, -1.0, 0.2, 2, 0.1, 2.0, 0.01,
2.0, 2
#define HUD_HELP2                           255, 255, 0, -1.0, 0.25, 2, 0.1, 2.0, 0.01,
2.0, 3
#define HUD_ANNOUNCE                        -1.0, 0.3, 0, 0.0, 3.0, 0.1, 1.0, 4
#define HUD_RESPAWN                         0, 255, 0, -1.0, 0.6, 2, 0.5, 0.1, 0.0, 1.0, 1
#define HUD_PROTECTION                      255, 255, 0, -1.0, 0.6, 2, 0.5, 0.1, 0.0,
1.0, 1
#define HUD_ADRENALINE                      255, 255, 255, -1.0, -0.1, 0, 0.0, 600.0,
0.0, 0.0, 1

#define entity_create(%1)           create_entity(%1)
#define entity_spawn(%1)            DispatchSpawn(%1)
#define entity_think(%1)            call_think(%1)
#define entity_remove(%1)           remove_entity(%1)
#define weapon_remove(%1)           call_think(%1)

#define task_set(%1)                set_task(%1)
#define task_remove(%1)                     remove_task(%1)

#define player_hasFlag(%1)          (g_iFlagHolder[TEAM_RED] == %1 ||
g_iFlagHolder[TEAM_BLUE] == %1)
```

```
#define player_allowChangeTeam(%1)		set_pdata_int(%1, 125, get_pdata_int(%1, 125) & ~(1<<8))

#define gen_color(%1,%2)		%1 == TEAM_RED ? %2 : 0, 0, %1 == TEAM_RED ? 0 : %2

#define get_opTeam(%1)		(%1 == TEAM_BLUE ? TEAM_RED : (%1 == TEAM_RED ? TEAM_BLUE : 0))
```

```
enum
{
	x,
	y,
	z
}

enum
{
	pitch,
	yaw,
	roll
}

enum (+= 64)
{
	TASK_RESPAWN = 64,
	TASK_PROTECTION,
	TASK_DAMAGEPROTECTION,
	TASK_EQUIPAMENT,
	TASK_PUTINSERVER,
	TASK_TEAMBALANCE,
	TASK_ADRENALINE,
	TASK_DEFUSE,
	TASK_CHECKHP
}
```

```
enum
{
	TEAM_NONE = 0,
	TEAM_RED,
	TEAM_BLUE,
	TEAM_SPEC
}

new const g_szCSTeams[][] =
{
	NULL,
	"TERRORIST",
	"CT",
	"SPECTATOR"
}

new const g_szTeamName[][] =
{
	NULL,
	"Red",
	"Blue",
	"Spectator"
}

new const g_szMLTeamName[][] =
{
	NULL,
	"TEAM_RED",
	"TEAM_BLUE",
	"TEAM_SPEC"
}

new const g_szMLFlagTeam[][] =
{
	NULL,
	"FLAG_RED",
	"FLAG_BLUE",
	NULL
}

enum
{
	FLAG_STOLEN = 0,
	FLAG_PICKED,
	FLAG_DROPPED,
	FLAG_MANUALDROP,
```

```
	FLAG_RETURNED,
	FLAG_CAPTURED,
	FLAG_AUTORETURN,
	FLAG_ADMINRETURN
}

enum
{
	EVENT_TAKEN = 0,
	EVENT_DROPPED,
	EVENT_RETURNED,
	EVENT_SCORE,
}

new const g_szSounds[][][] =
{
	{NULL, "red_flag_taken", "blue_flag_taken"},
	{NULL, "red_flag_dropped", "blue_flag_dropped"},
	{NULL, "red_flag_returned", "blue_flag_returned"},
	{NULL, "red_team_scores", "blue_team_scores"}
}

#if FEATURE_ADRENALINE == true
new const g_szAdrenalineUseML[][] =
{
	NULL,
	"ADR_SPEED",
	"ADR_BERSERK",
	"ADR_REGENERATE",
	"ADR_INVISIBILITY"
}
#endif // FEATURE_ADRENALINE

#if FEATURE_BUY == true

enum
{
	no_weapon,
	primary,
	secondary,
	he,
	flash,
	smoke,
	armor,
	nvg
}
```

```
new const g_szRebuyCommands[][] =
{
	NULL,
	"PrimaryWeapon",
	"SecondaryWeapon",
	"HEGrenade",
	"Flashbang",
	"SmokeGrenade",
	"Armor",
	"NightVision"
}

#endif // FEATURE_BUY

new const g_szRemoveEntities[][] =
{
	"func_buyzone",
	"armoury_entity",
	"func_bomb_target",
	"info_bomb_target",
	"hostage_entity",
	"monster_scientist",
	"func_hostage_rescue",
	"info_hostage_rescue",
	"info_vip_start",
	"func_vip_safetyzone",
	"func_escapezone",
	"info_map_parameters",
	"player_weaponstrip",
	"game_player_equip"
}

enum
{
	ZERO = 0,
	W_P228,
	W_SHIELD,
	W_SCOUT,
	W_HEGRENADE,
	W_XM1014,
	W_C4,
	W_MAC10,
	W_AUG,
	W_SMOKEGRENADE,
	W_ELITE,
```

```
	W_FIVESEVEN,
	W_UMP45,
	W_SG550,
	W_GALIL,
	W_FAMAS,
	W_USP,
	W_GLOCK18,
	W_AWP,
	W_MP5NAVY,
	W_M249,
	W_M3,
	W_M4A1,
	W_TMP,
	W_G3SG1,
	W_FLASHBANG,
	W_DEAGLE,
	W_SG552,
	W_AK47,
	W_KNIFE,
	W_P90,
	W_VEST,
	W_VESTHELM,
	W_NVG
}

new const g_iClip[] =
{
	0,		// (unknown)
	13,		// P228
	0,		// SHIELD (not used)
	10,		// SCOUT
	0,		// HEGRENADE (not used)
	7,		// XM1014
	0,		// C4 (not used)
	30,		// MAC10
	30,		// AUG
	0,		// SMOKEGRENADE (not used)
	30,		// ELITE
	20,		// FIVESEVEN
	25,		// UMP45
	30,		// SG550
	35,		// GALIL
	25,		// FAMAS
	12,		// USP
	20,		// GLOCK18
	10,		// AWP
```

```
	30,		// MP5NAVY
	100,		// M249
	8,		// M3
	30,		// M4A1
	30,		// TMP
	20,		// G3SG1
	0,		// FLASHBANG (not used)
	7,		// DEAGLE
	30,		// SG552
	30,		// AK47
	0,		// KNIFE (not used)
	50,		// P90
	0,		// Kevlar (not used)
	0,		// Kevlar + Helm (not used)
	0		// NVG (not used)
}

new const g_iBPAmmo[] =
{
	0,		// (unknown)
	52,		// P228
	0,		// SHIELD
	90,		// SCOUT
	0,		// HEGRENADE (not used)
	32,		// XM1014
	0,		// C4 (not used)
	100,		// MAC10
	90,		// AUG
	0,		// SMOKEGRENADE (not used)
	120,		// ELITE
	100,		// FIVESEVEN
	100,		// UMP45
	90,		// SG550
	90,		// GALIL
	90,		// FAMAS
	100,		// USP
	120,		// GLOCK18
	30,		// AWP
	120,		// MP5NAVY
	200,		// M249
	32,		// M3
	90,		// M4A1
	120,		// TMP
	90,		// G3SG1
	0,		// FLASHBANG (not used)
	35,		// DEAGLE
```

```
	90,		// SG552
	90,		// AK47
	0,		// KNIFE (not used)
	100,		// P90
	0,		// Kevlar (not used)
	0,		// Kevlar + Helm (not used)
	0		// NVG (not used)
}

#if FEATURE_BUY == true

new const g_iWeaponPrice[] =
{
	0,		// (unknown)
	600,		// P228
	10000, // SHIELD
	6000,		// SCOUT
	300,		// HEGRENADE
	3000,		// XM1014
	12000, // C4
	1400,		// MAC10
	3500,		// AUG
	100,		// SMOKEGRENADE
	1000,		// ELITE
	750,		// FIVESEVEN
	1700,		// UMP45
	6000,		// SG550
	2000,		// GALIL
	2250,		// FAMAS
	500,		// USP
	400,		// GLOCK18
	8000,		// AWP
	1500,		// MP5NAVY
	5000,		// M249
	1700,		// M3
	3100,		// M4A1
	1250,		// TMP
	7000,		// G3SG1
	200,		// FLASHBANG
	650,		// DEAGLE
	3500,		// SG552
	2500,		// AK47
	0,		// KNIFE (not used)
	2350,		// P90
	650,		// Kevlar
	1000,		// Kevlar + Helm
```

```
	1250		// NVG
}

#endif // FEATURE_BUY

#if FEATURE_BUY == true && FEATURE_ADRENALINE == true

new const g_iWeaponAdrenaline[] =
{
	0,		// (unknown)
	0,		// P228
	50,		// SHIELD
	50,		// SCOUT
	0,		// HEGRENADE
	0,		// XM1014
	80,		// C4
	0,		// MAC10
	0,		// AUG
	0,		// SMOKEGRENADE
	0,		// ELITE
	0,		// FIVESEVEN
	0,		// UMP45
	30,		// SG550
	0,		// GALIL
	0,		// FAMAS
	0,		// USP
	0,		// GLOCK18
	50,		// AWP
	0,		// MP5NAVY
	10,		// M249
	0,		// M3
	0,		// M4A1
	0,		// TMP
	30,		// G3SG1
	0,		// FLASHBANG
	0,		// DEAGLE
	0,		// SG552
	0,		// AK47
	0,		// KNIFE (not used)
	0,		// P90
	0,		// Kevlar
	0,		// Kevlar + Helm
	0		// NVG
}

#endif // FEATURE_ADRENALINE
```

```
new const Float:g_fWeaponRunSpeed[] = // CONFIGURABLE - weapon running speed (edit the numbers in the list)
{
	150.0,  // Zoomed speed with any weapon
	250.0,  // P228
	0.0,		// SHIELD (not used)
	260.0,  // SCOUT
	250.0,  // HEGRENADE
	240.0,  // XM1014
	250.0,  // C4
	250.0,  // MAC10
	240.0,  // AUG
	250.0,  // SMOKEGRENADE
	250.0,  // ELITE
	250.0,  // FIVESEVEN
	250.0,  // UMP45
	210.0,  // SG550
	240.0,  // GALIL
	240.0,  // FAMAS
	250.0,  // USP
	250.0,  // GLOCK18
	210.0,  // AWP
	250.0,  // MP5NAVY
	220.0,  // M249
	230.0,  // M3
	230.0,  // M4A1
	250.0,  // TMP
	210.0,  // G3SG1
	250.0,  // FLASHBANG
	250.0,  // DEAGLE
	235.0,  // SG552
	221.0,  // AK47
	250.0,  // KNIFE
	245.0,  // P90
	0.0,		// Kevlar (not used)
	0.0,		// Kevlar + Helm (not used)
	0.0		// NVG (not used)
}

#if FEATURE_BUY == true

new const g_iWeaponSlot[] =
{
	0,		// none
	2,		// P228
```

```
	1,		// SHIELD
	1,		// SCOUT
	4,		// HEGRENADE
	1,		// XM1014
	5,		// C4
	1,		// MAC10
	1,		// AUG
	4,		// SMOKEGRENADE
	2,		// ELITE
	2,		// FIVESEVEN
	1,		// UMP45
	1,		// SG550
	1,		// GALIL
	1,		// FAMAS
	2,		// USP
	2,		// GLOCK18
	1,		// AWP
	1,		// MP5NAVY
	1,		// M249
	1,		// M3
	1,		// M4A1
	1,		// TMP
	1,		// G3SG1
	4,		// FLASHBANG
	2,		// DEAGLE
	1,		// SG552
	1,		// AK47
	3,		// KNIFE (not used)
	1,		// P90
	0,		// Kevlar
	0,		// Kevlar + Helm
	0		// NVG
}

#endif // FEATURE_BUY

new const g_szWeaponEntity[][24] =
{
	NULL,
	"weapon_p228",
	"weapon_shield",
	"weapon_scout",
	"weapon_hegrenade",
	"weapon_xm1014",
	"weapon_c4",
	"weapon_mac10",
```

```
	"weapon_aug",
	"weapon_smokegrenade",
	"weapon_elite",
	"weapon_fiveseven",
	"weapon_ump45",
	"weapon_sg550",
	"weapon_galil",
	"weapon_famas",
	"weapon_usp",
	"weapon_glock18",
	"weapon_awp",
	"weapon_mp5navy",
	"weapon_m249",
	"weapon_m3",
	"weapon_m4a1",
	"weapon_tmp",
	"weapon_g3sg1",
	"weapon_flashbang",
	"weapon_deagle",
	"weapon_sg552",
	"weapon_ak47",
	"weapon_knife",
	"weapon_p90",
	"item_kevlar",
	"item_assaultsuit",
	NULL
}

#if FEATURE_BUY == true

new const g_szWeaponCommands[][] =
{
	{NULL,		NULL},
	{"p228",		"228compact"},
	{"shield",		NULL},
	{"scout",		NULL},
	{"hegren",		NULL},
	{"xm1014",		"autoshotgun"},
	{NULL,		NULL},
	{"mac10",		NULL},
	{"aug",		"bullpup"},
	{"sgren",		NULL},
	{"elites",		NULL},
	{"fiveseven",	"fn57"},
	{"ump45",		"sm"},
	{"sg550",		"krieg550"},
```

```
	{"galil",		"defender"},
	{"famas",		"clarion"},
	{"usp",		"km45"},
	{"glock",		"9x19mm"},
	{"awp",		"magnum"},
	{"mp5",		"mp"},
	{"m249",		NULL},
	{"m3",		"12gauge"},
	{"m4a1",		NULL},
	{"tmp",		NULL},
	{"g3sg1",		"d3au1"},
	{"flash",		NULL},
	{"deagle",		"nighthawk"},
	{"sg552",		"krieg552"},
	{"ak47",		"cv47"},
	{NULL,		NULL},
	{"p90",		"c90"},
	{"vest",		NULL},
	{"vesthelm",	NULL},
	{"nvgs",		NULL}
}

#endif // FEATURE_BUY




new g_iMaxPlayers
new g_szMap[32]
new g_szGame[16]
new g_iTeam[33]
new g_iScore[3]
new g_iFlagHolder[3]
new g_iFlagEntity[3]
new g_iBaseEntity[3]
new Float:g_fFlagDropped[3]

#if FEATURE_BUY == true

new g_iMenu[33]
new g_iRebuy[33][8]
new g_iAutobuy[33][64]
new g_iRebuyWeapons[33][8]
```

```
new pCvar_ctf_nospam_flash
new pCvar_ctf_nospam_he
new pCvar_ctf_nospam_smoke
new pCvar_ctf_spawn_prim
new pCvar_ctf_spawn_sec
new pCvar_ctf_spawn_knife

new gMsg_BuyClose

#endif // FEATURE_BUY

new g_iMaxArmor[33]
new g_iMaxHealth[33]
new g_iAdrenaline[33]
new g_iAdrenalineUse[33]
new bool:g_bRestarting
new bool:g_bBot[33]
new bool:g_bAlive[33]
new bool:g_bDefuse[33]
new bool:g_bLights[33]
new bool:g_bBuyZone[33]
new bool:g_bSuicide[33]
new bool:g_bFreeLook[33]
new bool:g_bAssisted[33][3]
new bool:g_bProtected[33]
new bool:g_bRestarted[33]
new bool:g_bFirstSpawn[33]

new Float:g_fFlagBase[3][3]
new Float:g_fFlagLocation[3][3]
new Float:g_fWeaponSpeed[33]
new Float:g_fLastDrop[33]
new Float:g_fLastBuy[33][4]

new pCvar_ctf_flagcaptureslay
new pCvar_ctf_flagheal
new pCvar_ctf_flagreturn
new pCvar_ctf_respawntime
new pCvar_ctf_protection
new pCvar_ctf_dynamiclights
new pCvar_ctf_glows
new pCvar_ctf_weaponstay
new pCvar_ctf_spawnmoney
new pCvar_ctf_itempercent

new pCvar_ctf_sound[4]
```

```
new pCvar_mp_winlimit
new pCvar_mp_startmoney
new pCvar_mp_fadetoblack
new pCvar_mp_forcecamera
new pCvar_mp_forcechasecam
new pCvar_mp_autoteambalance

#if FEATURE_C4 == true

new pCvar_mp_c4timer

new gMsg_BarTime
new gMsg_DeathMsg
new gMsg_SendAudio

#endif // FEATURE_C4

new gMsg_SayText
new gMsg_RoundTime
new gMsg_ScreenFade
new gMsg_HostageK
new gMsg_HostagePos
new gMsg_ScoreInfo
new gMsg_ScoreAttrib
new gMsg_TextMsg
new gMsg_TeamScore

new gHook_EntSpawn

#if FEATURE_ADRENALINE == true

new gSpr_trail
new gSpr_blood1
new gSpr_blood2

#endif // FEATURE_ADRENALINE

new gSpr_regeneration

new g_iForwardReturn
new g_iFW_flag


#if FEATURE_ORPHEU == true

new pCvar_ctf_infiniteround
```

```
new pCvar_ctf_flagendround

new g_pGameRules
new bool:g_bLinux
new OrpheuHook:g_oMapConditions
new OrpheuHook:g_oWinConditions
new OrpheuHook:g_oRoundTimeExpired
new MEMORY_ROUNDTIME[] = "roundTimeCheck"

#endif // FEATURE_ORPHEU



public plugin_precache()
{
	precache_model(FLAG_MODEL)
	precache_model(ITEM_MODEL_AMMO)
	precache_model(ITEM_MODEL_MEDKIT)

#if FEATURE_ADRENALINE == true

	precache_model(ITEM_MODEL_ADRENALINE)

	precache_sound(SND_GETADRENALINE)
	precache_sound(SND_ADRENALINE)

	gSpr_trail = precache_model("sprites/zbeam5.spr")
	gSpr_blood1 = precache_model("sprites/blood.spr")
	gSpr_blood2 = precache_model("sprites/bloodspray.spr")

#endif // FEATURE_ADRENALINE

	precache_sound(SND_GETAMMO)
	precache_sound(SND_GETMEDKIT)

	gSpr_regeneration = precache_model("sprites/th_jctf_heal.spr")

	for(new szSound[64], i = 0; i < sizeof g_szSounds; i++)
	{
		for(new t = 1; t <= 2; t++)
		{
			formatex(szSound, charsmax(szSound), "sound/ctf/%s.mp3",
g_szSounds[i][t])

			precache_generic(szSound)
		}
```

```
	}

#if FEATURE_C4 == true
	precache_sound(SND_C4DISARMED)

	new ent = entity_create(g_szRemoveEntities[11])

	if(ent)
	{
		DispatchKeyValue(ent, "buying", "0")
		DispatchKeyValue(ent, "bombradius", C4_RADIUS)
		DispatchSpawn(ent)
	}
#endif // FEATURE_C4

	gHook_EntSpawn = register_forward(FM_Spawn, "ent_spawn")


#if FEATURE_ORPHEU == true

	OrpheuRegisterHook(OrpheuGetFunction("InstallGameRules"),
"game_onInstallGameRules", OrpheuHookPost)

#endif // FEATURE_ORPHEU
}

public ent_spawn(ent)
{
	if(!is_valid_ent(ent))
		return FMRES_IGNORED

	static szClass[32]

	entity_get_string(ent, EV_SZ_classname, szClass, charsmax(szClass))

	for(new i = 0; i < sizeof g_szRemoveEntities; i++)
	{
		if(equal(szClass, g_szRemoveEntities[i]))
		{
			entity_remove(ent)

			return FMRES_SUPERCEDE
		}
	}

	return FMRES_IGNORED
```

```
}

public plugin_init()
{
	register_plugin(MOD_TITLE, MOD_VERSION, MOD_AUTHOR)
	set_pcvar_string(register_cvar("jctf_version", MOD_VERSION,
FCVAR_SERVER|FCVAR_SPONLY), MOD_VERSION)

	register_dictionary("jctf.txt")
	register_dictionary("common.txt")

	new const SEPARATOR_TEMP[] = " - - - - - - - - - - - - - - - - -"

	server_print(SEPARATOR_TEMP)
	server_print("    %s - v%s", MOD_TITLE, MOD_VERSION)
	server_print("    Mod by %s", MOD_AUTHOR)

#if FEATURE_ORPHEU == false
	server_print("[!] Orpheu module usage is disabled! (FEATURE_ORPHEU = false)")
#endif

#if FEATURE_BUY == false
	server_print("[!] Custom buy feature is disabled! (FEATURE_BUY = false)")
#endif

#if FEATURE_C4 == false
	server_print("[!] C4 feature is disabled! (FEATURE_BUYC4 = false)")
#endif

#if FEATURE_ADRENALINE == false
	server_print("[!] Adrenaline feature is disabled! (FEATURE_ADRENALINE = false)")
#endif

	server_print(SEPARATOR_TEMP)

	// Forwards, hooks, events, etc

	unregister_forward(FM_Spawn, gHook_EntSpawn)

	register_forward(FM_GetGameDescription, "game_description")

	register_touch(FLAG_CLASSNAME, PLAYER, "flag_touch")

	register_think(FLAG_CLASSNAME, "flag_think")
	register_think(BASE_CLASSNAME, "base_think")
```

```
	register_logevent("event_restartGame", 2, "1&Restart_Round",
"1&Game_Commencing")
	register_event("HLTV", "event_roundStart", "a", "1=0", "2=0")

	register_clcmd("fullupdate", "msg_block")

	register_event("TeamInfo", "player_joinTeam", "a")

	RegisterHam(Ham_Spawn, PLAYER, "player_spawn", 1)
	RegisterHam(Ham_Killed, PLAYER, "player_killed", 1)
	RegisterHam(Ham_TakeDamage, PLAYER, "player_damage")

	register_clcmd("say", "player_cmd_say")
	register_clcmd("say_team", "player_cmd_sayTeam")

#if FEATURE_ADRENALINE == true

	register_menucmd(register_menuid(MENU_ADRENALINE),
MENU_KEYS_ADRENALINE, "player_key_adrenaline")

	register_clcmd("adrenaline", "player_cmd_adrenaline")

#endif // FEATURE_ADRENALINE

#if FEATURE_BUY == true

	register_menucmd(register_menuid(MENU_BUY), MENU_KEYS_BUY,
"player_key_buy")

	register_event("StatusIcon", "player_inBuyZone", "be", "2=buyzone")

	register_clcmd("buy", "player_cmd_buy_main")
	register_clcmd("buyammo1", "player_fillAmmo")
	register_clcmd("buyammo2", "player_fillAmmo")
	register_clcmd("primammo", "player_fillAmmo")
	register_clcmd("secammo", "player_fillAmmo")
	register_clcmd("client_buy_open", "player_cmd_buyVGUI")

	register_clcmd("autobuy", "player_cmd_autobuy")
	register_clcmd("cl_autobuy", "player_cmd_autobuy")
	register_clcmd("cl_setautobuy", "player_cmd_setAutobuy")

	register_clcmd("rebuy", "player_cmd_rebuy")
	register_clcmd("cl_rebuy", "player_cmd_rebuy")
	register_clcmd("cl_setrebuy", "player_cmd_setRebuy")
```

```
	register_clcmd("buyequip", "player_cmd_buy_equipament")

#endif // FEATURE_BUY

	for(new w = W_P228; w <= W_NVG; w++)
	{
#if FEATURE_BUY == true
		for(new i = 0; i < 2; i++)
		{
			if(strlen(g_szWeaponCommands[w][i]))
				register_clcmd(g_szWeaponCommands[w][i],
"player_cmd_buyWeapon")
		}
#endif // FEATURE_BUY

		if(w != W_SHIELD && w <= W_P90)
			RegisterHam(Ham_Weapon_PrimaryAttack, g_szWeaponEntity[w],
"player_useWeapon", 1)
	}

	register_clcmd("ctf_moveflag", "admin_cmd_moveFlag", ADMIN_RCON, "<red/blue>
- Moves team's flag base to your origin (for map management)")
	register_clcmd("ctf_save", "admin_cmd_saveFlags", ADMIN_RCON)
	register_clcmd("ctf_return", "admin_cmd_returnFlag", ADMIN_RETURN)

	register_clcmd("dropflag", "player_cmd_dropFlag")

#if FEATURE_C4 == true

	RegisterHam(Ham_Use, GRENADE, "c4_defuse", 1)
	register_logevent("c4_planted", 3, "2=Planted_The_Bomb")
	register_logevent("c4_defused", 3, "2=Defused_The_Bomb")

	register_touch(WEAPONBOX, PLAYER, "c4_pickup")

#endif // FEATURE_C4

	register_touch(ITEM_CLASSNAME, PLAYER, "item_touch")

	register_event("CurWeapon", "player_currentWeapon", "be", "1=1")
	register_event("SetFOV", "player_currentWeapon", "be", "1>1")

	RegisterHam(Ham_Spawn, WEAPONBOX, "weapon_spawn", 1)

	RegisterHam(Ham_Weapon_SecondaryAttack, g_szWeaponEntity[W_KNIFE],
"player_useWeapon", 1) // not a typo
```

```
#if FEATURE_ADRENALINE == true

	RegisterHam(Ham_Weapon_SecondaryAttack, g_szWeaponEntity[W_USP],
"player_useWeaponSec", 1)
	RegisterHam(Ham_Weapon_SecondaryAttack, g_szWeaponEntity[W_FAMAS],
"player_useWeaponSec", 1)
	RegisterHam(Ham_Weapon_SecondaryAttack, g_szWeaponEntity[W_M4A1],
"player_useWeaponSec", 1)

#endif // FEATURE_ADRENALINE


#if FEATURE_C4 == true

	gMsg_BarTime = get_user_msgid("BarTime")
	gMsg_DeathMsg = get_user_msgid("DeathMsg")
	gMsg_SendAudio = get_user_msgid("SendAudio")

	register_message(gMsg_BarTime, "c4_used")
	register_message(gMsg_SendAudio, "msg_sendAudio")

#endif // FEATURE_C4

	gMsg_HostagePos = get_user_msgid("HostagePos")
	gMsg_HostageK = get_user_msgid("HostageK")
	gMsg_RoundTime = get_user_msgid("RoundTime")
	gMsg_SayText = get_user_msgid("SayText")
	gMsg_ScoreInfo = get_user_msgid("ScoreInfo")
	gMsg_ScoreAttrib = get_user_msgid("ScoreAttrib")
	gMsg_ScreenFade = get_user_msgid("ScreenFade")
	gMsg_TextMsg = get_user_msgid("TextMsg")
	gMsg_TeamScore = get_user_msgid("TeamScore")

	register_message(gMsg_TextMsg, "msg_textMsg")
	register_message(get_user_msgid("BombDrop"), "msg_block")
	register_message(get_user_msgid("ClCorpse"), "msg_block")
	register_message(gMsg_HostageK, "msg_block")
	register_message(gMsg_HostagePos, "msg_block")
	register_message(gMsg_RoundTime, "msg_roundTime")
	register_message(gMsg_ScreenFade, "msg_screenFade")
	register_message(gMsg_ScoreAttrib, "msg_scoreAttrib")
	register_message(gMsg_TeamScore, "msg_teamScore")
	register_message(gMsg_SayText, "msg_sayText")

	// CVARS
```

```
	pCvar_ctf_flagcaptureslay = register_cvar("ctf_flagcaptureslay", "0")
	pCvar_ctf_flagheal = register_cvar("ctf_flagheal", "1")
	pCvar_ctf_flagreturn = register_cvar("ctf_flagreturn", "120")
	pCvar_ctf_respawntime = register_cvar("ctf_respawntime", "10")
	pCvar_ctf_protection = register_cvar("ctf_protection", "5")
	pCvar_ctf_dynamiclights = register_cvar("ctf_dynamiclights", "1")
	pCvar_ctf_glows = register_cvar("ctf_glows", "1")
	pCvar_ctf_weaponstay = register_cvar("ctf_weaponstay", "15")
	pCvar_ctf_spawnmoney = register_cvar("ctf_spawnmoney", "1000")
	pCvar_ctf_itempercent = register_cvar("ctf_itempercent", "25")

#if FEATURE_ORPHEU == true

	register_srvcmd("ctf_infiniteround", "server_cmd_infiniteround")

	pCvar_ctf_infiniteround = register_cvar("_ctf_infiniteround_memory", "1")
	pCvar_ctf_flagendround = register_cvar("ctf_flagendround", "0")

#endif // FEATURE_ORPHEU


#if FEATURE_BUY == true

	pCvar_ctf_nospam_flash = register_cvar("ctf_nospam_flash", "20")
	pCvar_ctf_nospam_he = register_cvar("ctf_nospam_he", "20")
	pCvar_ctf_nospam_smoke = register_cvar("ctf_nospam_smoke", "20")
	pCvar_ctf_spawn_prim = register_cvar("ctf_spawn_prim", "m3")
	pCvar_ctf_spawn_sec = register_cvar("ctf_spawn_sec", "glock")
	pCvar_ctf_spawn_knife = register_cvar("ctf_spawn_knife", "1")

	gMsg_BuyClose = get_user_msgid("BuyClose")

#endif // FEATURE_BUY

	pCvar_ctf_sound[EVENT_TAKEN] = register_cvar("ctf_sound_taken", "1")
	pCvar_ctf_sound[EVENT_DROPPED] = register_cvar("ctf_sound_dropped", "1")
	pCvar_ctf_sound[EVENT_RETURNED] = register_cvar("ctf_sound_returned", "1")
	pCvar_ctf_sound[EVENT_SCORE] = register_cvar("ctf_sound_score", "1")

#if FEATURE_C4 == true

	pCvar_mp_c4timer = get_cvar_pointer("mp_c4timer")

#endif // FEATURE_C4
```

```
	pCvar_mp_winlimit = get_cvar_pointer("mp_winlimit")
	pCvar_mp_startmoney = get_cvar_pointer("mp_startmoney")
	pCvar_mp_fadetoblack = get_cvar_pointer("mp_fadetoblack")
	pCvar_mp_forcecamera = get_cvar_pointer("mp_forcecamera")
	pCvar_mp_forcechasecam = get_cvar_pointer("mp_forcechasecam")
	pCvar_mp_autoteambalance = get_cvar_pointer("mp_autoteambalance")

	// Plugin's forwards

	g_iFW_flag = CreateMultiForward("jctf_flag", ET_IGNORE, FP_CELL, FP_CELL,
FP_CELL, FP_CELL)


	// Variables

	new szGame[3]

	get_modname(szGame, charsmax(szGame))

	if(szGame[0] == 'c')
	{
		switch(szGame[1])
		{
			case 's': copy(g_szGame, charsmax(g_szGame), "CS 1.6 ") // leave
the space at the end
			case 'z': copy(g_szGame, charsmax(g_szGame), "CS:CZ ")
		}
	}

	get_mapname(g_szMap, charsmax(g_szMap))

	g_iMaxPlayers = get_maxplayers()


#if FEATURE_C4 == true
	// fake bomb target

	new ent = entity_create(g_szRemoveEntities[2])

	if(ent)
	{
		entity_spawn(ent)
		entity_set_size(ent, Float:{-8192.0, -8192.0, -8192.0}, Float:{8192.0, 8192.0,
8192.0})
	}
#endif // FEATURE_C4
```

```
#if FEATURE_ORPHEU == true
	g_bLinux = bool:is_linux_server()

	state disabled

	game_enableForwards()
#endif // FEATURE_ORPHEU
}

#if FEATURE_ORPHEU == true

public game_onInstallGameRules()
	g_pGameRules = OrpheuGetReturn();

public game_enableForwards() <> {}
public game_enableForwards() <disabled>
{
	g_oMapConditions =
OrpheuRegisterHook(OrpheuGetFunction("CheckMapConditions", "CHalfLifeMultiplay"),
"game_blockConditions")
	g_oWinConditions =
OrpheuRegisterHook(OrpheuGetFunction("CheckWinConditions", "CHalfLifeMultiplay"),
"game_blockConditions")

	if(g_bLinux)
		g_oRoundTimeExpired =
OrpheuRegisterHook(OrpheuGetFunction("HasRoundTimeExpired", "CHalfLifeMultiplay"),
"game_blockConditions")
	else
		game_memoryReplace(MEMORY_ROUNDTIME, {0x90, 0x90, 0x90})

	state enabled
}

public game_disableForwards() <> {}
public game_disableForwards() <enabled>
{
	OrpheuUnregisterHook(g_oMapConditions)
	OrpheuUnregisterHook(g_oWinConditions)

	if(g_bLinux)
		OrpheuUnregisterHook(g_oRoundTimeExpired)
	else
		game_memoryReplace(MEMORY_ROUNDTIME, {0xF6, 0xC4, 0x41})
```

```
	state disabled
}

public OrpheuHookReturn:game_blockConditions() <>
	return OrpheuIgnored

public OrpheuHookReturn:game_blockConditions() <enabled>
{
	OrpheuSetReturn(false)

	return OrpheuSupercede
}

game_memoryReplace(szID[], const iBytes[], const iLen = sizeof iBytes)
{
	new iAddress

	OrpheuMemoryGet(szID, iAddress)

	for(new i; i < iLen; i++)
	{
		OrpheuMemorySetAtAddress(iAddress, "roundTimeCheck|dummy", 1,
iBytes[i], iAddress)

		iAddress++
	}

	server_cmd("sv_restart 1")
}

public server_cmd_infiniteround()
{
	if(read_argc() == 2)
	{
		new szArg[2]

		read_argv(1, szArg, charsmax(szArg))

		new iSet = clamp(str_to_num(szArg), 0, 1)

		set_pcvar_num(pCvar_ctf_infiniteround, iSet)

		switch(iSet)
		{
			case 0: game_disableForwards()
			case 1: game_enableForwards()
```

```
                }
        }
        else
                server_print("^"ctf_infiniteround^" is ^"%d^"",
get_pcvar_num(pCvar_ctf_infiniteround))
}

#endif // FEATURE_ORPHEU

public game_description()
{
        new szFormat[32]

        formatex(szFormat, charsmax(szFormat), "%sjCTF v%s", g_szGame,
MOD_VERSION)
        forward_return(FMV_STRING, szFormat)

        return FMRES_SUPERCEDE
}

public plugin_cfg()
{
        new szFile[64]

        formatex(szFile, charsmax(szFile), FLAG_SAVELOCATION, g_szMap)

        new hFile = fopen(szFile, "rt")

        if(hFile)
        {
                new iFlagTeam = TEAM_RED
                new szData[24]
                new szOrigin[3][6]

                while(fgets(hFile, szData, charsmax(szData)))
                {
                        if(iFlagTeam > TEAM_BLUE)
                                break

                        trim(szData)
                        parse(szData, szOrigin[x], charsmax(szOrigin[]), szOrigin[y],
charsmax(szOrigin[]), szOrigin[z], charsmax(szOrigin[]))

                        g_fFlagBase[iFlagTeam][x] = str_to_float(szOrigin[x])
                        g_fFlagBase[iFlagTeam][y] = str_to_float(szOrigin[y])
                        g_fFlagBase[iFlagTeam][z] = str_to_float(szOrigin[z])
```

```
                        iFlagTeam++
                }

                fclose(hFile)
        }

        flag_spawn(TEAM_RED)
        flag_spawn(TEAM_BLUE)

        task_set(6.5, "plugin_postCfg")
}

public plugin_postCfg()
{
        set_cvar_num("mp_freezetime", 0)
        set_cvar_num("mp_limitteams", 0)
        set_cvar_num("mp_buytime", 99999999)
        server_cmd("sv_restart 1")
}

public plugin_natives()
{
      register_library("jctf")

        register_native("jctf_get_team", "native_get_team")
        register_native("jctf_get_flagcarrier", "native_get_flagcarrier")
        register_native("jctf_get_adrenaline", "native_get_adrenaline")
        register_native("jctf_add_adrenaline", "native_add_adrenaline")
        register_native("adre", "player_cmd_adrenaline", 1)
}

public plugin_end()
{
#if FEATURE_ORPHEU == true

        game_disableForwards()

#endif // FEATURE_ORPHEU

        DestroyForward(g_iFW_flag)
}
```

```
public native_get_team(iPlugin, iParams)
{
	/* jctf_get_team(id) */

	return g_iTeam[get_param(1)]
}

public native_get_flagcarrier(iPlugin, iParams)
{
	/* jctf_get_flagcarrier(id) */

	new id = get_param(1)

	return g_iFlagHolder[get_opTeam(g_iTeam[id])] == id
}

public native_get_adrenaline(iPlugin, iParams)
{
#if FEATURE_ADRENALINE == true

	/* jctf_get_adrenaline(id) */

	return g_iAdrenaline[get_param(1)]

#else // FEATURE_ADRENALINE

	log_error(AMX_ERR_NATIVE, "jctf_get_adrenaline() does not work ! main jCTF
plugin has FEATURE_ADRENALINE = false")

	return 0

#endif // FEATURE_ADRENALINE
}

public native_add_adrenaline(iPlugin, iParams)
{
#if FEATURE_ADRENALINE == true

	/* jctf_add_adrenaline(id, iAdd, szReason[]) */

	new id = get_param(1)
	new iAdd = get_param(2)
```

```
	new szReason[64]

	get_string(3, szReason, charsmax(szReason))

	if(strlen(szReason))
		player_award(id, 0, 0, iAdd, szReason)

	else
	{
		g_iAdrenaline[id] = clamp(g_iAdrenaline[id] + iAdd, 0, 100)

		player_hudAdrenaline(id)
	}

	return g_iAdrenaline[id]

#else // FEATURE_ADRENALINE

	log_error(AMX_ERR_NATIVE, "jctf_add_adrenaline() does not work ! main jCTF
plugin has FEATURE_ADRENALINE = false")

	return 0

#endif // FEATURE_ADRENALINE
}









public flag_spawn(iFlagTeam)
{
	if(g_fFlagBase[iFlagTeam][x] == 0.0 && g_fFlagBase[iFlagTeam][y] == 0.0 &&
g_fFlagBase[iFlagTeam][z] == 0.0)
	{
		new iFindSpawn = find_ent_by_class(g_iMaxPlayers, iFlagTeam ==
TEAM_BLUE ? "info_player_start" : "info_player_deathmatch")

		if(iFindSpawn)
		{
```

```
                        entity_get_vector(iFindSpawn, EV_VEC_origin,
g_fFlagBase[iFlagTeam])

                        server_print("[CTF] %s flag origin not defined, set on player spawn.",
g_szTeamName[iFlagTeam])
                        log_error(AMX_ERR_NOTFOUND, "[CTF] %s flag origin not defined,
set on player spawn.", g_szTeamName[iFlagTeam])
                }
                else
                {
                        server_print("[CTF] WARNING: player spawn for ^"%s^" team does
not exist !", g_szTeamName[iFlagTeam])
                        log_error(AMX_ERR_NOTFOUND, "[CTF] WARNING: player spawn
for ^"%s^" team does not exist !", g_szTeamName[iFlagTeam])
                        set_fail_state("Player spawn unexistent!")

                        return PLUGIN_CONTINUE
                }
        }
        else
                server_print("[CTF] %s flag and base spawned at: %.1f %.1f %.1f",
g_szTeamName[iFlagTeam], g_fFlagBase[iFlagTeam][x], g_fFlagBase[iFlagTeam][y],
g_fFlagBase[iFlagTeam][z])

        new ent
        new Float:fGameTime = get_gametime()

        // the FLAG

        ent = entity_create(INFO_TARGET)

        if(!ent)
                return flag_spawn(iFlagTeam)

        entity_set_model(ent, FLAG_MODEL)
        entity_set_string(ent, EV_SZ_classname, FLAG_CLASSNAME)
        entity_set_int(ent, EV_INT_body, iFlagTeam)
        entity_set_int(ent, EV_INT_sequence, FLAG_ANI_STAND)
        entity_spawn(ent)
        entity_set_origin(ent, g_fFlagBase[iFlagTeam])
        entity_set_size(ent, FLAG_HULL_MIN, FLAG_HULL_MAX)
        entity_set_vector(ent, EV_VEC_velocity, FLAG_SPAWN_VELOCITY)
        entity_set_vector(ent, EV_VEC_angles, FLAG_SPAWN_ANGLES)
        entity_set_edict(ent, EV_ENT_aiment, 0)
        entity_set_int(ent, EV_INT_movetype, MOVETYPE_TOSS)
        entity_set_int(ent, EV_INT_solid, SOLID_TRIGGER)
```

```
	entity_set_float(ent, EV_FL_gravity, 2.0)
	entity_set_float(ent, EV_FL_nextthink, fGameTime + FLAG_THINK)

	g_iFlagEntity[iFlagTeam] = ent
	g_iFlagHolder[iFlagTeam] = FLAG_HOLD_BASE

	// flag BASE

	ent = entity_create(INFO_TARGET)

	if(!ent)
		return flag_spawn(iFlagTeam)

	entity_set_string(ent, EV_SZ_classname, BASE_CLASSNAME)
	entity_set_model(ent, FLAG_MODEL)
	entity_set_int(ent, EV_INT_body, 0)
	entity_set_int(ent, EV_INT_sequence, FLAG_ANI_BASE)
	entity_spawn(ent)
	entity_set_origin(ent, g_fFlagBase[iFlagTeam])
	entity_set_vector(ent, EV_VEC_velocity, FLAG_SPAWN_VELOCITY)
	entity_set_int(ent, EV_INT_movetype, MOVETYPE_TOSS)

	if(get_pcvar_num(pCvar_ctf_glows))
		entity_set_int(ent, EV_INT_renderfx, kRenderFxGlowShell)

	entity_set_float(ent, EV_FL_renderamt, 100.0)
	entity_set_float(ent, EV_FL_nextthink, fGameTime + BASE_THINK)

	if(iFlagTeam == TEAM_RED)
		entity_set_vector(ent, EV_VEC_rendercolor, Float:{150.0, 0.0, 0.0})
	else
		entity_set_vector(ent, EV_VEC_rendercolor, Float:{0.0, 0.0, 150.0})

	g_iBaseEntity[iFlagTeam] = ent

	return PLUGIN_CONTINUE
}

public flag_think(ent)
{
	if(!is_valid_ent(ent))
		return

	entity_set_float(ent, EV_FL_nextthink, get_gametime() + FLAG_THINK)

	static id
```

```
static iStatus
static iFlagTeam
static iSkip[3]
static Float:fOrigin[3]
static Float:fPlayerOrigin[3]

iFlagTeam = (ent == g_iFlagEntity[TEAM_BLUE] ? TEAM_BLUE : TEAM_RED)

if(g_iFlagHolder[iFlagTeam] == FLAG_HOLD_BASE)
	fOrigin = g_fFlagBase[iFlagTeam]
else
	entity_get_vector(ent, EV_VEC_origin, fOrigin)

g_fFlagLocation[iFlagTeam] = fOrigin

iStatus = 0

if(++iSkip[iFlagTeam] >= FLAG_SKIPTHINK)
{
	iSkip[iFlagTeam] = 0

	if(1 <= g_iFlagHolder[iFlagTeam] <= g_iMaxPlayers)
	{
		id = g_iFlagHolder[iFlagTeam]

		set_hudmessage(HUD_HELP)
		show_hudmessage(id, "%L", id, "HUD_YOUHAVEFLAG")

		iStatus = 1
	}
	else if(g_iFlagHolder[iFlagTeam] == FLAG_HOLD_DROPPED)
		iStatus = 2

	message_begin(MSG_BROADCAST, gMsg_HostagePos)
	write_byte(0)
	write_byte(iFlagTeam)
	engfunc(EngFunc_WriteCoord, fOrigin[x])
	engfunc(EngFunc_WriteCoord, fOrigin[y])
	engfunc(EngFunc_WriteCoord, fOrigin[z])
	message_end()

	message_begin(MSG_BROADCAST, gMsg_HostageK)
	write_byte(iFlagTeam)
	message_end()

	static iStuck[3]
```

```
                if(g_iFlagHolder[iFlagTeam] >= FLAG_HOLD_BASE && !(entity_get_int(ent,
EV_INT_flags) & FL_ONGROUND))
                {
                        if(++iStuck[iFlagTeam] > 4)
                        {
                                flag_autoReturn(ent)

                                log_message("^"%s^" flag is outside world, auto-returned.",
g_szTeamName[iFlagTeam])

                                return
                        }
                }
                else
                        iStuck[iFlagTeam] = 0
        }

        for(id = 1; id <= g_iMaxPlayers; id++)
        {
                if(g_iTeam[id] == TEAM_NONE || g_bBot[id])
                        continue

                /* Check flag proximity for pickup */
                if(g_iFlagHolder[iFlagTeam] >= FLAG_HOLD_BASE)
                {
                        entity_get_vector(id, EV_VEC_origin, fPlayerOrigin)

                        if(get_distance_f(fOrigin, fPlayerOrigin) <= FLAG_PICKUPDISTANCE)
                                flag_touch(ent, id)
                }

                /* Send dynamic lights to players that have them enabled */
                if(g_iFlagHolder[iFlagTeam] != FLAG_HOLD_BASE && g_bLights[id])
                {
                        message_begin(MSG_ONE_UNRELIABLE, SVC_TEMPENTITY, _,
id)
                        write_byte(TE_DLIGHT)
                        engfunc(EngFunc_WriteCoord, fOrigin[x])
                        engfunc(EngFunc_WriteCoord, fOrigin[y])
                        engfunc(EngFunc_WriteCoord, fOrigin[z] + (g_iFlagHolder[iFlagTeam]
== FLAG_HOLD_DROPPED ? 32 : -16))
                        write_byte(FLAG_LIGHT_RANGE)
                        write_byte(iFlagTeam == TEAM_RED ? 100 : 0)
                        write_byte(0)
                        write_byte(iFlagTeam == TEAM_BLUE ? 155 : 0)
```

```
                        write_byte(FLAG_LIGHT_LIFE)
                        write_byte(FLAG_LIGHT_DECAY)
                        message_end()
                }

                /* If iFlagTeam's flag is stolen or dropped, constantly warn team players */
                if(iStatus && g_iTeam[id] == iFlagTeam)
                {
                        set_hudmessage(HUD_HELP2)
                        show_hudmessage(id, "%L", id, (iStatus == 1 ?
"HUD_ENEMYHASFLAG" : "HUD_RETURNYOURFLAG"))
                }
        }
}

flag_sendHome(iFlagTeam)
{
        new ent = g_iFlagEntity[iFlagTeam]

        entity_set_edict(ent, EV_ENT_aiment, 0)
        entity_set_origin(ent, g_fFlagBase[iFlagTeam])
        entity_set_int(ent, EV_INT_sequence, FLAG_ANI_STAND)
        entity_set_int(ent, EV_INT_movetype, MOVETYPE_TOSS)
        entity_set_int(ent, EV_INT_solid, SOLID_TRIGGER)
        entity_set_vector(ent, EV_VEC_velocity, FLAG_SPAWN_VELOCITY)
        entity_set_vector(ent, EV_VEC_angles, FLAG_SPAWN_ANGLES)

        g_iFlagHolder[iFlagTeam] = FLAG_HOLD_BASE
}

flag_take(iFlagTeam, id)
{
        if(g_bProtected[id])
                player_removeProtection(id, "PROTECTION_TOUCHFLAG")

        new ent = g_iFlagEntity[iFlagTeam]

        entity_set_edict(ent, EV_ENT_aiment, id)
        entity_set_int(ent, EV_INT_movetype, MOVETYPE_FOLLOW)
        entity_set_int(ent, EV_INT_solid, SOLID_NOT)

        g_iFlagHolder[iFlagTeam] = id

        message_begin(MSG_BROADCAST, gMsg_ScoreAttrib)
        write_byte(id)
        write_byte(g_iTeam[id] == TEAM_BLUE ? 4 : 2)
```

```
	message_end()

	player_updateSpeed(id)
}

public flag_touch(ent, id)
{
#if FLAG_IGNORE_BOTS == true

	if(!g_bAlive[id] || g_bBot[id])
		return

#else // FLAG_IGNORE_BOTS

	if(!g_bAlive[id])
		return

#endif // FLAG_IGNORE_BOTS

	new iFlagTeam = (g_iFlagEntity[TEAM_BLUE] == ent ? TEAM_BLUE : TEAM_RED)

	if(1 <= g_iFlagHolder[iFlagTeam] <= g_iMaxPlayers) // if flag is carried we don't care
		return

	new Float:fGameTime = get_gametime()

	if(g_fLastDrop[id] > fGameTime)
		return

	new iTeam = g_iTeam[id]

	if(!(TEAM_RED <= g_iTeam[id] <= TEAM_BLUE))
		return

	new iFlagTeamOp = get_opTeam(iFlagTeam)
	new szName[32]

	get_user_name(id, szName, charsmax(szName))

	if(iTeam == iFlagTeam) // If the PLAYER is on the same team as the FLAG
	{
		if(g_iFlagHolder[iFlagTeam] == FLAG_HOLD_DROPPED) // if the team's flag
is dropped, return it to base
		{
			flag_sendHome(iFlagTeam)
```

```
			task_remove(ent)

			player_award(id, REWARD_RETURN, "%L", id,
"REWARD_RETURN")

			ExecuteForward(g_iFW_flag, g_iForwardReturn, FLAG_RETURNED,
id, iFlagTeam, false)

			new iAssists = 0

			for(new i = 1; i <= g_iMaxPlayers; i++)
			{
				if(i != id && g_bAssisted[i][iFlagTeam] && g_iTeam[i] ==
iFlagTeam)
				{
					player_award(i, REWARD_RETURN_ASSIST, "%L", i,
"REWARD_RETURN_ASSIST")

					ExecuteForward(g_iFW_flag, g_iForwardReturn,
FLAG_RETURNED, i, iFlagTeam, true)

					iAssists++
				}

				g_bAssisted[i][iFlagTeam] = false
			}

			if(1 <= g_iFlagHolder[iFlagTeamOp] <= g_iMaxPlayers)
				g_bAssisted[id][iFlagTeamOp] = true

			if(iAssists)
			{
				new szFormat[64]

				format(szFormat, charsmax(szFormat), "%s + %d assists",
szName, iAssists)

				game_announce(EVENT_RETURNED, iFlagTeam, szFormat)
			}
			else
				game_announce(EVENT_RETURNED, iFlagTeam, szName)

			log_message("<%s>%s returned the ^"%s^" flag.",
g_szTeamName[iTeam], szName, g_szTeamName[iFlagTeam])

			set_hudmessage(HUD_HELP)
```

```
                        show_hudmessage(id, "%L", id, "HUD_RETURNEDFLAG")

                        if(g_bProtected[id])
                                player_removeProtection(id, "PROTECTION_TOUCHFLAG")
                }
                else if(g_iFlagHolder[iFlagTeam] == FLAG_HOLD_BASE &&
g_iFlagHolder[iFlagTeamOp] == id) // if the PLAYER has the ENEMY FLAG and the FLAG is
in the BASE make SCORE
                {
                        message_begin(MSG_BROADCAST, gMsg_ScoreAttrib)
                        write_byte(id)
                        write_byte(0)
                        message_end()

                        player_award(id, REWARD_CAPTURE, "%L", id,
"REWARD_CAPTURE")

                        ExecuteForward(g_iFW_flag, g_iForwardReturn, FLAG_CAPTURED,
id, iFlagTeamOp, false)

                        new iAssists = 0

                        for(new i = 1; i <= g_iMaxPlayers; i++)
                        {
                                if(i != id && g_iTeam[i] > 0 && g_iTeam[i] == iTeam)
                                {
                                        if(g_bAssisted[i][iFlagTeamOp])
                                        {
                                                player_award(i, REWARD_CAPTURE_ASSIST,
"%L", i, "REWARD_CAPTURE_ASSIST")

                                                ExecuteForward(g_iFW_flag,
g_iForwardReturn, FLAG_CAPTURED, i, iFlagTeamOp, true)

                                                iAssists++
                                        }
                                        else
                                                player_award(i, REWARD_CAPTURE_TEAM,
"%L", i, "REWARD_CAPTURE_TEAM")
                                }

                                g_bAssisted[i][iFlagTeamOp] = false
                        }

                        set_hudmessage(HUD_HELP)
                        show_hudmessage(id, "%L", id, "HUD_CAPTUREDFLAG")
```

```
                        if(iAssists)
                        {
                                new szFormat[64]

                                format(szFormat, charsmax(szFormat), "%s + %d assists",
szName, iAssists)

                                game_announce(EVENT_SCORE, iFlagTeam, szFormat)
                        }
                        else
                                game_announce(EVENT_SCORE, iFlagTeam, szName)

                        log_message("<%s>%s captured the ^"%s^" flag. (%d assists)",
g_szTeamName[iTeam], szName, g_szTeamName[iFlagTeamOp], iAssists)

                        emessage_begin(MSG_BROADCAST, gMsg_TeamScore)
                        ewrite_string(g_szCSTeams[iFlagTeam])
                        ewrite_short(++g_iScore[iFlagTeam])
                        emessage_end()

                        flag_sendHome(iFlagTeamOp)

                        player_updateSpeed(id)

                        g_fLastDrop[id] = fGameTime + 3.0

                        if(g_bProtected[id])
                                player_removeProtection(id, "PROTECTION_TOUCHFLAG")
                        else
                                player_updateRender(id)

                        if(0 < get_pcvar_num(pCvar_mp_winlimit) <= g_iScore[iFlagTeam])
                        {
                                emessage_begin(MSG_ALL, SVC_INTERMISSION) //
hookable mapend
                                emessage_end()

                                return
                        }

#if FEATURE_ORPHEU == true

                        new iFlagRoundEnd = get_pcvar_num(pCvar_ctf_flagendround)

                        if(iFlagRoundEnd)
```

```
{
	static OrpheuFunction:ofEndRoundMsg
	static OrpheuFunction:ofUpdateTeamScores
	static OrpheuFunction:ofCheckWinConditions

	if(!ofEndRoundMsg)
		ofEndRoundMsg =
OrpheuGetFunction("EndRoundMessage")

	if(!ofUpdateTeamScores)
		ofUpdateTeamScores =
OrpheuGetFunction("UpdateTeamScores", "CHalfLifeMultiplay")

	if(!ofCheckWinConditions)
		ofCheckWinConditions =
OrpheuGetFunction("CheckWinConditions", "CHalfLifeMultiplay")

	new iEvent
	new iWinStatus
	new szWinOffset[20]
	new szWinMessage[16]

	switch(iFlagTeam)
	{
		case TEAM_RED:
		{
			iEvent = 9
			iWinStatus = 2
			copy(szWinOffset, charsmax(szWinOffset),
"m_iNumTerroristWins")
			copy(szWinMessage,
charsmax(szWinMessage), "#Terrorists_Win")
		}

		case TEAM_BLUE:
		{
			iEvent = 8
			iWinStatus = 1
			copy(szWinOffset, charsmax(szWinOffset),
"m_iNumCTWins")
			copy(szWinMessage,
charsmax(szWinMessage), "#CTs_Win")
		}
	}

	OrpheuCallSuper(ofUpdateTeamScores, g_pGameRules)
```

```
                                OrpheuCallSuper(ofEndRoundMsg, szWinMessage, iEvent)

                                OrpheuMemorySetAtAddress(g_pGameRules,
"m_iRoundWinStatus", 1, iWinStatus)
                                OrpheuMemorySetAtAddress(g_pGameRules,
"m_fTeamCount", 1, get_gametime() + 3.0)
                                OrpheuMemorySetAtAddress(g_pGameRules,
"m_bRoundTerminating", 1, true)
                                OrpheuMemorySetAtAddress(g_pGameRules, szWinOffset, 1,
g_iScore[iFlagTeam])

                                OrpheuCallSuper(ofCheckWinConditions, g_pGameRules)
                        }

#else
                        new iFlagRoundEnd = 1

#endif // FEATURE_ORPHEU

                        if(iFlagRoundEnd && get_pcvar_num(pCvar_ctf_flagcaptureslay))
                        {
                                for(new i = 1; i <= g_iMaxPlayers; i++)
                                {
                                        if(g_iTeam[i] == iFlagTeamOp)
                                        {
                                                user_kill(i)
                                                player_print(i, i, "%L", i,
"DEATH_FLAGCAPTURED")
                                        }
                                }
                        }
                }
        }
        else
        {
                if(g_iFlagHolder[iFlagTeam] == FLAG_HOLD_BASE)
                {
                        player_award(id, REWARD_STEAL, "%L", id, "REWARD_STEAL")

                        ExecuteForward(g_iFW_flag, g_iForwardReturn, FLAG_STOLEN, id,
iFlagTeam, false)

                        log_message("<%s>%s stole the ^"%s^" flag.",
g_szTeamName[iTeam], szName, g_szTeamName[iFlagTeam])
                }
                else
```

```
		{
			player_award(id, REWARD_PICKUP, "%L", id, "REWARD_PICKUP")

			ExecuteForward(g_iFW_flag, g_iForwardReturn, FLAG_PICKED, id,
iFlagTeam, false)

			log_message("<%s>%s picked up the ^"%s^" flag.",
g_szTeamName[iTeam], szName, g_szTeamName[iFlagTeam])
		}

		set_hudmessage(HUD_HELP)
		show_hudmessage(id, "%L", id, "HUD_YOUHAVEFLAG")

		flag_take(iFlagTeam, id)

		g_bAssisted[id][iFlagTeam] = true

		task_remove(ent)

		if(g_bProtected[id])
			player_removeProtection(id, "PROTECTION_TOUCHFLAG")
		else
			player_updateRender(id)

		game_announce(EVENT_TAKEN, iFlagTeam, szName)
	}
}

public flag_autoReturn(ent)
{
	task_remove(ent)

	new iFlagTeam = (g_iFlagEntity[TEAM_BLUE] == ent ? TEAM_BLUE :
(g_iFlagEntity[TEAM_RED] == ent ? TEAM_RED : 0))

	if(!iFlagTeam)
		return

	flag_sendHome(iFlagTeam)

	ExecuteForward(g_iFW_flag, g_iForwardReturn, FLAG_AUTORETURN, 0,
iFlagTeam, false)

	game_announce(EVENT_RETURNED, iFlagTeam, NULL)

	log_message("^"%s^" flag returned automatically", g_szTeamName[iFlagTeam])
```

```
}

public base_think(ent)
{
	if(!is_valid_ent(ent))
		return

	if(!get_pcvar_num(pCvar_ctf_flagheal))
	{
		entity_set_float(ent, EV_FL_nextthink, get_gametime() + 10.0) /* recheck
each 10s seconds */

		return
	}

	entity_set_float(ent, EV_FL_nextthink, get_gametime() + BASE_THINK)

	new iFlagTeam = (g_iBaseEntity[TEAM_BLUE] == ent ? TEAM_BLUE : TEAM_RED)

	if(g_iFlagHolder[iFlagTeam] != FLAG_HOLD_BASE)
		return

	static id
	static iHealth

	id = -1

	while((id = find_ent_in_sphere(id, g_fFlagBase[iFlagTeam],
BASE_HEAL_DISTANCE)) != 0)
	{
		if(1 <= id <= g_iMaxPlayers && g_bAlive[id] && g_iTeam[id] == iFlagTeam)
		{
			iHealth = get_user_health(id)

			if(iHealth < g_iMaxHealth[id])
			{
				set_user_health(id, iHealth + 1)
```

```
                                player_healingEffect(id)
                        }
                }

                if(id >= g_iMaxPlayers)
                        break
        }
}
```

```
public client_putinserver(id)
{
        g_bBot[id] = (is_user_bot(id) ? true : false)

        g_iTeam[id] = TEAM_SPEC
        g_bFirstSpawn[id] = true
        g_bRestarted[id] = false
        g_bLights[id] = (g_bBot[id] ? false : (get_pcvar_num(pCvar_ctf_dynamiclights) ? true
: false));

        task_set(3.0, "client_putinserverPost", id - TASK_PUTINSERVER)
}

public client_putinserverPost(id)
{
        id += TASK_PUTINSERVER

        player_print(id, id, "%L", id, "JOIN_INFO", "^x04", MOD_TITLE, "^x01", "^x03",
MOD_AUTHOR, "^x01")

        client_print(id, print_console, "^n%s", SEPARATOR)
        client_print(id, print_console, "              %s v%s - %L", MOD_TITLE,
MOD_VERSION, id, "QH_TITLE")
        client_print(id, print_console, "                %L %s^n%s", id, "QH_MADEBY",
MOD_AUTHOR, SEPARATOR)
        client_print(id, print_console, "    %L", id, "QH_LINE1")
        client_print(id, print_console, "    %L", id, "QH_LINE2")
```

```
	client_print(id, print_console, "    %L", id, "QH_LINE3")
	client_print(id, print_console, "^n    %L", id, "QH_HELP")

#if FEATURE_ADRENALINE == true

	client_print(id, print_console, "^n    %L", id, "QH_ADRENALINE")

#endif // FEATURE_ADRENALINE

#if FEATURE_ORPHEU == false || FEATURE_BUY == false || FEATURE_C4 == false ||
FEATURE_ADRENALINE == false

	player_print(id, id, "%L", id, "JOIN_NOFEATURES")

#endif // FEATURE_ORPHEU || FEATURE_BUY || FEATURE_C4 ||
FEATURE_ADRENALINE
}

public client_disconnect(id)
{
	player_dropFlag(id)
	task_remove(id)

	g_iTeam[id] = TEAM_NONE
	g_iAdrenaline[id] = 0
	g_iAdrenalineUse[id] = 0

	g_bAlive[id] = false
	g_bLights[id] = false
	g_bFreeLook[id] = false
	g_bAssisted[id][TEAM_RED] = false
	g_bAssisted[id][TEAM_BLUE] = false
}

public player_joinTeam()
{
	new id = read_data(1)

	if(g_bAlive[id])
		return

	new szTeam[2]

	read_data(2, szTeam, charsmax(szTeam))

	switch(szTeam[0])
```

```
	{
		case 'T':
		{
			if(g_iTeam[id] == TEAM_RED && g_bFirstSpawn[id])
			{
				new iRespawn = get_pcvar_num(pCvar_ctf_respawntime)

				if(iRespawn > 0)
					player_respawn(id - TASK_RESPAWN, iRespawn + 1)

				task_remove(id - TASK_TEAMBALANCE)
				task_set(1.0, "player_checkTeam", id -
TASK_TEAMBALANCE)
			}

			g_iTeam[id] = TEAM_RED
		}

		case 'C':
		{
			if(g_iTeam[id] == TEAM_BLUE && g_bFirstSpawn[id])
			{
				new iRespawn = get_pcvar_num(pCvar_ctf_respawntime)

				if(iRespawn > 0)
					player_respawn(id - TASK_RESPAWN, iRespawn + 1)

				task_remove(id - TASK_TEAMBALANCE)
				task_set(1.0, "player_checkTeam", id -
TASK_TEAMBALANCE)
			}

			g_iTeam[id] = TEAM_BLUE
		}

		case 'U':
		{
			g_iTeam[id] = TEAM_NONE
			g_bFirstSpawn[id] = true
		}

		default:
		{
			player_screenFade(id, {0,0,0,0}, 0.0, 0.0, FADE_OUT, false)
			player_allowChangeTeam(id)
```

```
                        g_iTeam[id] = TEAM_SPEC
                        g_bFirstSpawn[id] = true
                }
        }
}

public player_spawn(id)
{
        if(!is_user_alive(id) || (!g_bRestarted[id] && g_bAlive[id]))
                return

        /* make sure we have team right */

        switch(cs_get_user_team(id))
        {
                case CS_TEAM_T: g_iTeam[id] = TEAM_RED
                case CS_TEAM_CT: g_iTeam[id] = TEAM_BLUE
                default: return
        }

        g_bAlive[id] = true
        g_bDefuse[id] = false
        g_bBuyZone[id] = true
        g_bFreeLook[id] = false
        g_fLastBuy[id] = Float:{0.0, 0.0, 0.0, 0.0}

        task_remove(id - TASK_PROTECTION)
        task_remove(id - TASK_EQUIPAMENT)
        task_remove(id - TASK_DAMAGEPROTECTION)
        task_remove(id - TASK_TEAMBALANCE)
        task_remove(id - TASK_ADRENALINE)
        task_remove(id - TASK_DEFUSE)

#if FEATURE_BUY == true

        task_set(0.1, "player_spawnEquipament", id - TASK_EQUIPAMENT)

#endif // FEATURE_BUY

        task_set(0.2, "player_checkVitals", id - TASK_CHECKHP)

#if FEATURE_ADRENALINE == true

        player_hudAdrenaline(id)

#endif // FEATURE_ADRENALINE
```

```
	new iProtection = get_pcvar_num(pCvar_ctf_protection)

	if(iProtection > 0)
		player_protection(id - TASK_PROTECTION, iProtection)

	message_begin(MSG_BROADCAST, gMsg_ScoreAttrib)
	write_byte(id)
	write_byte(0)
	message_end()

	if(g_bFirstSpawn[id] || g_bRestarted[id])
	{
		g_bRestarted[id] = false
		g_bFirstSpawn[id] = false

		cs_set_user_money(id, get_pcvar_num(pCvar_mp_startmoney))
	}
	else if(g_bSuicide[id])
	{
		g_bSuicide[id] = false

		player_print(id, id, "%L", id, "SPAWN_NOMONEY")
	}
	else
		cs_set_user_money(id, clamp((cs_get_user_money(id) +
get_pcvar_num(pCvar_ctf_spawnmoney)), get_pcvar_num(pCvar_mp_startmoney), 16000))
}

public player_checkVitals(id)
{
	id += TASK_CHECKHP

	if(!g_bAlive[id])
		return

	/* in case player is VIP or whatever special class that sets armor */
	new CsArmorType:iArmorType
	new iArmor = cs_get_user_armor(id, iArmorType)

	g_iMaxArmor[id] = (iArmor > 0 ? iArmor : 100)
	g_iMaxHealth[id] = get_user_health(id)
}

#if FEATURE_BUY == true
```

```
public player_spawnEquipament(id)
{
	id += TASK_EQUIPAMENT

	if(!g_bAlive[id])
		return

	strip_user_weapons(id)

	if(get_pcvar_num(pCvar_ctf_spawn_knife))
		give_item(id, g_szWeaponEntity[W_KNIFE])

	new szWeapon[3][24]

	get_pcvar_string(pCvar_ctf_spawn_prim, szWeapon[1], charsmax(szWeapon[]))
	get_pcvar_string(pCvar_ctf_spawn_sec, szWeapon[2], charsmax(szWeapon[]))

	for(new iWeapon, i = 2; i >= 1; i--)
	{
		iWeapon = 0

		if(strlen(szWeapon[i]))
		{
			for(new w = 1; w < sizeof g_szWeaponCommands; w++)
			{
				if(g_iWeaponSlot[w] == i && equali(szWeapon[i],
g_szWeaponCommands[w][0]))
				{
					iWeapon = w
					break
				}
			}

			if(iWeapon)
			{
				give_item(id, g_szWeaponEntity[iWeapon])
				cs_set_user_bpammo(id, iWeapon, g_iBPAmmo[iWeapon])
			}
			else
				log_error(AMX_ERR_NOTFOUND, "Invalid %s weapon:
^"%s^", please fix ctf_spawn_%s cvar", (i == 1 ? "primary" : "secondary"), szWeapon[i], (i ==
1 ? "prim" : "sec"))
		}
	}
}
```

```
#endif // FEATURE_BUY

public player_protection(id, iStart)
{
	id += TASK_PROTECTION

	if(!(TEAM_RED <= g_iTeam[id] <= TEAM_BLUE))
		return

	static iCount[33]

	if(iStart)
	{
		iCount[id] = iStart + 1

		g_bProtected[id] = true

		player_updateRender(id)
	}

	if(--iCount[id] > 0)
	{
		set_hudmessage(HUD_RESPAWN)
		show_hudmessage(id, "%L", id, "PROTECTION_LEFT", iCount[id])

		task_set(1.0, "player_protection", id - TASK_PROTECTION)
	}
	else
		player_removeProtection(id, "PROTECTION_EXPIRED")
}

public player_removeProtection(id, szLang[])
{
	if(!(TEAM_RED <= g_iTeam[id] <= TEAM_BLUE))
		return

	g_bProtected[id] = false

	task_remove(id - TASK_PROTECTION)
	task_remove(id - TASK_DAMAGEPROTECTION)

	set_hudmessage(HUD_PROTECTION)
	show_hudmessage(id, "%L", id, szLang)

	player_updateRender(id)
}
```

```
public player_currentWeapon(id)
{
	if(!g_bAlive[id])
		return

	static bool:bZoom[33]

	new iZoom = read_data(1)

	if(1 < iZoom <= 90) /* setFOV event */
		bZoom[id] = bool:(iZoom <= 40)

	else /* CurWeapon event */
	{
		if(!bZoom[id]) /* if not zooming, get weapon speed */
			g_fWeaponSpeed[id] = g_fWeaponRunSpeed[read_data(2)]

		else /* if zooming, set zoom speed */
			g_fWeaponSpeed[id] = g_fWeaponRunSpeed[0]

		player_updateSpeed(id)
	}
}

public client_PostThink(id)
{
	if(!g_bAlive[id])
		return

	static iOffset
	static iShield[33]

	iOffset = get_pdata_int(id, m_iUserPrefs)

	if(iOffset & (1<<24)) /* Shield available */
	{
		if(iOffset & (1<<16)) /* Uses shield */
		{
			if(iShield[id] < 2) /* Trigger only once */
			{
				iShield[id] = 2

				g_fWeaponSpeed[id] = 180.0

				player_updateSpeed(id)
```

```
                        }
                }
                else if(iShield[id] == 2) /* Doesn't use the shield anymore */
                {
                        iShield[id] = 1

                        g_fWeaponSpeed[id] = 250.0

                        player_updateSpeed(id)
                }
        }
        else if(iShield[id]) /* Shield not available anymore */
                iShield[id] = 0
}

public player_useWeapon(ent)
{
        if(!is_valid_ent(ent))
                return

        static id

        id = entity_get_edict(ent, EV_ENT_owner)

        if(1 <= id <= g_iMaxPlayers && g_bAlive[id])
        {
                if(g_bProtected[id])
                        player_removeProtection(id, "PROTECTION_WEAPONUSE")

#if FEATURE_ADRENALINE == true
                else if(g_iAdrenalineUse[id] == ADRENALINE_BERSERK)
                {
                        set_pdata_float(ent, m_flNextPrimaryAttack, get_pdata_float(ent,
m_flNextPrimaryAttack, 4) * BERSERKER_SPEED1)
                        set_pdata_float(ent, m_flNextSecondaryAttack, get_pdata_float(ent,
m_flNextSecondaryAttack, 4) * BERSERKER_SPEED2)
                }
#endif // FEATURE_ADRENALINE
        }
}

#if FEATURE_ADRENALINE == true

public player_useWeaponSec(ent)
{
        if(!is_valid_ent(ent))
```

```
		return

	static id

	id = entity_get_edict(ent, EV_ENT_owner)

	if(1 <= id <= g_iMaxPlayers && g_bAlive[id] && g_iAdrenalineUse[id] ==
ADRENALINE_BERSERK)
	{
		set_pdata_float(ent, m_flNextPrimaryAttack, get_pdata_float(ent,
m_flNextPrimaryAttack, 4) * BERSERKER_SPEED1)
		set_pdata_float(ent, m_flNextSecondaryAttack, get_pdata_float(ent,
m_flNextSecondaryAttack, 4) * BERSERKER_SPEED2)
	}
}

#endif // FEATURE_ADRENALINE


public player_damage(id, iWeapon, iAttacker, Float:fDamage, iType)
{
	if(g_bProtected[id])
	{
		player_updateRender(id, fDamage)

		task_remove(id - TASK_DAMAGEPROTECTION)
		task_set(0.1, "player_damageProtection", id -
TASK_DAMAGEPROTECTION)

		entity_set_vector(id, EV_VEC_punchangle, FLAG_SPAWN_ANGLES)

		return HAM_SUPERCEDE
	}

#if FEATURE_ADRENALINE == true

	else if(1 <= iAttacker <= g_iMaxPlayers && g_iAdrenalineUse[iAttacker] ==
ADRENALINE_BERSERK && g_iTeam[iAttacker] != g_iTeam[id])
	{
		SetHamParamFloat(4, fDamage * BERSERKER_DAMAGE)

		new iOrigin[3]

		get_user_origin(id, iOrigin)

		message_begin(MSG_PVS, SVC_TEMPENTITY, iOrigin)
```

```
                write_byte(TE_BLOODSPRITE)
                write_coord(iOrigin[x] + random_num(-15, 15))
                write_coord(iOrigin[y] + random_num(-15, 15))
                write_coord(iOrigin[z] + random_num(-15, 15))
                write_short(gSpr_blood2)
                write_short(gSpr_blood1)
                write_byte(248)
                write_byte(18)
                message_end()

                return HAM_OVERRIDE
        }

#endif // FEATURE_ADRENALINE

        return HAM_IGNORED
}

public player_damageProtection(id)
{
        id += TASK_DAMAGEPROTECTION

        if(g_bAlive[id])
                player_updateRender(id)
}

public player_killed(id, killer)
{
        g_bAlive[id] = false
        g_bBuyZone[id] = false

        task_remove(id - TASK_RESPAWN)
        task_remove(id - TASK_PROTECTION)
        task_remove(id - TASK_EQUIPAMENT)
        task_remove(id - TASK_DAMAGEPROTECTION)
        task_remove(id - TASK_TEAMBALANCE)
        task_remove(id - TASK_ADRENALINE)
        task_remove(id - TASK_DEFUSE)

        new szHint[10]

#if FEATURE_C4 == true && FEATURE_ADRENALINE == true

        formatex(szHint, charsmax(szHint), "HINT_%d", random_num(1, 12))

#else
```

```
		new iHint

		while((iHint = random_num(1, 12)))
		{
#if FEATURE_ADRENALINE == false
			if(iHint == 1 || iHint == 7 || iHint == 9)
				continue
#endif // FEATURE_ADRENALINE


#if FEATURE_C4 == false
			if(iHint == 4 || iHint == 8 || iHint == 10)
				continue
#endif // FEATURE_C4

			break
		}

		formatex(szHint, charsmax(szHint), "HINT_%d", iHint)

#endif // FEATURE_C4 || FEATURE_ADRENALINE

		set_hudmessage(HUD_HINT)
		show_hudmessage(id, "%L: %L", id, "HINT", id, szHint)
		client_print(id, print_console, "%s%L: %L", CONSOLE_PREFIX, id, "HINT", id,
szHint)

#if FEATURE_C4 == true

		new iWeapon = entity_get_edict(id, EV_ENT_dmg_inflictor)
		new szWeapon[10]
		new bool:bC4 = false

		if(iWeapon > g_iMaxPlayers && is_valid_ent(iWeapon))
		{
			entity_get_string(iWeapon, EV_SZ_classname, szWeapon,
charsmax(szWeapon))

			if(equal(szWeapon, GRENADE) && get_pdata_int(iWeapon, 96) & (1<<8))
			{
				message_begin(MSG_ALL, gMsg_DeathMsg)
				write_byte(killer)
				write_byte(id)
				write_byte(0)
				write_string("c4")
```

```
                        message_end()

                        bC4 = true
                }
        }

#endif // FEATURE_C4

        if(id == killer || !(1 <= killer <= g_iMaxPlayers))
        {
                g_bSuicide[id] = true

                player_award(id, PENALTY_SUICIDE, "%L", id, "PENALTY_SUICIDE")

#if FEATURE_C4 == true

                if(bC4)
                        player_setScore(id, -1, 1)

#endif // FEATURE_C4

        }
        else if(1 <= killer <= g_iMaxPlayers)
        {
                if(g_iTeam[id] == g_iTeam[killer])
                {

#if FEATURE_C4 == true

                        if(bC4)
                        {
                                player_setScore(killer, -1, 0)
                                cs_set_user_money(killer, clamp(cs_get_user_money(killer) -
3300, 0, 16000), 1)
                        }

#endif // FEATURE_C4

                        player_award(killer, PENALTY_TEAMKILL, "%L", killer,
"PENALTY_TEAMKILL")
                }
                else
                {

#if FEATURE_C4 == true
```

```
                        if(bC4)
                        {
                                player_setScore(killer, -1, 0)
                                player_setScore(id, 0, 1)

                                cs_set_user_money(killer, clamp(cs_get_user_money(killer) +
300, 0, 16000), 1)
                        }

#endif // FEATURE_C4

                        if(id == g_iFlagHolder[g_iTeam[killer]])
                        {
                                g_bAssisted[killer][g_iTeam[killer]] = true

                                player_award(killer, REWARD_KILLCARRIER, "%L", killer,
"REWARD_KILLCARRIER")

                                message_begin(MSG_BROADCAST, gMsg_ScoreAttrib)
                                write_byte(id)
                                write_byte(0)
                                message_end()
                        }
                        else
                        {
                                player_spawnItem(id)
                                player_award(killer, REWARD_KILL, "%L", killer,
"REWARD_KILL")
                        }
                }
        }

#if FEATURE_ADRENALINE == true

        if(g_iAdrenalineUse[id])
        {

                switch(g_iAdrenalineUse[id])
                {
                        case ADRENALINE_SPEED:
                        {
                                message_begin(MSG_BROADCAST, SVC_TEMPENTITY)
                                write_byte(TE_KILLBEAM)
                                write_short(id)
                                message_end()
                        }
```

```
                }

                g_iAdrenaline[id] = 0
                g_iAdrenalineUse[id] = 0

                player_updateRender(id)
                player_hudAdrenaline(id)
        }

#endif // FEATURE_ADRENALINE

        new iRespawn = get_pcvar_num(pCvar_ctf_respawntime)

        if(iRespawn > 0)
                player_respawn(id - TASK_RESPAWN, iRespawn)

        player_dropFlag(id)
        player_allowChangeTeam(id)

        task_set(1.0, "player_checkTeam", id - TASK_TEAMBALANCE)
}

public player_checkTeam(id)
{
        id += TASK_TEAMBALANCE

        if(!(TEAM_RED <= g_iTeam[id] <= TEAM_BLUE) || g_bAlive[id] ||
!get_pcvar_num(pCvar_mp_autoteambalance))
                return

        new iPlayers[3]
        new iTeam = g_iTeam[id]
        new iOpTeam = get_opTeam(iTeam)

        for(new i = 1; i <= g_iMaxPlayers; i++)
        {
                if(TEAM_RED <= g_iTeam[i] <= TEAM_BLUE)
                        iPlayers[g_iTeam[i]]++
        }

        if((iPlayers[iTeam] > 1 && !iPlayers[iOpTeam]) || iPlayers[iTeam] > (iPlayers[iOpTeam]
+ 1))
        {
                player_allowChangeTeam(id)

                engclient_cmd(id, "jointeam", (iOpTeam == TEAM_BLUE ? "2" : "1"))
```

```
                set_task(2.0, "player_forceJoinClass", id)

                player_print(id, id, "%L", id, "DEATH_TRANSFER", "^x04", id,
g_szMLTeamName[iOpTeam], "^x01")
        }
}

public player_forceJoinClass(id)
{
        engclient_cmd(id, "joinclass", "5")
}

public player_respawn(id, iStart)
{
        id += TASK_RESPAWN

        if(!(TEAM_RED <= g_iTeam[id] <= TEAM_BLUE) || g_bAlive[id])
                return

        static iCount[33]

        if(iStart)
                iCount[id] = iStart + 1

        set_hudmessage(HUD_RESPAWN)

        if(--iCount[id] > 0)
        {
                show_hudmessage(id, "%L", id, "RESPAWNING_IN", iCount[id])
                client_print(id, print_console, "%L", id, "RESPAWNING_IN", iCount[id])

                task_set(1.0, "player_respawn", id - TASK_RESPAWN)
        }
        else
        {
                show_hudmessage(id, "%L", id, "RESPAWNING")
                client_print(id, print_console, "%L", id, "RESPAWNING")

                entity_set_int(id, EV_INT_deadflag, DEAD_RESPAWNABLE)
                entity_set_int(id, EV_INT_iuser1, 0)
                entity_think(id)
                entity_spawn(id)
                set_user_health(id, 100)
        }
}
```

```
#if FEATURE_ADRENALINE == true

public player_cmd_buySpawn(id)
{
	if(g_bAlive[id] || !(TEAM_RED <= g_iTeam[id] <= TEAM_BLUE))
		player_print(id, id, "%L", id, "INSTANTSPAWN_NOTEAM")

	else if(g_iAdrenaline[id] < INSTANTSPAWN_COST)
		player_print(id, id, "%L", id, "INSTANTSPAWN_NOADRENALINE",
INSTANTSPAWN_COST)

	else
	{
		g_iAdrenaline[id] -= INSTANTSPAWN_COST

		player_print(id, id, "%L", id, "INSTANTSPAWN_BOUGHT",
INSTANTSPAWN_COST)

		task_remove(id)
		player_respawn(id - TASK_RESPAWN, -1)
	}

	return PLUGIN_HANDLED
}

#endif // FEATURE_ADRENALINE

public player_cmd_dropFlag(id)
{
	if(!g_bAlive[id] || id != g_iFlagHolder[get_opTeam(g_iTeam[id])])
		player_print(id, id, "%L", id, "DROPFLAG_NOFLAG")

	else
	{
		new iOpTeam = get_opTeam(g_iTeam[id])

		player_dropFlag(id)
		player_award(id, PENALTY_DROP, "%L", id, "PENALTY_MANUALDROP")

		ExecuteForward(g_iFW_flag, g_iForwardReturn, FLAG_MANUALDROP, id,
iOpTeam, false)

		g_bAssisted[id][iOpTeam] = false
	}
```

```
	return PLUGIN_HANDLED
}

public player_dropFlag(id)
{
	new iOpTeam = get_opTeam(g_iTeam[id])

	if(id != g_iFlagHolder[iOpTeam])
		return

	new ent = g_iFlagEntity[iOpTeam]

	if(!is_valid_ent(ent))
		return

	g_fLastDrop[id] = get_gametime() + 2.0
	g_iFlagHolder[iOpTeam] = FLAG_HOLD_DROPPED

	entity_set_edict(ent, EV_ENT_aiment, -1)
	entity_set_int(ent, EV_INT_movetype, MOVETYPE_TOSS)
	entity_set_int(ent, EV_INT_sequence, FLAG_ANI_DROPPED)
	entity_set_int(ent, EV_INT_solid, SOLID_TRIGGER)
	entity_set_origin(ent, g_fFlagLocation[iOpTeam])

	new Float:fReturn = get_pcvar_float(pCvar_ctf_flagreturn)

	if(fReturn > 0)
		task_set(fReturn, "flag_autoReturn", ent)

	if(g_bAlive[id])
	{
		new Float:fVelocity[3]

		velocity_by_aim(id, 200, fVelocity)

		fVelocity[z] = 0.0

		entity_set_vector(ent, EV_VEC_velocity, fVelocity)

		player_updateSpeed(id)
		player_updateRender(id)

		message_begin(MSG_BROADCAST, gMsg_ScoreAttrib)
		write_byte(id)
		write_byte(0)
		message_end()
```

```
	}
	else
		entity_set_vector(ent, EV_VEC_velocity, FLAG_DROP_VELOCITY)

	new szName[32]

	get_user_name(id, szName, charsmax(szName))

	game_announce(EVENT_DROPPED, iOpTeam, szName)

	ExecuteForward(g_iFW_flag, g_iForwardReturn, FLAG_DROPPED, id, iOpTeam,
false)

	g_fFlagDropped[iOpTeam] = get_gametime()

	log_message("<%s>%s dropped the ^"%s^" flag.", g_szTeamName[g_iTeam[id]],
szName, g_szTeamName[iOpTeam])
}

public player_cmd_say(id)
{
	static Float:fLastUsage[33]

	new Float:fGameTime = get_gametime()

	if((fLastUsage[id] + 0.5) > fGameTime)
		return PLUGIN_HANDLED

	fLastUsage[id] = fGameTime

	new szMsg[128]

	read_args(szMsg, charsmax(szMsg))
	remove_quotes(szMsg)
	trim(szMsg)

	if(equal(szMsg, NULL))
		return PLUGIN_HANDLED

	if(equal(szMsg[0], "@"))
		return PLUGIN_CONTINUE

	new szFormat[192]
	new szName[32]

	get_user_name(id, szName, charsmax(szName))
```

```
	switch(g_iTeam[id])
	{
		case TEAM_RED, TEAM_BLUE: formatex(szFormat, charsmax(szFormat), "^x01%s^x03%s ^x01:  %s", (g_bAlive[id] ? NULL : "^x01*DEAD* "), szName, szMsg)
		case TEAM_NONE, TEAM_SPEC: formatex(szFormat, charsmax(szFormat), "^x01*SPEC* ^x03%s ^x01:  %s", szName, szMsg)
	}

	for(new i = 1; i <= g_iMaxPlayers; i++)
	{
		if(i == id || g_iTeam[i] == TEAM_NONE || g_bAlive[i] == g_bAlive[id] || g_bBot[id])
			continue

		message_begin(MSG_ONE, gMsg_SayText, _, i)
		write_byte(id)
		write_string(szFormat)
		message_end()
	}

#if FEATURE_BUY == true

	if(equali(szMsg, "/buy"))
	{
		player_menu_buy(id, 0)

		return CHAT_SHOW_COMMANDS ? PLUGIN_CONTINUE : PLUGIN_HANDLED
	}

#endif // FEATURE_BUY

#if FEATURE_ADRENALINE == true

	if(equali(szMsg, "/spawn"))
	{
		player_cmd_buySpawn(id)

		return CHAT_SHOW_COMMANDS ? PLUGIN_CONTINUE : PLUGIN_HANDLED
	}

	if(equali(szMsg, "/adrenaline"))
	{
		player_cmd_adrenaline(id)
```

```
		return CHAT_SHOW_COMMANDS ? PLUGIN_CONTINUE :
PLUGIN_HANDLED
	}

#endif // FEATURE_ADRENALINE

	if(equali(szMsg, "/help"))
	{
		player_cmd_help(id)

		return CHAT_SHOW_COMMANDS ? PLUGIN_CONTINUE :
PLUGIN_HANDLED
	}

	if(equali(szMsg, "/dropflag"))
	{
		player_cmd_dropFlag(id)

		return CHAT_SHOW_COMMANDS ? PLUGIN_CONTINUE :
PLUGIN_HANDLED
	}

	if(equali(szMsg, "/lights", 7))
	{
		player_cmd_setLights(id, szMsg[8])

		return CHAT_SHOW_COMMANDS ? PLUGIN_CONTINUE :
PLUGIN_HANDLED
	}

	if(equali(szMsg, "/sounds", 7))
	{
		player_cmd_setSounds(id, szMsg[8])

		return CHAT_SHOW_COMMANDS ? PLUGIN_CONTINUE :
PLUGIN_HANDLED
	}

	return PLUGIN_CONTINUE
}

public player_cmd_sayTeam(id)
{
	static Float:fLastUsage[33]
```

```
	new Float:fGameTime = get_gametime()

	if((fLastUsage[id] + 0.5) > fGameTime)
		return PLUGIN_HANDLED

	fLastUsage[id] = fGameTime

	new szMsg[128]

	read_args(szMsg, charsmax(szMsg))
	remove_quotes(szMsg)
	trim(szMsg)

	if(equal(szMsg, NULL))
		return PLUGIN_HANDLED

	if(equal(szMsg[0], "@"))
		return PLUGIN_CONTINUE

	new szFormat[192]
	new szName[32]

	get_user_name(id, szName, charsmax(szName))

	switch(g_iTeam[id])
	{
		case TEAM_RED, TEAM_BLUE: formatex(szFormat, charsmax(szFormat),
"^x01%s(%L) ^x03%s ^x01:  %s", (g_bAlive[id] ? NULL : "*DEAD* "), LANG_PLAYER,
g_szMLFlagTeam[g_iTeam[id]], szName, szMsg)
		case TEAM_NONE, TEAM_SPEC: formatex(szFormat, charsmax(szFormat),
"^x01*SPEC*(%L) ^x03%s ^x01:  %s", LANG_PLAYER, g_szMLTeamName[TEAM_SPEC],
szName, szMsg)
	}

	for(new i = 1; i <= g_iMaxPlayers; i++)
	{
		if(i == id || g_iTeam[i] == TEAM_NONE || g_iTeam[i] != g_iTeam[id] ||
g_bAlive[i] == g_bAlive[id] || g_bBot[id])
			continue

		message_begin(MSG_ONE, gMsg_SayText, _, i)
		write_byte(id)
		write_string(szFormat)
		message_end()
	}
```

```
#if FEATURE_BUY == true

	if(equali(szMsg, "/buy"))
	{
		player_menu_buy(id, 0)

		return CHAT_SHOW_COMMANDS ? PLUGIN_CONTINUE :
PLUGIN_HANDLED
	}

#endif // FEATURE_BUY

#if FEATURE_ADRENALINE == true

	if(equali(szMsg, "/spawn"))
	{
		player_cmd_buySpawn(id)

		return CHAT_SHOW_COMMANDS ? PLUGIN_CONTINUE :
PLUGIN_HANDLED
	}

	if(equali(szMsg, "/adrenaline"))
	{
		player_cmd_adrenaline(id)

		return CHAT_SHOW_COMMANDS ? PLUGIN_CONTINUE :
PLUGIN_HANDLED
	}

#endif // FEATURE_ADRENALINE

	if(equali(szMsg, "/dropflag"))
	{
		player_cmd_dropFlag(id)

		return CHAT_SHOW_COMMANDS ? PLUGIN_CONTINUE :
PLUGIN_HANDLED
	}

	if(equali(szMsg, "/help"))
	{
		player_cmd_help(id)

		return CHAT_SHOW_COMMANDS ? PLUGIN_CONTINUE :
PLUGIN_HANDLED
```

```
	}

	return PLUGIN_CONTINUE
}

public player_cmd_help(id)
{
	client_cmd(id, "hideconsole;toggleconsole")

	client_print(id, print_console, "^n^n^n^n%s", SEPARATOR)
	client_print(id, print_console, "		%s v%s - %L^n
Mod by %s", MOD_TITLE, MOD_VERSION, id, "HELP_TITLE", MOD_AUTHOR)
	client_print(id, print_console, SEPARATOR)
	client_print(id, print_console, "    1. %L^n    2. %L^n    3. %L^n    4. %L", id,
"HELP_1", id, "HELP_2", id, "HELP_3", id, "HELP_4")
	client_print(id, print_console, "^n    --- 1: %L ---^n", id, "HELP_1")
	client_print(id, print_console, "%L", id, "HELP_1_LINE1")
	client_print(id, print_console, "%L", id, "HELP_1_LINE2")
	client_print(id, print_console, "%L", id, "HELP_1_LINE3")
	client_print(id, print_console, "%L", id, "HELP_1_LINE4")
	client_print(id, print_console, "^n    --- 2: %L ---", id, "HELP_2")
	client_print(id, print_console, "%L", id, "HELP_2_NOTE")
	client_print(id, print_console, "%L", id, "HELP_2_LINE1")
	client_print(id, print_console, "%L", id, "HELP_2_LINE2")

#if FEATURE_ADRENALINE == true
	client_print(id, print_console, "%L", id, "HELP_2_LINE3", INSTANTSPAWN_COST)
#endif // FEATURE_ADRENALINE

	client_print(id, print_console, "%L", id, "HELP_2_LINE4")

#if FEATURE_ADRENALINE == true
	client_print(id, print_console, "%L", id, "HELP_2_LINE5")
#endif // FEATURE_ADRENALINE

	client_print(id, print_console, "^n    --- 3: %L ---", id, "HELP_3")

#if FEATURE_ORPHEU == false
	client_print(id, print_console, " * %L", id, "HELP_3_INFROUND", id, "OFF")
	client_print(id, print_console, " * %L", id, "HELP_3_ROUNDEND", id, "OFF")
#else
	client_print(id, print_console, " * %L", id, "HELP_3_INFROUND", id,
get_pcvar_num(pCvar_ctf_infiniteround) ? "ON" : "OFF")
	client_print(id, print_console, " * %L", id, "HELP_3_ROUNDEND", id,
get_pcvar_num(pCvar_ctf_flagendround) ? "ON" : "OFF")
#endif
```

```
	client_print(id, print_console, " * %L", id, "HELP_3_CAPTURESLAY", id,
get_pcvar_num(pCvar_ctf_flagcaptureslay) ? "ON" : "OFF")

#if FEATURE_BUY == true
	client_print(id, print_console, " * %L", id, "HELP_3_BUY", id, "ON")
#else
	client_print(id, print_console, " * %L", id, "HELP_3_BUY", id, "OFF")
#endif

#if FEATURE_C4 == true
	client_print(id, print_console, " * %L", id, "HELP_3_C4", id, "ON")
#else
	client_print(id, print_console, " * %L", id, "HELP_3_C4", id, "OFF")
#endif

#if FEATURE_ADRENALINE == true
	client_print(id, print_console, " * %L", id, "HELP_3_ADRENALINE", id, "ON")
#else
	client_print(id, print_console, " * %L", id, "HELP_3_ADRENALINE", id, "OFF")
#endif

	client_print(id, print_console, " * %L", id, "HELP_3_FLAGHEAL", id,
get_pcvar_num(pCvar_ctf_flagheal) ? "ON" : "OFF")
	client_print(id, print_console, " * %L", id, "HELP_3_RESPAWN",
get_pcvar_num(pCvar_ctf_respawntime))
	client_print(id, print_console, " * %L", id, "HELP_3_PROTECTION",
get_pcvar_num(pCvar_ctf_protection))
	client_print(id, print_console, " * %L", id, "HELP_3_FLAGRETURN",
get_pcvar_num(pCvar_ctf_flagreturn))
	client_print(id, print_console, " * %L", id, "HELP_3_WEAPONSTAY",
get_pcvar_num(pCvar_ctf_weaponstay))
	client_print(id, print_console, " * %L", id, "HELP_3_ITEMDROP",
get_pcvar_num(pCvar_ctf_itempercent))

	client_print(id, print_console, "^n    --- 4: %L ---", id, "HELP_4")
	client_print(id, print_console, "          %L:
http://forums.alliedmods.net/showthread.php?t=132115", id, "HELP_4_LINE1")
	client_print(id, print_console, "          %L: http://thehunters.ro/jctf", id,
"HELP_4_LINE2")
	client_print(id, print_console, SEPARATOR)

	return PLUGIN_HANDLED
}

public player_cmd_setLights(id, const szMsg[])
```

```
{
	switch(szMsg[1])
	{
		case 'n':
		{
			g_bLights[id] = true
			player_print(id, id, "%L", id, "LIGHTS_ON", "^x04", "^x01")
		}

		case 'f':
		{
			g_bLights[id] = false
			player_print(id, id, "%L", id, "LIGHTS_OFF", "^x04", "^x01")
		}

		default: player_print(id, id, "%L", id, "LIGHTS_INVALID", "^x04", "^x01",
"^x04")
	}

	return PLUGIN_HANDLED
}

public player_cmd_setSounds(id, const szMsg[])
{
	if(equali(szMsg, "test"))
	{
		player_print(id, id, "%L", id, "SOUNDS_TEST", "^x04 Red Flag Taken^x01")
		client_cmd(id, "mp3 play ^"sound/ctf/red_flag_taken.mp3^"")

		return PLUGIN_HANDLED
	}

	new iVol = (strlen(szMsg) ? str_to_num(szMsg) : -1)

	if(0 <= iVol <= 10)
	{
		client_cmd(id, "mp3volume %.2f", iVol == 0 ? 0.0 : iVol * 0.1)
		player_print(id, id, "%L", id, "SOUNDS_SET", "^x04", iVol)
	}
	else
		player_print(id, id, "%L", id, "SOUNDS_INVALID", "^x04 0^x01", "^x04
10^x01", "^x04 test")

	return PLUGIN_HANDLED
}
```

```
#if FEATURE_ADRENALINE == true

public player_cmd_adrenaline(id)
{
	player_hudAdrenaline(id)

	if(!(TEAM_RED <= g_iTeam[id] <= TEAM_BLUE) || !g_bAlive[id])
		return player_print(id, id, "%L", id, "ADR_ALIVE")

	if(g_iAdrenalineUse[id])
		return player_print(id, id, "%L", id, "ADR_USING")

	if(g_iAdrenaline[id] < 100)
		return player_print(id, id, "%L", id, "ADR_MORE", 100)

	new szFormat[256]

	formatex(szFormat, charsmax(szFormat), "\r%L:^n^n\w1. \y%L \d(%L)^n\w2. \y%L \d(%L)^n\w3. \y%L \d(%L)^n\w4. \y%L \d(%L)^n^n\d\w0. %L",
		id, "ADR_MENU_TITLE",
		id, "ADR_SPEED", id, "ADR_SPEED_DESC",
		id, "ADR_BERSERK", id, "ADR_BERSERK_DESC",
		id, "ADR_REGENERATE", id, "ADR_REGENERATE_DESC",
		id, "ADR_INVISIBILITY", id, "ADR_INVISIBILITY_DESC",
		id, "EXIT"
	)

	show_menu(id, MENU_KEYS_ADRENALINE, szFormat, -1, MENU_ADRENALINE)

	return PLUGIN_HANDLED
}

public player_key_adrenaline(id, iKey)
{
	if(!(TEAM_RED <= g_iTeam[id] <= TEAM_BLUE) || !g_bAlive[id])
		return player_print(id, id, "%L", id, "ADR_ALIVE")
```

```
	if(g_iAdrenalineUse[id])
		return player_print(id, id, "%L", id, "ADR_USING")

	if(g_iAdrenaline[id] < 100)
		return player_print(id, id, "%L", id, "ADR_USING", 100)

	iKey += 1

	if(1 <= iKey <= 4)
		player_useAdrenaline(id, iKey)

	return PLUGIN_HANDLED
}

public player_useAdrenaline(id, iUse)
{
	if(!(1 <= iUse <= 4))
		return PLUGIN_HANDLED

	if(!(TEAM_RED <= g_iTeam[id] <= TEAM_BLUE) || !g_bAlive[id])
		return player_print(id, id, "%L", id, "ADR_ALIVE")

	if(g_iAdrenalineUse[id])
		return player_print(id, id, "%L", id, "ADR_USING")

	if(g_iAdrenaline[id] < 100)
		return player_print(id, id, "%L", id, "ADR_USING", 100)

	if(g_bProtected[id])
		player_removeProtection(id, "PROTECTION_ADRENALINE")

	g_iAdrenalineUse[id] = iUse

	task_set(0.25, "player_adrenalineDrain", id - TASK_ADRENALINE)

	if(iUse == ADRENALINE_SPEED)
	{
		message_begin(MSG_BROADCAST, SVC_TEMPENTITY)
		write_byte(TE_BEAMFOLLOW)
		write_short(id)
		write_short(gSpr_trail)
		write_byte(8) // life in 0.1's
		write_byte(6) // line width in 0.1's
		write_byte(255)
		write_byte(255)
```

```
                write_byte(0)
                write_byte(255) // brightness
                message_end()

                player_updateSpeed(id)
        }

        player_updateRender(id)

        new iOrigin[3]

        get_user_origin(id, iOrigin)

        message_begin(MSG_PVS, SVC_TEMPENTITY, iOrigin)
        write_byte(TE_IMPLOSION)
        write_coord(iOrigin[x])
        write_coord(iOrigin[y])
        write_coord(iOrigin[z])
        write_byte(128) // radius
        write_byte(32) // count
        write_byte(4) // life in 0.1's
        message_end()

        emit_sound(id, CHAN_ITEM, SND_ADRENALINE, VOL_NORM, ATTN_NORM, 0,
255)

        return PLUGIN_HANDLED
}

public player_adrenalineDrain(id)
{
        id += TASK_ADRENALINE

        if(!g_bAlive[id] || !g_iAdrenalineUse[id] || !(TEAM_RED <= g_iTeam[id] <=
TEAM_BLUE))
        {
                g_iAdrenaline[id] = 0
                return
        }

        if(g_iAdrenaline[id] > 0)
        {
                new iDrain = (player_hasFlag(id) ? 2 : 1)

                g_iAdrenaline[id] = clamp(g_iAdrenaline[id] - (g_iAdrenalineUse[id] ==
ADRENALINE_REGENERATE ? iDrain : iDrain * 2), 0, 100)
```

```
		switch(g_iAdrenalineUse[id])
		{
			case ADRENALINE_REGENERATE:
			{
				new iHealth = get_user_health(id)

				if(iHealth < (g_iMaxHealth[id] + REGENERATE_EXTRAHP))
					set_user_health(id, iHealth + 1)

				else
				{
					new CsArmorType:ArmorType
					new iArmor = cs_get_user_armor(id, ArmorType)

					if(iArmor < g_iMaxArmor[id])
						cs_set_user_armor(id, iArmor + 1, ArmorType)
				}

				player_healingEffect(id)
			}
		}

		task_set(0.25, "player_adrenalineDrain", id - TASK_ADRENALINE)
	}
	else
	{
		new iUsed = g_iAdrenalineUse[id]

		g_iAdrenaline[id] = 0
		g_iAdrenalineUse[id] = 0 /* to allow player_updateSpeed() to work correctly */

		switch(iUsed)
		{
			case ADRENALINE_SPEED:
			{
				player_updateSpeed(id)

				message_begin(MSG_BROADCAST, SVC_TEMPENTITY)
				write_byte(TE_KILLBEAM)
				write_short(id)
				message_end()
			}

			case ADRENALINE_BERSERK, ADRENALINE_INVISIBILITY:
player_updateRender(id)
```

```
                }
        }

        player_hudAdrenaline(id)
}

#endif // FEATURE_ADRENALINE









public admin_cmd_moveFlag(id, level, cid)
{
        if(!cmd_access(id, level, cid, 2))
                return PLUGIN_HANDLED

        new szTeam[2]

        read_argv(1, szTeam, charsmax(szTeam))

        new iTeam = str_to_num(szTeam)

        if(!(TEAM_RED <= iTeam <= TEAM_BLUE))
        {
                switch(szTeam[0])
                {
                        case 'r', 'R': iTeam = 1
                        case 'b', 'B': iTeam = 2
                }
        }

        if(!(TEAM_RED <= iTeam <= TEAM_BLUE))
                return PLUGIN_HANDLED

        entity_get_vector(id, EV_VEC_origin, g_fFlagBase[iTeam])

        entity_set_origin(g_iBaseEntity[iTeam], g_fFlagBase[iTeam])
```

```
		entity_set_vector(g_iBaseEntity[iTeam], EV_VEC_velocity,
FLAG_SPAWN_VELOCITY)

		if(g_iFlagHolder[iTeam] == FLAG_HOLD_BASE)
		{
			entity_set_origin(g_iFlagEntity[iTeam], g_fFlagBase[iTeam])
			entity_set_vector(g_iFlagEntity[iTeam], EV_VEC_velocity,
FLAG_SPAWN_VELOCITY)
		}

		new szName[32]
		new szSteam[48]

		get_user_name(id, szName, charsmax(szName))
		get_user_authid(id, szSteam, charsmax(szSteam))

		log_amx("Admin %s<%s><%s> moved %s flag to %.2f %.2f %.2f", szName,
szSteam, g_szTeamName[g_iTeam[id]], g_szTeamName[iTeam], g_fFlagBase[iTeam][0],
g_fFlagBase[iTeam][1], g_fFlagBase[iTeam][2])

		show_activity_key("ADMIN_MOVEBASE_1", "ADMIN_MOVEBASE_2", szName,
LANG_PLAYER, g_szMLFlagTeam[iTeam])

		client_print(id, print_console, "%s%L", CONSOLE_PREFIX, id,
"ADMIN_MOVEBASE_MOVED", id, g_szMLFlagTeam[iTeam])

		return PLUGIN_HANDLED
}

public admin_cmd_saveFlags(id, level, cid)
{
		if(!cmd_access(id, level, cid, 1))
			return PLUGIN_HANDLED

		new iOrigin[3][3]
		new szFile[96]
		new szBuffer[1024]

		FVecIVec(g_fFlagBase[TEAM_RED], iOrigin[TEAM_RED])
		FVecIVec(g_fFlagBase[TEAM_BLUE], iOrigin[TEAM_BLUE])

		formatex(szBuffer, charsmax(szBuffer), "%d %d %d^n%d %d %d",
iOrigin[TEAM_RED][x], iOrigin[TEAM_RED][y], iOrigin[TEAM_RED][z],
iOrigin[TEAM_BLUE][x], iOrigin[TEAM_BLUE][y], iOrigin[TEAM_BLUE][z])
		formatex(szFile, charsmax(szFile), FLAG_SAVELOCATION, g_szMap)
```

```
	if(file_exists(szFile))
		delete_file(szFile)

	write_file(szFile, szBuffer)

	new szName[32]
	new szSteam[48]

	get_user_name(id, szName, charsmax(szName))
	get_user_authid(id, szSteam, charsmax(szSteam))

	log_amx("Admin %s<%s><%s> saved flag positions.", szName, szSteam,
g_szTeamName[g_iTeam[id]])

	client_print(id, print_console, "%s%L %s", CONSOLE_PREFIX, id,
"ADMIN_MOVEBASE_SAVED", szFile)

	return PLUGIN_HANDLED
}

public admin_cmd_returnFlag(id, level, cid)
{
	if(!cmd_access(id, level, cid, 2))
		return PLUGIN_HANDLED

	new szTeam[2]

	read_argv(1, szTeam, charsmax(szTeam))

	new iTeam = str_to_num(szTeam)

	if(!(TEAM_RED <= iTeam <= TEAM_BLUE))
	{
		switch(szTeam[0])
		{
			case 'r', 'R': iTeam = 1
			case 'b', 'B': iTeam = 2
		}
	}

	if(!(TEAM_RED <= iTeam <= TEAM_BLUE))
		return PLUGIN_HANDLED

	if(g_iFlagHolder[iTeam] == FLAG_HOLD_DROPPED)
	{
		if(g_fFlagDropped[iTeam] < (get_gametime() - ADMIN_RETURNWAIT))
```

```
		{
			new szName[32]
			new szSteam[48]

			new Float:fFlagOrigin[3]

			entity_get_vector(g_iFlagEntity[iTeam], EV_VEC_origin, fFlagOrigin)

			flag_sendHome(iTeam)

			ExecuteForward(g_iFW_flag, g_iForwardReturn,
FLAG_ADMINRETURN, id, iTeam, false)

			game_announce(EVENT_RETURNED, iTeam, NULL)

			get_user_name(id, szName, charsmax(szName))
			get_user_authid(id, szSteam, charsmax(szSteam))

			log_message("^"%s^" flag returned by admin %s<%s><%s>",
g_szTeamName[iTeam], szName, szSteam, g_szTeamName[g_iTeam[id]])
			log_amx("Admin %s<%s><%s> returned %s flag from %.2f %.2f
%.2f", szName, szSteam, g_szTeamName[g_iTeam[id]], g_szTeamName[iTeam],
fFlagOrigin[0], fFlagOrigin[1], fFlagOrigin[2])

			show_activity_key("ADMIN_RETURN_1", "ADMIN_RETURN_2",
szName, LANG_PLAYER, g_szMLFlagTeam[iTeam])

			client_print(id, print_console, "%s%L", CONSOLE_PREFIX, id,
"ADMIN_RETURN_DONE", id, g_szMLFlagTeam[iTeam])
		}
		else
			client_print(id, print_console, "%s%L", CONSOLE_PREFIX, id,
"ADMIN_RETURN_WAIT", id, g_szMLFlagTeam[iTeam], ADMIN_RETURNWAIT)
	}
	else
		client_print(id, print_console, "%s%L", CONSOLE_PREFIX, id,
"ADMIN_RETURN_NOTDROPPED", id, g_szMLFlagTeam[iTeam])

	return PLUGIN_HANDLED
}
```

```
#if FEATURE_BUY == true

public player_inBuyZone(id)
{
	if(!g_bAlive[id])
		return

	g_bBuyZone[id] = (read_data(1) ? true : false)

	if(!g_bBuyZone[id])
		set_pdata_int(id, 205, 0) // no "close menu upon exit buyzone" thing
}

public player_cmd_setAutobuy(id)
{
	new iIndex
	new szWeapon[24]
	new szArgs[1024]

	read_args(szArgs, charsmax(szArgs))
	remove_quotes(szArgs)
	trim(szArgs)

	while(contain(szArgs, WHITESPACE) != -1)
	{
		strbreak(szArgs, szWeapon, charsmax(szWeapon), szArgs,
charsmax(szArgs))

		for(new bool:bFound, w = W_P228; w <= W_NVG; w++)
		{
			if(!bFound)
			{
				for(new i = 0; i < 2; i++)
				{
					if(!bFound && equali(g_szWeaponCommands[w][i],
szWeapon))
					{
						bFound = true

						g_iAutobuy[id][iIndex++] = w
```

```
                                        }
                                }
                        }
                }
        }

        player_cmd_autobuy(id)

        return PLUGIN_HANDLED
}

public player_cmd_autobuy(id)
{
        if(!g_bAlive[id])
                return PLUGIN_HANDLED

        if(!g_bBuyZone[id])
        {
                client_print(id, print_center, "%L", id, "BUY_NOTINZONE")
                return PLUGIN_HANDLED
        }

        new iMoney = cs_get_user_money(id)

        for(new bool:bBought[6], iWeapon, i = 0; i < sizeof g_iAutobuy[]; i++)
        {
                if(!g_iAutobuy[id][i])
                        return PLUGIN_HANDLED

                iWeapon = g_iAutobuy[id][i]

                if(bBought[g_iWeaponSlot[iWeapon]])
                        continue

#if FEATURE_ADRENALINE == true

                if((g_iWeaponPrice[iWeapon] > 0 && g_iWeaponPrice[iWeapon] > iMoney) ||
(g_iWeaponAdrenaline[iWeapon] > 0 && g_iWeaponAdrenaline[iWeapon] >
g_iAdrenaline[id]))
                        continue

#else // FEATURE_ADRENALINE

                if(g_iWeaponPrice[iWeapon] > 0 && g_iWeaponPrice[iWeapon] > iMoney)
                        continue
```

```
#endif // FEATURE_ADRENALINE

		player_buyWeapon(id, iWeapon)
		bBought[g_iWeaponSlot[iWeapon]] = true
	}

	return PLUGIN_HANDLED
}

public player_cmd_setRebuy(id)
{
	new iIndex
	new szType[18]
	new szArgs[256]

	read_args(szArgs, charsmax(szArgs))
	replace_all(szArgs, charsmax(szArgs), "^"", NULL)
	trim(szArgs)

	while(contain(szArgs, WHITESPACE) != -1)
	{
		split(szArgs, szType, charsmax(szType), szArgs, charsmax(szArgs),
WHITESPACE)

		for(new i = 1; i < sizeof g_szRebuyCommands; i++)
		{
			if(equali(szType, g_szRebuyCommands[i]))
				g_iRebuy[id][++iIndex] = i
		}
	}

	player_cmd_rebuy(id)

	return PLUGIN_HANDLED
}

public player_cmd_rebuy(id)
{
	if(!g_bAlive[id])
		return PLUGIN_HANDLED

	if(!g_bBuyZone[id])
	{
		client_print(id, print_center, "%L", id, "BUY_NOTINZONE")
		return PLUGIN_HANDLED
	}
```

```
	new iBought

	for(new iType, iBuy, i = 1; i < sizeof g_iRebuy[]; i++)
	{
		iType = g_iRebuy[id][i]

		if(!iType)
			continue

		iBuy = g_iRebuyWeapons[id][iType]

		if(!iBuy)
			continue

		switch(iType)
		{
			case primary, secondary: player_buyWeapon(id, iBuy)

			case armor: player_buyWeapon(id, (iBuy == 2 ? W_VESTHELM :
W_VEST))

			case he: player_buyWeapon(id, W_HEGRENADE)

			case flash:
			{
				player_buyWeapon(id, W_FLASHBANG)

				if(iBuy == 2)
					player_buyWeapon(id, W_FLASHBANG)
			}

			case smoke: player_buyWeapon(id, W_SMOKEGRENADE)

			case nvg: player_buyWeapon(id, W_NVG)
		}

		iBought++

		if(iType == flash && iBuy == 2)
			iBought++
	}

	if(iBought)
		client_print(id, print_center, "%L", id, "BUY_REBOUGHT", iBought)
```

```
	return PLUGIN_HANDLED
}

public player_addRebuy(id, iWeapon)
{
	if(!g_bAlive[id])
		return

	switch(g_iWeaponSlot[iWeapon])
	{
		case 1: g_iRebuyWeapons[id][primary] = iWeapon
		case 2: g_iRebuyWeapons[id][secondary] = iWeapon

		default:
		{
			switch(iWeapon)
			{
				case W_VEST: g_iRebuyWeapons[id][armor] =
(g_iRebuyWeapons[id][armor] == 2 ? 2 : 1)
				case W_VESTHELM: g_iRebuyWeapons[id][armor] = 2
				case W_FLASHBANG: g_iRebuyWeapons[id][flash] =
clamp(g_iRebuyWeapons[id][flash] + 1, 0, 2)
				case W_HEGRENADE: g_iRebuyWeapons[id][he] = 1
				case W_SMOKEGRENADE: g_iRebuyWeapons[id][smoke] =
1
				case W_NVG: g_iRebuyWeapons[id][nvg] = 1
			}
		}
	}
}

public player_cmd_buy_main(id)
	return player_menu_buy(id, 0)

public player_cmd_buy_equipament(id)
	return player_menu_buy(id, 8)

public player_cmd_buyVGUI(id)
{
	message_begin(MSG_ONE, gMsg_BuyClose, _, id)
	message_end()

	return player_menu_buy(id, 0)
}

public player_menu_buy(id, iMenu)
```

```
{
	if(!g_bAlive[id])
		return PLUGIN_HANDLED

	if(!g_bBuyZone[id])
	{
		client_print(id, print_center, "%L", id, "BUY_NOTINZONE")
		return PLUGIN_HANDLED
	}

	static szMenu[1024]

	new iMoney = cs_get_user_money(id)

	switch(iMenu)
	{
		case 1:
		{
			formatex(szMenu, charsmax(szMenu),
				"\y%L: %L^n^n\d1. \%sGlock 18\R$%d^n\d2.
\%sUSP\R$%d^n\d3. \%sP228\R$%d^n\d4. \%sDesert Eagle\R$%d^n\d5.
\%sFiveseven\R$%d^n\d6. \%sDual Elites\R$%d^n^n\d0. \w%L",
				id, "BUYMENU_TITLE", id, "BUYMENU_PISTOLS",
				(iMoney >= g_iWeaponPrice[W_GLOCK18] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_GLOCK18],
				(iMoney >= g_iWeaponPrice[W_USP] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_USP],
				(iMoney >= g_iWeaponPrice[W_P228] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_P228],
				(iMoney >= g_iWeaponPrice[W_DEAGLE] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_DEAGLE],
				(iMoney >= g_iWeaponPrice[W_FIVESEVEN] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_FIVESEVEN],
				(iMoney >= g_iWeaponPrice[W_ELITE] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_ELITE],
				id, "EXIT"
			)
		}

		case 2:
		{
			formatex(szMenu, charsmax(szMenu),
				"\y%L: %L^n^n\d1. \%sM3 Super90\R$%d^n\d2.
\%sXM1014\R$%d^n^n\d0. \w%L",
				id, "BUYMENU_TITLE", id, "BUYMENU_SHOTGUNS",
```

```
                                (iMoney >= g_iWeaponPrice[W_M3] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_M3],
                                (iMoney >= g_iWeaponPrice[W_XM1014] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_XM1014],
                                id, "EXIT"
                        )
                }

                case 3:
                {
                        formatex(szMenu, charsmax(szMenu),
                                "\y%L: %L^n^n\d1. \%sTMP\R$%d^n\d2.
\%sMac-10\R$%d^n\d3. \%sMP5 Navy\R$%d^n\d4. \%sUMP-45\R$%d^n\d5.
\%sP90\R$%d^n^n\d0. \w%L",
                                id, "BUYMENU_TITLE", id, "BUYMENU_SMGS",
                                (iMoney >= g_iWeaponPrice[W_TMP] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_TMP],
                                (iMoney >= g_iWeaponPrice[W_MAC10] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_MAC10],
                                (iMoney >= g_iWeaponPrice[W_MP5NAVY] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_MP5NAVY],
                                (iMoney >= g_iWeaponPrice[W_UMP45] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_UMP45],
                                (iMoney >= g_iWeaponPrice[W_P90] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_P90],
                                id, "EXIT"
                        )
                }

                case 4:
                {
                        formatex(szMenu, charsmax(szMenu),
                                "\y%L: %L^n^n\d1. \%sGalil\R$%d^n\d2.
\%sFamas\R$%d^n\d3. \%sAK-47\R$%d^n\d4. \%sM4A1\R$%d^n\d5. \%sAug\R$%d^n\d6.
\%sSG552\R$%d^n^n\d0. \w%L",
                                id, "BUYMENU_TITLE", id, "BUYMENU_RIFLES",
                                (iMoney >= g_iWeaponPrice[W_GALIL] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_GALIL],
                                (iMoney >= g_iWeaponPrice[W_FAMAS] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_FAMAS],
                                (iMoney >= g_iWeaponPrice[W_AK47] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_AK47],
                                (iMoney >= g_iWeaponPrice[W_M4A1] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_M4A1],
                                (iMoney >= g_iWeaponPrice[W_AUG] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_AUG],
```

```
                                (iMoney >= g_iWeaponPrice[W_SG552] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_SG552],
                                id, "EXIT"
                        )
                }



                case 5:
                {

#if FEATURE_ADRENALINE == true

#if FEATURE_C4 == true

                        formatex(szMenu, charsmax(szMenu),
                                "\y%L: %L^n^n\d1. \%sM249 \w(\%s%d %L\w)\R\%s$%d^n\d3.
\%sSG550\w(\%s%d %L\w)\R\%s$%d^n\d3. \%sG3SG1 \w(\%s%d %L\w)\R\%s$%d^n\d4.
\%sScout \w(\%s%d %L\w)\R\%s$%d^n\d5. \%sAWP \w(\%s%d %L\w)\R\%s$%d^n\d6.
\%s%L \w(\%s%d %L\w)\R\%s$%d^n\d7. \%s%L \w(\%s%d %L\w)\R\%s$%d^n^n\d0.
\w%L",
                                id, "BUYMENU_TITLE", id, "BUYMENU_SPECIAL",
                                (iMoney >= g_iWeaponPrice[W_M249] && g_iAdrenaline[id] >=
g_iWeaponAdrenaline[W_M249] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
(g_iAdrenaline[id] >= g_iWeaponAdrenaline[W_M249] ? BUY_ITEM_AVAILABLE2 :
BUY_ITEM_DISABLED), g_iWeaponAdrenaline[W_M249], id, "ADRENALINE", (iMoney >=
g_iWeaponPrice[W_M249] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
g_iWeaponPrice[W_M249],
                                (iMoney >= g_iWeaponPrice[W_SG550] && g_iAdrenaline[id]
>= g_iWeaponAdrenaline[W_SG550] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
(g_iAdrenaline[id] >= g_iWeaponAdrenaline[W_SG550] ? BUY_ITEM_AVAILABLE2 :
BUY_ITEM_DISABLED), g_iWeaponAdrenaline[W_SG550], id, "ADRENALINE", (iMoney >=
g_iWeaponPrice[W_SG550] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
g_iWeaponPrice[W_SG550],
                                (iMoney >= g_iWeaponPrice[W_G3SG1] && g_iAdrenaline[id]
>= g_iWeaponAdrenaline[W_G3SG1] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
(g_iAdrenaline[id] >= g_iWeaponAdrenaline[W_G3SG1] ? BUY_ITEM_AVAILABLE2 :
BUY_ITEM_DISABLED), g_iWeaponAdrenaline[W_G3SG1], id, "ADRENALINE", (iMoney
>= g_iWeaponPrice[W_G3SG1] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
g_iWeaponPrice[W_G3SG1],
                                (iMoney >= g_iWeaponPrice[W_SCOUT] && g_iAdrenaline[id]
>= g_iWeaponAdrenaline[W_SCOUT] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
(g_iAdrenaline[id] >= g_iWeaponAdrenaline[W_SCOUT] ? BUY_ITEM_AVAILABLE2 :
BUY_ITEM_DISABLED), g_iWeaponAdrenaline[W_SCOUT], id, "ADRENALINE", (iMoney
>= g_iWeaponPrice[W_SCOUT] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
g_iWeaponPrice[W_SCOUT],
```

```
                                (iMoney >= g_iWeaponPrice[W_AWP] && g_iAdrenaline[id] >=
g_iWeaponAdrenaline[W_AWP] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
(g_iAdrenaline[id] >= g_iWeaponAdrenaline[W_AWP] ? BUY_ITEM_AVAILABLE2 :
BUY_ITEM_DISABLED), g_iWeaponAdrenaline[W_AWP], id, "ADRENALINE", (iMoney >=
g_iWeaponPrice[W_AWP] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
g_iWeaponPrice[W_AWP],
                                (iMoney >= g_iWeaponPrice[W_SHIELD] && g_iAdrenaline[id]
>= g_iWeaponAdrenaline[W_SHIELD] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
id, "BUYMENU_ITEM_SHIELD", (g_iAdrenaline[id] >= g_iWeaponAdrenaline[W_SHIELD] ?
BUY_ITEM_AVAILABLE2 : BUY_ITEM_DISABLED), g_iWeaponAdrenaline[W_SHIELD], id,
"ADRENALINE", (iMoney >= g_iWeaponPrice[W_SHIELD] ? BUY_ITEM_AVAILABLE :
BUY_ITEM_DISABLED), g_iWeaponPrice[W_SHIELD],
                                (iMoney >= g_iWeaponPrice[W_C4] && g_iAdrenaline[id] >=
g_iWeaponAdrenaline[W_C4] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), id,
"BUYMENU_ITEM_C4", (g_iAdrenaline[id] >= g_iWeaponAdrenaline[W_C4] ?
BUY_ITEM_AVAILABLE2 : BUY_ITEM_DISABLED), g_iWeaponAdrenaline[W_C4], id,
"ADRENALINE", (iMoney >= g_iWeaponPrice[W_C4] ? BUY_ITEM_AVAILABLE :
BUY_ITEM_DISABLED), g_iWeaponPrice[W_C4],
                                id, "EXIT"
                        )

#else // FEATURE_C4

                        formatex(szMenu, charsmax(szMenu),
                                "\y%L: %L^n^n\d1. \%sM249 \w(\%s%d %L\w)\R\%s$%d^n\d3.
\%sSG550 \w(\%s%d %L\w)\R\%s$%d^n\d3. \%sG3SG1 \w(\%s%d %L\w)\R\%s$%d^n\d4.
\%sScout \w(\%s%d %L\w)\R\%s$%d^n\d5. \%sAWP \w(\%s%d %L\w)\R\%s$%d^n\d6.
\%s%L \w(\%s%d %L\w)\R\%s$%d^n^n\d0. \w%L",
                                id, "BUYMENU_TITLE", id, "BUYMENU_SPECIAL",
                                (iMoney >= g_iWeaponPrice[W_M249] && g_iAdrenaline[id] >=
g_iWeaponAdrenaline[W_M249] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
(g_iAdrenaline[id] >= g_iWeaponAdrenaline[W_M249] ? BUY_ITEM_AVAILABLE2 :
BUY_ITEM_DISABLED), g_iWeaponAdrenaline[W_M249], id, "ADRENALINE", (iMoney >=
g_iWeaponPrice[W_M249] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
g_iWeaponPrice[W_M249],
                                (iMoney >= g_iWeaponPrice[W_SG550] && g_iAdrenaline[id]
>= g_iWeaponAdrenaline[W_SG550] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
(g_iAdrenaline[id] >= g_iWeaponAdrenaline[W_SG550] ? BUY_ITEM_AVAILABLE2 :
BUY_ITEM_DISABLED), g_iWeaponAdrenaline[W_SG550], id, "ADRENALINE", (iMoney >=
g_iWeaponPrice[W_SG550] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
g_iWeaponPrice[W_SG550],
                                (iMoney >= g_iWeaponPrice[W_G3SG1] && g_iAdrenaline[id]
>= g_iWeaponAdrenaline[W_G3SG1] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
(g_iAdrenaline[id] >= g_iWeaponAdrenaline[W_G3SG1] ? BUY_ITEM_AVAILABLE2 :
BUY_ITEM_DISABLED), g_iWeaponAdrenaline[W_G3SG1], id, "ADRENALINE", (iMoney
```

```
>= g_iWeaponPrice[W_G3SG1] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
g_iWeaponPrice[W_G3SG1],
                                        (iMoney >= g_iWeaponPrice[W_SCOUT] && g_iAdrenaline[id]
>= g_iWeaponAdrenaline[W_SCOUT] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
(g_iAdrenaline[id] >= g_iWeaponAdrenaline[W_SCOUT] ? BUY_ITEM_AVAILABLE2 :
BUY_ITEM_DISABLED), g_iWeaponAdrenaline[W_SCOUT], id, "ADRENALINE", (iMoney
>= g_iWeaponPrice[W_SCOUT] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
g_iWeaponPrice[W_SCOUT],
                                        (iMoney >= g_iWeaponPrice[W_AWP] && g_iAdrenaline[id] >=
g_iWeaponAdrenaline[W_AWP] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
(g_iAdrenaline[id] >= g_iWeaponAdrenaline[W_AWP] ? BUY_ITEM_AVAILABLE2 :
BUY_ITEM_DISABLED), g_iWeaponAdrenaline[W_AWP], id, "ADRENALINE", (iMoney >=
g_iWeaponPrice[W_AWP] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
g_iWeaponPrice[W_AWP],
                                        (iMoney >= g_iWeaponPrice[W_SHIELD] && g_iAdrenaline[id]
>= g_iWeaponAdrenaline[W_SHIELD] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
id, "BUYMENU_ITEM_SHIELD", (g_iAdrenaline[id] >= g_iWeaponAdrenaline[W_SHIELD] ?
BUY_ITEM_AVAILABLE2 : BUY_ITEM_DISABLED), g_iWeaponAdrenaline[W_SHIELD], id,
"ADRENALINE", (iMoney >= g_iWeaponPrice[W_SHIELD] ? BUY_ITEM_AVAILABLE :
BUY_ITEM_DISABLED), g_iWeaponPrice[W_SHIELD],
                                        id, "EXIT"
                                )

#endif // FEATURE_C4

#else // FEATURE_ADRENALINE

#if FEATURE_C4 == true

                                formatex(szMenu, charsmax(szMenu),
                                        "\y%L: %L^n^n\d1. \%sM249\R\%s$%d^n\d3.
\%sSG550\R\%s$%d^n\d3. \%sG3SG1\R\%s$%d^n\d4. \%sScout\R\%s$%d^n\d5.
\%sAWP\R\%s$%d^n\d6. \%s%L\R\%s$%d^n\d7. \%s%L\R\%s$%d^n^n\d0. \w%L",
                                        id, "BUYMENU_TITLE", id, "BUYMENU_SPECIAL",
                                        (iMoney >= g_iWeaponPrice[W_M249] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),(iMoney >= g_iWeaponPrice[W_M249] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_M249],
                                        (iMoney >= g_iWeaponPrice[W_SG550] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), (iMoney >= g_iWeaponPrice[W_SG550]
? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_SG550],
                                        (iMoney >= g_iWeaponPrice[W_G3SG1] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), (iMoney >= g_iWeaponPrice[W_G3SG1]
? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_G3SG1],
                                        (iMoney >= g_iWeaponPrice[W_SCOUT] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), (iMoney >=
```

```
g_iWeaponPrice[W_SCOUT] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
g_iWeaponPrice[W_SCOUT],
                                (iMoney >= g_iWeaponPrice[W_AWP] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), (iMoney >= g_iWeaponPrice[W_AWP] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_AWP],
                                (iMoney >= g_iWeaponPrice[W_SHIELD] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), id, "BUYMENU_ITEM_SHIELD",
(iMoney >= g_iWeaponPrice[W_SHIELD] ? BUY_ITEM_AVAILABLE :
BUY_ITEM_DISABLED), g_iWeaponPrice[W_SHIELD],
                                (iMoney >= g_iWeaponPrice[W_C4] ? BUY_ITEM_AVAILABLE
: BUY_ITEM_DISABLED), id, "BUYMENU_ITEM_C4", (iMoney >= g_iWeaponPrice[W_C4]
? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_C4],
                                id, "EXIT"
                        )

#else // FEATURE_C4

                        formatex(szMenu, charsmax(szMenu),
                                "\y%L: %L^n^n\d1. \%sM249\R\%s$%d^n\d3.
\%sSG550\R\%s$%d^n\d3. \%sG3SG1\R\%s$%d^n\d4. \%sScout\R\%s$%d^n\d5.
\%sAWP\R\%s$%d^n\d6. \%s%L\R\%s$%d^n^n\d0. \w%L",
                                id, "BUYMENU_TITLE", id, "BUYMENU_SPECIAL",
                                (iMoney >= g_iWeaponPrice[W_M249] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),(iMoney >= g_iWeaponPrice[W_M249] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_M249],
                                (iMoney >= g_iWeaponPrice[W_SG550] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), (iMoney >= g_iWeaponPrice[W_SG550]
? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_SG550],
                                (iMoney >= g_iWeaponPrice[W_G3SG1] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), (iMoney >= g_iWeaponPrice[W_G3SG1]
? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_G3SG1],
                                (iMoney >= g_iWeaponPrice[W_SCOUT] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), (iMoney >=
g_iWeaponPrice[W_SCOUT] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED),
g_iWeaponPrice[W_SCOUT],
                                (iMoney >= g_iWeaponPrice[W_AWP] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), (iMoney >= g_iWeaponPrice[W_AWP] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), g_iWeaponPrice[W_AWP],
                                (iMoney >= g_iWeaponPrice[W_SHIELD] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), id, "BUYMENU_ITEM_SHIELD",
(iMoney >= g_iWeaponPrice[W_SHIELD] ? BUY_ITEM_AVAILABLE :
BUY_ITEM_DISABLED), g_iWeaponPrice[W_SHIELD],
                                id, "EXIT"
                        )

#endif // FEATURE_C4
```

```
#endif // FEATURE_ADRENALINE

		}

		case 6:
		{
			formatex(szMenu, charsmax(szMenu),
				"\y%L: %L^n^n\d1. \%s%L\R$%d^n\d2. \%s%L\R$%d^n^n\d3.
\%s%L\R$%d^n\d4. \%s%L\R$%d^n\d5. \%s%L\R$%d^n^n\d7. \%s%L\R$%d^n^n\d0.
\w%L",
				id, "BUYMENU_TITLE", id, "BUYMENU_EQUIPAMENT",
				(iMoney >= g_iWeaponPrice[W_VEST] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), id, "BUYMENU_ITEM_VEST",
g_iWeaponPrice[W_VEST],
				(iMoney >= g_iWeaponPrice[W_VESTHELM] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), id, "BUYMENU_ITEM_VESTHELM",
g_iWeaponPrice[W_VESTHELM],
				(iMoney >= g_iWeaponPrice[W_FLASHBANG] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), id, "BUYMENU_ITEM_FLASHBANG",
g_iWeaponPrice[W_FLASHBANG],
				(iMoney >= g_iWeaponPrice[W_HEGRENADE] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), id, "BUYMENU_ITEM_HE",
g_iWeaponPrice[W_HEGRENADE],
				(iMoney >= g_iWeaponPrice[W_SMOKEGRENADE] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), id, "BUYMENU_ITEM_SMOKE",
g_iWeaponPrice[W_SMOKEGRENADE],
				(iMoney >= g_iWeaponPrice[W_NVG] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), id, "BUYMENU_ITEM_NVG",
g_iWeaponPrice[W_NVG],
				id, "EXIT"
			)
		}

		default:
		{

#if FEATURE_ADRENALINE == true

			formatex(szMenu, charsmax(szMenu),
				"\y%L^n^n\d1. \%s%L^n\d2. \%s%L^n\d3. \%s%L^n\d4.
\%s%L^n\d5. \%s%L^n^n\d6. \w%L\R$0^n^n\d8. \%s%L^n^n\d0. \w%L",
				id, "BUYMENU_TITLE",
				(iMoney >= g_iWeaponPrice[W_GLOCK18] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), id, "BUYMENU_PISTOLS",
```

```
                                (iMoney >= g_iWeaponPrice[W_M3] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), id, "BUYMENU_SHOTGUNS",
                                (iMoney >= g_iWeaponPrice[W_TMP] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), id, "BUYMENU_SMGS",
                                (iMoney >= g_iWeaponPrice[W_GALIL] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), id, "BUYMENU_RIFLES",
                                (iMoney >= g_iWeaponPrice[W_M249] && g_iAdrenaline[id] >=
g_iWeaponAdrenaline[W_M249] ? BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), id,
"BUYMENU_SPECIAL",
                                id, "BUYMENU_AMMO",
                                (iMoney >= g_iWeaponPrice[W_FLASHBANG] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), id, "BUYMENU_EQUIPAMENT",
                                id, "EXIT"
                        )

#else // FEATURE_ADRENALINE

                        formatex(szMenu, charsmax(szMenu),
                                "\y%L^n^n\d1. \%s%L^n\d2. \%s%L^n\d3. \%s%L^n\d4.
\%s%L^n\d5. \%s%L^n^n\d6. \w%L\R$0^n^n\d8. \%s%L^n^n\d0. \w%L",
                                id, "BUYMENU_TITLE",
                                (iMoney >= g_iWeaponPrice[W_GLOCK18] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), id, "BUYMENU_PISTOLS",
                                (iMoney >= g_iWeaponPrice[W_M3] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), id, "BUYMENU_SHOTGUNS",
                                (iMoney >= g_iWeaponPrice[W_TMP] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), id, "BUYMENU_SMGS",
                                (iMoney >= g_iWeaponPrice[W_GALIL] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), id, "BUYMENU_RIFLES",
                                (iMoney >= g_iWeaponPrice[W_M249] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), id, "BUYMENU_SPECIAL",
                                id, "BUYMENU_AMMO",
                                (iMoney >= g_iWeaponPrice[W_FLASHBANG] ?
BUY_ITEM_AVAILABLE : BUY_ITEM_DISABLED), id, "BUYMENU_EQUIPAMENT",
                                id, "EXIT"
                        )

#endif // FEATURE_ADRENALINE

                }
        }

        g_iMenu[id] = iMenu

        show_menu(id, MENU_KEYS_BUY, szMenu, -1, MENU_BUY)
```

```
	return PLUGIN_HANDLED
}

public player_key_buy(id, iKey)
{
	iKey += 1

	if(!g_bAlive[id] || iKey == 10)
		return PLUGIN_HANDLED

	if(!g_bBuyZone[id])
	{
		client_print(id, print_center, "%L", id, "BUY_NOTINZONE")
		return PLUGIN_HANDLED
	}

	switch(g_iMenu[id])
	{
		case 1:
		{
			switch(iKey)
			{
				case 1: player_buyWeapon(id, W_GLOCK18)
				case 2: player_buyWeapon(id, W_USP)
				case 3: player_buyWeapon(id, W_P228)
				case 4: player_buyWeapon(id, W_DEAGLE)
				case 5: player_buyWeapon(id, W_FIVESEVEN)
				case 6: player_buyWeapon(id, W_ELITE)
			}
		}

		case 2:
		{
			switch(iKey)
			{
				case 1: player_buyWeapon(id, W_M3)
				case 2: player_buyWeapon(id, W_XM1014)
			}
		}

		case 3:
		{
			switch(iKey)
			{
				case 1: player_buyWeapon(id, W_TMP)
				case 2: player_buyWeapon(id, W_MAC10)
```

```
                case 3: player_buyWeapon(id, W_MP5NAVY)
                case 4: player_buyWeapon(id, W_UMP45)
                case 5: player_buyWeapon(id, W_P90)
        }
}

case 4:
{
        switch(iKey)
        {
                case 1: player_buyWeapon(id, W_GALIL)
                case 2: player_buyWeapon(id, W_FAMAS)
                case 3: player_buyWeapon(id, W_AK47)
                case 4: player_buyWeapon(id, W_M4A1)
                case 5: player_buyWeapon(id, W_AUG)
                case 6: player_buyWeapon(id, W_SG552)
        }
}

case 5:
{
        switch(iKey)
        {
                case 1: player_buyWeapon(id, W_M249)
                case 2: player_buyWeapon(id, W_SG550)
                case 3: player_buyWeapon(id, W_G3SG1)
                case 4: player_buyWeapon(id, W_SCOUT)
                case 5: player_buyWeapon(id, W_AWP)
                case 6: player_buyWeapon(id, W_SHIELD)
                case 7: player_buyWeapon(id, W_C4)
        }
}

case 8:
{
        switch(iKey)
        {
                case 1: player_buyWeapon(id, W_VEST)
                case 2: player_buyWeapon(id, W_VESTHELM)
                case 3: player_buyWeapon(id, W_FLASHBANG)
                case 4: player_buyWeapon(id, W_HEGRENADE)
                case 5: player_buyWeapon(id, W_SMOKEGRENADE)
                case 7: player_buyWeapon(id, W_NVG)
        }
}
```

```
		default:
		{
			switch(iKey)
			{
				case 1,2,3,4,5,8: player_menu_buy(id, iKey)
				case 6,7: player_fillAmmo(id)
			}
		}
	}

	return PLUGIN_HANDLED
}

public player_cmd_buyWeapon(id)
{
	if(!g_bBuyZone[id])
	{
		client_print(id, print_center, "%L", id, "BUY_NOTINZONE")
		return PLUGIN_HANDLED
	}

	new szCmd[12]

	read_argv(0, szCmd, charsmax(szCmd))

	for(new w = W_P228; w <= W_NVG; w++)
	{
		for(new i = 0; i < 2; i++)
		{
			if(equali(g_szWeaponCommands[w][i], szCmd))
			{
				player_buyWeapon(id, w)
				return PLUGIN_HANDLED
			}
		}
	}

	return PLUGIN_HANDLED
}

public player_buyWeapon(id, iWeapon)
{
	if(!g_bAlive[id])
		return

	new CsArmorType:ArmorType
```

```
	new iArmor = cs_get_user_armor(id, ArmorType)

	new iMoney = cs_get_user_money(id)

	/* apply discount if you already have a kevlar and buying a kevlar+helmet */
	new iCost = g_iWeaponPrice[iWeapon] - (ArmorType == CS_ARMOR_KEVLAR &&
iWeapon == W_VESTHELM ? 650 : 0)

#if FEATURE_ADRENALINE == true

	new iCostAdrenaline = g_iWeaponAdrenaline[iWeapon]

#endif // FEATURE_ADRENALINE

	if(iCost > iMoney)
	{
		client_print(id, print_center, "%L", id, "BUY_NEEDMONEY", iCost)
		return
	}

#if FEATURE_ADRENALINE == true

	else if(!(iCostAdrenaline <= g_iAdrenaline[id]))
	{
		client_print(id, print_center, "%L", id, "BUY_NEEDADRENALINE",
iCostAdrenaline)
		return
	}

#endif // FEATURE_ADRENALINE

	switch(iWeapon)
	{

#if FEATURE_C4 == true

		case W_C4:
		{
			if(user_has_weapon(id, W_C4))
			{
				client_print(id, print_center, "%L", id, "BUY_HAVE_C4")
				return
			}

			player_giveC4(id)
		}
```

```
#endif // FEATURE_C4

		case W_NVG:
		{
			if(cs_get_user_nvg(id))
			{
				client_print(id, print_center, "%L", id, "BUY_HAVE_NVG")
				return
			}

			cs_set_user_nvg(id, 1)
		}

		case W_VEST:
		{
			if(iArmor >= 100)
			{
				client_print(id, print_center, "%L", id, "BUY_HAVE_KEVLAR")
				return
			}
		}

		case W_VESTHELM:
		{
			if(iArmor >= 100 && ArmorType == CS_ARMOR_VESTHELM)
			{
				client_print(id, print_center, "%L", id,
"BUY_HAVE_KEVLARHELM")
				return
			}
		}

		case W_FLASHBANG:
		{
			new iGrenades = cs_get_user_bpammo(id, W_FLASHBANG)

			if(iGrenades >= 2)
			{
				client_print(id, print_center, "%L", id,
"BUY_NOMORE_FLASH")
				return
			}

			new iCvar = get_pcvar_num(pCvar_ctf_nospam_flash)
			new Float:fGameTime = get_gametime()
```

```
                        if(g_fLastBuy[id][iGrenades] > fGameTime)
                        {
                                client_print(id, print_center, "%L", id, "BUY_DELAY_FLASH",
iCvar)
                                return
                        }

                        g_fLastBuy[id][iGrenades] = fGameTime + iCvar

                        if(iGrenades == 1)
                                g_fLastBuy[id][0] = g_fLastBuy[id][iGrenades]
                }

                case W_HEGRENADE:
                {
                        if(cs_get_user_bpammo(id, W_HEGRENADE) >= 1)
                        {
                                client_print(id, print_center, "%L", id, "BUY_NOMORE_HE")
                                return
                        }

                        new iCvar = get_pcvar_num(pCvar_ctf_nospam_he)
                        new Float:fGameTime = get_gametime()

                        if(g_fLastBuy[id][2] > fGameTime)
                        {
                                client_print(id, print_center, "%L", id, "BUY_DELAY_HE",
iCvar)
                                return
                        }

                        g_fLastBuy[id][2] = fGameTime + iCvar
                }

                case W_SMOKEGRENADE:
                {
                        if(cs_get_user_bpammo(id, W_SMOKEGRENADE) >= 1)
                        {
                                client_print(id, print_center, "%L", id,
"BUY_NOMORE_SMOKE")
                                return
                        }

                        new iCvar = get_pcvar_num(pCvar_ctf_nospam_smoke)
                        new Float:fGameTime = get_gametime()
```

```
                        if(g_fLastBuy[id][3] > fGameTime)
                        {
                                client_print(id, print_center, "%L", id, "BUY_DELAY_SMOKE",
iCvar)
                                return
                        }

                        g_fLastBuy[id][3] = fGameTime + iCvar
                }
        }

        if(1 <= g_iWeaponSlot[iWeapon] <= 2)
        {
                new iWeapons
                new iWeaponList[32]

                get_user_weapons(id, iWeaponList, iWeapons)

                if(cs_get_user_shield(id))
                        iWeaponList[iWeapons++] = W_SHIELD

                for(new w, i = 0; i < iWeapons; i++)
                {
                        w = iWeaponList[i]

                        if(1 <= g_iWeaponSlot[w] <= 2)
                        {
                                if(w == iWeapon)
                                {
                                        client_print(id, print_center, "%L", id,
"BUY_HAVE_WEAPON")
                                        return
                                }

                                if(iWeapon == W_SHIELD && w == W_ELITE)
                                        engclient_cmd(id, "drop",
g_szWeaponEntity[W_ELITE]) // drop the dual elites too if buying a shield

                                if(iWeapon == W_ELITE && w == W_SHIELD)
                                        engclient_cmd(id, "drop",
g_szWeaponEntity[W_SHIELD]) // drop the too shield if buying dual elites

                                if(g_iWeaponSlot[w] == g_iWeaponSlot[iWeapon])
                                {
```

```
                                        if(g_iWeaponSlot[w] == 2 && iWeaponList[iWeapons-1]
== W_SHIELD)
                                        {
                                                engclient_cmd(id, "drop",
g_szWeaponEntity[W_SHIELD]) // drop the shield

                                                new ent = find_ent_by_owner(g_iMaxPlayers,
g_szWeaponEntity[W_SHIELD], id)

                                                if(ent)
                                                {
                                                        entity_set_int(ent, EV_INT_flags,
FL_KILLME) // kill the shield
                                                        call_think(ent)
                                                }

                                                engclient_cmd(id, "drop", g_szWeaponEntity[w])
// drop the secondary

                                                give_item(id, g_szWeaponEntity[W_SHIELD]) //
give back the shield
                                        }
                                        else
                                                engclient_cmd(id, "drop", g_szWeaponEntity[w])
// drop weapon if it's of the same slot
                                }
                        }
                }
        }

        if(iWeapon != W_NVG && iWeapon != W_C4)
                give_item(id, g_szWeaponEntity[iWeapon])

        player_addRebuy(id, iWeapon)

        if(g_iWeaponPrice[iWeapon])
                cs_set_user_money(id, iMoney - iCost, 1)

#if FEATURE_ADRENALINE == true

        if(iCostAdrenaline)
        {
                g_iAdrenaline[id] -= iCostAdrenaline
                player_hudAdrenaline(id)
        }
```

```
#endif // FEATURE_ADRENALINE

	if(g_iBPAmmo[iWeapon])
		cs_set_user_bpammo(id, iWeapon, g_iBPAmmo[iWeapon])
}

public player_fillAmmo(id)
{
	if(!g_bAlive[id])
		return PLUGIN_HANDLED

	if(!g_bBuyZone[id])
	{
		client_print(id, print_center, "%L", id, "BUY_NOTINZONE")
		return PLUGIN_HANDLED
	}

	if(player_getAmmo(id))
	{
		emit_sound(id, CHAN_ITEM, SND_GETAMMO, VOL_NORM, ATTN_NORM,
0, PITCH_NORM)
		client_print(id, print_center, "%L", id, "BUY_FULLAMMO")
	}

	return PLUGIN_HANDLED
}

#endif // FEATURE_BUY









#if FEATURE_C4 == true

public c4_used(msgid, dest, id)
{
	if(g_iTeam[id])
		player_updateSpeed(id)
```

```
	if(get_msg_arg_int(1) == 0)
		g_bDefuse[id] = false

	return PLUGIN_HANDLED
}

public c4_planted()
{
	new szLogUser[80], szName[32]

	read_logargv(0, szLogUser, charsmax(szLogUser))
	parse_loguser(szLogUser, szName, charsmax(szName))

	new id = get_user_index(szName)
	new ent = g_iMaxPlayers
	new Float:fAngles[3]
	new szFormat[40]

	entity_get_vector(id, EV_VEC_angles, fAngles)

	fAngles[pitch] = 0.0
	fAngles[yaw] += 90.0
	fAngles[roll] = 0.0

	new iC4Timer = get_pcvar_num(pCvar_mp_c4timer)

	client_print(id, print_center, "%L", id, "C4_ARMED", iC4Timer)

	formatex(szFormat, charsmax(szFormat), "%L", LANG_PLAYER,
"C4_ARMED_RADIO", iC4Timer)

	for(new i = 1; i <= g_iMaxPlayers; i++)
	{
		if(g_iTeam[i] == g_iTeam[id] && !g_bBot[id])
		{
			/* fully fake hookable radio message and event */

			emessage_begin(MSG_ONE, gMsg_TextMsg, _, i)
			ewrite_byte(3)
			ewrite_string("#Game_radio")
			ewrite_string(szName)
			ewrite_string(szFormat)
			emessage_end()

			emessage_begin(MSG_ONE, gMsg_SendAudio, _, i)
			ewrite_byte(id)
```

```
                ewrite_string("%!MRAD_BLOW")
                ewrite_short(100)
                emessage_end()
            }
        }

        while((ent = find_ent_by_owner(ent, GRENADE, id)))
        {
            if(get_pdata_int(ent, 96) & (1<<8))
            {
                entity_set_int(ent, EV_INT_solid, SOLID_NOT)
                entity_set_int(ent, EV_INT_movetype, MOVETYPE_TOSS)
                entity_set_float(ent, EV_FL_gravity, 1.0)
                entity_set_vector(ent, EV_VEC_angles, fAngles)

                return
            }
        }
}

public c4_defuse(ent, id, activator, iType, Float:fValue)
{
        if(g_bAlive[id] && get_pdata_int(ent, 96) & (1<<8))
        {
            new iOwner = entity_get_edict(ent, EV_ENT_owner)

            if(id != iOwner && 1 <= iOwner <= g_iMaxPlayers && g_iTeam[id] ==
g_iTeam[iOwner])
            {
                client_print(id, print_center, "%L", id, "C4_NODEFUSE")
                client_cmd(id, "-use")

                return HAM_SUPERCEDE
            }

            if(g_iTeam[id] == TEAM_RED)
            {
                set_pdata_int(id, 114, TEAM_BLUE, 5)

                ExecuteHam(Ham_Use, ent, id, activator, iType, fValue)

                set_pdata_int(id, 114, TEAM_RED, 5)
            }

            if(!g_bDefuse[id])
            {
```

```
                        client_print(id, print_center, "%L", id, "C4_DEFUSING",
C4_DEFUSETIME)

                        message_begin(MSG_ONE_UNRELIABLE, gMsg_BarTime, _, id)
                        write_short(C4_DEFUSETIME)
                        message_end()

                        set_pdata_float(ent, 99, get_gametime() + C4_DEFUSETIME, 5)

                        g_bDefuse[id] = true
                }
        }

        return HAM_IGNORED
}

public c4_defused()
{
        new szLogUser[80], szName[32]

        read_logargv(0, szLogUser, charsmax(szLogUser))
        parse_loguser(szLogUser, szName, charsmax(szName))

        new id = get_user_index(szName)

        if(!g_bAlive[id])
                return

        g_bDefuse[id] = false

        player_giveC4(id)
        client_print(id, print_center, "%L", id, "C4_DEFUSED")
}

public c4_pickup(ent, id)
{
        if(g_bAlive[id] && is_valid_ent(ent) && (entity_get_int(ent, EV_INT_flags) &
FL_ONGROUND))
        {
                static szModel[32]

                entity_get_string(ent, EV_SZ_model, szModel, charsmax(szModel))

                if(equal(szModel, "models/w_backpack.mdl"))
                {
                        if(user_has_weapon(id, W_C4))
```

```
                                return PLUGIN_HANDLED

                        player_giveC4(id)

                        weapon_remove(ent)

                        return PLUGIN_HANDLED
                }
        }

        return PLUGIN_CONTINUE
}

#endif // FEATURE_C4 == true









public weapon_spawn(ent)
{
        if(!is_valid_ent(ent))
                return

        new Float:fWeaponStay = get_pcvar_float(pCvar_ctf_weaponstay)

        if(fWeaponStay > 0)
        {
                task_remove(ent)
                task_set(fWeaponStay, "weapon_startFade", ent)
        }
}

public weapon_startFade(ent)
{
        if(!is_valid_ent(ent))
                return

        new szClass[32]
```

```
	entity_get_string(ent, EV_SZ_classname, szClass, charsmax(szClass))

	if(!equal(szClass, WEAPONBOX) && !equal(szClass, ITEM_CLASSNAME))
		return

	entity_set_int(ent, EV_INT_movetype, MOVETYPE_FLY)
	entity_set_int(ent, EV_INT_rendermode, kRenderTransAlpha)

	if(get_pcvar_num(pCvar_ctf_glows))
		entity_set_int(ent, EV_INT_renderfx, kRenderFxGlowShell)

	entity_set_float(ent, EV_FL_renderamt, 255.0)
	entity_set_vector(ent, EV_VEC_rendercolor, Float:{255.0, 255.0, 0.0})
	entity_set_vector(ent, EV_VEC_velocity, Float:{0.0, 0.0, 20.0})

	weapon_fadeOut(ent, 255.0)
}

public weapon_fadeOut(ent, Float:fStart)
{
	if(!is_valid_ent(ent))
	{
		task_remove(ent)
		return
	}

	static Float:fFadeAmount[4096]

	if(fStart)
	{
		task_remove(ent)
		fFadeAmount[ent] = fStart
	}

	fFadeAmount[ent] -= 25.5

	if(fFadeAmount[ent] > 0.0)
	{
		entity_set_float(ent, EV_FL_renderamt, fFadeAmount[ent])

		task_set(0.1, "weapon_fadeOut", ent)
	}
	else
	{
		new szClass[32]
```

```
		entity_get_string(ent, EV_SZ_classname, szClass, charsmax(szClass))

		if(equal(szClass, WEAPONBOX))
			weapon_remove(ent)
		else
			entity_remove(ent)
	}
}
```

```
public item_touch(ent, id)
{
	if(g_bAlive[id] && is_valid_ent(ent) && entity_get_int(ent, EV_INT_flags) &
FL_ONGROUND)
	{
		new iType = entity_get_int(ent, EV_INT_iuser2)

		switch(iType)
		{
			case ITEM_AMMO:
			{
				if(!player_getAmmo(id))
					return PLUGIN_HANDLED

				client_print(id, print_center, "%L", id, "PICKED_AMMO")

				emit_sound(id, CHAN_ITEM, SND_GETAMMO, VOL_NORM,
ATTN_NORM, 0, PITCH_NORM)
			}

			case ITEM_MEDKIT:
			{
				new iHealth = get_user_health(id)

				if(iHealth >= g_iMaxHealth[id])
					return PLUGIN_HANDLED

				set_user_health(id, clamp(iHealth + ITEM_MEDKIT_GIVE, 0,
100))
```

```
                                client_print(id, print_center, "%L", id, "PICKED_HEALTH",
ITEM_MEDKIT_GIVE)

                                emit_sound(id, CHAN_ITEM, SND_GETMEDKIT,
VOL_NORM, ATTN_NORM, 0, 110)
                        }

#if FEATURE_ADRENALINE == true

                        case ITEM_ADRENALINE:
                        {
                                if(g_iAdrenaline[id] >= 100)
                                        return PLUGIN_HANDLED

                                g_iAdrenaline[id] = clamp(g_iAdrenaline[id] +
ITEM_ADRENALINE_GIVE, 0, 100)

                                player_hudAdrenaline(id)

                                client_print(id, print_center, "%L", id,
"PICKED_ADRENALINE", ITEM_ADRENALINE_GIVE)

                                emit_sound(id, CHAN_ITEM, SND_GETADRENALINE,
VOL_NORM, ATTN_NORM, 0, 140)
                        }

#endif // FEATURE_ADRENALINE == true
                }

                task_remove(ent)
                entity_remove(ent)
        }

        return PLUGIN_CONTINUE
}
```

```
public event_restartGame()
	g_bRestarting = true

public event_roundStart()
{
	new ent = -1

	while((ent = find_ent_by_class(ent, WEAPONBOX)) > 0)
	{
		task_remove(ent)
		weapon_remove(ent)
	}

	ent = -1

	while((ent = find_ent_by_class(ent, ITEM_CLASSNAME)) > 0)
	{
		task_remove(ent)
		entity_remove(ent)
	}

	for(new id = 1; id < g_iMaxPlayers; id++)
	{
		if(!g_bAlive[id])
			continue

		g_bDefuse[id] = false
		g_bFreeLook[id] = false
		g_fLastBuy[id] = Float:{0.0, 0.0, 0.0, 0.0}

		task_remove(id - TASK_EQUIPAMENT)
		task_remove(id - TASK_TEAMBALANCE)
		task_remove(id - TASK_DEFUSE)

		if(g_bRestarting)
		{
			task_remove(id)
			task_remove(id - TASK_ADRENALINE)
```

```
                        g_bRestarted[id] = true
                        g_iAdrenaline[id] = 0
                        g_iAdrenalineUse[id] = 0
                }

                player_updateSpeed(id)
        }

        for(new iFlagTeam = TEAM_RED; iFlagTeam <= TEAM_BLUE; iFlagTeam++)
        {
                flag_sendHome(iFlagTeam)

                task_remove(g_iFlagEntity[iFlagTeam])

                log_message("%s, %s flag returned back to base.", (g_bRestarting ? "Game
restarted" : "New round started"), g_szTeamName[iFlagTeam])
        }

        if(g_bRestarting)
        {
                g_iScore = {0,0,0}
                g_bRestarting = false
        }
}
```

```
public msg_block()
        return PLUGIN_HANDLED

#if FEATURE_C4 == true
```

```
public msg_sendAudio()
{
	new szAudio[14]

	get_msg_arg_string(2, szAudio, charsmax(szAudio))

	return equal(szAudio, "%!MRAD_BOMB", 11) ? PLUGIN_HANDLED :
PLUGIN_CONTINUE
}

#endif // FEATURE_C4 == true

public msg_screenFade(msgid, dest, id)
	return (g_bProtected[id] && g_bAlive[id] && get_msg_arg_int(4) == 255 &&
get_msg_arg_int(5) == 255 && get_msg_arg_int(6) == 255 && get_msg_arg_int(7) > 199 ?
PLUGIN_HANDLED : PLUGIN_CONTINUE)

public msg_scoreAttrib()
	return (get_msg_arg_int(2) & (1<<1) ? PLUGIN_HANDLED : PLUGIN_CONTINUE)

public msg_teamScore()
{
	new szTeam[2]

	get_msg_arg_string(1, szTeam, 1)

	switch(szTeam[0])
	{
		case 'T': set_msg_arg_int(2, ARG_SHORT, g_iScore[TEAM_RED])
		case 'C': set_msg_arg_int(2, ARG_SHORT, g_iScore[TEAM_BLUE])
	}
}

public msg_roundTime()
	set_msg_arg_int(1, ARG_SHORT, get_timeleft())

public msg_sayText(msgid, dest, id)
{
	new szString[32]

	get_msg_arg_string(2, szString, charsmax(szString))

	new iTeam = (szString[14] == 'T' ? TEAM_RED : (szString[14] == 'C' ? TEAM_BLUE :
TEAM_SPEC))
	new bool:bDead = (szString[16] == 'D' || szString[17] == 'D')
```

```
	if(TEAM_RED <= iTeam <= TEAM_BLUE && equali(szString, "#Cstrike_Chat_", 14))
	{
		formatex(szString, charsmax(szString), "^x01%s(%L)^x03 %%s1^x01 :
%%s2", (bDead ? "*DEAD* " : NULL), id, g_szMLFlagTeam[iTeam])
		set_msg_arg_string(2, szString)
	}
}

public msg_textMsg(msgid, dest, id)
{
	static szMsg[48]

	get_msg_arg_string(2, szMsg, charsmax(szMsg))

	if(equal(szMsg, "#Spec_Mode", 10) && !get_pcvar_num(pCvar_mp_fadetoblack) &&
(get_pcvar_num(pCvar_mp_forcecamera) || get_pcvar_num(pCvar_mp_forcechasecam)))
	{
		if(TEAM_RED <= g_iTeam[id] <= TEAM_BLUE && szMsg[10] == '3')
		{
			if(!g_bFreeLook[id])
			{
				player_screenFade(id, {0,0,0,255}, 0.25, 9999.0, FADE_IN,
true)
				g_bFreeLook[id] = true
			}

			formatex(szMsg, charsmax(szMsg), "%L", id,
"DEATH_NOFREELOOK")

			set_msg_arg_string(2, szMsg)
		}
		else if(g_bFreeLook[id])
		{
			player_screenFade(id, {0,0,0,255}, 0.25, 0.0, FADE_OUT, true)
			g_bFreeLook[id] = false
		}
	}
	else if(equal(szMsg, "#Terrorists_Win") || equal(szMsg, "#CTs_Win"))
	{
		static szString[32]

		formatex(szString, charsmax(szString), "%L", LANG_PLAYER,
"STARTING_NEWROUND")

		set_msg_arg_string(2, szString)
	}
```

```
	else if(equal(szMsg, "#Only_1", 7))
	{
		formatex(szMsg, charsmax(szMsg), "%L", id, "DEATH_ONLY1CHANGE")

		set_msg_arg_string(2, szMsg)
	}

#if FEATURE_C4 == true

	else if(equal(szMsg, "#Defusing", 9) || equal(szMsg, "#Got_bomb", 9) || equal(szMsg,
"#Game_bomb", 10) || equal(szMsg, "#Bomb", 5) || equal(szMsg, "#Target", 7))
		return PLUGIN_HANDLED

#endif // FEATURE_C4 == true

	return PLUGIN_CONTINUE
}









player_award(id, iMoney, iFrags, iAdrenaline, szText[], any:...)
{
#if FEATURE_ADRENALINE == false

	iAdrenaline = 0

#endif // FEATURE_ADRENALINE

	if(!g_iTeam[id] || (!iMoney && !iFrags && !iAdrenaline))
		return

	new szMsg[48]
	new szMoney[24]
	new szFrags[48]
```

```
	new szFormat[192]
	new szAdrenaline[48]

	if(iMoney != 0)
	{
		cs_set_user_money(id, clamp(cs_get_user_money(id) + iMoney, 0, 16000),
1)

		formatex(szMoney, charsmax(szMoney), "%s%d$", iMoney > 0 ? "+" : NULL,
iMoney)
	}

	if(iFrags != 0)
	{
		player_setScore(id, iFrags, 0)

		formatex(szFrags, charsmax(szFrags), "%s%d %L", iFrags > 0 ? "+" : NULL,
iFrags, id, (iFrags > 1 ? "FRAGS" : "FRAG"))
	}

#if FEATURE_ADRENALINE == true

	if(iAdrenaline != 0)
	{
		g_iAdrenaline[id] = clamp(g_iAdrenaline[id] + iAdrenaline, 0, 100)

		player_hudAdrenaline(id)

		formatex(szAdrenaline, charsmax(szAdrenaline), "%s%d %L", iAdrenaline > 0
? "+" : NULL, iAdrenaline, id, "ADRENALINE")
	}

#endif // FEATURE_ADRENALINE == true

	vformat(szMsg, charsmax(szMsg), szText, 6)
	formatex(szFormat, charsmax(szFormat), "%s%s%s%s%s %s", szMoney,
(szMoney[0] && (szFrags[0] || szAdrenaline[0]) ? ", " : NULL), szFrags, (szFrags[0] &&
szAdrenaline[0] ? ", " : NULL), szAdrenaline, szMsg)

	client_print(id, print_console, "%s%L: %s", CONSOLE_PREFIX, id, "REWARD",
szFormat)
	client_print(id, print_center, szFormat)
}

#if FEATURE_ADRENALINE == true
```

```
player_hudAdrenaline(id)
{
	set_hudmessage(HUD_ADRENALINE)

	if(g_iAdrenalineUse[id])
		show_hudmessage(id, "%L", id, "HUD_ADRENALINECOMBO", id,
g_szAdrenalineUseML[g_iAdrenalineUse[id]], g_iAdrenaline[id], 100)

	else if(g_iAdrenaline[id] >= 100)
		show_hudmessage(id, "%L", id, "HUD_ADRENALINEFULL")

	else
		show_hudmessage(id, "%L", id, "HUD_ADRENALINE", g_iAdrenaline[id],
100)
}

#endif // FEATURE_ADRENALINE == true

player_print(id, iSender, szMsg[], any:...)
{
	if(g_bBot[id] || (id && !g_iTeam[id]))
		return PLUGIN_HANDLED

	new szFormat[192]

	vformat(szFormat, charsmax(szFormat), szMsg, 4)
	format(szFormat, charsmax(szFormat), "%s%s", CHAT_PREFIX, szFormat)

	if(id)
		message_begin(MSG_ONE, gMsg_SayText, _, id)
	else
		message_begin(MSG_ALL, gMsg_SayText)

	write_byte(iSender)
	write_string(szFormat)
	message_end()

	return PLUGIN_HANDLED
}

bool:player_getAmmo(id)
{
	if(!g_bAlive[id])
		return false

	new iWeapons
```

```
	new iWeaponList[32]
	new bool:bGotAmmo = false

	get_user_weapons(id, iWeaponList, iWeapons)

	for(new iAmmo, iClip, ent, w, i = 0; i < iWeapons; i++)
	{
		w = iWeaponList[i]

		if(g_iBPAmmo[w])
		{
			ent = find_ent_by_owner(g_iMaxPlayers, g_szWeaponEntity[w], id)

			iAmmo = cs_get_user_bpammo(id, w)
			iClip = (ent ? cs_get_weapon_ammo(ent) : 0)

			if((iAmmo + iClip) < (g_iBPAmmo[w] + g_iClip[w]))
			{
				cs_set_user_bpammo(id, w, g_iBPAmmo[w] + (g_iClip[w] -
iClip))
				bGotAmmo = true
			}
		}
	}

	return bGotAmmo
}

player_setScore(id, iAddFrags, iAddDeaths)
{
	new iFrags = get_user_frags(id)
	new iDeaths = cs_get_user_deaths(id)

	if(iAddFrags != 0)
	{
		iFrags += iAddFrags

		set_user_frags(id, iFrags)
	}

	if(iAddDeaths != 0)
	{
		iDeaths += iAddDeaths

		cs_set_user_deaths(id, iDeaths)
	}
```

```
	message_begin(MSG_BROADCAST, gMsg_ScoreInfo)
	write_byte(id)
	write_short(iFrags)
	write_short(iDeaths)
	write_short(0)
	write_short(g_iTeam[id])
	message_end()
}

player_spawnItem(id)
{
#if FEATURE_ADRENALINE == true
	if(!ITEM_DROP_AMMO && !ITEM_DROP_MEDKIT &&
!ITEM_DROP_ADRENALINE)
		return
#else
	if(!ITEM_DROP_AMMO && !ITEM_DROP_MEDKIT)
		return
#endif

	if(random_num(1, 100) > get_pcvar_float(pCvar_ctf_itempercent))
		return

	new ent = entity_create(INFO_TARGET)

	if(!ent)
		return

	new iType
	new Float:fOrigin[3]
	new Float:fAngles[3]
	new Float:fVelocity[3]

	entity_get_vector(id, EV_VEC_origin, fOrigin)

	fVelocity[x] = random_float(-100.0, 100.0)
	fVelocity[y] = random_float(-100.0, 100.0)
	fVelocity[z] = 50.0

	fAngles[yaw] = random_float(0.0, 360.0)

#if FEATURE_ADRENALINE == true
	while((iType = random(3)))
#else
	while((iType = random(2)))
```

```
#endif
	{
		switch(iType)
		{
			case ITEM_AMMO:
			{
				if(ITEM_DROP_AMMO)
				{
					entity_set_model(ent, ITEM_MODEL_AMMO)
					break
				}
			}

			case ITEM_MEDKIT:
			{
				if(ITEM_DROP_MEDKIT)
				{
					entity_set_model(ent, ITEM_MODEL_MEDKIT)
					break
				}
			}

#if FEATURE_ADRENALINE == true
			case ITEM_ADRENALINE:
			{
				if(ITEM_DROP_ADRENALINE)
				{
					entity_set_model(ent, ITEM_MODEL_ADRENALINE)
					entity_set_int(ent, EV_INT_skin, 2)
					break
				}
			}
#endif // FEATURE_ADRENALINE == true
		}
	}

	entity_set_string(ent, EV_SZ_classname, ITEM_CLASSNAME)
	entity_spawn(ent)
	entity_set_size(ent, ITEM_HULL_MIN, ITEM_HULL_MAX)
	entity_set_origin(ent, fOrigin)
	entity_set_vector(ent, EV_VEC_angles, fAngles)
	entity_set_vector(ent, EV_VEC_velocity, fVelocity)
	entity_set_int(ent, EV_INT_movetype, MOVETYPE_TOSS)
	entity_set_int(ent, EV_INT_solid, SOLID_TRIGGER)
	entity_set_int(ent, EV_INT_iuser2, iType)
```

```
	task_remove(ent)
	task_set(get_pcvar_float(pCvar_ctf_weaponstay), "weapon_startFade", ent)
}

#if FEATURE_C4 == true

player_giveC4(id)
{
	give_item(id, g_szWeaponEntity[W_C4])

	cs_set_user_plant(id, 1, 1)
}

#endif // FEATURE_C4

player_healingEffect(id)
{
	new iOrigin[3]

	get_user_origin(id, iOrigin)

	message_begin(MSG_PVS, SVC_TEMPENTITY, iOrigin)
	write_byte(TE_PROJECTILE)
	write_coord(iOrigin[x] + random_num(-10, 10))
	write_coord(iOrigin[y] + random_num(-10, 10))
	write_coord(iOrigin[z] + random_num(0, 30))
	write_coord(0)
	write_coord(0)
	write_coord(15)
	write_short(gSpr_regeneration)
	write_byte(1)
	write_byte(id)
	message_end()
}

player_updateRender(id, Float:fDamage = 0.0)
{
	new bool:bGlows = (get_pcvar_num(pCvar_ctf_glows) == 1)
	new iTeam = g_iTeam[id]
	new iMode = kRenderNormal
	new iEffect = kRenderFxNone
	new iAmount = 0
	new iColor[3] = {0,0,0}

	if(g_bProtected[id])
	{
```

```
			if(bGlows)
				iEffect = kRenderFxGlowShell

			iAmount = 200

			iColor[0] = (iTeam == TEAM_RED ? 155 : 0)
			iColor[1] = (fDamage > 0.0 ? 100 - clamp(floatround(fDamage), 0, 100) : 0)
			iColor[2] = (iTeam == TEAM_BLUE ? 155 : 0)
		}

#if FEATURE_ADRENALINE == true
		switch(g_iAdrenalineUse[id])
		{
			case ADRENALINE_BERSERK:
			{
				if(bGlows)
					iEffect = kRenderFxGlowShell

				iAmount = 160
				iColor = {55, 0, 55}
			}

			case ADRENALINE_INVISIBILITY:
			{
				iMode = kRenderTransAlpha

				if(bGlows)
					iEffect = kRenderFxGlowShell

				iAmount = 10
				iColor = {15, 15, 15}
			}
		}
#endif // FEATURE_ADRENALINE == true

		if(player_hasFlag(id))
		{
			if(iMode != kRenderTransAlpha)
				iMode = kRenderNormal

			if(bGlows)
				iEffect = kRenderFxGlowShell

			iColor[0] = (iTeam == TEAM_RED ? (iColor[0] > 0 ? 200 : 155) : 0)
			iColor[1] = (iAmount == 160 ? 55 : 0)
			iColor[2] = (iTeam == TEAM_BLUE ? (iColor[2] > 0 ? 200 : 155) : 0)
```

```
                iAmount = (iAmount == 160 ? 50 : (iAmount == 10 ? 20 : 30))
        }

        set_user_rendering(id, iEffect, iColor[0], iColor[1], iColor[2], iMode, iAmount)
}

player_updateSpeed(id)
{
        new Float:fSpeed = 1.0

        if(player_hasFlag(id))
                fSpeed *= SPEED_FLAG

#if FEATURE_ADRENALINE == true

        if(g_iAdrenalineUse[id] == ADRENALINE_SPEED)
                fSpeed *= SPEED_ADRENALINE

#endif // FEATURE_ADRENALINE

        set_user_maxspeed(id, g_fWeaponSpeed[id] * fSpeed)
}

player_screenFade(id, iColor[4] = {0,0,0,0}, Float:fEffect = 0.0, Float:fHold = 0.0, iFlags =
FADE_OUT, bool:bReliable = false)
{
        if(id && !g_iTeam[id])
                return

        static iType

        if(1 <= id <= g_iMaxPlayers)
                iType = (bReliable ? MSG_ONE : MSG_ONE_UNRELIABLE)
        else
                iType = (bReliable ? MSG_ALL : MSG_BROADCAST)

        message_begin(iType, gMsg_ScreenFade, _, id)
        write_short(clamp(floatround(fEffect * (1<<12)), 0, 0xFFFF))
        write_short(clamp(floatround(fHold * (1<<12)), 0, 0xFFFF))
        write_short(iFlags)
        write_byte(iColor[0])
        write_byte(iColor[1])
        write_byte(iColor[2])
        write_byte(iColor[3])
        message_end()
```

```
}

game_announce(iEvent, iFlagTeam, szName[])
{
	new iColor = iFlagTeam
	new szText[64]

	switch(iEvent)
	{
		case EVENT_TAKEN:
		{
			iColor = get_opTeam(iFlagTeam)
			formatex(szText, charsmax(szText), "%L", LANG_PLAYER,
"ANNOUNCE_FLAGTAKEN", szName, LANG_PLAYER, g_szMLFlagTeam[iFlagTeam])
		}

		case EVENT_DROPPED: formatex(szText, charsmax(szText), "%L",
LANG_PLAYER, "ANNOUNCE_FLAGDROPPED", szName, LANG_PLAYER,
g_szMLFlagTeam[iFlagTeam])

		case EVENT_RETURNED:
		{
			if(strlen(szName) != 0)
				formatex(szText, charsmax(szText), "%L", LANG_PLAYER,
"ANNOUNCE_FLAGRETURNED", szName, LANG_PLAYER, g_szMLFlagTeam[iFlagTeam])
			else
				formatex(szText, charsmax(szText), "%L", LANG_PLAYER,
"ANNOUNCE_FLAGAUTORETURNED", LANG_PLAYER, g_szMLFlagTeam[iFlagTeam])
		}

		case EVENT_SCORE: formatex(szText, charsmax(szText), "%L",
LANG_PLAYER, "ANNOUNCE_FLAGCAPTURED", szName, LANG_PLAYER,
g_szMLFlagTeam[get_opTeam(iFlagTeam)])
	}

	set_hudmessage(iColor == TEAM_RED ? 255 : 0, 0, iColor == TEAM_BLUE ? 255 :
0, HUD_ANNOUNCE)
	show_hudmessage(0, szText)

	client_print(0, print_console, "%s%L: %s", CONSOLE_PREFIX, LANG_PLAYER,
"ANNOUNCEMENT", szText)

	if(get_pcvar_num(pCvar_ctf_sound[iEvent]))
		client_cmd(0, "mp3 play ^"sound/ctf/%s.mp3^"",
g_szSounds[iEvent][iFlagTeam])
```