8DTuber

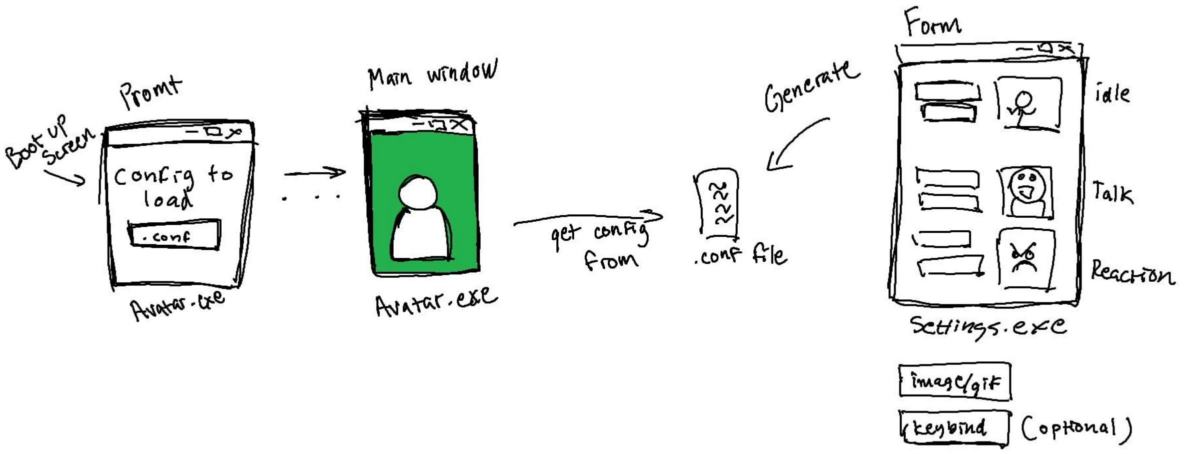


Illustration 1:
"Early preliminary concept art of the relation between windows and files. Also a preliminary preview of how the program will look."

Main window

Boot-up screen

The boot-up screen is mainly used to specify which config file to load when booting up the 2D sprite/avatar screen. This will allow the user to display multiple sprite/avatar windows at the same time. The windows can display different sprites depending on the config file the user decides to load. The boot-up screen will only read the local folder.

Alternatively, the boot-up screen might not be needed. You could just drag and drop the config file into the 2D-sprite/avatar screen.

2D sprite/avatar screen

This screen is where all the sprites are animated/displayed. As of writing this it's planned that it should have 2-3 layers. The 2nd layer is where the sprite/avatar is rendered. The layer above that, the 3thd layer, is where facial features, hats and such can be rendered. The first layer is the background. It's green by default to act like a green screen.

The main intention of this window is that OBS, or some other kind of window catching software, is going to catch this window and filter out the green screen. This is to make it so that the sprite will look like a transparent animated gif.

The Main Window reads the input levels from the microphone. Depending on the percentage (%), the talk animation will play. The percentage can be configured/defined in the Forms application or directly in the config file.

Form

The form is a seperate program designed to generate the config file used by the Main window. Here the user can drop gif/jpg/png in the various categories that the form displays. The categories are;

- Idle_pose *(layer 2)*
- talk_pose *(layer 2)*
- Custom_pose_1 *(layer 2)*
- Custom_pose_2 *(layer 2)*
- …
- Custom_pose_n *(layer 2)*
- Custom_pose_n+1 *(layer 2)*
- Cosmetic *(layer 3)*
- background *(layer 1)*

At the top of the program is an input-field which asks the user what the config file should be called.

The custom poses can be assigned to hotkeys and will override either the Idle_pose or the talk_pose. This is if the user wants the character to be able to portray different emotions. In the custom pose “menu”, the user can select whether it should replace the Idle_pose or the talk_pose.

For each row of the form there exists a preview window of how the sprite will look. To the left of that preview window are two input boxes. The first box is where you drag and drop the file into, or define the path to the file. The second box is where you input the hotkey for the sprite.

In the Custom_pose form area there are two radio buttons. They decide whether the pose should override the idle pose or the talk pose. For those who don't know what a radio button is, an example is shown below; Only one can be selected at once. By default the override for the idle pose is selected.

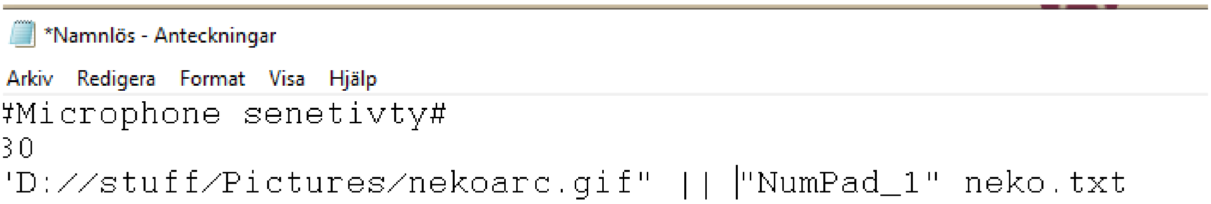
○ HTML
● CSS
○ JavaScript

When the form is filled and everything is done, the user can click the “done” button. The form will then generate a config file containing all the information needed for the Main Window to work.

Config file

The config file contains all the necessary data for the Main Window to display what the end user wants it to display. The file itself is very simple. The Main Window parses the information in this configuration file.

The config file contains the file paths and hotkeys assigned to the image files. Also the input sensitivity level of the microphone which decides when the talking animation will play.



```
#Microphone senetivty#
30
'D://stuff/Pictures/nekoarc.gif" || "NumPad_1" neko.txt
```

Illustration 2:
Example of how a config file looks, kinda.

Tillvägagång

Road map:

- Create the main window

    - Make a light version which displays a hard-coded avatar.
- When that’s done, create a draft of how the config file should look
- After that, create the form.
- After, you’re done! Heat up a pizza.
- And after that… who knows…