RealRTCW guide for modders and mappers
Last edited 24/05/2022

New weapons script files references (“name – altname”):

weapon_dagger – dagger (vampiric knife)

weapon_delisle – delisle (delisle carbine) – *shares ammo with thompson,colt.*

weapon_delislescope – delislescope (scoped delisle) – *shares ammo with thompson,colt.*

weapon_mp34 – mp34 – *shares ammo with mp40*

weapon_tt33 – tt33

weapon_p38 – p38 – *shares ammo with mp40*

weapon_ppsh – ppsh

weapon_mosin – mosin

weapon_g43 – g43 – shares ammo with mauser,fg42

weapon_m1garand – m1garand

weapon_m7 – m7 (rifle grenade for m1 garand)

weapon_bar – bar

weapon_mp44 – mp44

weapon_mg42m – mg42m – *shares ammo with venom*

weapon_browning – browning

weapon_m97 – m97 (Ithaca shotgun)

weapon_m30 – m30

weapon_welrod – welrod – *shares ammo with snooper*

weapon_holycross – holy cross

weapon_revolver – revolver

weapon_grenadesmoke – smokeGrenade (airstrike signal)

weapon_poisongas – poison gas

New ammo script files references:

ammo_poison_gas – poison gas

ammo_m7 – m7_ammo

ammo_holyspirit – spirit – *used by holycross*

ammo_ttammo – ttammo *– used by ppsh,tt33*

ammo_ttammo_l – ttammol

ammo_mosina – mosina *– used by mosin rifle*

ammo_barammo – barammo *– used by bar,garand*

ammo_barammo_l – barammol

ammo_44ammo – 44ammo *– used by mp44*

ammo_44ammo_l – 44ammol

ammo_m97ammo – m97ammo

ammo_revolver – revolverammo

New holdables references:

holdable_adrenaline

holdable_bandages

Re-enabled Q3 powerups references:

item_quad – quad damage

item_haste – speed powerup

item_enviro – protective suit

item_invis – invisibility

Atmospheric effects:

RealRTCW implements atmospheric effects system from Wolfenstein: Enemy Territory. It includes rain and snow. To add snow into your map add **“atmosphere”** key parameter to your **worldspawn** entity (click on any geo non scripted brush).

As a value for this key parameter you can use these presets:

T=SNOW,B=5 10,C=0.5,G=0.3 2,BV=50 50,GV=30 80,W=1 2,D=15000 (strong snow - used on norway)

T=SNOW,B=5 10,C=0.5,G=0.3 2,BV=20 30,GV=25 40,W=3 5,D=5000 (weaker snow - used on escape1)

T=RAIN,B=5 10,C=0.5,G=0.5 2,BV=50 50,GV=200 200,W=1 2,D=5000 (strong rain - used on dark)

Atmoshperic effects CVARs:

cg_atmoshpericeffects - disables\enables atmospheric effects

cg_forceatmosphericeffects - 0- no force, 1- rain, 2- snow. This way you can force atmoshpericeffects in-game on any map. Requires vid_restart.

cg_lowAtmosphericEffects – 0 - high quality (dont recommend that, cause it will cause particles bugs), 1- medium quality(no raindrops), 2- disables completely

Automatic AI attributes system:

RealRTCW introduces an easier way to rebalance AI in the game.

In vanilla game, if you do not specify certain attribute(for example *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.

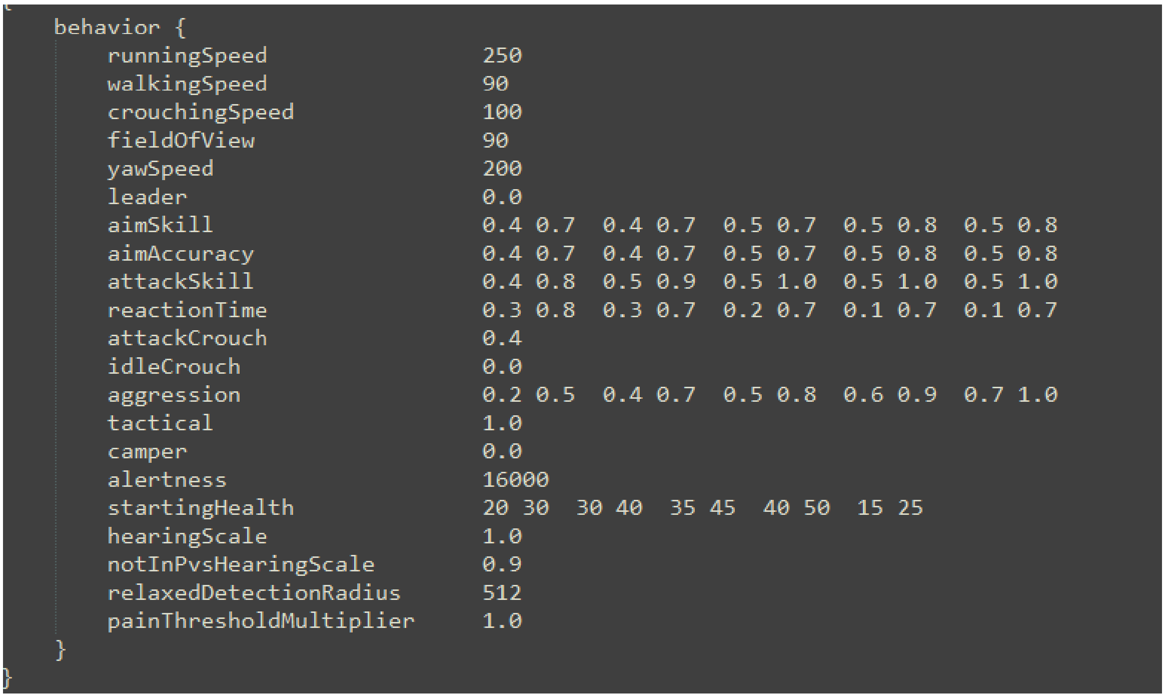
However RealRTCW takes it to another level in different ways.

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.



```
    behavior {
        runningSpeed                250
        walkingSpeed                90
        crouchingSpeed              100
        fieldOfView                 90
        yawSpeed                    200
        leader                      0.0
        aimSkill                    0.4 0.7  0.4 0.7  0.5 0.7  0.5 0.8  0.5 0.8
        aimAccuracy                 0.4 0.7  0.4 0.7  0.5 0.7  0.5 0.8  0.5 0.8
        attackSkill                 0.4 0.8  0.5 0.9  0.5 1.0  0.5 1.0  0.5 1.0
        reactionTime                0.3 0.8  0.3 0.7  0.2 0.7  0.1 0.7  0.1 0.7
        attackCrouch                0.4
        idleCrouch                  0.0
        aggression                  0.2 0.5  0.4 0.7  0.5 0.8  0.6 0.9  0.7 1.0
        tactical                    1.0
        camper                      0.0
        alertness                   16000
        startingHealth              20 30  30 40  35 45  40 50  15 25
        hearingScale                1.0
        notInPvsHearingScale        0.9
        relaxedDetectionRadius      512
        painThresholdMultiplier     1.0
    }
}
```

So basically, if you want to use this system you do not need to specify needed attributes in the .ai files. **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.

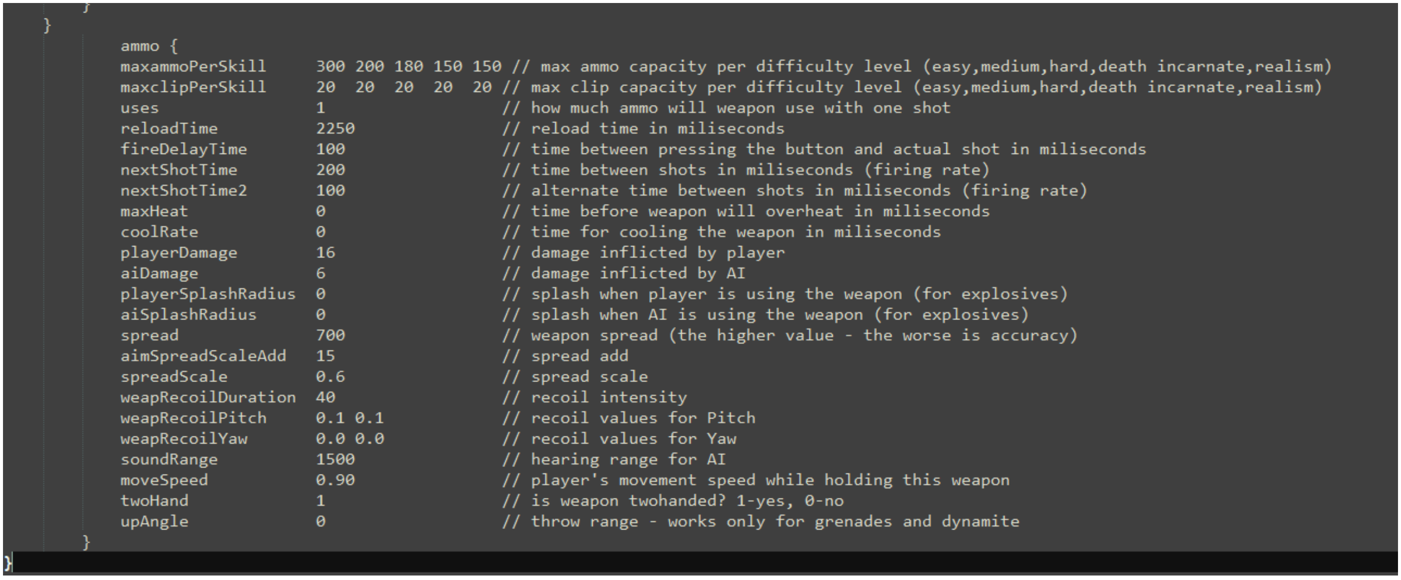
.weap files system:

All weapons media including sounds,icons,models,etc. is now defined in **.weap files** instead of the code. This was ported from Wolfenstein: Enemy Territory.

However, this system was greatly expanded, since all weapon parameters like fire rate,damage,reloading time was parsed out of the code into .weap as well.

All .weap files are located in z_zrealrtcw_scripts.pk3/weapons/

It’s structure should be self-explanatory. Take a look at the BAR ammo section:



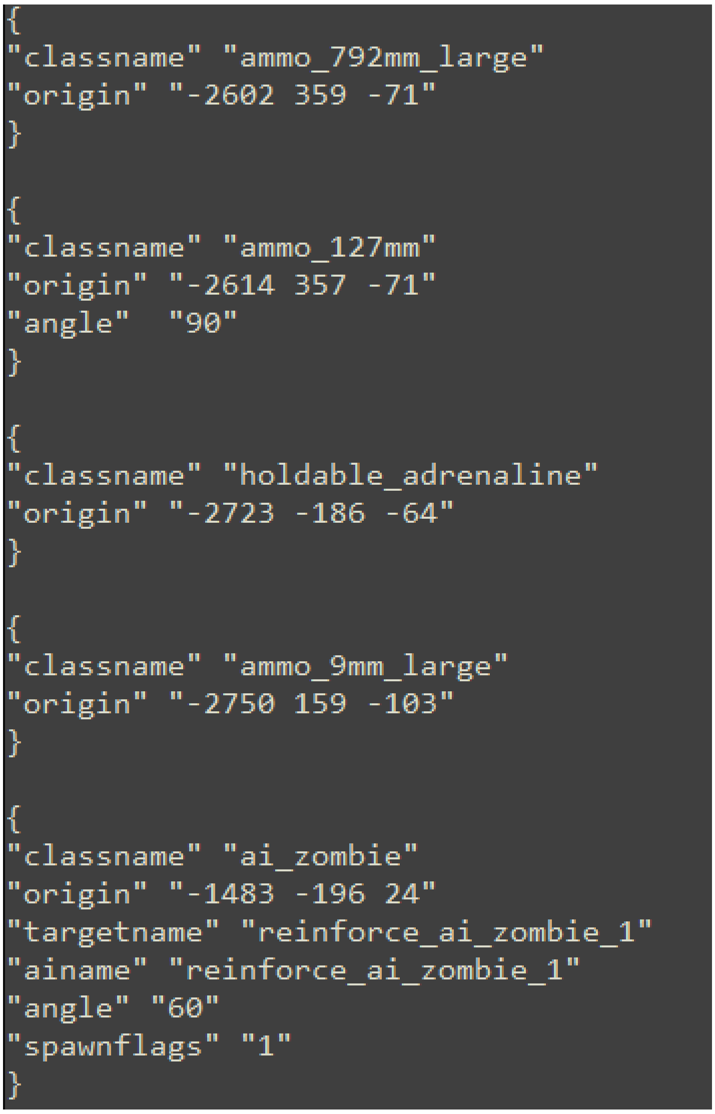
```
        }
    }
            ammo {
            maxammoPerSkill     300 200 180 150 150 // max ammo capacity per difficulty level (easy,medium,hard,death incarnate,realism)
            maxclipPerSkill     20  20  20  20  20 // max clip capacity per difficulty level (easy,medium,hard,death incarnate,realism)
            uses                1                  // how much ammo will weapon use with one shot
            reloadTime          2250               // reload time in miliseconds
            fireDelayTime       100                // time between pressing the button and actual shot in miliseconds
            nextShotTime        200                // time between shots in miliseconds (firing rate)
            nextShotTime2       100                // alternate time between shots in miliseconds (firing rate)
            maxHeat             0                  // time before weapon will overheat in miliseconds
            coolRate            0                  // time for cooling the weapon in miliseconds
            playerDamage        16                 // damage inflicted by player
            aiDamage            6                  // damage inflicted by AI
            playerSplashRadius  0                  // splash when player is using the weapon (for explosives)
            aiSplashRadius      0                  // splash when AI is using the weapon (for explosives)
            spread              700                // weapon spread (the higher value - the worse is accuracy)
            aimSpreadScaleAdd   15                 // spread add
            spreadScale         0.6                // spread scale
            weapRecoilDuration  40                 // recoil intensity
            weapRecoilPitch     0.1 0.1            // recoil values for Pitch
            weapRecoilYaw       0.0 0.0            // recoil values for Yaw
            soundRange          1500               // hearing range for AI
            moveSpeed           0.90               // player's movement speed while holding this weapon
            twoHand             1                  // is weapon twohanded? 1-yes, 0-no
            upAngle             0                  // throw range - works only for grenades and dynamite
        }
}
```

.ents files system

This was ported from RTCWCoop and allows you to add more entities onto your maps without recompiling them (very useful for vanilla maps). 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. For RealRTCW specific classnames take a look at the list I specified in the beginning.

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.



```
{
"classname" "ammo_792mm_large"
"origin" "-2602 359 -71"
}

{
"classname" "ammo_127mm"
"origin" "-2614 357 -71"
"angle"  "90"
}

{
"classname" "holdable_adrenaline"
"origin" "-2723 -186 -64"
}

{
"classname" "ammo_9mm_large"
"origin" "-2750 159 -103"
}

{
"classname" "ai_zombie"
"origin" "-1483 -196 24"
"targetname" "reinforce_ai_zombie_1"
"ainame" "reinforce_ai_zombie_1"
"angle" "60"
"spawnflags" "1"
}
```

Subtitles

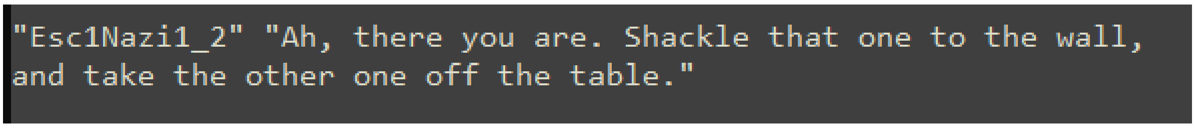
Specific CVAR: cg_drawsubtitles – enables/disables subtitles

Located in **z_realrtcw_text.pk3/text/EnglishUSA/maps**

Each map has its specific file for subtitles.

Basically 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. Just go to the next string like that:

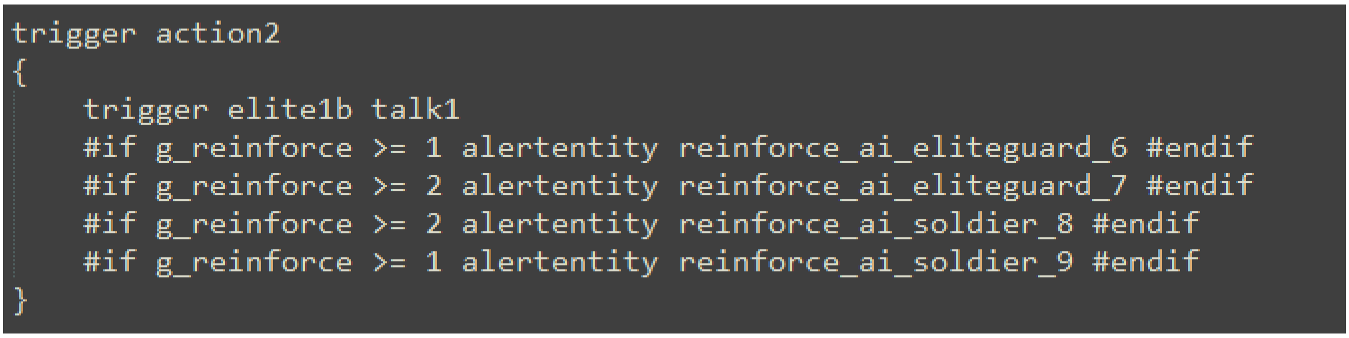


```
"Esc1Nazi1_2" "Ah, there you are. Shackle that one to the wall,
and take the other one off the table."
```

I strongly suggest you to take a look at RealRTCW subtitles files and do your subtitles accordingly to its structure.

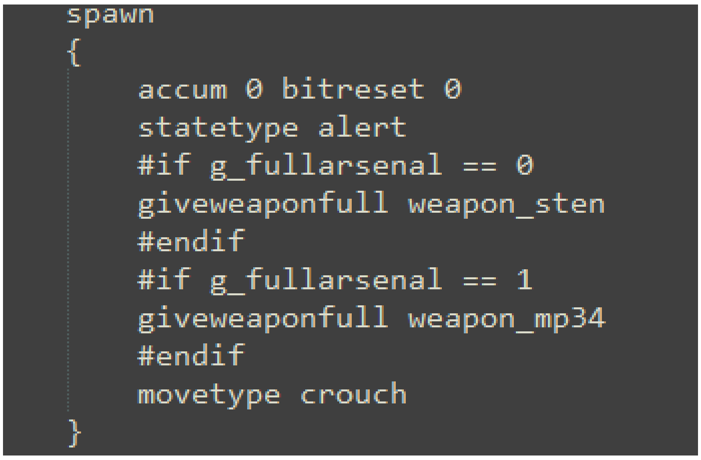
Expanded scripting

You can now reference CVARs in the .ai and .script files. For example you can specify spawn of the certain enemis depending on the CVAR value:



```
trigger action2
{
    trigger elite1b talk1
    #if g_reinforce >= 1 alertentity reinforce_ai_eliteguard_6 #endif
    #if g_reinforce >= 2 alertentity reinforce_ai_eliteguard_7 #endif
    #if g_reinforce >= 2 alertentity reinforce_ai_soldier_8 #endif
    #if g_reinforce >= 1 alertentity reinforce_ai_soldier_9 #endif
}
```

Or you can give AI certain weapon:



```
    spawn
    {
        accum 0 bitreset 0
        statetype alert
        #if g_fullarsenal == 0
        giveweaponfull weapon_sten
        #endif
        #if g_fullarsenal == 1
        giveweaponfull weapon_mp34
        #endif
        movetype crouch
    }
```

New script functions:

giveweaponfull – basically do multiple things in one go. Takes away all AIs weapon, gives him specified weapon, fills both reserve ammo and current clip to the max and selects the weapon itself.

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.

accumprint – ?

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.

New CVARs:

- g_jumptime - enable/disable strafejumping
- cg_drawsubtitles - enable/disable subtitles
- cg_solidcrosshair - enable/disable solid crosshair (ported from RTCWCoop)
- cg_bloodblend - enable/disable blood on HUD (ported from RTCWCoop)
- cg_bobbing - enable/disable bobbing while crouching (ported from ETLegacy)
- 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 (ported from ET)
- g_bodysink - enable/disable bodysink
- g_gunposlock - varied view weapon positions
- g_airespawn - enable/disable AI respawn (ported from RTCWCoop)
- g_reinforce - enable/disable AI reinforce (ported from RTCWCoop)
- 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 SMGs and pistols
- g_aicanheadshot - allows AI to do headshots