RealRTCW Guide for Mappers and Modders
Last edited 13/08/2023

New weapons:
weapon_dagger
weapon_delisle
weapon_delislescope
weapon_mp34
weapon_tt33
weapon_p38
weapon_ppsh
weapon_mosin
weapon_g43
weapon_m1garand
weapon_m7
weapon_bar
weapon_mp44
weapon_mg42m
weapon_browning
weapon_m97
weapon_m30
weapon_welrod
weapon_holycross
weapon_revolver
weapon_grenadesmoke
weapon_poisongas

New ammo items:
ammo_poison_gas
ammo_m7
ammo_holyspirit
ammo_ttammo
ammo_ttammo_l
ammo_mosina
ammo_barammo
ammo_barammo_l
ammo_44ammo -(mp44 ammo)
ammo_44ammo_l
ammo_m97ammo
ammo_revolver

New holdable items:
holdable_adrenaline
holdable_bandages

Re-enabled Q3 powerups:
item_quad – quad damage
item_haste – speed powerup
item_enviro – protective suit
item_invis – invisibility

New enemies types:
ai_dog
ai_priest
ai_xshepherd

New radiant entities:

Health and Ammo cabinets.
trigger_ammo
Any entity that touches this will get additional ammo a specified rate up to a specified maximum.

Keys:

"ammorate" rate of ammo clips per second. default 1. (whole number only)
"ammototal" the maximum clips of ammo this trigger can add. if <= 0, it's unlimited.
default 0 (whole numbers only)
"target" cabinet that this entity is linked to

trigger_heal
Any entity that touches this will be healed at a specified rate up to a specified maximum.

Keys:

"healrate" rat of healing per second, default 5 (whole numbers only)
"healtotal" the maximum of healing this trigger can do. if <= 0, it's unlimited.
default 0 (whole numbers only)
"target" cabinet that this entity is linked to

misc_cabinet_supply
Ammo cabinet. Must be targeted by trigger_ammo.

misc_cabinet_health
Health cabinet. Must be targeted by trigger_heal.

New vehicle pathing system.
info_train_spline_control
Train spline.

info_train_spline_main
Train spline.

Atmospheric effects
Includes rain and snow. To add atmospheric effect into your map add “atmosphere” key parameter to your worldspawn entity.

As a value for this key parameter you can use these presets:
Strong snow - used on norway:
«T=SNOW,B=5 10,C=0.5,G=0.3 2,BV=50 50,GV=30 80,W=1 2,D=15000»
Weaker snow - used on escape1:
«T=SNOW,B=5 10,C=0.5,G=0.3 2,BV=20 30,GV=25 40,W=3 5,D=5000»
Strong rain - used on dark:
«T=RAIN,B=5 10,C=0.5,G=0.5 2,BV=50 50,GV=200 200,W=1 2,D=5000»

Atmoshperic effects CVARs:
cg_atmoshpericeffects — disable/enable atmospheric effects
cg_forceatmosphericeffects — no force (0), rain (1), snow (2). Requires vid_restart.
cg_lowAtmosphericEffects — high (0), medium (1), 2 (disabled)

Automatic AI attributes system — easier way to rebalance AI in the game
In vanilla game, if you do not specify certain AI attribute like “starting_health” or “aim_accuracy” in the .ai file – game will take its value from the code. Those values are specified in the aidefaults functions accessible only in the code.

However RealRTCW takes it to another level in different ways.

First of all aidefaults values parsed out of the code into **txt files with .aidefaults extension.** They located in **z_zrealrtcw_scripts.pk3/aidefaults.** All default attributes values are specified there for each AI type.

Secondly, certain values like **aimSkill, aimAccuracy, attackSkill, reactionTime, aggression and startingHealth could be randomized** in certain range for each of five RealRTCW difficulty levels. It goes from easy to realism.

So basically, if you want to use this system you **DO NOT NEED to specify** needed attributes in the .ai files for each ai individually. As I said – if you do not specify attribute in .ai file it will take it from .aidefaults. This allows you to do massive balance changes without editing each .ai on every map. But it is still optional and you can do it like in the old days.

.weap files system
All weapons media including sounds,icons,models,etc. are now defined in .weap files instead of the code. All weapon parameters like fire rate,damage,reloading time were parsed out of the code into .weap as well.
All .weap files are located in z_zrealrtcw_scripts.pk3/weapons/

.ents files system
Allows you to add more entities onto your maps without recompiling them.
You add entities into .ents the same way Radiant adds them onto your map – simply define its classname and origin. Classnames could be looked up in Radiant.

As for the origin, you can simply launch the map in the game, stand on the point you want to add your entity and type in the console “/where”. This will give you the exact coordinates of the spot. Additional AIs must be specified in .ai file as well. Just like you normally do while creating the map.

Example from maps/assault.ents:

```
{
"classname" "ai_soldier"
"origin" "3280 3130 472"
"ainame" "reinforce_ai_soldier_2"
"angle" "-177"
"spawnflags" "1"
"skin" "infantryss/assault1"
"head" "assault2"
}
```

Difficulty specified .ai and .ents files
You can use specific .ai and .ents files for your maps. For this you need to create additional folders for each difficulty inside maps folder in .pk3. You should name those folders like that: easy, medium, hard, max, realism.
Now throw your .ai and .ents in those folders and game will use separate file for each difficulty level.

Subtitles
cg_drawsubtitles – enable/disable subtitles
Subtitle files are located in z_realrtcw_text.pk3/text/EnglishUSA/maps
Each map has its specific file for subtitles. You reference script name of the audio file, where AI speaks, and after that specify the text.

Be aware of the **50 symbols** limit in the single string. Going over it will lead to visual subtitles glitches. I strongly suggest you to take a look at RealRTCW subtitles files and do your subtitles accordingly to its structure.

Expanded scripting
You can now reference CVARs in the .ai and .script files.
For example you can specify weapon of the enemies depending on the CVAR value:

```
#if g_fullarsenal == 0
giveweaponfull weapon_sten
#endif
#if g_fullarsenal > 0
giveweaponfull weapon_mp34
 #endif
```

New script functions
giveweaponfull – Takes away all AIs weapon, gives specified weapon, fills both reserve ammo and current clip to the max and selects the weapon itself.
giveweaponfull weapon_random - gives AI random weapon
giveweaponfull pistol_random - gives AI random pistol
giveweaponfull smg_random - gives AI random smg
giveweaponfull rifle_random - gives AI random rifle
giveweaponfull ar_random - gives AI random assault rifle
giveweaponfull heavy_random - gives AI random heavy weapon

drop_weapon – makes AI to toss his current weapon.
changeaiteam – change team of the AI on the fly.
changeainame – changes AI script name on the fly.
burn – make em burn.

giveammo – gives player an actual ammo item
givehealth – gives player an actual health item
givearmor – gives player an actual armor item
giveinventory – gives holdable item

setmovespeed - changes player movespeed through the triggers.
Values - veryslow, slow, default, fast, veryfast.

screenfade - fades player screen in/out. Syntax - screenfade <fadetime> <in/out>.
Example: "screenfade 5000 in"

accumaction - math operations with accum buffers.
Examples:
accumaction 3 1 plus 0 // accum 3 equals accum 1 plus accum 0
accumaction 3 1 minus 0 // accum 3 equals accum 1 minus accum 0
accumaction 3 1 mul 0 // accum 3 equals accum 1 multiply on accum 0
accumaction 3 1 div 0 // accum 3 equals accum1 divided on accum 0

accumgametime - store game time into accum buffer.
Example: "accumgametime 1" - will store game time in accum 1

printlabel - change parameters of displayed label
Syntax:
printlabel txt <string> // change text part of the label
printlabel param <value_buffer_index> <x> <y> // change numeric part of the label and position it with x y coordinates
printlabel state <state_buffer_index> // show the label, if value of the accum is higher than 0, otherwise hide
printlabel on // show the label
printlabel off // hide the label
printlabel format <formatstring> // label formatting. Assing them in any order and divide them with space.
formatstring:
pulse – pulsing string
string – enable text part
accum – enable numeric part
timer – numeric part will be displayed in seconds
inline – one string

New CVARs

g_jumptime - enable/disable strafejumping

cg_drawsubtitles - enable/disable subtitles

cg_solidcrosshair - enable/disable solid crosshair

cg_bloodblend - enable/disable blood on HUD

cg_bobbing - enable/disable bobbing while crouching

cg_sniperscrosshair - enable/disable crosshairs for unscoped sniper rifles

cg_atmosphericEffects - enable/disable atmospheric effects

cg_lowAtmosphericEffects - enable/disable low quality atmospheric effects

cg_forceAtmosphericEffects - force specific type of atmospheric effects

cg_autoReload - enable/disable autoreload

g_bodysink - enable/disable bodysink

g**_gunposlock** - varied view weapon positions

g_airespawn - enable/disable AI respawn

g_reinforce - enable/disable AI reinforce

g_fullarsenal - affects set of weapons player will recieve during campaign

g_fireonthemove - allows AI to shoot rifles on the move

g_weaponfalloff - enables damage falloff for weapons

g_aicanheadshot - allows AI to do headshots

g_spawnpriests – enable/disable Priest AI spawn

g_spawndogs – enable/disable Dog AI spawn

g_spawnxshepherds – enable/disable X-Shepherd AI spawn

cg_hudstatus – choose hud type

cg_hudWeapIcon – enable/disable weapon icon

cg_hudAmmoClip – enable/disable ammo icon

cg_hudStamina – enable/disable stamina bar

cg_journalstyle – enable/disable journal notifications

cg_hudStyle - choose style of the HUD