Subject: Re: Basic explanation about Huds
Posted by Spyder on Mon, 21 Jul 2008 19:16:18 GMT
View Forum Message <> Reply to Message

BHS.TXT
Also, hud.ini now contains features to totally customize the HUD.
Any time you are setting a screen position (e.g. HealthXPos), if you pass a negative number, it will move that many units in from the bottom right of the screen, otherwise it moves from the top left of the screen)
This is so you can build HUDs with items in bottom and left corners whilst remaining independant of screen resolution.
Unless specified otherwise, all entries go in the [General] section of hud.ini
Also, any time you specify a texture, always use .tga on the end even if the texture is a dds file.
Where it specifies a font number from stylemgr.ini, 0 is FONT_TITLE and so on. Adding new fonts is not possible.

Firstly, you can define custom colors that will be used later on in hud.ini
You set ColorCount to specifiy how many colors to add.
Then you create sections labeled [Color1] and so on on up. Each section has entries Red, Green and Blue to specify the color (values go from 0 to 255)
In all the entries below, anytime a color is mentioned, it is one of these colors specified here.
Color zero is the default color and is black.

You can define customization for the number that shows your current health.
Set HealthEnabled=true to turn it on (and turn the default westwood logic off)
Set HealthVisible=false to completly hide the number otherwise it will be drawn
Set HealthXPos and HealthYPos to specify where on the screen to draw the number.
Set HealthXOffset and HealthYOffset to specify the health offset (basicly, this number is multiplied by the current % health the player has and added to the health position when drawing)
Set HealthFont to a texture name (i.e. a texture similar to FONT12x16.TGA) to specify the font to use.
Set HealthColor to a color index to specify the color to use.
There is also a way (which overrides HealthColor) to set colors depending on how much health you have)
HealthColorCount specifies how many color entries there are.
Then you create sections like [HealthColor0] etc to specify the colors.
Each section has keywords Color which is the color index to use and Value which is the percentage of health to start drawing at.
The sections must be organized so that the smallest value is section 0 and so on on up.

You can define customization for the number that shows your current shield.
Set ShieldEnabled=true to turn it on (and turn the default westwood logic off)
Set ShieldVisible=false to completly hide the number otherwise it will be drawn
Set ShieldXPos and ShieldYPos to specify where on the screen to draw the number.
Set ShieldXOffset and ShieldYOffset to specify the shield offset (basicly, this number is multiplied by the current % shield the player has and added to the shield position when drawing)
Set ShieldFont to a texture name (i.e. a texture similar to FONT12x16.TGA) to specify the font to use.

Set ShieldColor to a color index to specify the color to use.
There is also a way (which overrides ShieldColor) to set colors depending on how much shield you have)
ShieldColorCount specifies how many color entries there are.
Then you create sections like [ShieldColor0] etc to specify the colors.
Each section has keywords Color which is the color index to use and Value which is the percentage of shield to start drawing at.
The sections must be organized so that the smallest value is section 0 and so on on up.

You can define customization for the number that shows the current bullets in your gun
Set BulletEnabled=true to turn it on (and turn the default westwood logic off)
Set BulletVisible=false to completly hide the number otherwise it will be drawn
Set BulletXPos and BulletYPos to specify where on the screen to draw the number.
Set BulletFont to a texture name (i.e. a texture similar to FONT12x16.TGA) to specify the font to use.
Set BulletColor to a color index to specify the color to use.

You can define customization for the number that shows the current bullets in your "backpack"
Set ClipEnabled=true to turn it on (and turn the default westwood logic off)
Set ClipVisible=false to completly hide the number otherwise it will be drawn
Set ClipXPos and ClipYPos to specify where on the screen to draw the number.
Set ClipFont to a texture name (i.e. a texture similar to FONT12x16.TGA) to specify the font to use.
Set ClipColor to a color index to specify the color to use.

You can define customization for the weapon name.
Set WeaponEnabled=true to turn it on (and turn the default westwood logic off)
Set WeaponVisible=false to completly hide the name otherwise it will be drawn
Set WeaponVisibleNonVehicle=false to hide the name whilst not in a vehicle
Set WeaponXPos and WeaponYPos to specify where on the screen to draw the name.
Set WeaponFont to a font number from stylemgr.ini
Set WeaponColor to a color index to specify the color to use.

You can define customization for the weapon image.
Set WeaponImageEnabled=true to turn it on (and turn the default westwood logic off)
Set WeaponImageVisible=false to completly hide the image otherwise it will be drawn
Set WeaponImageVisibleNonVehicle=false to hide the image whilst not in a vehicle (i.e. you see the steering wheel, gun and seat icons but not the weapon icon)
Set WeaponImageXPos and WeaponImageYPos to specify where on the screen to draw the name.
Set WeaponImageColor to a color index to specify the color to use.
Set WeaponImageVehicleColor to a color index to specify the color to use when inside a vehicle.

You can hide the names of enemy soldiers.
Set HideEnemySoldiers=true to enable this feature
Then, set HideExceptionCount to define the count of how many exceptions (exceptions are for presets like spies etc that you want to keep visible)
Set HideException0 and so on to the preset IDs of the presets you want to hide

You can define arbitrary textures to be drawn on the screen. (this is for the general parts of the HUD)
Set TextureCount to the count of how many textures you want
Create a section e.g. [Texture0] for each texture
Then, under that section, set TextureName to the texture name.
Set QuadCount to the count of how many rectangles to draw with this texture
Set Quad0Color (and on up) to the color for that rectangle
Set Quad0XPos (and on up) to the X screen position for that rectangle
Set Quad0YPos (and on up) to the Y screen position for that rectangle
Quad0Top, Quad0Left, Quad0Bottom & Quad0Right (and on up) define a rectangle within the texture which is what is drawn at the x,y screen position.

You can customize the health bar.
Set HealthBarEnabled=true to turn it on (and turn the default westwood logic off)
Set HealthBarVisible=false to completly hide the health bar otherwise it will be drawn
Set HealthBarEmptyVisible=false to hide the empty part of the healt bar otherwise it will be drawn
Set HealthBarColor to define the color index to use for the health bar
Set HealthBarEmptyColor to define the color index to use the for empty health bar
There is also a way (which overrides HealthBarColor) to set colors depending on how much health you have)
HealthBarColorCount specifies how many color entries there are.
Then you create sections like [HealthBarColor0] etc to specify the colors.
Each section has keywords Color which is the color index to use and Value which is the percentage of health to start drawing at.
The sections must be organized so that the smallest value is section 0 and so on on up.
Set HealthBarTexture to define the texture to use for the health bar
Set HealthBarXPos and HealthBarYPos to define the position on the screen of the health bar
HealthBarTop, HealthBarLeft, HealthBarBottom & HealthBarRight define a rectangle within the texture for the health bar
HealthBarEmptyTop & HealthBarEmptyLeft define the position of the empty health bar on the texture. The size is the same as the full health bar (i.e. when the health is not full to that spot)
Set HealthBarStyle=0 to enable a bar similar to the renegade health bar
Set HealthBarStyle=1 to enable a bar similar to the renegade shield strength bar
Set HealthBarDirection to determine which direction to draw for style 2 (1 is the way renegade draws it, 0 is the opposite direction)
Set HealthBarOffset to determine the offset to draw the icons at for style 2
Set HealthBarLength to set the length for the entire health bar for style 2
When the style is style 2, the HealthBarTop etc keywords define the icon that is to be used (e.g. like the shield in renegade)
The x and y position determines the position of the bar
The length determines the horizontal size and the height defined by the HealthBarTop etc keywords determine the vertical size

You can customize the shield bar.
Set ShieldBarEnabled=true to turn it on (and turn the default westwood logic off)
Set ShieldBarVisible=false to completly hide the shield bar otherwise it will be drawn
Set ShieldBarEmptyVisible=false to hide the empty part of the healt bar otherwise it will be drawn

Set ShieldBarColor to define the color index to use for the shield bar
Set ShieldBarEmptyColor to define the color index to use the for empty shield bar
There is also a way (which overrides ShieldBarColor) to set colors depending on how much shield you have)
ShieldBarColorCount specifies how many color entries there are.
Then you create sections like [ShieldBarColor0] etc to specify the colors.
Each section has keywords Color which is the color index to use and Value which is the percentage of shield to start drawing at.
The sections must be organized so that the smallest value is section 0 and so on on up.
Set ShieldBarTexture to define the texture to use for the shield bar
Set ShieldBarXPos and ShieldBarYPos to define the position on the screen of the shield bar
ShieldBarTop, ShieldBarLeft, ShieldBarBottom & ShieldBarRight define a rectangle within the texture for the shield bar
ShieldBarEmptyTop & ShieldBarEmptyLeft define the position of the empty shield bar on the texture. The size is the same as the full shield bar (i.e. when the shield is not full to that spot)
Set ShieldBarStyle=0 to enable a bar similar to the renegade health bar
Set ShieldBarStyle=1 to enable a bar similar to the renegade shield strength bar
Set ShieldBarDirection to determine which direction to draw for style 2 (1 is the way renegade draws it, 0 is the opposite direction)
Set ShieldBarOffset to determine the offset to draw the icons at for style 2
Set ShieldBarLength to set the length for the entire shield bar for style 2
When the style is style 2, the ShieldBarTop etc keywords define the icon that is to be used (e.g. like the shield in renegade)
The x and y position determines the position of the bar
The length determines the horizontal size and the height defined by the ShieldBarTop etc keywords determine the vertical size

Set HideWeaponBox=true to hide the background piece that is underneath the weapon information
Set HideInfoBox to hide the background piece that is over the other side (this covers everything except the radar that is not text including the health and shield bars)

You can also customize the compass (i.e. the N,E,S,W etc text)
Set EnableCompass=true to turn it on (and turn the default westwood logic off)
Set CompassVisible=false to completly hide the compass otherwise it will be drawn
Set CompassFont to a font number from stylemgr.ini
Set CompassColor to the color index to use when drawing the compass
Set CompassXPos and CompassYPos to define where to draw the compass

You can also customize the credits text.
Set EnableCredits=true to turn it on
Set CreditsXPos and CreditsYPos to define where to draw the credits display
Set CreditsFont to a font number from stylemgr.ini
Set CreditsColor to the color index to use when drawing the credits
Set CreditsStringID to a string ID from strings.tdb. This string should use %d at the spot where you want the credits number to go.
For example, "Current Credits is %d now" will replace the %d with the current credits

You can also customize the time remaining text (it will not be drawn if the level has unlimited time)
Set EnableTime=true to turn it on
Set TimeXPos and TimeYPos to define where to draw the time display
Set TimeFont to a font number from stylemgr.ini
Set TimeColor to the color index to use when drawing the time
Set TimeStringID to a string ID from strings.tdb. This string should use %s at the spot where you want the time to go.
For example, "There is %s remaining" will replace the %s with the current time formatted as hours:minutes:seconds just like the default display)

You can also display an icon for the health (like the + symbol in normal renegade)
Set HealthIconEnabled=true to turn it on
Set HealthIconColor to define the color index to use for the health icon
There is also a way (which overrides HealthIconColor) to set colors depending on how much health you have)
HealthIconColorCount specifies how many color entries there are.
Then you create sections like [HealthIconColor0] etc to specify the colors.
Each section has keywords Color which is the color index to use and Value which is the percentage of health to start drawing at.
The sections must be organized so that the smallest value is section 0 and so on on up.
Set HealthIconTexture to the texture to use for the health icon
Set HealthIconXPosition and HealthIconYPosition to the position to draw the health icon
HealthIconTop, HealthIconLeft, HealthIconBottom and HealthIconRight define the rectangle on the texture to use for the icon

You can also display an icon for the shield (like the + symbol in normal renegade)
Set ShieldIconEnabled=true to turn it on
Set ShieldIconColor to define the color index to use for the shield icon
There is also a way (which overrides ShieldIconColor) to set colors depending on how much shield you have)
ShieldIconColorCount specifies how many color entries there are.
Then you create sections like [ShieldIconColor0] etc to specify the colors.
Each section has keywords Color which is the color index to use and Value which is the percentage of shield to start drawing at.
The sections must be organized so that the smallest value is section 0 and so on on up.
Set ShieldIconTexture to the texture to use for the shield icon
Set ShieldIconXPosition and ShieldIconYPosition to the position to draw the shield icon
ShieldIconTop, ShieldIconLeft, ShieldIconBottom and ShieldIconRight define the rectangle on the texture to use for the icon

Unlike the renegade code, you cant implement it so that the health/shield icon will flash when you have low health/shield

You can also customize the radar.
Set EnableRadar=true to turn it on and turn the default westwood logic off.
Set RadarRotate=true to make the radar rotate so that "up" is always the direction you are facing.
Set RadarRotate=false to make up always = north. (this is needed later on when I implement the scrolling map logic)

Set DrawStar=true to draw you (i.e. the current player) on the radar
Set DrawCompassLine=true to draw a line in the direction the player is facing (this is disabled if RadarRotate is set to true)
Set CompassLineColor to the color to use for the compass line
Set CompassLineWidth to the width for the compass line
Set RadarVisible=false to turn off the radar completly
Set RadarSize to the screen size in pixels to use for the radar (its always going to be a circle)
Set RadarWorldSize to the size in the game world that is to correspond to the radius of the radar circle
Set BackgroundColor to the color to use when drawing the background
Set BackgroundTexture to the texture to use for the background
Set BlipTexture to the texture to use for the radar blips
Set BackgroundTop and BackgroundLeft to the top left position on the background texture to draw from.
Set RadarX and RadarY to define the top left position of the square where the radar is to be drawn
Set ScrollingRadarMap=true to enable the scrolling map background feature (where you see a top-down view of the map under the radar blips which scrolls based on the location of the player)
Set RadarBlipnTop and RadarBlipnLeft (starting from RadarBlip1Top/RadarBlip1Left) to cover the UV positions for the radar blips
Blips should go from 1 to 5
The blips are always 8 pixels in size
Set RadarBlipColor0 and so on up to RadarBlipColor7 to define the blip colors
Blip zero is blank
Blip 1 is for humans (by default its a circle)
Blip 2 is for vehicles (by default its a triangle)
Blip 3 is for stationary objects (by default its a square)
Blip 4 is for objectives (by default its a star)
Blip 5 is for the bracket that is drawn when you are targeting an object
The blip colors are as follows:
Color 0 is red (Nod color)
Color 1 is gold (GDI color)
Color 2 is white
Color 3 is dark green
Color 4 is blue
Color 5 is green (and is used when the bracket is drawn)
Color 6 is light blue
Color 7 is purple
Right now, this code is not perfect, I plan to extend it later (for example, to make it draw objectives and radar markers)

How the scrolling map feature works:
If you turn it on with ScrollingRadarMap=true, you then need to set the settings for each map with something like JFW_Change_Radar_Map.
Basicly, it draws the radar background the same as the normal radar code with the exception of the texture coordinates for the background texture.
By default, the center of the map texture is assumed to match with 0,0 in the game world. Use the offsetx and offsety values to specify where on the texture (relative to the center) 0,0 in the game

world is.
The scale field codes for how many pixels on the texture 1 unit in the game world is equal to.
Just play around with the offset and scale asnd see what looks right for your map.
As of 3.0, you can create an ini file called mapname.ini (e.g. C&C_Islands.ini) which contains the following tags in the [General] section
ScrollingMapTexture
ScrollingMapOffsetX
ScrollingMapOffsetY
ScrollingMapScale
These correspond to the parameters that get passed to Change_Radar_Map
This ini file goes on the client. If the map/server makes a call to Change_Radar_Map, it will override anything set by the ini file.

Set HideBottomText=true to disable the credits and time remaining dislay
Set HidePlayerList=true to hide the player and team list
Set HidePowerupIcons=true to hide the icons that show powerups (e.g. when you pick one up)
Set DisableCostMultiplier=true to disable the 2x cost multiplier that applies if the base power is down (both the display and the effect)
Set BuildTimeDelay=some number to change the vehicle build time multiplier (default is 2x) that applies if base power is down
Set VehicleOwnershipDisable=true to disable the effect where only the person who bought a vehicle can get into it for a while after its bought
Set UseExtraPTPages=true to enable use of the extra "hidden" PT pages (the ones you access with the alt key) as normal PT pages. This disables the extras console command, disables the laddered server check and enables the PT pages feature.
Set NewUnpurchaseableLogic=true to enable new logic for when vehicles and characters are being bought. This is needed if you wish to use Set_Can_Generate_Soldiers or Set_Can_Generate_Vehicles after the relavent buildings have gone down.
However, with it enabled, the text under the vehicle and character buttons will always say "unavailable", never "destroyed"
Also, vehicles dont become available again if the vehicle factory is offline (however, the vehicle purchase hook will be triggered) since there is no vehicle factory to deliver the vehicles
Set VehicleBuildingDisable=true to disable the code that makes the vehicle button show "building" when a vehicle is bought from the weapons factory.
set the folowing keywords to change the colors associated with the various UI elements. Alpha is the indicator of how opaque that element is (FF = completly opaque). The value after each keyword is the default.
TitleColorAlpha=255
TitleColorRed=255
TitleColorGreen=255
TitleColorBlue=255
TitleHilightColorAlpha=255
TitleHilightColorRed=255
TitleHilightColorGreen=255
TitleHilightColorBlue=0
TitleShadowColorAlpha=255
TitleShadowColorRed=0
TitleShadowColorGreen=0

TitleShadowColorBlue=0
TextColorAlpha=255
TextColorRed=255
TextColorGreen=213
TextColorBlue=40
TextShadowColorAlpha=200
TextShadowColorRed=0
TextShadowColorGreen=0
TextShadowColorBlue=0
LineColorAlpha=255
LineColorRed=255
LineColorGreen=174
LineColorBlue=40
BkColorAlpha=40
BkColorRed=255
BkColorGreen=174
BkColorBlue=40
DisabledTextColorAlpha=140
DisabledTextColorRed=255
DisabledTextColorGreen=213
DisabledTextColorBlue=40
DisabledTextShadowColorAlpha=96
DisabledTextShadowColorRed=0
DisabledTextShadowColorGreen=0
DisabledTextShadowColorBlue=0
DisabledLineColorAlpha=128
DisabledLineColorRed=230
DisabledLineColorGreen=160
DisabledLineColorBlue=35
DisabledBkColorAlpha=30
DisabledBkColorRed=255
DisabledBkColorGreen=174
DisabledBkColorBlue=40
HilightColorAlpha=255
HilightColorRed=70
HilightColorGreen=70
HilightColorBlue=70
TabTextColorAlpha=255
TabTextColorRed=255
TabTextColorGreen=255
TabTextColorBlue=255
TabGlowColorAlpha=255
TabGlowColorRed=16
TabGlowColorGreen=10
TabGlowColorBlue=0
DialogTextTitleColorAlpha=255
DialogTextTitleColorRed=255
DialogTextTitleColorGreen=255

DialogTextTitleColorBlue=36
DialogTextTitleGlowColorAlpha=255
DialogTextTitleGlowColorRed=14
DialogTextTitleGlowColorGreen=0
DialogTextTitleGlowColorBlue=0
MenuHiliteColorAlpha=255
MenuHiliteColorRed=0
MenuHiliteColorGreen=0
MenuHiliteColorBlue=0
MerchandiseTextColorAlpha=0
MerchandiseTextColorRed=255
MerchandiseTextColorGreen=255
MerchandiseTextColorBlue=255