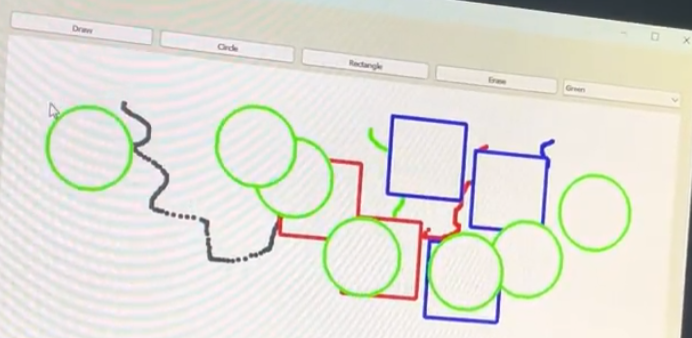
Python Note App Final Report

Submitted by: Nicholas Rawicz, Christian Rodriguez, Shane Holmes, Jake Karas

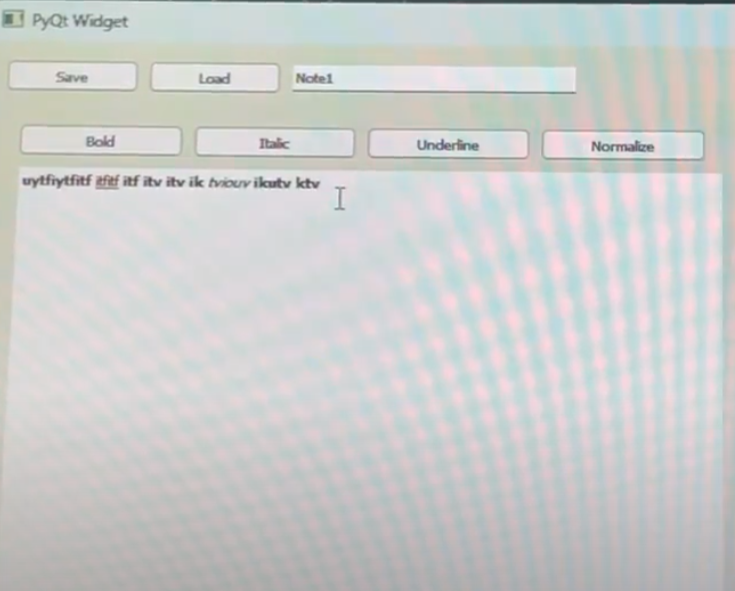
**Revised Deliverables:**

The team sought to create a simple note-taking app, capable of storing and loading both text and drawings made by the user in the application.

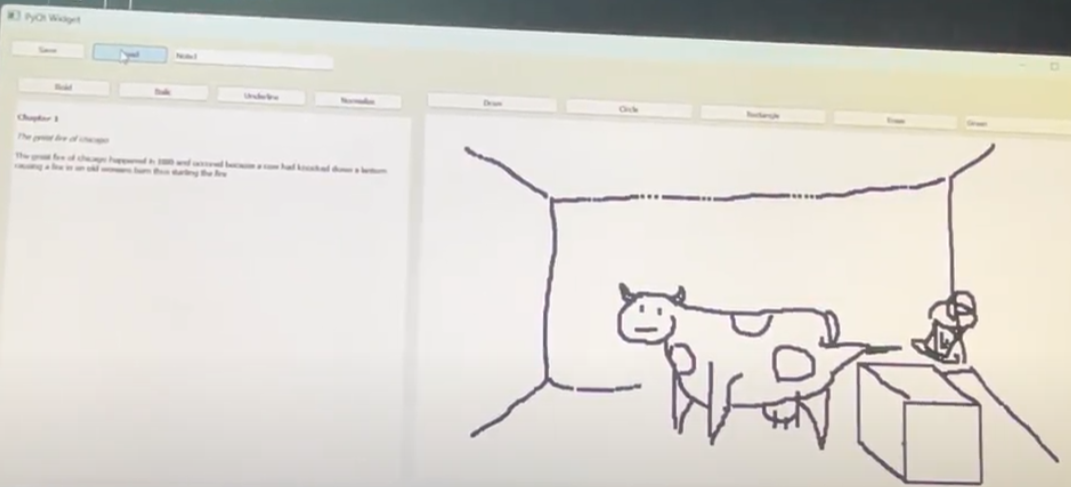
Drawings consist of freehand sketches by the user’s mouse input and the ability to include shapes such as squares, triangles, circles, and rectangles with the ability to change color of freehand and shapes to black, green, blue, or red. An erase tool is provided.



For text, different modes can be selected such as italic, bold, and underline. Additionally, the “normalize” mode returns text to its default setting.



Finally, the application can save files with a user-entered title, and load files using the appropriate title.



**Lesson Learned:**

Estimated hours: 37.5 hours

Actual hours: 5 hours

About the hours: The difficulty of utilizing the Pyqt5 library was severely overestimated, so programming with it took about ⅙ of the originally estimated time as a result.

The team obtained valuable experience in creating complete documents such as SDS, SRS, Project Management Plan. The team followed the SDLC of Agile Methods, focusing on completing the program rather than focusing on the formal documents. Effort estimation was calculated with a chart, an activity graph was created, a WBS was created, Critical Path was determined, a design-level class diagram (instead of a use-case diagram) was made, a sequence diagram was made, a UML class diagram was created, a System Interfaces diagram was formed, and dependencies view and execution view diagrams were made. All these are essential parts of the planning section of assembling software.

**Testing Results:**

Unit Test: Run each individual button alone and see if the corresponding interfaces react

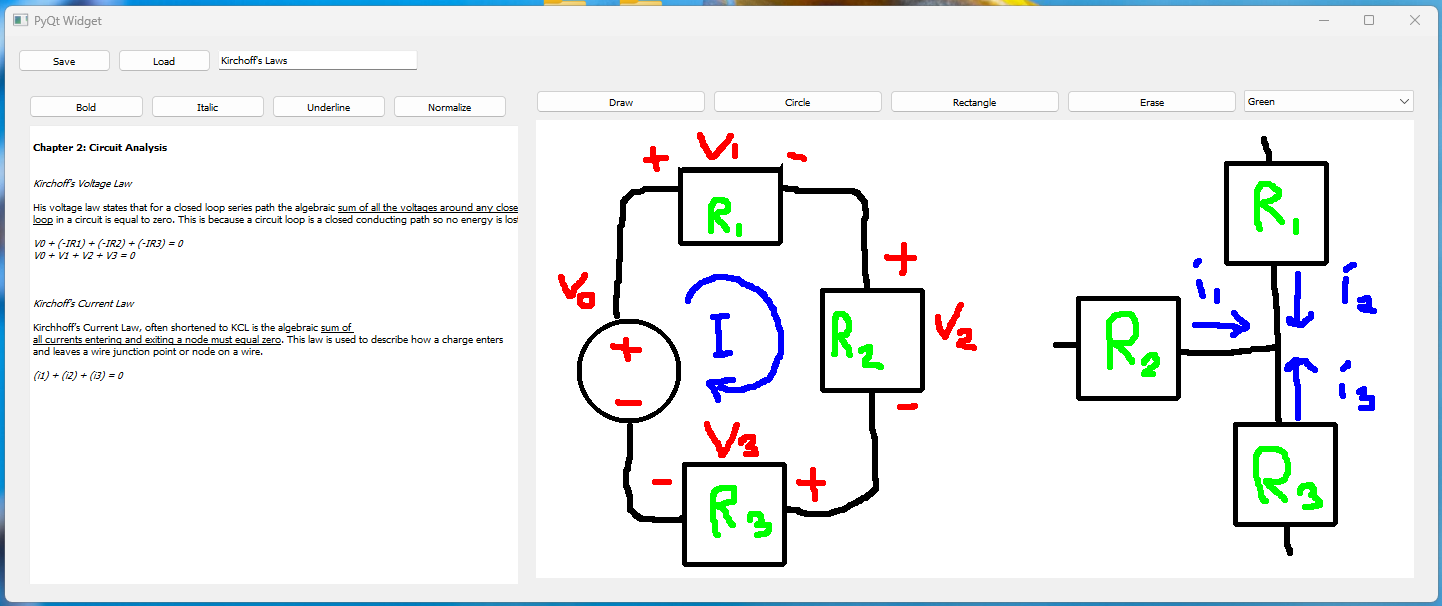
| Button Name | Description | Functional? |
| --- | --- | --- |
| Save | Saves Current note to be loaded later under the name specified in the textbox next to load | Yes |
| Load | Loads file from name specified in the textbox next to load | Yes |
| Bold | Bolds selected text | Yes |
| Italic | Italisices selected text | Yes |
| Underline | Underlines selected text | Yes |
| Normalize | Removes font-weight from selected text | Yes |
| Draw | Selects the draw tool in the drawing space | Yes |
| Circle | Selects the circle draw tool in the drawing space, create a circle of a defined size | Yes |
| Rectangle | Selects the rectangle draw tool in the drawing space, create a rectangle of a defined size | Yes |
| Erase | Selects the erase tool in the drawing space | Yes |
| Color ComboBox(black) | Any shape in the drawing space will be black | Yes |
| Color ComboBox(red) | Any shape in the drawing space will be red | Yes |
| Color ComboBox(green) | Any shape in the drawing space will be green | Yes |
| Color ComboBox(blue) | Any shape in the drawing space will be blue | Yes |

Integration testing: run each of the interfaces (drawing, text, and saving/loading) independently of each other The only interfaces are comprised of interfacing with the FileManager class to the DrawingUIWidget, the FileManager class to the TextUIWidget, and the Drawing Widget class to the Drawing UI Widget.

| Interface | Description | Functional? |
| --- | --- | --- |
| FileManager class to the DrawingUIWidget class | This is the interface between the saving/loading system and the DrawingUIWidget | Yes |
| FileManager class to the TextUIWidget class | This is the interface between the saving/loading system and the TextUIWidget | Yes |
| DrawingWidget class to the DrawingUIWidget class | This is the interface between the drawing space and the GUI for that interface | Yes |

System Testing: run a sample scenario of a user booting the program, drawing writing, and saving before closing

The user wants to take notes from his physics lesson, the lesson goes over Kirchoff’s voltage and current laws. The user will want to write about what he learned about each in the left-hand window. The student will use weighted styles, including bolding, italicizing, and underlining text to highlight different relevant parts of his notes. He will use the drawing function to draw out a diagram to convey the laws. He will use circles, rectangles, and different colors to convey the meaning.



We were able to accomplish all sought out tasks with the program