**Steganography Phase** II

Linden Crandall, Jonathan Mainhart, Zhihua Zheng

University of Maryland Global Campus

CMIS 495: Current Trends and Projects in Computer Science

Prof. Majid Shaalan

April 26, 2022

**Overview**

For Phase 2, we began preliminary testing of the software and are still working on the image file saving error handling which is proving to be more complicated than originally thought. Our GUI can now implement image selection via filechooser from the user’s default directory, loading the chosen image and rendering it onto the GUI window, extracting the image’s pixels into bytes for steganography processing, calculating the max number of character allowance for the user’s secret message, encoding, reset, and save buttons have been implemented. We are still working on small bugs and fixes like overwriting files on save but we are still ahead of schedule.

Phase I (Week 5) Milestone Status Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Scope** | **Milestone** | **Status** | **Notes** |
| Week5-6 | Main GUI | incomplete | GUI Display is done.  Will need to dynamically display input and output from other py. classes. |
| Week5-6 | Popup windows | incomplete | Still implementing popups for specific actions like saving image file to machine, resetting image to original state, etc. |
| Week5-6 | ImageObject Class | complete | Window Display is done.  Will need to dynamically display input and output from other py. classes. |
| Week5 | File Input | complete | Successfully opens files of image type only, throws error in the form of popup dialog box if user attempts to open a non-image file |
| Week5-6 | File Output | complete | Successfully opens files of image type only, throws error in the form of popup dialog box if user attempts to open a non-image file |
| Week5 | Message conversion | complete |  |
| Week6 | Image encoding | complete | Will need to refactor to simplify some of the code. Button enabled |
| Week6 | Image decoding | complete |  |
| Week6 | Image reset | complete |  |
| Week6 | Image save | incomplete | Filechooser popup implemented, image file isolated and can now successfully save to user’s machine. Need to implement overwriting functionality. |

**Problems Encountered and Reevaluation of the Decisions**

This week’s errors mostly were contained to the image file choosing/saving functionality. There were AttributeErrors and Invalid type errors when a user would inadvertently or deliberately choose a non-image file type on “Open Image” button click. The errors were handled by attempting to open the image using the Image module from the PIL package and surrounding that within a try/except block. Currently, obtaining the user-selected directory in which the image file should be saved is an ongoing issue that is being actively worked. Should be completed by Phase III

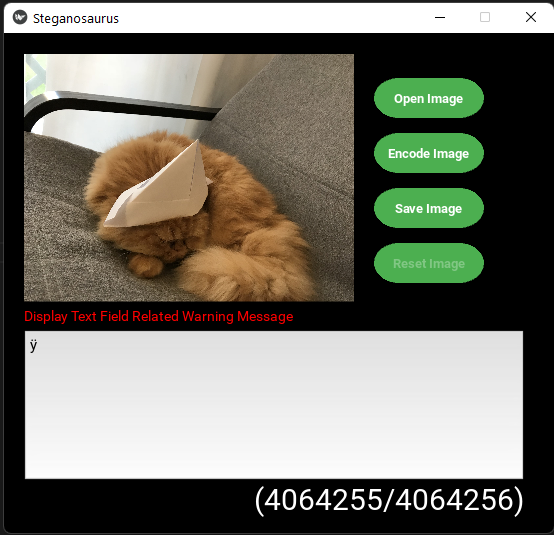
**Changes of the previous document**

For image file saving functionality, an additional Filechooser class, named ImageSaverPopup, was created and is responsible for opening up the file explorer, capturing a user-selected directory, and then saving the currently loaded image that should have previously been chosen by the user to that directory.

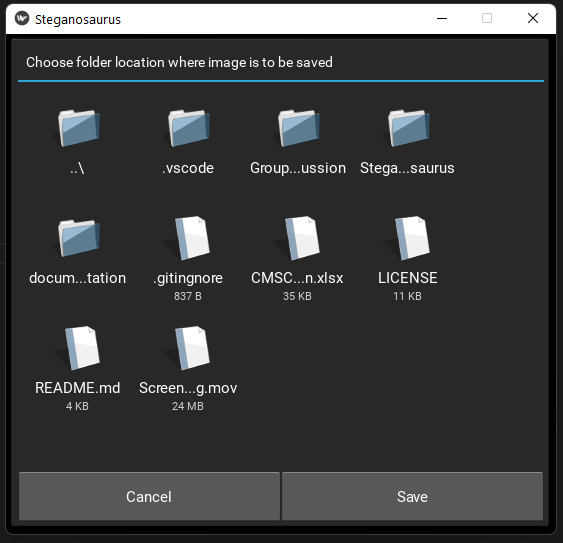
The GUI was also updated to include the “Save Image” button and “Encode Image” button.

**Phase II Software Implementation Result**

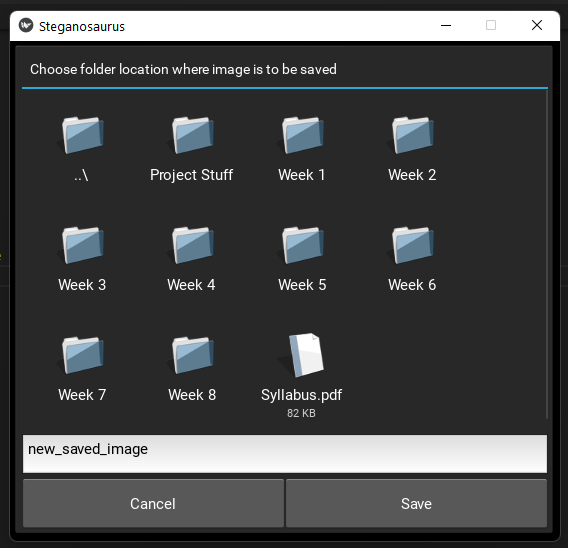
1. Updated Main GUI with “Save Image” button and “Encode Image” button.



1. FileChooser - Save Image Dialog



1. Save button within filechooser



Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

1. “Reset” Button implementation which reverts textfield to default and erases any encoding done to current image

A screenshot of a dog lying in grass

Description automatically generated with low confidenceA dog lying in the grass

Description automatically generated with low confidence

