Comic Lab

Namuuntuul Bat-Erdene
Mosfiqur Rahman

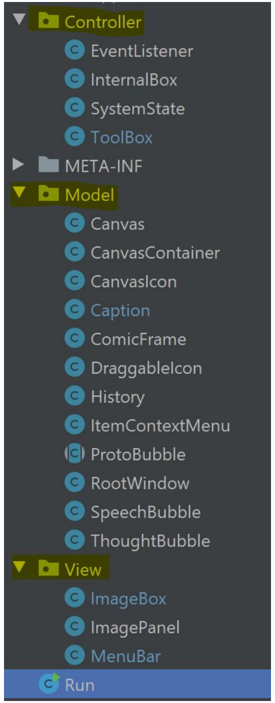
IDEA/USERS

Comic Lab allows users to create comic strips fast and efficiently. When people have quick stories to convey, our system allows users to easily create their story with different types of provided assets: background, avatars, and speech bubbles. Our approach to GUI was very minimalistic. Users can select different types of background or change its color, and add avatars onto it, as well as speech bubbles to narrate the story, as well as adding own image. Making use of those assets, user can create, edit and save their stories to share with their friends, on social media, etc.

The main audience for our system will be mostly people who are familiar with comic or manga, and would love to make tell their own story in that kind of format.

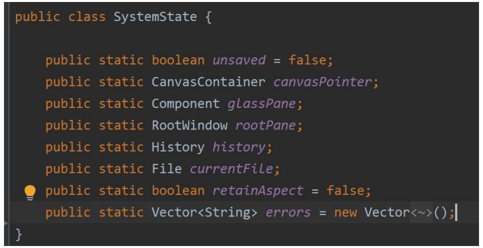
IMPLEMENTATION

For implementation we followed MVC:



```
▼ Controller
    EventListener
    InternalBox
    SystemState
    ToolBox
▶ META-INF
▼ Model
    Canvas
    CanvasContainer
    CanvasIcon
    Caption
    ComicFrame
    DraggableIcon
    History
    ItemContextMenu
    ProtoBubble
    RootWindow
    SpeechBubble
    ThoughtBubble
▼ View
    ImageBox
    ImagePanel
    MenuBar
  Run
```

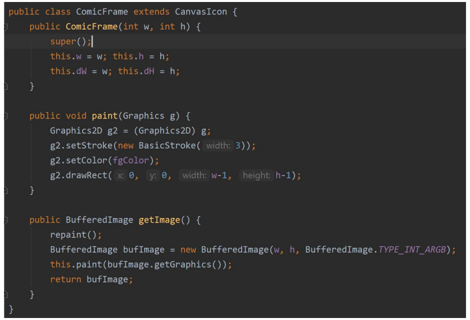
For controllers we have EventListener class, as well SystemState class :



```java
public class SystemState {

    public static boolean unsaved = false;
    public static CanvasContainer canvasPointer;
    public static Component glassPane;
    public static RootWindow rootPane;
    public static History history;
    public static File currentFile;
    public static boolean retainAspect = false;
    public static Vector<String> errors = new Vector<~>();
}
```

For model we have classes that creates all the necessary components for our apps: Canvas, Frame, Windows, etc:

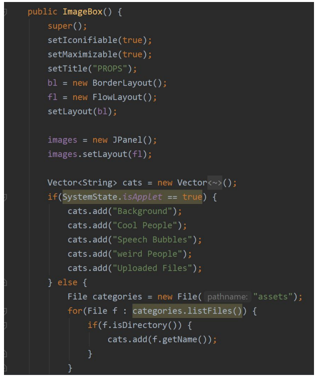


```java
public class ComicFrame extends CanvasIcon {
    public ComicFrame(int w, int h) {
        super();
        this.w = w; this.h = h;
        this.dW = w; this.dH = h;
    }

    public void paint(Graphics g) {
        Graphics2D g2 = (Graphics2D) g;
        g2.setStroke(new BasicStroke( width: 3));
        g2.setColor(fgColor);
        g2.drawRect( x: 0,  y: 0,  width: w-1,  height: h-1);
    }

    public BufferedImage getImage() {
        repaint();
        BufferedImage bufImage = new BufferedImage(w, h, BufferedImage.TYPE_INT_ARGB);
        this.paint(bufImage.getGraphics());
        return bufImage;
    }
}
```

For out view we have classes that provide with assets, and menubar:



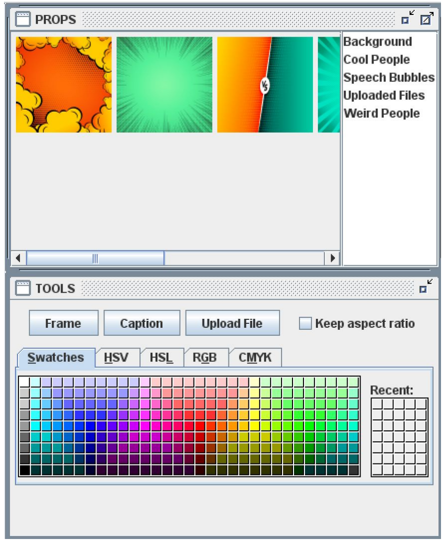
```java
    public ImageBox() {
        super();
        setIconifiable(true);
        setMaximizable(true);
        setTitle("PROPS");
        bl = new BorderLayout();
        fl = new FlowLayout();
        setLayout(bl);

        images = new JPanel();
        images.setLayout(fl);

        Vector<String> cats = new Vector<~>();
        if(SystemState.isApplet == true) {
            cats.add("Background");
            cats.add("Cool People");
            cats.add("Speech Bubbles");
            cats.add("weird People");
            cats.add("Uploaded Files");
        } else {
            File categories = new File( pathname: "assets");
            for(File f : categories.listFiles()) {
                if(f.isDirectory()) {
                    cats.add(f.getName());
                }
            }
```

SYSTEM FEATURES

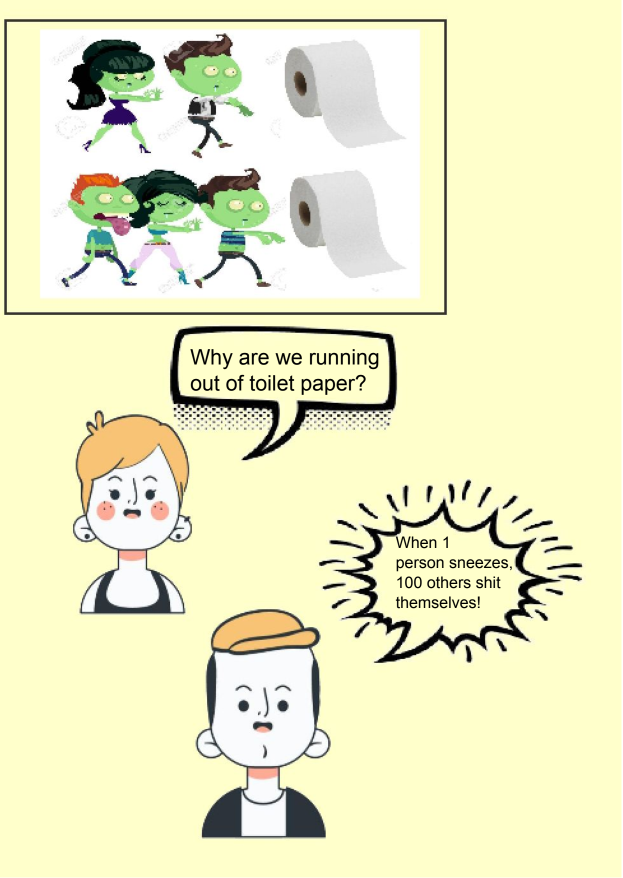
One of our main system features is being minimalist, we provide with minimal necessary assets for users to create short comic. Hence our, GUI provide users with only the following (Props & Tools):



Props box have different types of assets users can use, as well user uploaded ones.

Tools Box have frame, caption and option to upload file which will be saved to “Uploaded Files” in props.

Users can create short strips like this picture, conveying their thoughts about what stupid things people are doing in a funny way:


Why are we running out of toilet paper?
When 1 person sneezes, 100 others shit themselves!