# Fugui Font Preparator

## 1 – Workflow

Main soft is ‘FuguiFontPreparator’, It take some file as inputs, you can find them into ‘**inputs’** folder.

- **fontData.txt**
  - Height (height in em of each glyphs in the generated font)
  - csName (name of the Icons.cs file and class)
  - emSize (em font size (according to DPI)
  - baseline (baseline height percent from emSize)

- **iconList.txt**
  - List of all icons you want in your generated font
  - Each line must be as follow (without space)
    - Row 1 : name of the glyph in the fontawesome or custom source (You can find names on fontawesome website or in your custom file)
    - Row 2 : font type of the icon, can be stacked with ‘,’ (Fontawesome have following types : solid, regular, light, thin and duotone). If you want to take an icon from a custom file, juste set type as ‘custom’
    - Row 3 : name of the Icon in the generated Icons.cs file (C# class)
    - Row 4+: whatever you want if you need to add commentaries

- **fontAwesomeIcons.json**
  - do not edit this file, it’s a fontawesome pro generated file. You can find it on github to update fontawesome version if needed.

- **customIcons.json**
  - It’s a IcoMoon.io/app generated json file. You can create your own using icoMoon app. Whenever your font is ready, just export it as json (burger-menu / Download JSON).
  - Remember to edit each glyph of your custom font to add names on your glyphs.

Whenever your input files are ready, just run FuguiFontPreparator.exe. it will generate 2 files into ‘**output’** folder.

- **Fugui_icomoon.json**
  - It contains data that icomoon.io/app need to generate a font.
  - Go to icomoon.io/app, make sure to remove any set (burger-menu/remove set).
  - Click on Import Icons button (on top of the app) and select this file.
  - To export the font, just click on ‘Generate Font’ (on the bottom of the app), click again on it (on ‘download’ button).
  - You just downloaded a zip that contain a folder ‘**fonts’.** Into this folder, copy the **icomoon.ttf** file and past it into **StreamingAssets/Fugui/font/current/icons.ttf** (rename it)

- **Icons.cs**
    - This file a C# class compatible with fugue
    - Just copy it content into the one of your project.
    - Whenever you want to use an Icon on your project, just draw a text (eg : layout.Button / layout.Text, etx…) and use it like : layout.Text(“my text” + Icons.MyIcon);

# 2 – Limitations

Fugui handle Duotone icons. To do so, it need to draw 2 glyph at the same position.

You can use duotone almost everywhere except :

- Menu Item / Contextmenu Items :
    - It will only work if the duotone icon is the first char to draw (otherwise it will missdraw the secondary glyph
- Window Name :
    - Duotone does not work on window names (basically it does not work on any text NOT draw by Fugui)

# 3 – IcoMoonAutoPreparator

If you use custom icons, you may notice that generated icons on your font has wrong sizes.

IcoMoonAutoPreparator.exe if here to handle it for you.

Basicaly it’s a macro automate that will click for you on icomoon.io/app UI to center and scale your glyphs.

When you launch it, you can see there is two textbox (left is input, right is output).

You can create your own macro or use the one for icoMoon already build in.

- Icomoon macro :
    - Whenever a macro start by ‘g:’, it’s a icomoon macro. It only need 4 parameters
        - Number of glyphs to resize
        - Speed (in millisecond) between two clicks
        - Number of scale out to do after resizing icons (after resizing, you may notice icons are smaller that fontawesome once)
        - Number of scale in to do after resizing icons (after resizing, you may notice icons are smaller that fontawesome once)
    - Open icomoon and load the fugui_icomoon.json file
    - Click on the edition tool (the ped on top bar)
    - Scroll to your custom icons range and click on the first
    - Count how many icons you have to resize and give it as parameter to the macro line (see up there)
    - Click on the glyph to enter edit mode (a popup will show).
    - If you have multiple screens, the icomoon.io app must be on first and fullscreen (you may test with only one iteration first)

  - Place the IcoMoonAutoPreparator window NOT on the icomoon popup and just click ‘Xecute macro’ button
  - Here you are, icons are nice now! (Maybe. Depending on how a monkey you are, or you luck.)
  - You can export you font on past ttf file into unity’s streaming asset (see path on first chapter of this doc).
- Custom macro :
  - You can add your own custom macro:
    - If macro input does not start by ‘g:’, it will parse and execute macro.
    - Each line will be an instruction
      - Set mouse position : ‘sp:Xpos:Ypos’
      - Wait : ‘w:msDuration’
      - Click : ‘c:Xpos,Ypos’
    - Exemple of custom macro :

      *sp:500:500*  
      *w:50*  
      *c:500:500*

      This will place mouse on pos 500;500  
      Wait 50 ms  
      Click on pos 500;500