

## Changing the GUI of Ground Control II

This gives you a VERY brief introduction to the GUI-system of Ground Control II.

### The files that control the GUI are:

- directory.juice
- GUIs\GUI\_frontend.juice
- GUIs\ingame.juice
- GUIs\ingame\_slim.juice
- GUIs\mos.juice
- GUIs\multiplayer.juice
- GUIs\GUIskins.juice
- gametext.juice
- Units\unittypes.juice

#### **directory.juice**

The main repository for GCII and it also contains many GUI settings as well. It is scripted to use the files **GUIs\GUI\_frontend.juice**, **GUIs\ingame.juice**, **GUIs\ingame\_slim.juice**, **GUIs\mos.juice** and **GUIs\multiplayer.juice**, and these are the files containing the actual GUI screens. Look also in this file to change which prefabs are used for mouse cursors, and a lot of other things that aren't part of any specific GUI screen.

**ingame.juice** contains all in-game screens, and **ingame\_slim.juice** is a slimmed down version that allows you to see more of the game.

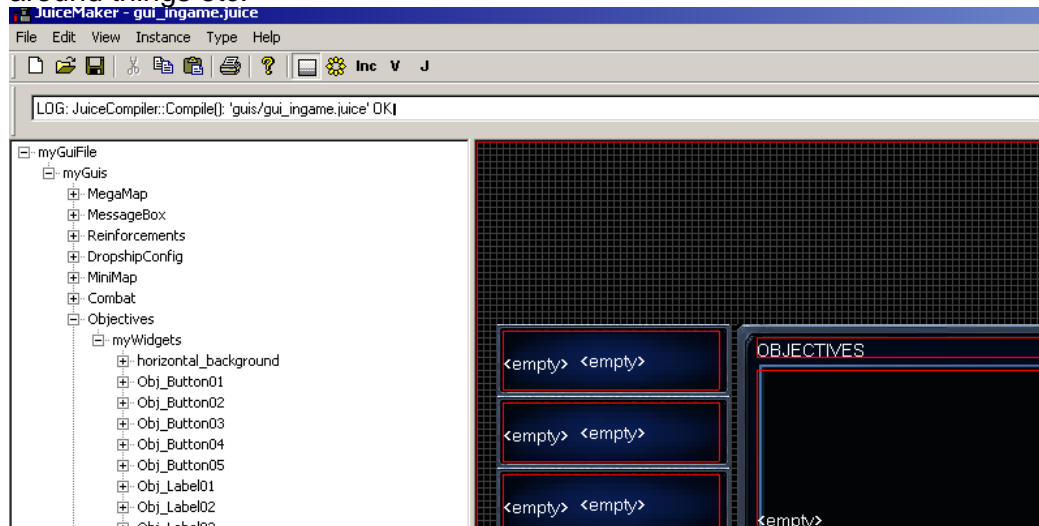
**GUIskins.juice** defines the graphics "prefabs" available in the GUI screens. It basically defines prefab names and coordinates into the graphics texture file used.

**gametext.juice** contain text strings used in the game.

**units\unittypes.juice** contains the unit definitions and specific secondary mode icons.

## How to change things

The juicemaker is the preferred tool to use when changing these files. When opening a GUI file, it enables you to edit the GUI's graphically, dragging around things etc.



If you open up **ingame.juice** you will see all the GUI screens that are used in-game. Note that several of these screens may be active at any one time. It's important to note that the ordering of GUI's, and also widgets within them, decide the drawing order! The first item (GUI or widget) will be drawn last and be on top of all the other items below.

If you right-click on the myWidgets text a small dialog will appear. Here you can re-order the widgets drawing order and also add new widgets. There are different widgets available and if you just want to add some decorating graphics, you will probably want to add an EXP\_Plate which is the simplest type of widget. This dialog will also appear if you right-click on the myGUIs text, and will enable you to reorder the GUI's if for some reason, this is nessecary.

When you click on the name of a GUI-screen it will be laid out to the right. If you click on a widget name in the GUI, there will be drawn a green square around it in the layout view so you can easily find it. You can also select a widget by right-clicking on it directly in the layout view. When a widget is selected you can drag it around using the left mousebutton. If you drag with the left mousebutton outside of a selected widget you will resize it. When moving or resizing a widget with the mouse it will snap the visible grid. The gridsize can be changed by pressing + and - on the numpad. You can also enter values manually by right-clicking on a value in the tree view.

While a widget has several attributes, I will just mention a few:

### **MySkinfile**

This points to the file defining what graphics prefabs are available. Note that it points to a .ice file, not the .juice file.

### **MySkin**

This defines the set of prefabs within the skinfile to use.

### **MyIgnoreEvents**

Specifies that the widget shall ignore all mouse events, and thus not be clickable in any way (all clicks on it may be caught by whatever widget or 3d lies below).

### **Bound to code**

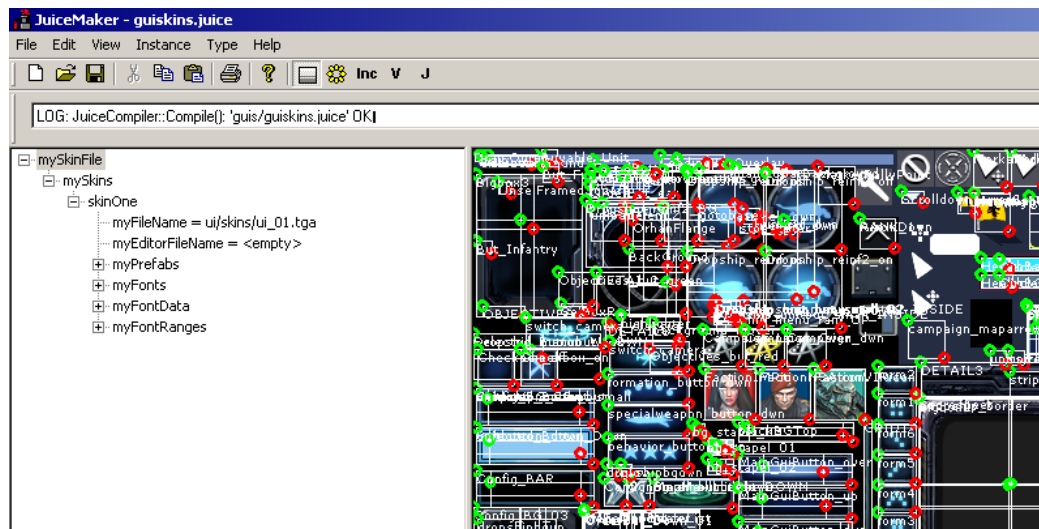
A lot of the widgets for any given screen will be referenced by the code, and so must not be deleted or renamed! Some widgets have their status (ENABLED/DISABLED/HIDDEN) changed by code, and so any changes made might be ignored in-game. If for some reason things don't act as you expect, chances are something is being controlled by code. Make sure to test often, and look in the debug output file!!! In the folder debug you will find the file `exodus_YOURCOMPUTERNAME_debug.txt`.

### **Reinforcements GUI**

In the Reinforcements screen you will notice that the buttons for ordering different units follow the naming Infantry00, Infantry01 etc... Here you can change the number of on-screen units by adding/removing buttons. As long as you keep the naming standard, the code will take care of detecting it! For example, if you want five infantry to be available at the same time select and copy one of the other buttons (Ctrl-C in the tree view), click on myWidgets and paste it in (Ctrl-V). Rename it to Infantry04 by selecting it and pressing Ctrl-R. Note that there also exists widgets with the naming InfantryFG00 etc. This means that you must also duplicate/remove this in order for the game to work as there must be an equal amount present! You must also make sure you have an equal amount of widgets for the other types of buyable units. Keep this in mind whenever you change something with numbers in the name!

I know this is messy and it is hard (impossible) to know when the code uses something and when it doesn't. This is where the debug output comes in. In the folder debug you will find the file `exodus_YOURCOMPUTERNAME_debug.txt`. This file is your friend. All errors are placed in this file and GUI errors have the name of the offending widget ID. Don't panic when you open it and see errors like the game not finding Infantry04 by default as this is related to how it auto-detects the amount of widgets in the GUI (It enumerates until it fails).

## GUI skins



Opening the file GUIskins.juice in the Juicemaker reveals the texture used for the GUI. It has a lot of prefabs defined that are the graphics available for the GUI widgets.

**Warning:** do not click in the layout view unless you have already selected the desired prefab to work on! Otherwise you won't know what you are changing!! You cannot select a prefab in the layout view, ONLY in the tree view!

myfileName decided which texture to use. When .tga file is specified, the game will automatically look for a .dds file of the same name for use in-game. A prefab has a start and end coordinate into the texture. It also has an inner start and end coordinate. This is to enable scaling to different screen resolutions. With the inner coordinates you define the corners and framing of the prefab. This will not be scaled up as resolution changes. The area within will be sized/stretched so the widget will have the same size in all screen resolutions. If you want the whole graphic to scale (no borders/frames), set the inner coordinates to the same value as the outer!

You can change the coordinates of a selected prefab by click-dragging in the layout view. This requires some practice though. Click-drag near the blob/corner you want to change. By pressing space you switch between three modes: DEFINE, OUTER, and INNER. This decides which blobs/corners/coordinates you can change. DEFINE is special in that it allows you to “redraw” the area/coordinates of the prefab from scratch. Inner and outer coordinates will be set to the same values.

## Fonts

Fonts and font-sizes are specified in GUIskins.juice.

## Important notes on using a slim GUI

When you check the “Use slim GUI” checkbox in the options in the game the slim version of the in-game GUI screen file is used (ingame\_slim.juice). The

game renders the 3d in full-screen mode and slimmer versions of the GUI graphics textures for the game are used. These graphics files are specified in directory.juice under

**ExodusData\myPlayState\myPlayer\myGUIskinTextures.**

If you don't want to support both normal and slim GUI's, you can just specify the same files for normal and slim. Just keep in mind that the game doesn't render the 3d-view in the lower 19% of the screen when the slim checkbox is unchecked making it really ugly if you don't have any graphics there!

### **Secondary mode icons**

These are specified directly in the units in Units\unittypes.juice.

### **That's it**

Good luck with your mod!