

## ChatGPT Log Week 1

Gave ChatGPT the initial App.java starter code and asked it to "create a tab called "canvas" that has buttons to resize, zoom in, and zoom out.

Add mnemonics as such,

save : ctrl + s

load : ctrl + l

undo : ctrl + z

redo : ctrl + x

clear:ctrl + c"

```
private void initUI() {
    // Existing code...

    // Edit Menu
    JMenu editMenu = new JMenu("Edit");

    JMenuItem undoItem = new JMenuItem("Undo");
    undoItem.setMnemonic(KeyEvent.VK_Z);
    undoItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_Z, InputEvent.CTRL
    undoItem.addActionListener(e -> undo());
    editMenu.add(undoItem);

    JMenuItem redoItem = new JMenuItem("Redo");
    redoItem.setMnemonic(KeyEvent.VK_X);
    redoItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_X, InputEvent.CTRL
    redoItem.addActionListener(e -> redo());
    editMenu.add(redoItem);

    JMenuItem clearItem = new JMenuItem("Clear");
    clearItem.setMnemonic(KeyEvent.VK_C);
    clearItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_C, InputEvent.CTRL
    clearItem.addActionListener(e -> clearCanvas());
    editMenu.add(clearItem);

    // Canvas Menu
    JMenu canvasMenu = new JMenu("Canvas");

    JMenuItem resizeItem = new JMenuItem("Resize");
    resizeItem.addActionListener(e -> resizeCanvas());
    canvasMenu.add(resizeItem);

    JMenuItem zoomInItem = new JMenuItem("Zoom In");
    zoomInItem.addActionListener(e -> zoomIn());
    canvasMenu.add(zoomInItem);

    JMenuItem zoomOutItem = new JMenuItem("Zoom Out");
    zoomOutItem.addActionListener(e -> zoomOut());
    canvasMenu.add(zoomOutItem);

    // Help Menu
    // Existing code...

    // Adding menus to menu bar
    JMenuBar menuBar = new JMenuBar();
    menuBar.add(fileMenu);
    menuBar.add(editMenu);
    menuBar.add(canvasMenu);
    menuBar.add(helpMenu);

    setJMenuBar(menuBar);
}
```

It also included some attempts at implementing the new buttons functionality but these had numerous bugs so I discarded that for now deciding to implement functionality once the layout of the application was solved.

It forgot to add save and load mnemonics but I added those using the others as reference.

I asked ChatGPT to add a function called `setupFeatureMenu` which creates a java swing toolbar on the left hand side of the application which has a title called Feature Menu and a search bar underneath that where the user can input strings.

ChatGPT added this to `initUI()`

```
// Setting up feature menu toolbar
JToolBar toolbar = setupFeatureMenu();
mainPanel.add(toolbar, BorderLayout.WEST);
```

And gave this `setupFeatureMenu()` function

```
private JToolBar setupFeatureMenu() {
    JToolBar toolbar = new JToolBar(JToolBar.VERTICAL);
    toolbar.setFloatable(false);

    JLabel featureMenuLabel = new JLabel("Feature Menu");
    toolbar.add(featureMenuLabel);

    // Search bar
    JTextField searchBar = new JTextField();
    searchBar.setPreferredSize(new Dimension(150, 25));
    searchBar.setMaximumSize(new Dimension(150, 25));
    toolbar.add(searchBar);

    return toolbar;
}
```

I followed up by prompting it with “underneath the search bar, add the following tabs: general, structural, kitchen, bedroom, dining room, living room, and bathroom. Also center the “Feature Menu” label in the toolbar.

```

/**
 * Sets up the feature menu toolbar on the left side of the application.
 */
private JToolBar setupFeatureMenu() {
    JToolBar toolbar = new JToolBar(JToolBar.VERTICAL);
    toolbar.setFloatable(false);

    // Panel for the label (centered)
    JPanel labelPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));
    JLabel featureMenuLabel = new JLabel("Feature Menu");
    labelPanel.add(featureMenuLabel);
    toolbar.add(labelPanel);

    // Search bar
    JTextField searchBar = new JTextField();
    searchBar.setPreferredSize(new Dimension(150, 25));
    searchBar.setMaximumSize(new Dimension(150, 25));
    toolbar.add(searchBar);

    // Tabbed pane for feature categories
    JTabbedPane tabbedPane = new JTabbedPane();
    tabbedPane.setPreferredSize(new Dimension(150, 300)); // Adjust the size as needed

    // Add tabs
    tabbedPane.addTab("General", createPanel("General Features"));
    tabbedPane.addTab("Structural", createPanel("Structural Features"));
    tabbedPane.addTab("Kitchen", createPanel("Kitchen Features"));
    tabbedPane.addTab("Bedroom", createPanel("Bedroom Features"));
    tabbedPane.addTab("Dining Room", createPanel("Dining Room Features"));
    tabbedPane.addTab("Living Room", createPanel("Living Room Features"));
    tabbedPane.addTab("Bathroom", createPanel("Bathroom Features"));

    toolbar.add(tabbedPane);

    return toolbar;
}

```

```

/**
 * Creates a simple JPanel with a label for the specified category.
 */
private JPanel createPanel(String category) {
    JPanel panel = new JPanel();
    panel.add(new JLabel(category));
    return panel;
}

```

Edited the dimensions for the tabbedPane to fit the application size better.

I told ChatGPT "Inside the Structural Features tab there should be three buttons : 'Wall', 'Window', and 'Door'"

Chatgpt modified the createPanel method to add these buttons.

```
private JPanel createPanel(String title) {
    JPanel panel = new JPanel(new BorderLayout());

    // Add your specific features or buttons based on the title
    if (title.equals("Structural Features")) {
        JPanel structuralPanel = new JPanel(new GridLayout(3, 1));

        JButton wallButton = new JButton("Wall");
        JButton windowButton = new JButton("Window");
        JButton doorButton = new JButton("Door");

        structuralPanel.add(wallButton);
        structuralPanel.add(windowButton);
        structuralPanel.add(doorButton);

        panel.add(structuralPanel, BorderLayout.CENTER);
    } else {
        // Add other features or buttons for different categories if needed
        // For other categories, you may want to customize the content accordingly
        // For example, you might add different buttons or components.
    }

    return panel;
}
```

Do the same thing with the following buttons in the specific tabs:

Dining Room Features : Table, Chair

Living Room Features: Couch, Sofa, TV

Bathroom Features: Shower, Toilet, Sink

Kitchen Features: Sink, Counter, Washing Machine

Bedroom Features: Bed, Desk, Chair

I followed up with the prompt

And it gave the createPanel method code:

```
private JPanel createPanel(String title) {
    JPanel panel = new JPanel(new BorderLayout());

    // Add your specific features or buttons based on the title
    JPanel specificPanel = new JPanel(new GridLayout(3, 1));

    if (title.equals("Dining Room Features")) {
        JButton tableButton = new JButton("Table");
        JButton chairButton = new JButton("Chair");
        specificPanel.add(tableButton);
        specificPanel.add(chairButton);
    } else if (title.equals("Living Room Features")) {
        JButton couchButton = new JButton("Couch");
        JButton sofaButton = new JButton("Sofa");
        JButton tvButton = new JButton("TV");
        specificPanel.add(couchButton);
        specificPanel.add(sofaButton);
        specificPanel.add(tvButton);
    } else if (title.equals("Bathroom Features")) {
        JButton showerButton = new JButton("Shower");
        JButton toiletButton = new JButton("Toilet");
        JButton sinkButton = new JButton("Sink");
        specificPanel.add(showerButton);
        specificPanel.add(toiletButton);
        specificPanel.add(sinkButton);
    } else if (title.equals("Kitchen Features")) {
        JButton sinkButton = new JButton("Sink");
        JButton counterButton = new JButton("Counter");
        JButton washingMachineButton = new JButton("Washing Machine");
        specificPanel.add(sinkButton);
        specificPanel.add(counterButton);
        specificPanel.add(washingMachineButton);
    } else if (title.equals("Bedroom Features")) {
        JButton bedButton = new JButton("Bed");
        JButton deskButton = new JButton("Desk");
        JButton chairButton = new JButton("Chair");
        specificPanel.add(bedButton);
        specificPanel.add(deskButton);
        specificPanel.add(chairButton);
    } else if (title.equals("Structural Features")) {
        JButton wallButton = new JButton("Wall");
        JButton windowButton = new JButton("Window");
        JButton doorButton = new JButton("Door");
        specificPanel.add(wallButton);
        specificPanel.add(windowButton);
        specificPanel.add(doorButton);
    }

    panel.add(specificPanel, BorderLayout.CENTER);

    return panel;
}
```

Which was modified slightly to include a final else statement adding a panel with the given title parameter should none of the conditions checked be met.

Moving on to developing the grid canvas I told ChatGPT

Currently we are setting up the layout of the application before implementing features. The application has a menu bar on the north edge and a feature menu on the left side. Add code so that the remainder of the application, the large portion in the bottom right corner, is a gride with a white background and black lines creating the grid.

And it gave code which one implemented yielded no noticeable change. After prompting ChatGPT as to why this might be and asking it to ensure the grid can still be drawn on with the mouse it gave this:

All of these changes were made to the App.java file. I refactored by moving createPanel into a PanelCreation.java file, setupMenuBar into a MenuBarSetup.java file, and setupFeatureMenu into a FeatureMenuSetup.java file each with their own class. App.java was edited to call setupMenuBar as such:

```
// call setupMenuBar to create the top application menu bar
private void setupMenuBar(JPanel gridPanel) {
    JMenuBar menuBar = MenuBarSetup.setupMenuBar();
    setJMenuBar(menuBar); // Set the menuBar for the application
}
```

panelCreation was edited to return the JPanel it creates which can then be added to the toolbar.

```
private void initUI() {
    canvas = new BufferedImage(800, 600, BufferedImage.TYPE_INT_ARGB);
    clearCanvas();

    JPanel mainPanel = new JPanel(new BorderLayout());

    // Setting up feature menu toolbar
    JToolBar toolbar = setupFeatureMenu();
    mainPanel.add(toolbar, BorderLayout.WEST);

    JPanel gridPanel = new JPanel() {
        @Override
        protected void paintComponent(Graphics g) {
            super.paintComponent(g);
            // Draw grid lines with a white background
            g.setColor(Color.WHITE);
            g.fillRect(0, 0, getWidth(), getHeight());

            g.setColor(Color.BLACK);

            int gridSize = 40; // You can adjust the grid size as needed
            int width = getWidth();
            int height = getHeight();

            // Draw vertical grid lines
            for (int x = 0; x <= width; x += gridSize) {
                g.drawLine(x, 0, x, height);
            }

            // Draw horizontal grid lines
            for (int y = 0; y <= height; y += gridSize) {
                g.drawLine(0, y, width, y);
            }

            // Draw the existing canvas on top of the grid
            g.drawImage(canvas, 0, 0, null);
        }
    };

    gridPanel.setPreferredSize(new Dimension(800, 600));
    gridPanel.addMouseListener(new MouseAdapter() {
        @Override
        public void mousePressed(MouseEvent e) {
            lastPoint = e.getPoint();
        }
    });

    gridPanel.addMouseMotionListener(new MouseMotionAdapter() {
        @Override
        public void mouseDragged(MouseEvent e) {
            drawLine(lastPoint, e.getPoint());
            lastPoint = e.getPoint();
            gridPanel.repaint(); // Repaint the gridPanel to reflect the drawing
        }
    });

    mainPanel.add(gridPanel, BorderLayout.CENTER);

    add(mainPanel);
    setupMenuBar();
    pack();
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLocationRelativeTo(null);
}
```