Graphical asset modding - Hearts of Iron 4 Wiki

This is a community maintained wiki. If you spot a mistake then you are welcome to fix it.

Graphical assets are image files that have been defined in a `.gfx` file. This binds the image to a script name that is then referred to in the `.gui` interface files.

Graphical asset files can be found in /Hearts of Iron IV/interface/*.gfx.

Note that for many assets, such as technology and equipment asset definitions, the image can be specified for a specific tag, i.e. `GFX_ENG_<name>`, which only displays the asset for the specified tag. This will override the base asset of the same name. This also applies to graphical culture, i.e. `GFX_african_<name>`.

Sprite Types[edit | edit source]

spriteType[edit | edit source]

A **spriteType** and **textSpriteType** asset entry follows this format:

```
spriteTypes = {
  spriteType = {
    name = "GFX_<name>"
    texturefile = <path>
    noOfFrames = <int>
    effectFile = <path>

    allwaystransparent = <bool>
    legacy_lazy_load = <bool>
    transparencecheck = <bool>

    animation = {
      animationmaskfile = <path>
      animationtexturefile = <path>
      animationrotation = <float>
      animationlooping = <bool>
      animationtime = <float>
      animationdelay = <float>
      animationblendmode = <mode>
      animationtype = <type>
      animationrotationoffset = {
        x = <float>
        y = <float>
      }
      animationtexturescale = {
        x = <float>
        y = <float>
      }
      animationframes = {
        <int> <int> # etc
      }
    }
  }
}
```

name is the name you have given to the asset. This may be prefixed with various other segments, such as *idea* for idea images, etc.

texturefile is the path to the image, based of the /Hearts of Iron IV/, meaning you only include the folders within this folder, not the full filepath.

noOfFrames is used for images with multiple *frames* within, for example `GFX_unit_level` has five images within for each of the unit level images, and so would have `noOfFrames = 5` set within its asset definition.

effectFile is used to define an effect state to apply to the image, i.e. the button press effect on a button image. Effects are found in /Hearts of Iron IV/gfx/FX/*.lua. Multiple can be defined to add multiple effects.

allwaystransparent defines whether or not an image can be clicked on and interacted with. If yes, the image will have no tooltip and can not be clicked on when used in game.

legacy_lazy_load defines whether the image is lazy loaded.

transparencecheck defines whether the image is bound by its alpha channel with regards to player clicks. Otherwise the bounding box is the image dimensions.

animation scope is used to add animation to an image. It uses the following attributes:

animationmaskfile controls the mask image for the animation.

animationtexturefile controls the color image used for the animation.

animationrotation controls the rotation of the animation image. By default it is -90 clockwise.

animationlooping controls whether the animation loops.

animationtime controls the duration of the animation. It is measured in seconds.

animationdelay controls the delay between animation loops. It is measured in seconds.

animationblendmode controls the blend mode for the animation image. Can be *add*, *multiply* or *overlay*.

animationtype controls the type of animation. Can be *scrolling*, *rotating* or *pulsing*.

animationrotationoffset controls the *x* and *y* offsets for the rotation.

animationtexturescale controls the scale of the animation image.

animationframes sets the number of frames within the animation image.

Be wary that the maximum file size for a sprite .dds files is around 16MB

frameAnimatedSpriteType[edit | edit source]

A **frameAnimatedSpriteType** asset entry follows this format:

```
spriteTypes = {
  frameAnimatedSpriteType = {
    name = "GFX_<name>"
    texturefile = "<path>"
    noOfFrames = <int>
    effectFile = "<path>"

    animation_rate_fps = <int>
    looping = <bool>
    play_on_show = <bool>
    pause_on_loop = <float>

    allwaystransparent = <bool>
  }
}
```

name is the name you have given to the asset. This may be prefixed with various other segments, such as *idea* for idea images, etc.

texturefile is the path to the image, based of the /Hearts of Iron IV/, meaning you only include the folders within this folder, not the full filepath.

noOfFrames is used for images with multiple *frames* within, for example `GFX_unit_level` has five images within for each of the unit level images, and so would have `noOfFrames = 5` set within its asset definition.

effectFile is used to define an effect state to apply to the image, i.e. the button press effect on a button image. Effects are found in /Hearts of Iron IV/gfx/FX/*.lua. Multiple can be defined to add multiple effects.

animation_rate_fps is the rate at which the animation plays.

looping defines whether the animation loops.

play_on_show defines whether the animations starts playing when visible.

pause_on_loop defines the length of time to pause for after an animation loop.

allwaystransparent defines whether the image is bound by its alpha channel with regards to player clicks. Otherwise the bounding box is the image dimensions.

progressbartype[edit | edit source]

A **progressbartype** asset entry is used for progress bars. It follows this format:

```
spriteTypes = {
  progressbartype = {
    name = "GFX_<name>"
    textureFile1 = "<path>"
    textureFile2 = "<path>"
    color = { <r> <g> <b> }
    colortwo = { <r> <g> <b> }
    size = {
      x = <int>
      y = <int>
    }
    effectFile = <path>

    horizontal = <bool>
  }
}
```

name is the name you have given to the asset. This may be prefixed with various other segments, such as *idea* for idea images, etc.

textureFile1 is the image to use for the progress bar.

textureFile2 is the image to use for the background of the progress bar.

color defines the unprogressed color of the bar. It is decimal RGB.

colortwo defines the progressed color of the bar. It is decimal RGB.

size defines the size of the progress bar section in height and width.

effectFile is used to define an effect state to apply to the image, i.e. the button press effect on a button image. Effects are found in /Hearts of Iron IV/gfx/FX/*.lua. Multiple can be defined to add multiple effects.

horizontal defines whether the progress bar is horizontal or not. By default a bar is horizontal.

corneredTileSpriteType[edit | edit source]

A **corneredTileSpriteType** asset entry is used for tiling textures that tile to fill the size of the interface element they are used in. It follows this format:

```
spriteTypes = {
  corneredTileSpriteType = {
    name = "GFX_<name>"
    texturefile = "<path>"
    noOfFrames = <int>
    size = {
      x = <int>
      y = <int>
    }
    borderSize = {
      x = <int>
```

```
        y = <int>
      }
      effectFile = <path>

      allwaystransparent = <bool>
      tilingCenter = <bool>

      looping = <bool>
      animation_rate_spf = <int>
    }
}
```

name is the name you have given to the asset. This may be prefixed with various other segments, such as *idea* for idea images, etc.

texturefile is the path to the image, based of the /Hearts of Iron IV/, meaning you only include the folders within this folder, not the full filepath.

noOfFrames is used for images with multiple *frames* within, for example `GFX_unit_level` has five images within for each of the unit level images, and so would have `noOfFrames = 5` set within its asset definition.

size determines the size of the tile.

borderSize determines the size of the border.

effectFile is used to define an effect state to apply to the image, i.e. the button press effect on a button image. Effects are found in /Hearts of Iron IV/gfx/FX/*.lua. Multiple can be defined to add multiple effects.

allwaystransparent defines whether the image is bound by its alpha channel with regards to player clicks. Otherwise the bounding box is the image dimensions.

tilingCenter defines whether the center of the image is used for tiling, rather than the top left corner.

looping defines whether the animation loops.

animation_rate_spf defiens the framerate at which the animation plays.

maskedShieldType[edit | edit source]

A **maskedShieldType** asset entry is used for country shields. It follows this format:

```
spriteTypes = {
  corneredTileSpriteType = {
    name = "GFX_<name>"
    textureFile1 = <path>
    textureFile2 = <path>
    effectFile = <path
  }
}
```

name is the name you have given to the asset. This may be prefixed with various other segments, such as *idea* for idea images, etc.

textureFile1 is the image to use for the shield background

textureFile2 is the image to use for the shield mask.

effectFile is used to define an effect state to apply to the image, i.e. the button press effect on a button image. Effects are found in /Hearts of Iron IV/gfx/FX/*.lua. Multiple can be defined to add multiple effects.

Object Types[edit | edit source]

animatedmaptext[edit | edit source]

A **animatedmaptext** asset entry is used for text that appears from the map. It follows this format:

```
objectTypes = {
  animatedmaptext = {
    name = <name>
    speed = <float>
    textblock = {
      text = <string>
      color = { <r> <g> <b> }
      font = <path>
      position = {
        x = <int>
        y = <int>
      }
      size = {
        x = <int>
        y = <int>
      }
      format = <type>
      cull_distance = <float>f
    }
  }
}
```

name is the name of the map text. It is not possible to add custom ones.

speed is the fade time of the text.

textblock controls the attributes of the text that appears.

Font Types[edit | edit source]

textcolors[edit | edit source]

A **textcolors** entry defines the text colors used in localization with the § symbol. It follows this format:

```
bitmapfonts = {
  textcolors = {
    <symbol> = { <r> <g> <b> }
  }
}
```

symbol is the character used in the localization, i.e. **H**.

RGB is the color that the symbol applies. It is integer RGB.

bitmapfont[edit | edit source]

A **bitmapfont** entry defines a font used in the game. It follows this format:

```
bitmapfonts = {
  bitmapfont = {
    name = <name>
    path = <path>
    color = <hex>
    border_color = <hex>
    fontfiles = {
      <path>
      <path>
    }
    textcolors = {
      <symbol> = { <r> <g> <b> }
    }
    cursor_offset = {
      <int>
      <int>
    }
  }
}
```

name is the reference name of the font for other graphics files.

path is the path to the font. Fonts are found in /Hearts of Iron IV/gfx/fonts/.

color is the default color of the font. It is a hexadecimal RGB code.

border_color is the border color of the font. It is a hexadecimal RGB code.

fontfiles is used when there are multiple font images for one font, i.e. to support English and Cyrillic. This is used in place of **path**.

textcolors alters the global textcolors for the font it is nested within.

cursor_offset offsets the text cursor for the font. Defines the *x* position and then the *y* position.

bitmapfont_override[edit | edit source]

A **bitmapfont_override** entry is used to override a font for a specific locale. It follows this format:

```
bitmapfonts = {
  bitmapfont_override = {
    name = <name>
    fontfiles = {
      <path>
      <path>
    }
    languages = {
      <language>
      <language>
    }
  }
}
```

name is the name of the font to override.

fontfiles is used when their are multiple font images for one font, i.e. to support English and Cryllic. This is used inplace of **path**.

languages defines which languages this override applies for.