

Pixel Intelligence Part 3 Reflections

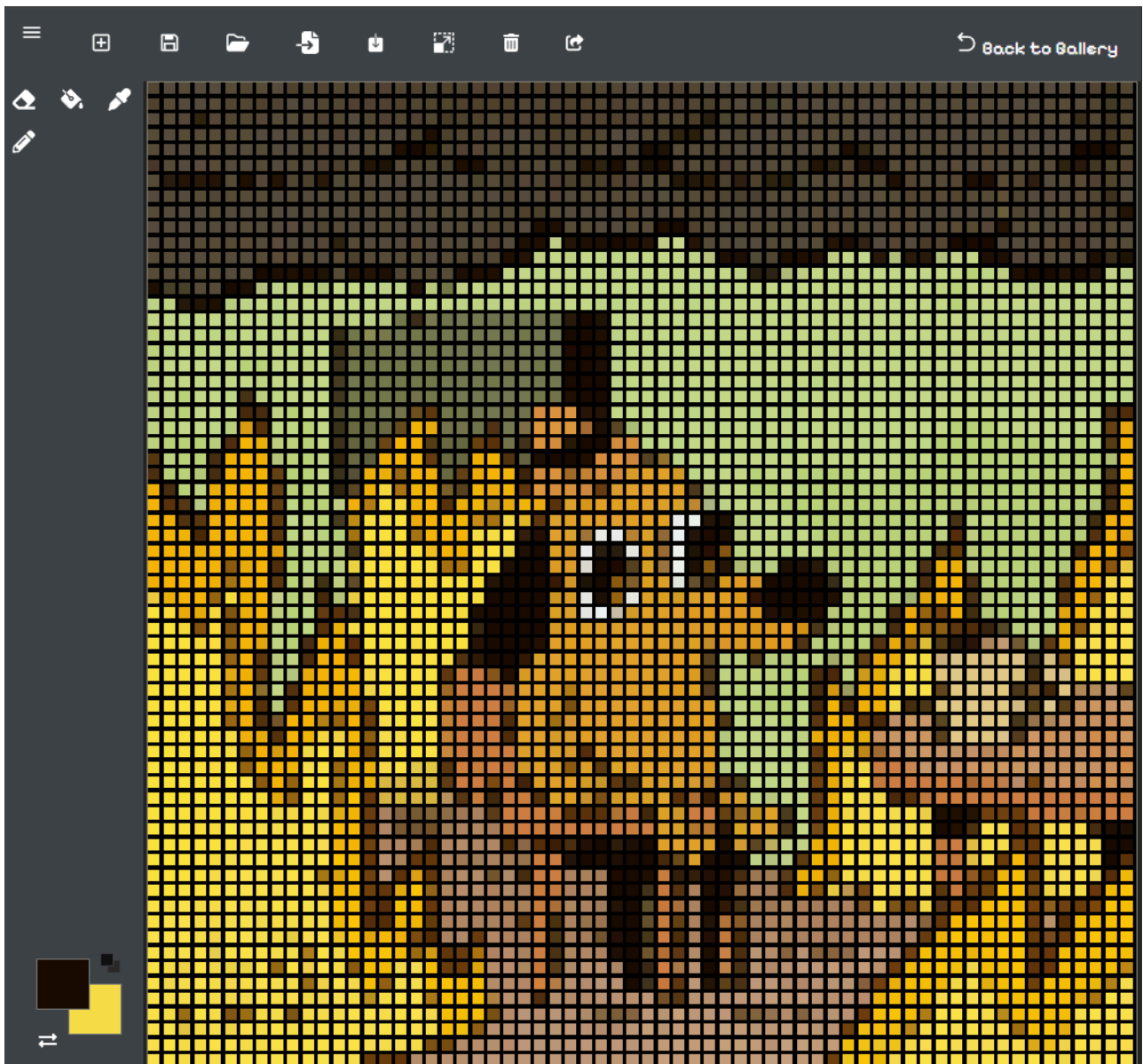


A Pixel Art Editor with generative AI functionality, image downscale processing, image export, and a showcase gallery. Comp229 Group Project (Fall 2024).

Latest live demo can be viewed on [Netlify](#)

Group Members:

- Connor "[ConnorBP](#)" Postma (301005412)
- Yoon "[superYM222](#)" Min (301317593)
- Sanjeevkumar "[sanjeev-cs](#)" Chauhan (301480021)
- Parmila "[Parmilashams](#)" Shams (301426195)



Connor "ConnorBP" Postma (301005412)

Changes to original concept:

We pivoted which API to use from polinations ai to horde ai for its more flexible api, and better image generation results for our purposes of pixel art generation. We also decided that storing "actual" images on the database would be redundant, so instead the json project files themselves are all that is stored, and image representations are generated on the fly and cached.

My Experience:

During this section I spent a lot of time doing code review, tweaking stable diffusion settings to get wicked pixel art, setting up an image generation node on my gaming pc, messing with broken Netlify deployments, and hunting for crazy obscure javascripty bugs or *repo regressions* 🤔. I also went through all the extended functionalities on the editor and canvas backends so that all editor context options work, project loading and saving works, image import and export works, the k-means clustering based scaling algorithm

works, transparency is supported (though not in the brushes yet), and the frontend can communicate with the backend while caching gallery results to avoid api spam. I was able to re-use a lot of the code I created for rendering the pixel art images to canvases for the gallery display. Using some html canvas trickery, I can turn my array of html color codes into a png data URI in an instant (which i guess kind of counts as having an encoded share url).

This project has been an interesting learning experience with the challenges that come with working in a team of varying technical experience. With hindsight, I may have suggested a project idea that was a wee bit complex in some parts. I was fully aware the whole time that I may have to complete certain tasks that might be a bit too intensive for newer coders however, and tasks were distributed with that in mind. Even with that, I probably gave my group quite a head scratcher though hehe... We managed to get a fully working app though, and some stretch goals like image import, extended metadata, and downscaling or upscaling projects in the editor as you work.

Currently missing

We didn't end up completely finishing the image generation connection to the frontend by the 7th of December 2024, but it will definitely be done this week for presentations. Many of the stretch goals proved to be not attainable at this time, as there is already so much to do with the nearly fully featured image editor and gallery system. Other goals changed into different things as the program matured as well. Some things I still want to add though include: Clone to editor from gallery (this will be easy due to the pivot using project files already there). As for file formats, we only have png and json right now but I would like to have svg as well if we can. The stretch goal i'm most sad about is undo-redo transaction support. It would probably be a fairly large refactor, but maybe it would work well with a useReducer hook.

Sanjeevkumar "sanjeev-cs" Chauhan (301480021)

In part 3 of this project the gallery page needed to efficiently handle large numbers of images. In this regard, I implemented a pagination API so that images are loaded in manageable chunks, hence improving performance and usability. This API allows the frontend to request a specific page of images from the backend, hence reducing the load on the server and improving the overall user experience. I designed the API to carefully accept parameters, such as the current page number and the number of images per page, to be flexible in the way images could be displayed. Then the backend processes these requests for a set of images to be returned and displays them on the gallery page.

I'm also working on the development of the Image Generation API, which is supposed to create a pixel art image according to the input or prompts of users. I found it a bit complicated task in the starting but after doing some google search and watching some tutorials on YouTube I think that I can do it finally.

Overall, the progress of the Pixel Intelligence project has been great, and everything is going as planned. All team members are contributing to the project, thus enabling the efficient completion of all tasks at the right time. The collaboration has been smooth, with everyone working to achieve our goals in this project.

Yoon "superYM222" Min (301317593)

Technical Challenges

I am working on the left toolbar tools section.

I have faced many errors. Even though Conner set up a canvas background algorithm, It is challenging to pass the color from Canvas.jsx to ColorPickerToolbar.jsx via props.

However, I am learning how to use gitHub and pixel algorithm.

The reason is that I also made a mistake gitHub PR...

Parmila "Parmilashams" Shams (301426195)

I made some changes to improve the functionality and usability of the application. On the gallery page, I added new components to display additional details about images, such as their titles and descriptions, the gives users more context and making the gallery more informative. Also for make navigation easier, I added a navbar to the main page, updated its layout, and applied styling to make it visually appealing and consistent with the rest of the app. I added a Share Popup feature to the editor page, allowing users to share their images directly to the gallery, making the process more seamless and user-friendly. I also made some minor adjustments, like removing the logo and cleaning up parts of the code to ensure everything runs smoothly. Finally, I'd like to thank Connor for his collaboration and support in addressing these issues, I appreciate it. His help was invaluable in completing this work in the group project.