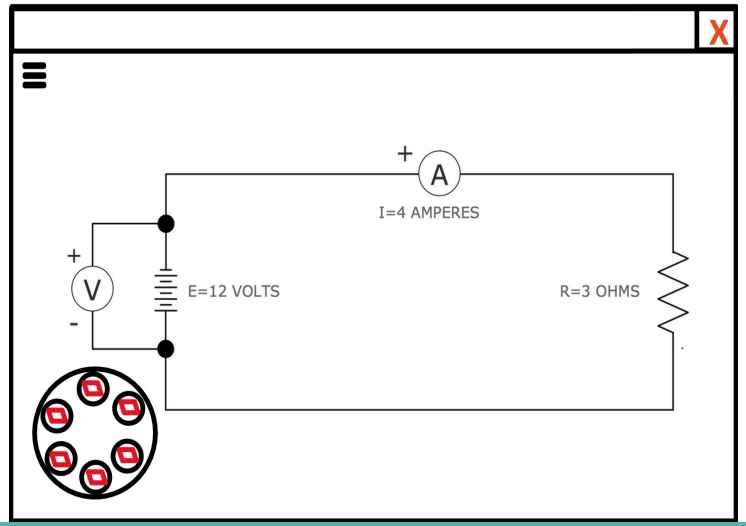
Circuit Board Designer

Joseph Spear, Kenneth Shipley,
Jason Rivas

Designing a Circuit

- The first of three primary features of this software is the ability to drag and drop different electrical components (Voltage source, Resistor, Capacitor, etc.) onto a gridded workspace.

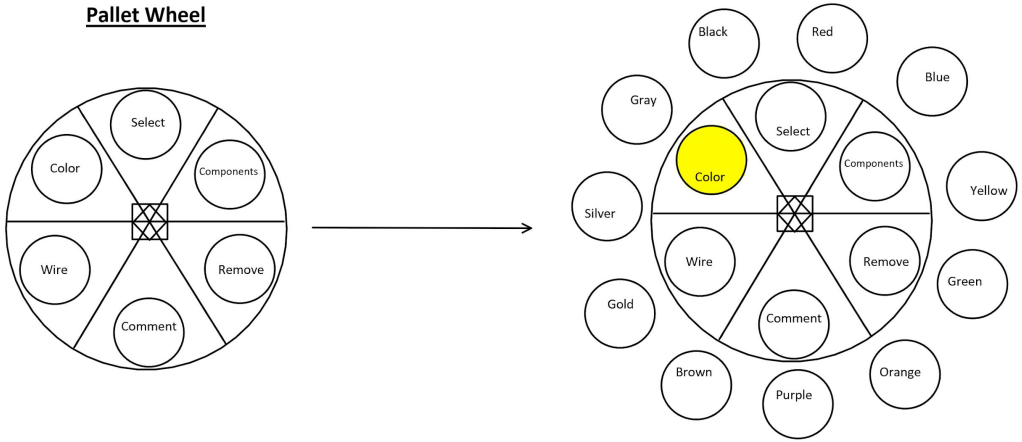
- There will be an internal database of components to choose from. These will be selected and placed into the desired location on the board.
  - Wiring is a special component which will auto-generate a straight line between the initial components leaving terminal, and the connecting components entering terminal.



Pallet Wheel GUI

- One of the major design choices is the implementation of a pallet wheel which expands in accordance with the selected item in the base wheel.
  - The wheel will include a “Select”, “Components”, “Remove”, “Label”, “Comment”, “Wire”, and “Color” button which expands.
  - Can collapse into a small square when the center is clicked.

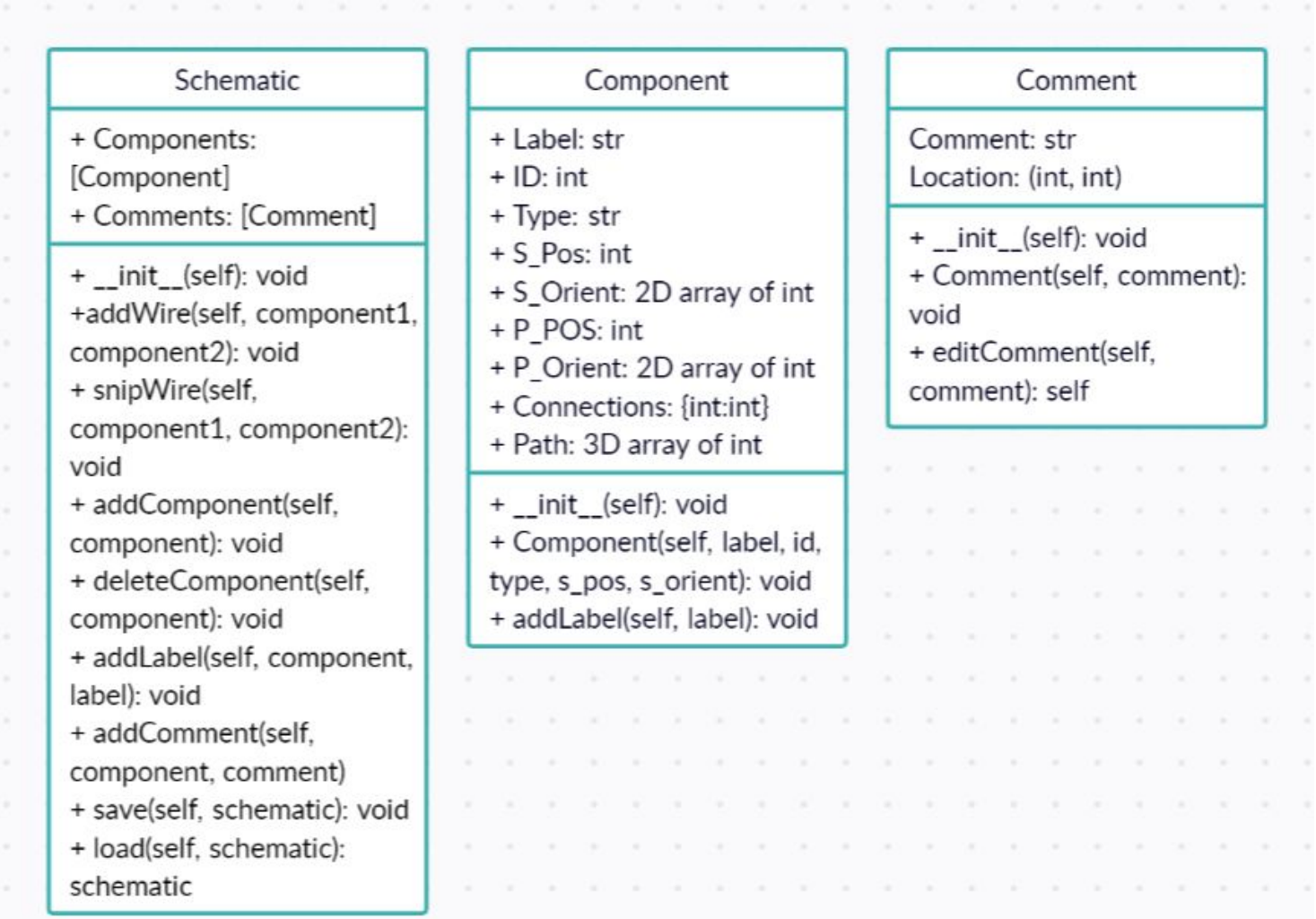
- The pallet wheel can also be moved around the workspace by grabbing the square and moving it to the desired location.



Other Workspace features

- Zoom by scroll, keyboard input, and GUI button.
- Hamburger Menu which includes Save, Save As, Load, and Create New Project.
  - When selected, a slide out menu will appear from the left side of the screen, displaying the above features.
- A “Schematic Class” will be used to generate different components (of the “Component Class”) and save their data later.
  - Will include, Label, Type, ID, Position, Connections, and Paths

Classes

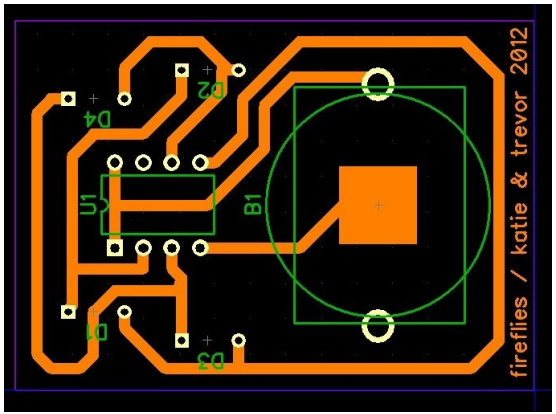

Schematic
+ Components: [Component]
+ Comments: [Comment]
+ __init__(self): void
+addWire(self, component1, component2): void
+ snipWire(self, component1, component2): void
+ addComponent(self, component): void
+ deleteComponent(self, component): void
+ addLabel(self, component, label): void
+ addComment(self, component, comment)
+ save(self, schematic): void
+ load(self, schematic): schematic
Component
+ Label: str
+ ID: int
+ Type: str
+ S_Pos: int
+ S_Orient: 2D array of int
+ P_POS: int
+ P_Orient: 2D array of int
+ Connections: {int:int}
+ Path: 3D array of int
+ __init__(self): void
+ Component(self, label, id, type, s_pos, s_orient): void
+ addLabel(self, label): void
Comment
Comment: str
Location: (int, int)
+ __init__(self): void
+ Comment(self, comment): void
+ editComment(self, comment): self

Optimization of Printed Circuit Board Space

- The second primary feature is that of producing a printed circuit board (PCB) layout from the input design.
  - A Monte Carlo algorithm will be used to randomize the components' location
    - Input a list of Components
    - Output a list of Components with updated path data.
  - The A* algorithm will be used to find the optimized paths for the traces connecting each component.
- The actual information will be stored in a .obj file which will then go through the exporter to produce an image.

Exporting Design to an image

- The third primary feature is the ability to take in the Optimal PCB Data, convert it to an image, and export it to an image file such as JPEG or PNG.
- The program will extract the data from the .obj file (such as the component data and location data) and use that to determine where the pixels will go on the image.



Source: http://www.trevorshp.com/creations/fireflies.htm
firefly PCB layout

File Management

- Ability to save a project into a .obj file using Python's Pickle library
- It will save the entire Schematic class to a separate file containing all the component data of the project
- This will allow the user to open up an old project they once worked on