Pyxel Morph Manual

Hello and Welcome to Pyxel Morph! Pyxel Morph is a software for graphic design in game development, that given an input image, outputs a green shaded preview, that when colored following the mapping instructions, gives exactly, or approximately, the image given in input. This tool was originally intended for speeding the process of scene generation in gbstudio, where the user disposes of 8 palettes per scene, and can change palettes between different scenes, but could be used for any game development platform where there is a limit on the maximum number of 4 color palettes a scene can use.

Limitations:

"free": {"max_palettes": 5, "max_input_colors": 10, "can_complex": False},

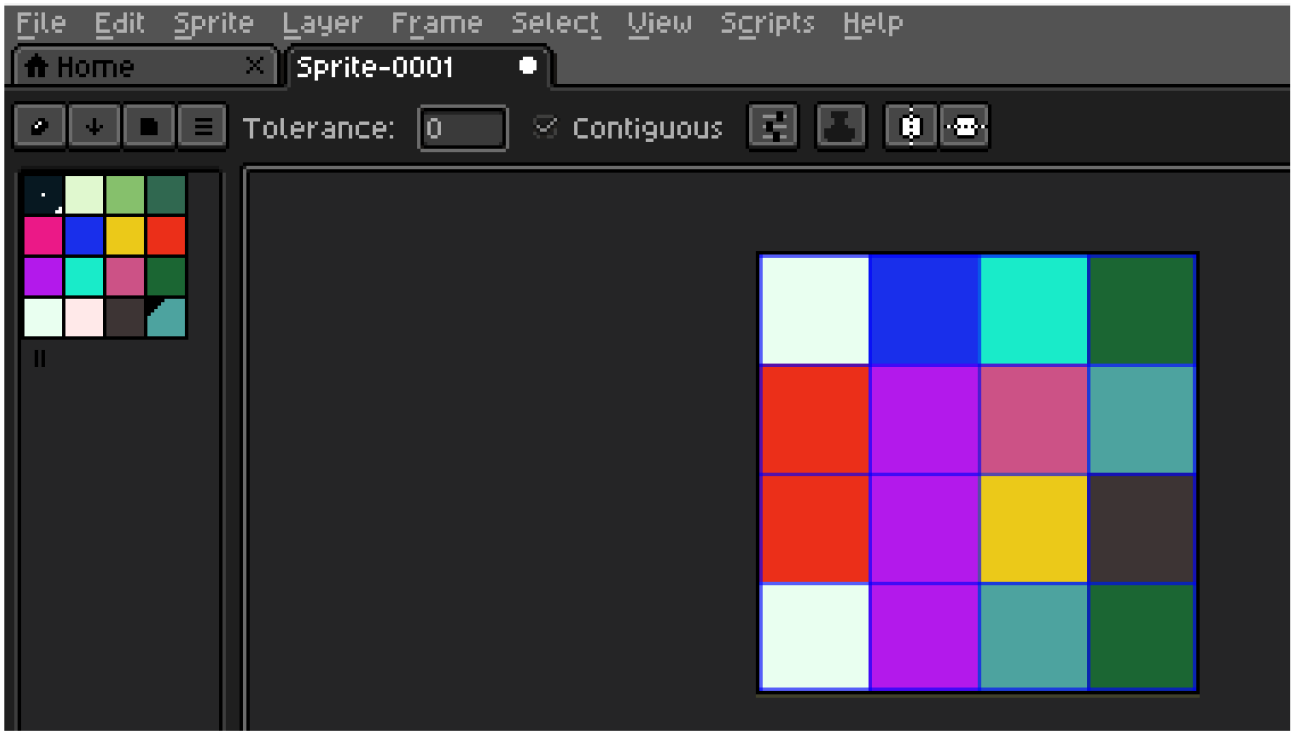
"pro": {"max_palettes": 7, "max_input_colors": 11, "can_complex": True}, COST: €1.99/month

"diamond": {"max_palettes": 999, "max_input_colors": 999, "can_complex": True}

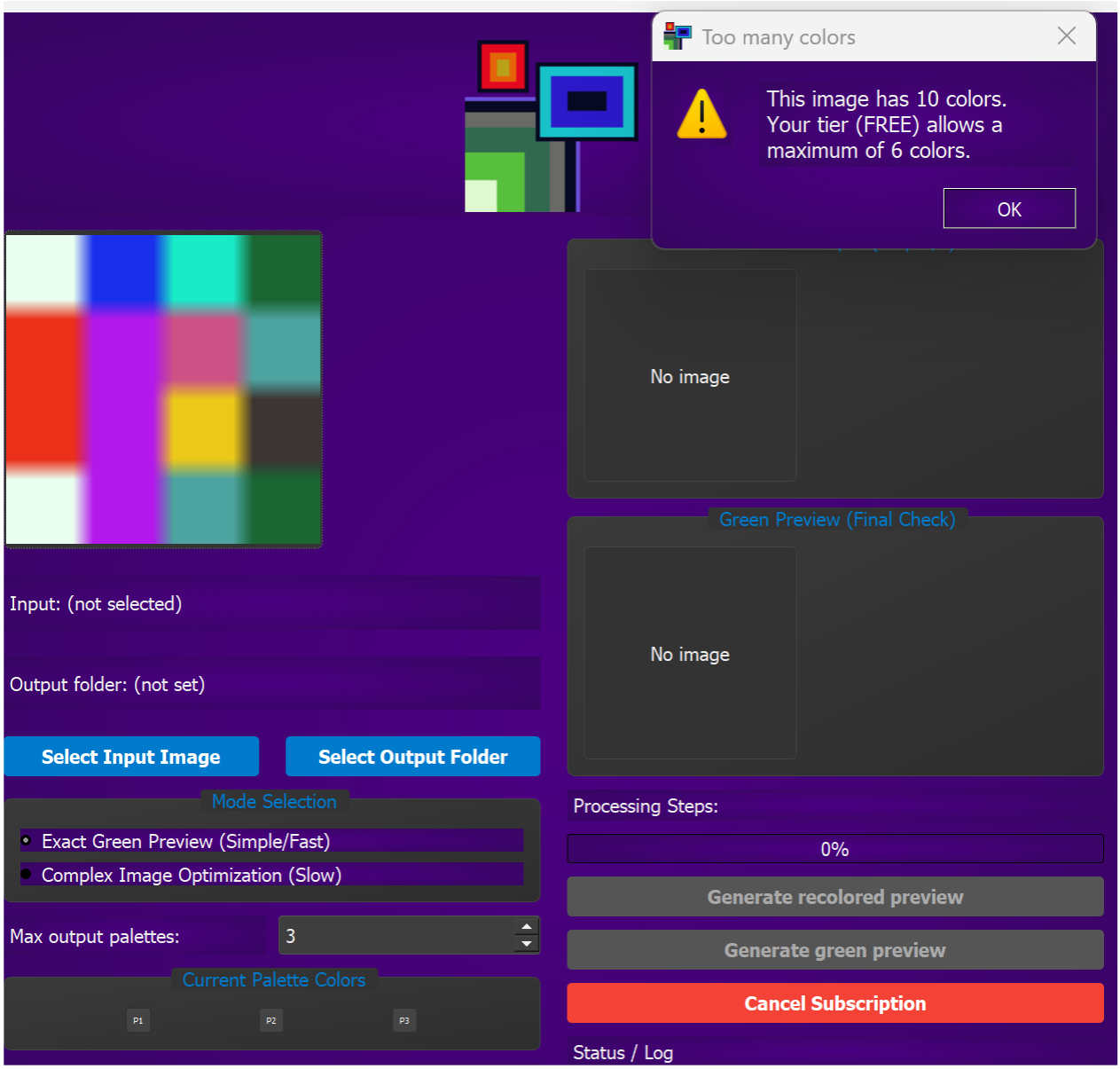
COST: €2.50/month

For all licenses: maximum 2 priority colors, only 8x8 tiles, image must be representable only through combinations of 8x8 tiles (if one side is 12 pixels, image cannot be computed by the app; we are working on adding a 3rd priority color for diamond.

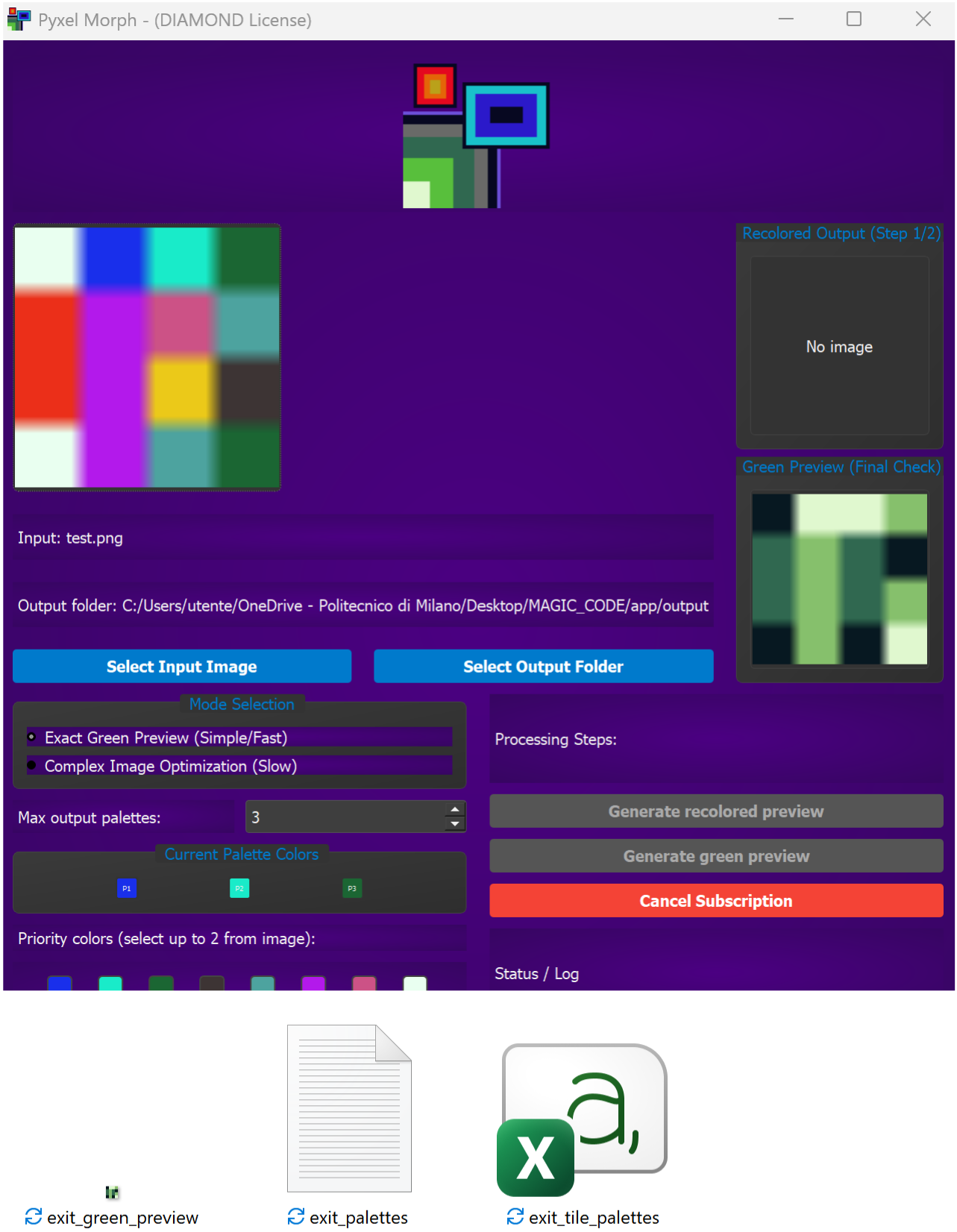
To cancel subscription in following month (auto billing after end of month) click the cancel subscription button, and you will be redirected to a page where by pressing once again cancel subscription, you will not be billed in following month, and will terminate your license 30 days from start of subscription. If this were not the case, please contact us, and we will do what we can to refund or adjust subscription details accordingly.



In the screenshot above the input image seems fuzzy: do not worry, that is simply how png displays images with few pixels; your original image is exactly as in the previous image. Because the original image contains more than 6 colors, it is not possible to convert with a free license. Therefore the user must upgrade to a pro or diamond license to be able to process this image in particular. Once updated to a higher tier, diamond in this case, the image can be processed, by directly pressing the button generate green preview in exact green preview mode. As shown in the following screenshot the image displayed at end of process is the input image expressed in green shades, that must be filled on whichever platform one uses for game development using the files released in the output folder selected at start of process.



In the screenshot above the input image seems fuzzy: do not worry, that is simply how png displays images with few pixels; your original image is exactly as in the previous image. Because the original image contains more than 6 colors, it is not possible to convert with a free license. Therefore the user must upgrade to a pro or diamond license to be able to process this image in particular. Once updated to a higher tier, diamond in this case, the image can be processed, by directly pressing the button generate green preview in exact green preview mode. As shown in the following screenshot the image displayed at end of process is the input image expressed in green shades, that must be filled on whichever platform one uses for game development using the files released in the output folder selected at start of process.



As one can see, in the output folder there are 3 or 4 files: the green preview png file (which is displayed inside the app as well), the palettes json file (which contains palette information),

and the tile palettes mapping csv file (which associates the palette indexes to the tile positions throughout the image), and occasionally the recolored preview (which shows how the image should look after painting it with the palettes in correct order). The green preview must be uploaded to the game development platform, as is standard procedure for uploading scenes for game development. The palette file instead, contains the colors in hex value, where the 4 color values in each palettes from top to bottom, correspond to the 4 colors from first to fourth in the in-game palette; it is advised to copy and paste each color separately in each palette, to avoid glitches of game development software in finding nearest color available in the game development platform, which varies from software to software. However, for quick scene making, one can try to manually change the file suffix of the palettes file simply from .json to .gbsres for example, and move the file to the palette folder in-app; however it is not guaranteed to work, adjustments may have to be made for successful upload, depending on game development software: below a screenshot of the palette json file.


```
[
  [
    "#192FEB",
    "#B319EB",
    "#E9FFF0",
    "#EB2F19"
  ],
  [
    "#19EBC9",
    "#1B6633",
    "#4DA39F",
    "#CC5286"
  ],
  [
    "#1B6633",
    "#3D3434",
    "#4DA39F",
    "#EBC919"
  ]
]
```


Once the palette have been uploaded to the game development platform, the user can proceed to color the green scene following the instructions on the csv tile mapping file: in this file, each palette index, ranging from index 0 to index “set input palettes – 1” , is associated to an 8x8 tile in the scene; the tiles are represented both in pixel coordinates ((8,16), (0,48)) and in tile coordinates ((1,2),(0,6)) matching gb studio’s tile coordinate system to facilitate the filling procedure (tile 0,0 is at top left corner), that is still manual and time consuming, especially when using many palettes in a scene to render a more accurate or characteristic image.

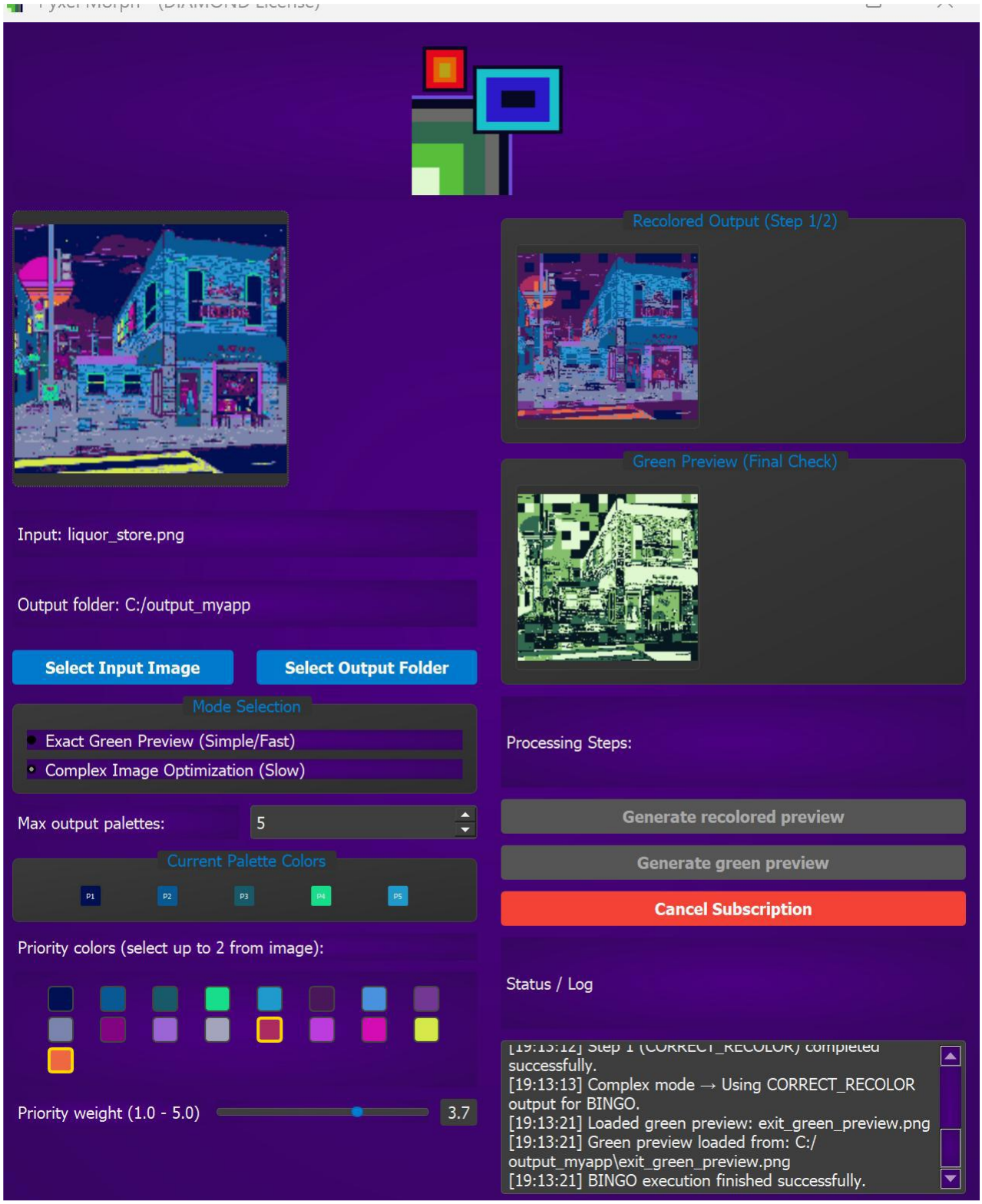
pixel_x	pixel_y	tile_x	tile_y	palette_index
0	0	0	0	0
8	0	1	0	1
0	8	0	1	0
8	8	1	1	2

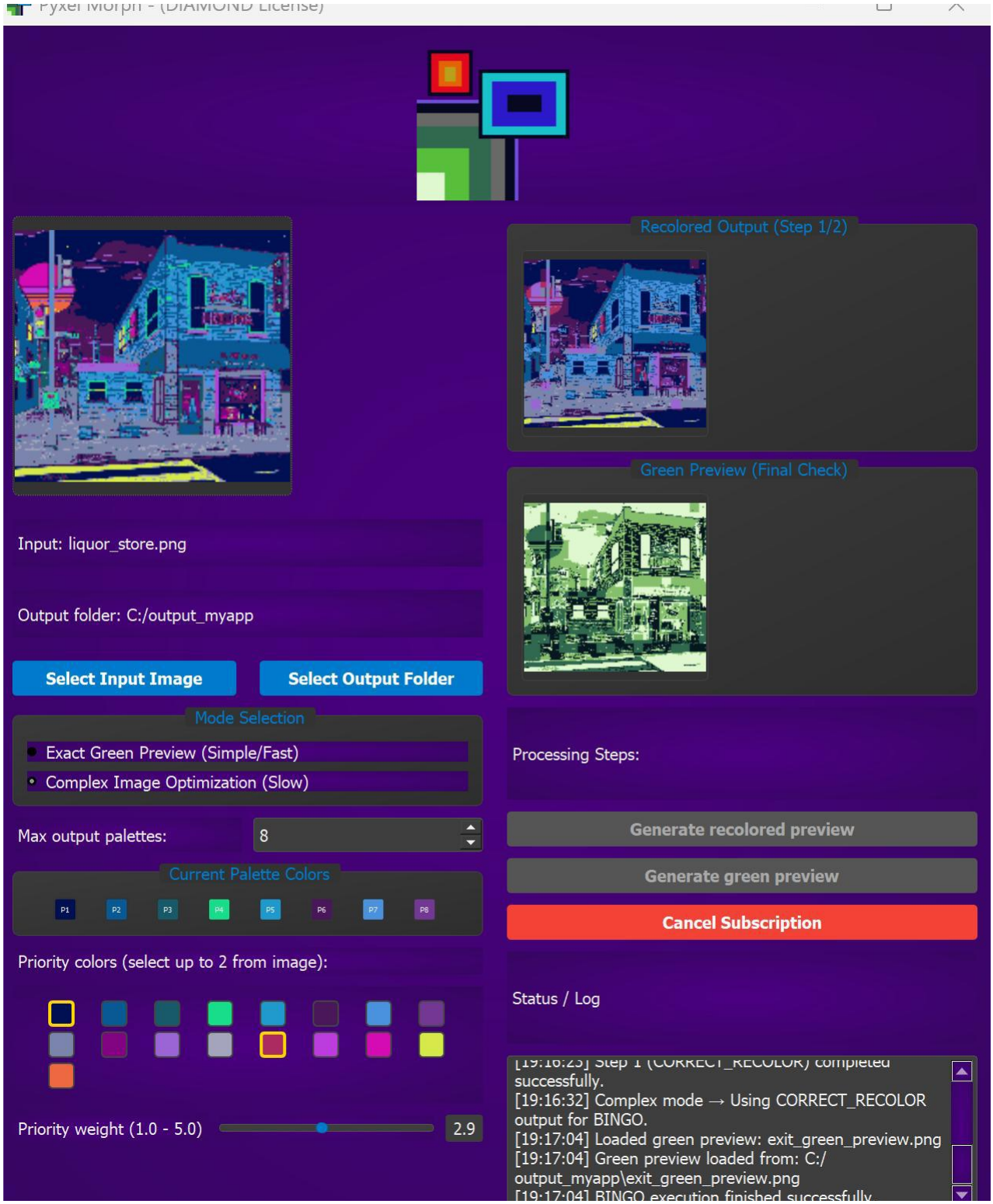
Important note: if one tried to generate an “exact green preview” for an image that for excess of colors or color randomness does not allow an exact green preview, the green preview would not be accurate at representing the input image when recolored, simply because you are forcing the mode to perform a function it is not meant to do.

COMPLEX IMAGE OPTIMIZATION (for PRO and DIAMOND users only)

Through this feature, user can recreate an image of high complexity (for example satellite image) on gba format, using extremely limited range of colors, but with outstanding similarities with respect to original image. The process is iterative and uses palette merging to achieve the optimal choice of palettes and color composition to recreate the closest thing to the actual image: infact, images with elevated number of palettes and extreme size could require a minute to compute; if this happens, please do not shut down the “non responsive” app that is simply performing the necessary iterations to generate your image. To perform this procedure, first set the mode from “exact green preview” to “complex image optimization”; next, select input image and output folder. Notice that in this mode one can’t go directly to green preview generation, as the meaning of complex image optimization is that the image can’t be created with the number of palettes you preset. Once the generate recolored preview button is pressed, a recolored image preview will be generated and displayed in app interface; do NOT open output folder yet, as the process is still halfway. Afterwards, press the “generate green preview” button, and the software will load the green preview in the interface, and load all needed files in chosen output folder. Repainting procedure is same as for exact green preview mode. Note that (especially for user with diamond license) changing priority colors and weights can greatly influence the accuracy of your image, even with an extremely limited number of palettes. Note that image accuracy of reproduction increases with image size, and decreases with amount of colors in original image, therefore more colors in input image does not necessarily mean higher accuracy, on the contrary! Also note that the recolored preview is not exactly what you will see when you finish coloring the green preview, this depends from initial image complexity and color range compatibility with your game development software; usually accuracy is between 80% and 100%, but could plummet to 60% in extreme cases; if the image reached through complex image optimization does not reach your standards, please try changing priority colors and weights, and if that doesn’t work, please contact pixel morph, so that we can improve the functionality of the app

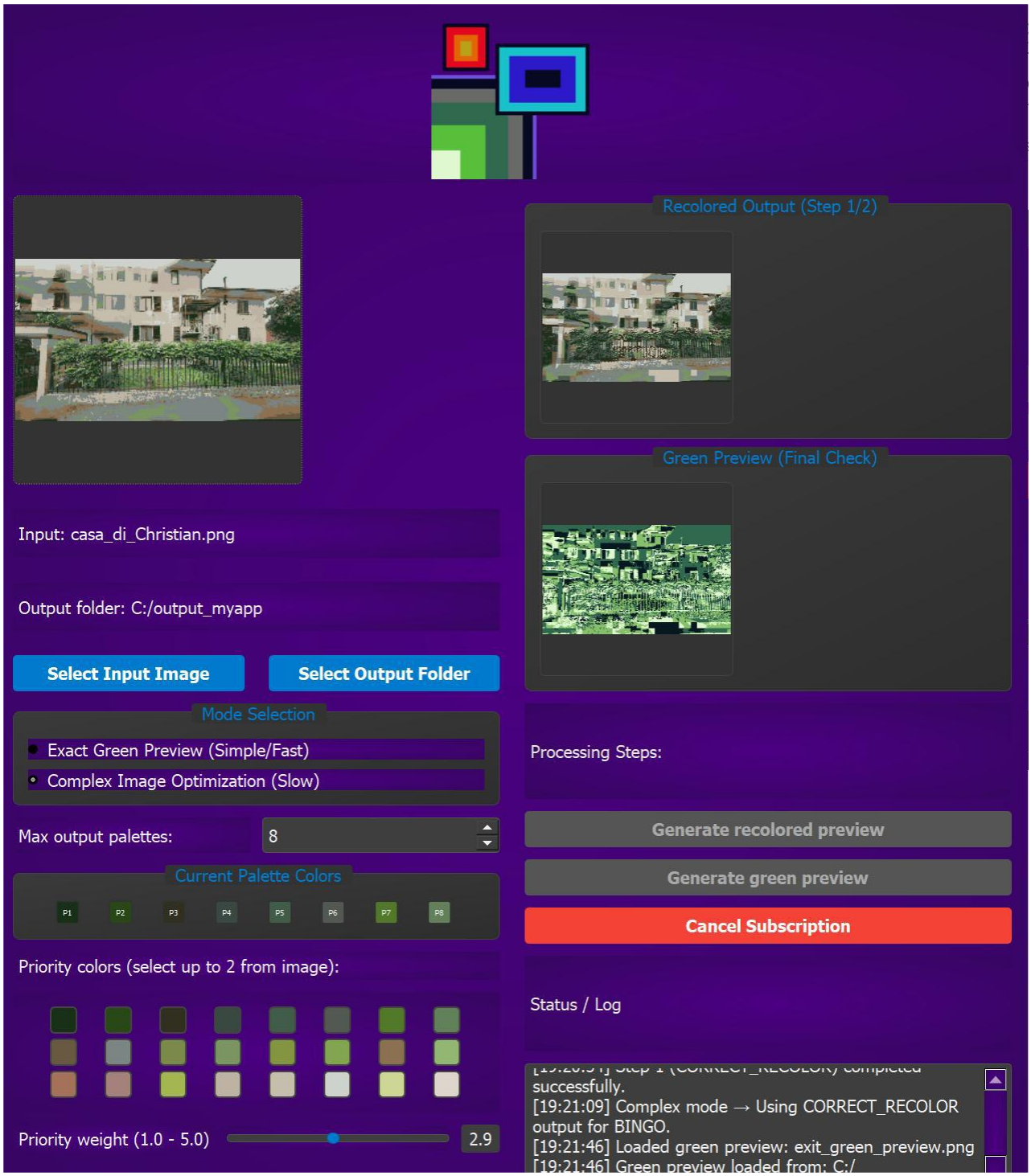
NOTE: images with 8 palettes of game boy screen size already require 10 seconds to generate green preview; if app is not responding after you press the generation button for high number of palettes, don't panik! Your computer is simply executing all the needed iterations.





Note: it is true there exist other software that perform palette quantization, and I must say rilden does a good job in doing so; color priority can steer the computation towards an image that you might like most. If you think it's not worth just don't buy pro, but you will be limited to

the 5 palettes also in exact green image: I'm sure that if you upload the rilden image on here, it will generate correct mapping. The main and basically ONLY purpose to use Pyxel Morph, is to map palettes in image, all other functions are collateral; thank you for reading this manual!



Server metadata: to access your app interface simply register first with username and password, and then login with those credentials; due to slow server, if one has been upgrade to a superior license, it will take approximately 50 seconds for it to apply to your app upon

startup: we apologize for the inconvenience; when we get at least 2 pro subscriptions to the app, investments will be made on a quicker server to guarantee optimal user experience.

More advanced features will be given in later versions of the app: stay tuned!

For support send an email to contact.pyxelmorph@gmail.com and we will get to you shortly