Java-Draw Project

To draw in Java, a Java Frame and a Java Panel is needed. The Java Frame makes the window while the Java Panel is placed on the Java Frame and objects are drawn onto the panel.

JFrame

- javax.swing.JFrame -

Some of the basic commands to setup a JFrame are:

```
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

This command sets the JFrame to **exit on close**.

```
this.setSize(width,height);
```

This command **sets the size** of the JFrame to the desired width and height.

```
this.setBackground(Color.color);
```

This command **sets the colour of the background** of the JFrame.

```
this.setVisible(true);
```

This command creates and **makes the JFrame window appear** on the screen.

```
this.setLocationRelativeTo(null);
```

This command **locates the JFrame to the centre** of the screen. And most important of all:

```
this.getContentPane().add(new DrawPanel());
```

adds the JPanel to the content pane.

JPanel

- javax.swing.JPanel -

To setup the JPanel, the same commands to setup the background color and size of the JFrame are needed:

```
this.setSize(width,height);
```

and

```
this.setBackground(Color.color);
```

Objects to be drawn onto the JPanel are then defined as functions. These paint functions *do not* need to be called.

Drawing Lines:

```
protected void paintComponent(Graphics g){
      g.setColor(Color.color);
      \\ this command sets the colour of the line
      Graphics2D g2D = (Graphics2D) g;
      g2D.setStroke(new BasicStroke(10));
      \\ this command sets the width of the stroke
      g2D.drawLine(x1,y1,x2,y2);
      \\ this command sets where the line is to be drawn
}
```

Adding Menus

Menus are added onto the JFrame object. There are 3 Java objects that are associated with Menus: **JMenuBar, JMenu** and **JMenuItem**. These are all part of the **javax.swing** packages.

JMenuBar is the menu bar on the top of the window, JMenu is the menu objects located on the menu bar and JMenuItem are the list of items that are shown when JMenu objects are clicked. These are implemented into a *private void* method named addMenus(), which is called in the default setup method using the command:

```
this.addMenus();
```

JMenu Objects

- javax.swing.JMenu -

Firstly, we will create JMenu objects:

```
JMenu button_name = new JMenu("Button Name");
```

then we will **add action listeners**:

```
Button_name.addActionListener(new NameOfListener());
```

where these action listeners are defined separately as a private class at the bottom of the file:

- java.awt.event.ActionListener -

```
private class NameOfListener implements ActionListener{
      public void actionPerformed(ActionEvent action){
            \\ do something here;
      }
}
```

Lastly, we **add shortcuts** to these JMenu objects using the following commands:

For Ctrl + KEY shortcuts:

```
Button_name.setAccelerator(KeyStroke.getKeyStroke(java.awt.eve
nt.KeyEvent.VK_KEY, java.awt.Event.CTRL_MASK));
```

For Alt + key shortcuts:

```
Button_name.setMnemonic('KEY');
```

JMenuBar Objects

After JMenu objects are created and their corresponding action listeners are added, we can create a JMenuBar object and add JMenu objects to it:

```
JMenuBar menu_bar = new JMenuBar();

menu_bar.add(menu_name);
```

JMenuItem Objects

JMenuItems objects are created and added onto JMenu objects in the same way as JMenu objects are created and added to JMenuBar objects:

```
menu_name.add(menu_item_name);
```

Adding Mouse Listeners/ Mouse Motion Listeners

MouseListener

```
- java.awt.event.MouseEvent -
- java.awt.event.MouseListener -
```

The MouseListener class handles mouse events such as the mouse being **pressed**, **released**, **clicked** on the DrawPanel object as well as when the mouse **enters** and **exits** the DrawPanel object.

Mouse listeners are defined separately as a private class implementing MouseListener at the bottom of the file with public void functions:

```
mousePressed(MouseEvent e)

mouseReleased(MouseEvent e)

mouseClicked(MouseEvent e)

mouseEntered(MouseEvent e)

mouseExited(MouseEvent e)
```

MouseMotionListener

```
- java.awt.event.MouseMotionListener -
```

The MouseListener class handles mouse events where the mouse is **moved (without being pressed)** or **dragged (when pressed)**.

Mouse motion listeners are defined separately as a private class implementing MouseMotionListener at the bottom of the file with public void functions:

```
mouseMoved(MouseEvent e)

mouseDragged(MouseEvent e)
```