Alumni Video Donations M8

Simone Davison, Andrew Lanum, and Austin Morin

May 4, 2022

1 General Progress Report

Our work since Milestone 7 has primarily been comprised of fixing bugs that popped up once we began testing our program on the lab machines rather than on our personal computers. Because the lab machines run older versions of Python and we lack administrative rights to install all of the necessary dependencies that made our program work on our personal machines, we had to overhaul how we implemented text clips into our weaving program. Imagemagick dependencies had to be entirely removed from our code and were replaced with ImageClips which were native to moviepy. Also there was a significant multiprocessing bug that prevented videos from being uploaded to Firebase. An infinite loop was fixed in order have processes delete a video weaving temp directory that was stalling the program. Also our code used to access Firebase was refined so that it pinged Firebase for files one at a time, which came at a cost of slower performance but fixed a host of other issues with videos never getting weaved and also causing deadcode on runtime. As of this writing our program is in near complete condition. Small patches such as background music, aspect ratio fixes, and a credits list still need to be made to the video content. More minor and patches in the source code still need to be addressed. A lot of stress-testing remains to be finished with chunks of increasingly large samples of donor files. Code will be cleaned and more extensive documentation will be written up in the coming weeks.

2 Team Accomplishments

- Successfully removed Imagemagick libraries as they do not work on the lab computers.
- Got access to the give day URL.
- Got access to the entire file of donors and have been able to test on it.
- Fixed many bugs.
- Fixed numerous dependency errors on lab machines.
- Improved videos and changed their content in preparation for Give Day

3 Goals Met

- Videos have been refined and stylized.
- Personalize videos and make them more aesthetically pleasing and accessible.
- Migrate all back-up code out of the main() function in weave.py to make program more readable and efficient with multiprocessing
- Complete all video weaving bugfixes.
- Analyze runtimes on both lab and personal computers.

4 Goals Not Met

- Clean all redundant and unused code
- Testing on 1,000+ videos.
- Add credits to our video content.
- Have not started refining documentation.
- Client has not had the chance to try the program, only experienced the output.
- Migrate all code from test-csv.py into weave.py and remove the splitter.py program so that there will be only two program files remaining: weave.py and storage-access.py.
- Fix all multiprocessing errors.

5 Updated Expectations

See section Quarter Plan below for more details

- Weeks 1-4:
  - Add multiprocessing
  - Refine videos
  - Add access tokens
  - Test on small amounts of videos
  - Clean up website and move hosting to Firebase
- Weeks 5-9:
  - Begin large scale testing
  - Continue to refine videos
  - Develop documentation further
- Weeks 10-11:
  - Continue website and code cleanup
  - Customer testing
  - Prepare and give 493 presentations

6 Branch Description

- master - All the most current project work is pushed to this branch (code, resources, documentation, etc.). The code-test folder is being cleaned up soon so that all photo and video content exists solely on our Google Firebase account, wwucsgiving.
- MS-2 - All project work pushed before Milestone 2's date.
- MS-3 - All project work pushed before Milestone 3's date.
- MS-4 - All project work pushed before Milestone 4's date.
- MS-5 - All project work pushed before Milestone 5's date.
- MS-6 - All project work pushed before Milestone 6's date.
- MS-7 - All project work pushed before Milestone 7's date.
- MS-8 - All project work pushed before Milestone 8's date.

7 Quarter Plan

- Week 1 (Mar 28 - Apr 1)
  - Multiprocessing
  - Refine videos
  - Update CSV reference in program
  - ~~Add access tokens~~ (unnecessary for our project)
  - Test on videos 2-30 (with upload turned off)
- Week 2 (Apr 4 - Apr 8)
  - MS6 (Have access token working, make headway on multiprocessing)
  - Multiprocessing
  - Refine videos
  - Begin hosting website on Firebase
  - Test on videos 2-30 (with upload turned off)
- Week 3 (Apr 11 - Apr 15)
  - Multiprocessing
  - Refine videos
  - Refine website hosted by Firebase
  - ~~Test on videos 1000 (with upload turned off)~~ (Moved to week 4)
- Week 4 (Apr 18 - Apr 22)
  - MS7 (Have multiprocessing working and video content well organized)
  - Refine videos
  - Refine website
  - Begin development on an installable executable
  - Test on videos 1000 (with upload turned off)
  - Begin full scale testing (10,000 videos) (expected to take a while)
- Week 5 (Apr 25 - Apr 29)
  - Monitor video making process (fix bugs)
  - Let client try product
  - Inspect output videos
  - Improve program
  - Refine videos
  - Refine website
  - ~~Begin development on an installable executable~~ (Moved to week 4)
- Week 6 (May 2 - May 6)
  - MS8 (Have begun refining documentation, ~~have website hosted by Firebase~~ (already accomplished), have an installable executable)
  - ~~Full scale test again (10,000 videos) (expected to take a while)~~ Began small scale testing and expanding to bigger scale testing

  - Refine videos
  - Refine website
  - Fix bugs
  - Verify project can run on lab computers (to future-proof)
- Week 7 (May 9 - May 13)
  - Documentation
  - Refine videos
  - Refine website
  - Fix bugs
  - Begin making the final videos (expected to take a while)
- Week 8 (May 16 - May 20)
  - MS9 (Have all the videos made and website finished)
  - Documentation
  - Monitor video making process
- Week 9 (May 23 - May 27)
  - Verify final videos
  - Videos sent out for give day (May 26)
  - Prepare for presentation
- Week 10 (May 30 - Jun 3)
  - Documentation
  - Use feedback to from give day to improve product
  - Present + watch presentations
- Week 11 (Jun 6 - Jun 10)
  - Watch presentations

8 Team Member Accomplishments

- **Simone Davison - (Hours Worked: 19)**
  - **Organize dependencies and put them in a requirements.txt file.**
  - **Debugged the program on the lab computers.**
    - **Removed the infinite loop when processes can't delete a folder and hold up the program.**
    - **Updated storage_access to ping Firebase for files one time which was causing the program to crash if there were too many processes running.**
  - **Replaced the functions that need Imagemagick which cannot be run on the lab computers with custom functions powered by the library Pillow.**
  - **Updated processes to only use one folder instead of two.**
- **Andrew Lanum - (Hours Worked: 8)**
  - **Reworked weave code to make videos look more professional**

    - **Separated thank you clips from other clips so they are played only at the end**
    - **Added checks for alumni's grad year to make message more personal**
    - **Added multiple text slides to make the video more coherent**
  - **Ran test case on home computer to ensure the project could still be delivered should something go terribly wrong, and sent feedback back to team to hopefully improve project efficiency**
  - **Updated the ad banner that plays at the end of the video to align more with client specifications**

- **Austin Morin - (Hours Worked: 16)**
  - **Debugged firebase token, Imagemagick dependency errors, and moviepy dependency errors on the lab machines.**
  - **Performed runtime timing tests on 1, 15, 50, 250, and 500 donors using lab machines. There is still a bug that needs to be patched when running program on 50-donor sample.**
  - **Re-deployed website with updated QR codes and hyperlinks**
  - **Looked into the aspect ratio bugs when Imageclips were implemented into the program in place of Imagemagick dependencies.**
  - **Wrote a Python program that takes in a .csv file or .xlsx file and eliminates rows that have either a missing first name or a missing email in accordance with the customer's preferred policy and outputs a cleaned .csv or .xlsx file.**
  - **Debugged a path issue when having our program read our "thank you" clips by fixing a path variable and re-organizing Firebase.**