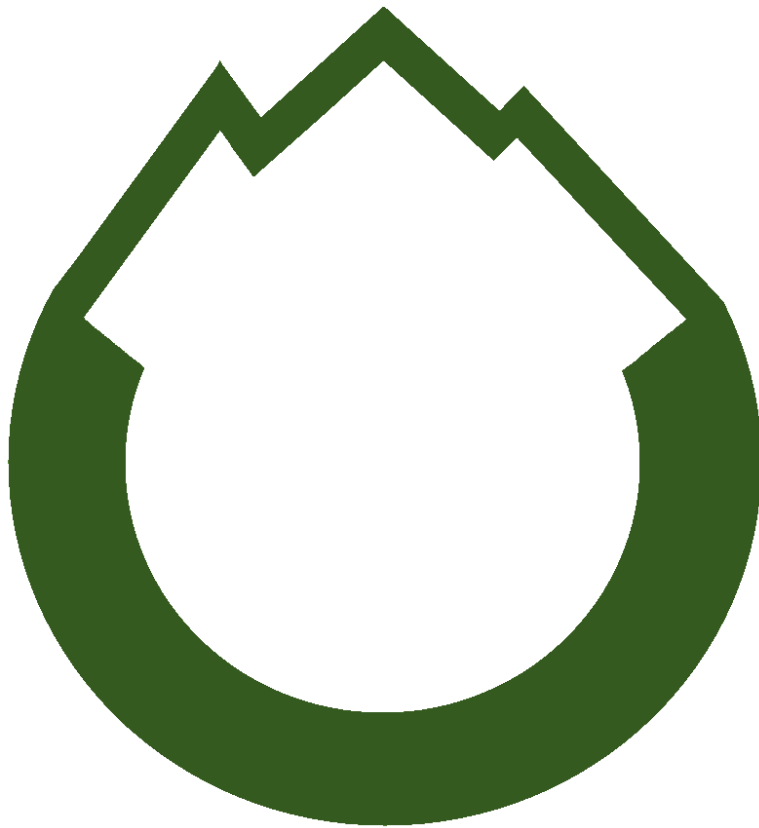


Topo Technologies

Bruce Clark, Justin McGowen, Gage O'Neill

Group 16

URL: TBD

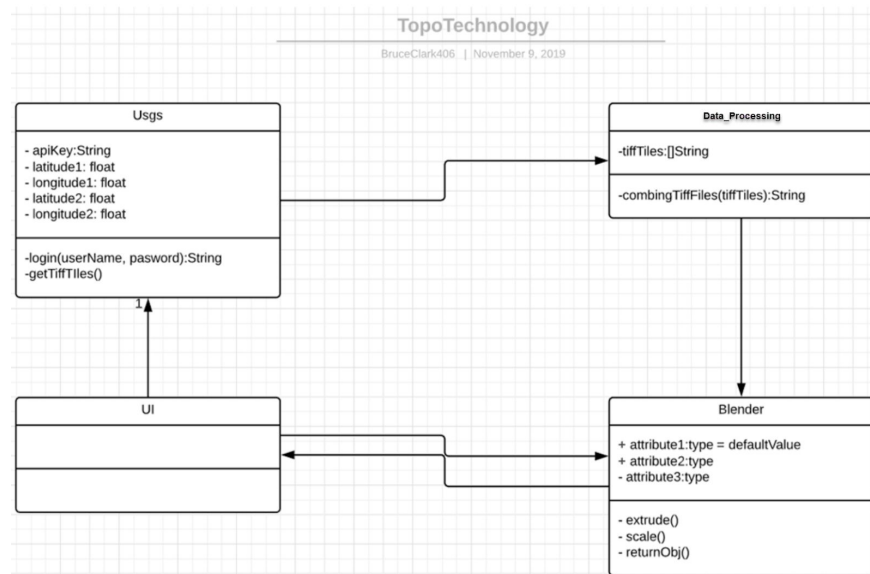


Objective:

We wanted to create a website that would output the topography of any geographic location on the planet. We wanted to do this leveraging the massive amounts of free geographical information, building and object file showing the selected geographic topography, display the object file, and then be able to download the file for use in game design or 3d printing. There are several programs that have functionality to do this, but the process to do so is often difficult and not laid out well. These websites either do not allow the user enough options for the type of file they want and often, they are paid tools which deters a lot of traffic from do-it-yourselfers and hobbyists who won't pay for a tool they aren't using often.

Summary:

In order to accomplish our goal, we needed to be able to retrieve map data, then generate a model that is viewed online, and finally allow the user to download this model. There are many extra modules we would like to add, but to get a minimum viable product, these are the functions we need.



In summary, there were four discrete categories of modules to our architecture: the websites UI, grabbing the appropriate data from the appropriate source, merging and cleaning that data to as we needed and as the user had specified, and then turning that data into an object that we can view and download on our website.

Currently, we have coded all of the necessary functionality in order to build a website with our minimum requirements. Our website is able to get user inputted location, then with that location, grab height-map data from USGS including the given location. We are also able to automatically create and render models without any intelligent input from the user. We can then allow the user to download an object file. Currently, we don't have downloading options because we don't have modeling options, but we hope to implement these in the near future.

Tools

We used a combination of tools to build our website:

- GitHub
 - Used as a version control and as a base we could push and pull our code from
- Discord
 - Used to communicate between team members
- Code Editor
 - Different people on our team used different editors including VS Code, PyCharm, and Atom

API

- Google Maps
 - Used to select the geographic land the user wanted to be modeled
- USGS
 - Used to grab a varying number of TIFF images
- Pillow
 - Used to crop, manipulate, and join our TIFF images
- Blender
 - Used to turn a TIFF image, into a 3d object
- 3JS
 - Used to display our 3d object

We used several tools to accomplish our goals. The two most important were the Blender and USGS apis. USGS has elevation maps of the entire earth, making it a really good jumping off point for us to generate models. We are aware that there might be higher resolution maps exist, but not of the whole world. Using google maps api, we are able to display a map to the user, then retrieve inputted coordinates for later use. The coordinates can be sent to USGS and will automatically return image files that are usable in the model creation. We are unsure how we will dynamically store these image files, so we need to work out a storage solution for these in the future. Once these pictures are downloaded though, we have a python script that can take an image and output a model. Then, user input will allow the user to download this model with blender. This part will not require downloading files to the server host, but may use server resources to create the model. This will be considered in making the website scalable in the future.

The blender api is really easy to use in python. For those looking to learn blender scripting, the easiest way is to first learn how to do basic stuff in blender. There is a learning curve, but there are also tons of videos on how to do things in blender. The next step after that is to turn on the Info Editor in your views. From here, whenever you do something in blender, the Info Tab will show you what you need to type in python in order to cause that action to happen using scripting.

Frameworks

- Flask
 - We used Flask as our web framework. Flask is an incredibly easy to use, python based, microframework.
- Bootstrap
 - We used Bootstrap 3 to make our site responsive.

Because we were looking at using blender from the start, we looked for python frameworks for web development. This influenced our decision to use flask. Flask is a bit more lightweight than django, but will not set up some extraneous things that we didn't need like a server. This gave us the control we wanted for our web app. Using flask is very simple, you will start by writing a python script with flask as an import. Flask uses templates, so you will then create a template, written in pure HTML, that flask will grab to display using its navigation system. With these templates, you can also inherit other templates, pass parameters, and use python variables and code in order to make each page dynamic.

Roles

Bruce was in charge of accessing the USGS and google maps API. He wrote a script to pull down geographical information from USGS and then another script to merge and clean the data, according to the user's specification. He also worked on the responsiveness of the site using bootstrap.

Justin used the other apis for developing 3D models. He wrote the automated blender script to be able to generate models based off of the data Bruce pulled. He also worked out the rendering, using three.js to allow the user to interact with the model.

Gage worked on styling on the front end as well as getting some of the blender stuff to communicate better with the website like downloading the models.

Summary

Altogether, we learned a lot about web development through this project. There is not too much that we would do different in the future. We are going through a very iterative process, starting with the most important parts, then merging them together as we accomplish different tasks. There are obviously a lot of things that we still would like to do, but we are hoping to continue this project and finish these.

References

Blender api : <https://docs.blender.org/manual/en/latest/>
Bootstrap: <https://getbootstrap.com/>

Google Maps: <https://developers.google.com/maps/documentation>

three.js : <https://threejs.org/>

USGS : <https://www.usgs.gov/>