ENGL 4710: Capstone: Literature & the Visual Arts
Erika Boeckeler
Making Story Maps Using Knight Lab

About

This handout breaks down the steps for hosting the folders with the processed image for use in Knight Lab's StoryMap Gigapixel platform. The hosting service used for this purpose is GitHub, which requires setting up a free account on GitHub, and (optionally but recommended) downloading the GitHub Desktop program, which enables quicker storage of the processed files.

The concrete goal is to create a link to the processed image that can be used in Knight Lab's StoryMap.

Relevant links

- StoryMap for images website: https://storymap.knightlab.com/gigapixel/
- StoryMap instructions for hosting your image (with screenshots!): https://github.com/NUKnightLab/StoryMapJS/blob/master/GITHUB_HOSTING/GITHUB_HOSTING.md
- GitHub website: https://github.com/
- GitHub desktop download site: https://desktop.github.com/

Keywords

- **GitHub** - a hosting platform for version control and collaborative software development
- **Repository** - the directory, storage space, or folder, where the project lives: this is where we will store our tiles. Initially it is created online
- **Cloning** - A command in GitHub Desktop for creating the version of the repository on your computer
- **Fetching/pulling** - A command in GitHub desktop for synchronizing the versions of the repository online and on your computer
- **Committing** - A command with which you are adding content to the repository on your computer
- **Pushing** - A command with which you are publishing online the content you just added to your repository

Initial Preparation

- Go to https://github.com/ and create an account for free
- Go to https://desktop.github.com/ and download GitHub Desktop for your OS
    - If you don't want to download another program, you can avoid using it, but it will make uploading the tiles through the browser significantly more tedious. Instructions for that case can be found at the end of this handout under the section "II: Storing the processed image and getting the link without downloading GitHub Desktop."

I. Storing the processed image and getting the link

Creating the repository

1. Go to GitHub website and log in
2. In the menu on the left side, click the green button **New**
3. Name your repository in the **Repository name** space
4. Check the **Public** option
5. Repositories can never be empty. Therefore, under the **Initialize this repository with** section, check the **Add a README file**. It will be a blank document and for now it can stay that way.
6. Click on the green button **Create**
7. You will be sent to the repository webpage. Copy its link.

Cloning the repository (putting a version of it onto your computer)

1. Open the GitHub desktop program
2. Go to **File**, and select **Clone repository**
3. Within the opened window, you can go to the **URL** tab, and paste the link of your repository there
4. Note that below the space for pasting, there is a specified path where your repositories are stored. GitHub Desktop automatically creates that folder. Your repository will be a folder within that folder.
5. Click on the **Clone** button. This means that you can work on your repository on your computer, and then publish that work on the web using the GitHub Desktop program.

Uploading your materials

1. Before you start working on the repository on your computer, click on the **Fetch** tab in GitHub Desktop. By doing that, you are making sure that the online version and your local clone of the repository are fully in sync.
2. Below that tab, still on the front page of GitHub Desktop, click on the **Show in Explorer/Finder** button. This command will open the folder for the repository on your computer.
3. Now you can paste your TileGroup folders into that folder.
4. Go back to GitHub Desktop. You will see it has registered changes in your repository.
5. In the bar to the left, in the lower part you will see a space for a short description (2-3 words) with the default suggestion made by the program. Fill it out.
    a. Important note for **Mac users!** Sometimes, in the space to the left you will see files that are called ".DS_Store". If you see any of these, un-select them. You do not want to add these files to your repository.
6. After that, press the blue **Commit** button (or **Commit to main**). Processing this commit will take a few seconds.
7. After that, in the central space of the program you will see an option to **Push origin**. Click on it. With this, you have published the new content of your repository online.

Getting the link for the StoryMap

1. Go back to the web version of your repository, and click on the **Settings** button above the list of content of your repository.
2. When in the Settings page, scroll down until you find the **GitHub Pages** section.
3. You will see a dropdown button which says None. Click on it and select **Main**.
4. Then click **Save**, and the page will refresh.
5. When it refreshes, scroll down to the same section, and you will see an URL toward the top of that section (at the end of the sentence: "Your site is ready to be published at"). **That is the URL you will use at the StoryMap website.**

II. Storing the processed image and getting the link without downloading GitHub Desktop

If you would like to avoid downloading one more program to your computer, you can in principle upload your images directly to the GitHub website, but with some limitations that make the process more labor intensive.

This handout will first go through creating the folders through the website, and then through uploading the files.

Creating the folder for your tiles

You cannot add folders to GitHub website, only files. You can, however, indirectly create new folders while specifying the path of the file you are adding through the GitHub website.

1. The same instructions for creating the repository, setting it to public and initializing it with a README file apply here as outlined above.
2. Once you are in the repository, you can click on **Add file**, and select **Create new file**. Even though we are actually interested in uploading files, we have to go this way.
3. A new page will open in which you will automatically be set up to add the name of the file.
4. You create a new folder indirectly by adding **"some_folder_name/"** before the name of the file. By doing that, you are actually creating a folder that will contain the files. So in our case, it makes the most sense to start writing the name with for example **TileGroup0/**
5. However, we are interested in uploading our files rather than creating any new files, so this file is actually a placeholder file. Conventionally those files are named **"some_folder_name.gitkeep"**. So in our case, the name of our file could be **TileGroup0.gitkeep**
6. Therefore, in our case, the full thing that we want to type is **TileGroup0/TileGroup0.gitkeep**
7. Now you can scroll to the bottom of the page, and click on the **Commit new file** button.
8. Since no two tiles share the same name, you can use this folder to upload all the images from your different tile folders.

Uploading the files

1. Once you are in the newly created (initialized) folder, go again to **Add file**, but now select **Upload files**.
2. Here is the main issue: GitHub allows only for uploading up to 100 files per upload. Given that you will likely have significantly more than that, you will have to redo this step a few times.
3. You can either drag your file selection to the designated window, or browse them.
4. Once you add them (it takes a few seconds), scroll down and click the **Commit changes** button.

5. It will take some time to get the images processed and uploaded. After uploading, GitHub will return you to the webpage of your whole repository.
6. You have to remember on your own what was the last file that you have uploaded. If you reupload the same files (files with same names), however, it is not an issue, since GitHub will just add the same files again.
7. Now you repeat these steps, but with selecting new files each time. You have to pay attention on your own that you have not selected more than 100 files each time.
8. In the end, do not forget to add the **Image properties file** generated during the zoomifying of the image.

Now that you have all your files uploaded, you can follow the steps outlined in the **Getting the link for the StoryMap** section above, to get the link you will use for the StoryMap itself.

III. Adding the link to the StoryMap

Now that you have the link to your image, you can go and use it to start creating your StoryMap.

1. Go to the StoryMap website, and click on the **Make a StoryMap** button towards the top of the webpage. After that, log in.
2. In the newly opened window click on the green button **New**. Give it a name, and click on the **Create** button.
3. Once you are inside, go to the **Options** button toward the upper right corner.
4. When the options window opens, do the following:
    a. Check **Image** in the **Treat As** options
    b. Select **Gigapixel** in the **Map Type** dropdown menu
    c. Copy the **GitHub link** you created after uploading the images into the **Zoomify URL** window
    d. Finally, insert the width and height of the original image in pixels in the **Max Image Size** windows

Congratulations! From now on, you are free to explore all the additional features of the Knight Lab's StoryMap, and are in full creative control over your StoryMap!