

Note, the first 3 weeks consisted of finding a supervisor and getting a project approved. I wasn't completely aware of the daily and weekly logs yet. As such, the exact numbers of hours spent during these weeks are unknown.

Week 1, Monday 2/1/21 to Sunday 2/7/21 - ? hours

- Contacted my computer graphics professor, Halevi, via email about CISC 4900
 - Informed them of my project idea to create a voxel editor
- Spoke with professor Halevi over zoom after class on the 2nd and 4th about CISC 4900
 - She informed me that they are not too certain how CISC 4900 will operate this semester and would like for me to have my project approved before committing to be my supervisor
- Sat in on info-session about CISC 4900
- Spent time studying the GUI framework Qt and combining it with OpenGL
 - At this point in time, the initial idea for my project was to use C++, Qt, and OpenGL to create a voxel editor
- Began drafting project proposal

Week 2, Monday 2/8/21 to Sunday 2/14/21 - ? hours

- Finished project proposal and submitted for approval
- Continued to study Qt, OpenGL, and C++
 - Managed to get Qt to render an OpenGL context and have various sliders and buttons interact with it
 - Created a simple program to rotate a triangle and change the colors of its vertices
 - Began learning how to produce voxels in OpenGL
- Continued to speak with professor Halevi and informed them that I am waiting for project approval

Week 3, Monday 2/15/21 to Sunday 2/21/21 - ? hours

- Began working on final draft of project proposal
 - I have not heard back yet about project approval but felt it best to begin working on it anyways
 - Ended up scrapping the initial final draft and utilizing the template given later in the week
 - New project proposal largely different than the last due to a new tech stack
 - The goal of building a voxel editor remains the same though
- Spoke with professor Halevi and decided to change the tech stack behind the voxel editor
 - Initially, the program was going to consist of Qt, OpenGL, and C++
 - Instead, the program will make use of JavaScript and React so as to better align with Brooklyn College's computer graphics class
- Began studying Three.js and researching web-based alternatives to Qt and OpenGL
 - Instead of OpenGL, Three.js is a perfect replacement and even a lot easier to use

- For the GUI, I decided to utilize React

Week 4, Monday 2/22/21 to Sunday 2/28/21 - 15 hours

- Double checked final draft of project proposal and submitted it
 - The creation of the draft perhaps took way longer than it should have and ended up being 26 pages long
- Began both the daily and weekly logs
 - I had assumed I was supposed to begin these after the project is approved and the final draft submitted
 - Just in case, I took the time to recap the previous three weeks
- Began studying React
 - Finished [this](#) crash course tutorial
 - Began [this](#) React tutorial on Udemy
 - Only the first 16 hours of the tutorial are needed
 - Managed to finish 10 of the 15 sections over the weekend
 - Skipped section 11 as it is a review of the previous sections
 - Will finish the course early next week
 - Took some time to plan the architecture of Voxel Edit with React's component system

Week 5, Monday 3/01/21 to Sunday 3/07/21 - 22 hours

- Continued working through [this](#) React tutorial on Udemy
 - As of March 2nd, all needed tutorials have been completed.
- Began work on basic-cube-edit project to learn how to combine React and Three.js
 - Devised plans for the project's React component architecture and drew design with the painting program Krita
 - Architecture consists of a Viewport component for handling the 3D scene and it's surrounding UI and a ColorSelect component for updating the color of the cubes
 - Included [semantic-ui-react](#) to speed up development time with CSS and common React components such as modals
 - Added [OrbitControls](#) from Three.js
 - Also, included [three](#) via npm
 - Added lighting and phong material
 - Added [React Color](#) to ease creation of color pickers
 - Tested the program via browsers on desktops/laptops and phone browsers
 - Works on PC
 - Works on iPhone and Android devices as well
 - Researched documenting JavaScript code
 - Documentation is generated as a website with [JSDoc](#)
 - Deployed the application and it's documentation via [Netlify](#)
 - Considered GitHub pages but Netlify will automatically run build scripts
 - Not sure if GitHub can also do this but Netlify will do for now
 - Live [basic-cube-edit](#) website

- Live [documentation](#) website

Week 6, Monday 3/08/21 to Sunday 3/14/21 - 17 hours

- Now that there is a sufficient understanding of React and Three.js, we can begin the actual voxel editor
 - To learn how to implement voxels with Three.js, [this](#) tutorial will be completed
- Created a test repo called voxel-tutorial and began following the aforementioned Three.js voxel tutorial
 - The majority of the week was spent following this tutorial and studying/documenting what everything does
- Upon completing the tutorial, I learned how to:
 - Add voxels
 - Remove voxels
 - Choose a texture for a voxel
 - Optimize rendering voxels to the screen
 - Used an algorithm that takes an array representing all of the voxels within a cell (a single chunk of the voxel world) and turns it into a *single* mesh
 - For a 32x32x32 cell, this means we only make a single draw call as opposed to 32,768 for each voxel!
 - Implement a ray casting algorithm to pick individual voxels adding/removing
 - Said algorithm is optimized for voxels and based on [this](#) research paper
 - Also took some time to print this paper and then annotate it so as to better understand how the algorithm works in case I need to modify it
- Updated the previous basic-cube-edit project's README.md with an installation guide for the project
 - Same instructions are being reused for voxel-edit
- Created the official voxel-edit repo!!!
 - It can be seen [here](#)
 - Also, decided to use GitHub Pages over Netlify for hosting the live site
 - This is done with the help of [gh-pages](#)
 - The live version of the site can be seen [here](#)
- Lastly, before I could begin work on writing the code for Voxel Edit I took a moment to create a rough draft of the UI
 - The rough draft can be seen [here](#)
 - This draft isn't final but it's a good idea of how to start structuring the application
 - The architecture folder of the voxel-edit repo shall be where application architecture and design related files go
 - It can be seen [here](#)
- Began making use of the voxel-edit repo's issues tab on GitHub to track what needs to be done

Week 7, Monday 3/15/21 to Sunday 3/21/21 - ~23 hours

- A lot more was done this week than expected

- This week and the next are midterm weeks. As such, I didn't expect to get much done due to studying
 - The project is at a point where the groundwork is in place and features can be added a lot more rapidly than previous weeks
- Started making heavy use of the issues tab for the project's GitHub page
 - Issues such as add favicon, noted bugs from iPhone users, etc.
- Drafted the React component architecture to meet the UI specifications from [this](#) sketch
 - The makeup of the React components can be seen [here](#)
- Added dev branch to keep main clean since it corresponds to the live website
- Started and finished writing code for the basic application skeleton
 - Wrote the GUIController code and made it responsive to desktop vs mobile devices
 - Managed to roughly replicate the UI specifications for both desktop and mobile from [this](#) picture
- Added supervisor as a collaborator to the official voxel-edit repo
- Had others test the software on their mobile devices to see if it works and that they can add voxels to the scene
 - iPhone 6 devices have a bug that prevents some touch events
 - iPhone 7 and beyond are fine
 - Android is fine as well
- Begun and finished the Brush component
 - Mobile and desktop users can be actively switch between adding, removing, and painting voxels
- Refactored Voxel.js module
 - Previously, all Three.js code was in a single Voxel.js file
 - The code has been split into various classes and made a lot more flexible
 - Brush class
 - ColorPalette class
 - Color class
 - VoxelEditor class
 - The major class instantiated by the VoxelManager React component that holds the main render loop
 - VoxelWorld class
 - The second most important class that handles the voxels themselves and generating the geometry for the scene
- Replaced the default scene with voxels in a sine-wave shape to just a flat ground
- Begun and finished the ColorPalette component
 - Instead of the previously used textures, the voxels now just appear with plain colors via BufferGeometry colors attribute
 - ColorPalette UI works on both desktop and mobile devices
 - Users can select a color and then add/paint voxels to be the color they selected
 - Users can have up to 128 unique colors on their palette to use
 - Choosing a new color for a pre-existing color on the palette will also update all voxels in the scene that had the old color with the new color

- Fixed a bug where the VoxelManager would destroy the scene when going from desktop to mobile UI and vice-versa
 - This is also rather useful for the same code can be reused to produce save files that the user use to resume their sessions
 - I also considered caching projects but the size of a single cell in the voxel world would take up too much space
- Lastly, I have more or less hit the first major milestone of the project!
 - A basic voxel editor that allows users to add/remove/paint voxels of whatever color they want regardless of platform
 - Next, the desktop UI needs to be polished a bit followed by eliminating bugs and adding additional features
 - Additional features such as being able to save projects and whatnot as detailed in the project proposal

Week 8, Monday 3/22/21 to Sunday 3/28/21 - 1 hour

- Didn't get anything done this week. Focused more on midterms and homework. At most, added a few issues to the GitHub repo.
 - I'm rather disappointed by my productivity this week regardless of how much precedence midterms should have. I wanted to get a lot more done than what I was able to do. Since next week is spring break, I hope to put in upwards 30 hours to make up for this.
- Despite not getting much done, I did showcase the now working voxel editor to my supervisor
 - I haven't been listing the time spent with my supervisor in my daily logs. I usually speak with her about the project after her graphics lectures on Tuesday and Thursdays.
 - I also got some feedback on the project from my supervisor and added it to my GitHub issues

Week 9, Monday 3/29/21 to Sunday 4/04/21 - 38 hours and 15 minutes

- Managed to complete a lot of different things this week
- As mentioned in week 8, I wasn't able to do much because of midterms...
 - However, I more than made up for it by putting in over 38 hours this week thanks to Spring break! A new record in productivity!
- Added many different issues to the project's GitHub that was either completed or is on the todo list of things that would be nice to have later on
 - Added issues include but aren't limited to:
 - Ray tracing for in browser rendering
 - Export Obj
 - File Size
 - Interim Report
 - Save Project
 - Render Scene
 - The list of issues currently being worked on can always be viewed [here](#)

- The first portion of the week was dedicated to the [Polish Desktop UI](#) issue
 - The UI for mobile devices was looking rather nice but desktop was really lacking
 - Much of what was done here was improving the overall look and feel of the desktop experience
 - A lot of time was spent experimenting with CSS and different semantic-ui-react components
- In addition to the Polish Desktop UI issue, the ColorPalette component has a major rework at the request of one of my testers
 - Originally, the ColorPalette consisted of a ton of grey squares that users could click on then adjust its color
 - This was overwhelming and confusing though!
 - Users were confused about how the color on the palette correlates the the scene
 - As such, only **one** color is displayed in a new project
 - That color is a green-ish color that is associated with the starting plane
 - In order to add new colors, users now click this plus symbol on the ColorPalette which then adds a new color they can work with!
 - This new color is a bright red which will change color upon moving the bottom slider of the color picker
 - My supervisor didn't like the grey color because you has to click twice to change its color so this fixes that issue
 - In addition, users only add as many colors as they need
 - No more overwhelming grey squares!
 - In my opinion, this was perhaps one of the best UI additions to the program yet as it greatly reduces the clutter on the screen
- One other nice addition to the desktop UI is the use of accordions to reduce screen clutter
 - Also, everything is nicely labeled with icons
- The mobile UI does feel as though its lacking a bit compared to the desktop UI now though
 - Unfortunately, the mobile UI just doesn't have the screen space for a more pleasant editing experience so some things won't be as nice
- Next, I implemented one of the most important features: **the ability to save your project!**
 - Previously, everytime you would revisit the site or refresh the page your hard work would be gone
 - All voxels, colors, etc would just disappear
 - Now, users have the option to click File->Save Project
 - Doing so will save their ColorPalette and placed voxels within the scene to a .json file
 - Said file the users can name themselves via an input!
 - After saving a project, a user can also work on it again by going File->Load Project
 - They will be prompted to supply a .json file with their projects data

- After loading, every single color and voxel they placed will be exactly the same and the user can continue working where they left off!
- I'm very happy to see this feature in place for now I can work on more complicated models and no longer be afraid of losing all my work
 - In addition, there's this sense of permanence that comes with the ability to save your work that makes the program feel a lot more *valid* if that makes sense
 - Users can also share their files with others to open as well whether on desktop or mobile
 - I found this to be really fun to take something I made and give it to someone else to inspect and play with instead of just giving them a static image of what I made
- After adding the save project stuff, I began work on the first draft of my interim report
 - The draft feels a little rough and will need some major changes as I've been rapidly adding more and more to Voxel Edit so it's quickly becoming out of date
 - The report I shall continue to tweak until it is due next week
- The next major addition is **rendering the scene to an image!**
 - Users can now position the scene however they wish and click the render dropdown and then export to image to save their works as a .png
 - However, this feature is still a bit lacking
 - It would be nice to specify screen dimensions and the like
 - Also better lighting or adding alpha to empty screen space
 - For now, rendering works and can be improved upon later as new features are added
- Lastly, I began work on the **Export Obj** issue
 - Whilst working on the interim report, I realized that the minimum viable product should have the ability to export models to other 3D file formats
 - I had completely forgotten that this was a part of the MVP and not just an extra feature!
 - Being able to take your work and use it on other projects is pretty important
 - Desktop applications are far more powerful than web so even I would like to model my voxel art in Voxel Edit but then take the result to a program like Blender for more advanced lighting and rendering
 - I was able to find some exporter code for Three.js
 - With it, I can now export to:
 - .obj
 - .ply
 - .stl
 - .dae
 - However, the exporter code does not include the color data of each vertex!!!
 - I can export the geometry of my models but **not** the color...
 - It appears that I will have to write the code for exporting color manually for each 3D file format

- This may be rather difficult and/or time consuming
 - As such, I'll just stick to .obj
- I might put this issue on the back burner as well to overcome some other low hanging fruit so to speak within the project
 - Basic camera settings such as projection vs ortho
 - Simple lighting settings
 - Etc.

Week 10, Monday 4/05/21 to Sunday 4/11/21 - 16 hours and 15 minutes

- Made export 3D file format code live
- Documentation is now hosted via Netlify [here](#)
 - Previously, I wanted to use GitHub pages. However, you cannot have multiple sites for multiple branches of the same repo at the same time. Netlify allows this though
 - Also, whenever the dev branch is updated Netlify will automatically generate the documentation to go with it on the live docs site
 - This works well since dev consists of code that developers currently have access to and need to know about
- Had someone test the application and discovered a bug with loading projects
 - The original mesh for other projects was not completely removed. It would be overwritten instead. But if the newly loaded project doesn't overwrite it, then it doesn't go away. This has been fixed
- Added a New Project button
- The input used for naming renders, 3D file exports, and projects was refactored into a single SubmitTextModal component
 - This greatly simplified the code and will likely be reused in the future as more features are added
- All React components now have .jsx for their extension instead of .js
- Updated metadata about the site and added a new favicon
- Continued to work on and finish the interim report along with prepare the logs for submission
- Began application for Brooklyn College's Research Day
 - The application requires an abstract to be written due on the 12th of April. This will be worked on after the submission of the interim report
- Added link to the project's GitHub repo to the Voxel Edit program
- Last but not least, emailed supervisor with Interim report, daily log, weekly log, and the interim form all as pdfs
 - After doing so, submitted those same files to the CISC 4900 Dropbox link

Week 11, Monday 4/12/21 to Sunday 4/18/21 - 16 hours

- Finished [Rework Mobile Modal](#)
 - Previously, mobile users had a single modal that would display on their screen with the ColorPalette. However, it was limited to only the ColorPalette

- There is now a single modal that conditionally renders content based on what the user selected to see such as
 - ColorPalette
 - Additional brush options
- More settings are planned to be added such as for adjusting materials or lighting
 - These have plenty of space on the desktop UI but not mobile hence the need for the refactor
- Also, reworked colors so that the modal no longer has a white background but a dark one
- [Blue Border Colors](#) have been addressed
 - MacOS users on the Firefox browser had an issue where the currently selected color on the ColorPalette was blue instead of white and without rounded edges
 - The issue was setting outline-style to auto
 - This gave the desired effect on Chrome but no on other browsers
 - This also served as a lesson to always test on multiple browsers before a pull request. Do not assume it'll just work everywhere
 - Albeit, I do not have access to a Mac but my supervisor does hence the discovery of this bug
 - The box-shadow CSS property gave the desired effect on all browsers over outline and was used instead
 - Tested with the supervisor and this worked!
- Since the semester is quickly coming to an end, the [Final Roadmap](#) issue was created to detail what objectives there are for these coming final weeks
 - Since the final report is due May 10th and the presentation May 12th, there isn't much time to actually develop the project itself
 - As such, the Final Roadmap addresses what might be possible in this time
 - The final features include:
 - Extrude brush
 - Lighting controls
 - Undo/Redo
 - Sample models that users can load in to look at
 - 3D grid
 - As will be mentioned, only the extrude brush and sample models have been added thus far
 - Export still needs to include vertex color for models
 - This might take a moment to implement and is being saved for if there is time
- Finished application for Research Day (Science Day) at Brooklyn College
- Finished [Sample Models](#) issue
 - This was a lot simpler to implement than first thought
 - Just a few json files with the models that are loaded in via the fetch api
 - There is now an examples dropdown where users can look at some previously created voxel models to get an idea of what they can make

- Began and finished the [Extrude Brush](#) issue
 - This was perhaps one of the most exciting additions to the program since adding a single voxel at a time is very tedious
 - The extrude brush is essentially the fill bucket from programs like MS Paint but in 3D!
 - Users can paint all voxels of the same color that are adjacent on a plane to a new color
 - Users can also stack a layer of voxels on top of adjacent voxels of the same color
 - Users can also remove all adjacent voxels of the same color similar to how painting works
 - Finishing this issue had two parts
 - First, the UI needed to be adjusted to allow for additional Brush Types
 - Brush Types now include “single” and “extrude”
 - Single is placing one voxel at a time which was default
 - Extrude works as previously described
 - The Brush component was split into BrushActions and BrushTypes
 - BrushTypes are single and extrude
 - BrushActions are add, remove, and paint
 - The internal Brush class now handles not just brush actions but the different brush types as well!
 - And it can be easily expanded to more brush types are new editing tools are added
 - Desktop and Mobile UI had to be reworked to accommodate the new BrushActions and BrushTypes components
 - For mobile, there is now a “Brush Settings” option instead of just the “add,” “remove,” and “paint” actions
 - This is a little cumbersome but reduces screen clutter by a lot
 - Second, the actual algorithm for extrude was implemented
 - This was a bit worrisome at first as I wasn’t sure how difficult this might prove to be
 - However it was a lot more simple than initially thought
 - While searching for possible solutions, I found this [Flood Fill](#) algorithm
 - The same algorithm that painting programs like photoshop use for when the user wishes to use the paint bucket tool
 - I implemented a simple version of the flood fill algorithm and then adapted it to work in 3D
 - Depending on which face of a voxel the user clicks, the flood fill algorithm will work on that plane that correlates with it

- For example, clicking on top of a voxel will have the algorithm work on the xz-plane
- The extrude brush now paints and removes voxels of the same color and are adjacent to one another along a plane
- Likewise, the extrude brush adds voxels above where the user clicked following the same flood fill rules

Week 12, Monday 4/19/21 to Sunday 4/25/21 - 21 hours

- The majority of this week was spent on writing
- I had written an abstract back in week 11 to submit to Brooklyn College's Research (Science) Day
 - Unfortunately, the abstract was rejected
 - There was an opportunity to create another one for submission
 - I was feeling too overwhelmed at the time and decided not to go through with the event
 - Even still, my supervisor requested I go ahead and create a new version of the abstract anyways
 - That new abstract can be found [here](#) under the revised abstract section
- My supervisor also requested that write a report detailing everything that I have done with the project thus far
 - As the due date for the final report is closing in, I decided to also have this report coincide with the requirements of the actual final report as well
 - That report can be found [here](#)
 - As of the end of week 12, that report is mostly done but still requires some info
 - Thus far, the report is 35 pages long and contains a lot of information that will likely need to be cut down as well

Week 13, Monday 4/26/21 to Sunday 5/02/21 - 15 hours

- Shared the report my supervisor requested
 - The report also doubles as the final report for CISC 4900
 - The report isn't quite done yet. There are some sections that need more filling out
- Continued to work on the final report
 - The report is close to being done. I had some trouble really focusing but still managed to get it closer to where it needs to be
 - Report should be finished during next week
- Began work on the CISC 4900 presentation
 - As with the report, I had a lot of trouble really focusing
 - I felt a bit lost on where to start and a little overwhelmed
 - Managed to get a decent first draft of the presentation done
 - The presentation can be seen [here](#)

Week 14, Monday 5/03/21 to Sunday 5/09/21 - 15 hours and 15 minutes

- Finished final report

- There's a ton of information in the report with detailed explanations on my thinking, progress, tech stack, etc.
- Also took the time to review the report and clean grammar mistakes
- Outside of perhaps adding a few more links to resources, the report is done
- The report is due on the 10th and shall be submitted then
- Continued work on the presentation
 - The presentation is mostly done but I would like to clean it up a bit more
 - I did a few practice presentations by myself and everything appears to work well enough
 - The presentation is due on the 12th so there is some time for a few more additions
 - Also, the actual recording of the presentation is due by the 19th
- Showed supervisor both the final report and the presentation
 - As for the final report, they said I wrote in the first person too much (words like "I" for example)
 - Even still, it appears to be okay
 - They said that the slides appear to be okay as well
 - Be sure to explain certain things in detail though they said which I plan on doing as I present
- Began work on an abstract
 - My supervisor requested another abstract
 - This one should be up to 800 words and I should reference [this](#) website
- Prepared files for the final report submission on the 10th

Week 15, Monday 5/10/21 to Sunday 5/16/21 - ? hours (logs were submitted on the 10th so no exact time is known)

- This is the final week! As such, it mostly consists of submitting CISC 4900 related files
- Submitted report related files by May 10th
- Submitted slides by May 12
- Recorded and submitted final presentation by May 14th
 - This is actually due by the 19th but I aim to do it sooner so that I can focus on studying for finals over the weekend