

## CS124 Assignment – A Java Comic Book Maker

By:

- Owain Jones [odj@aber.ac.uk]
- Dean Allen [deal@aber.ac.uk]
- Matthew Hathaway [mah19@aber.ac.uk]

### Who did what:

- **Owain:** majority of implementing code and debugging.
- **Dean:** testing; showing screenshots of progress, filling in the test table, lots of ideas, doing majority of documentation.
- **Matthew:** drew us lots of assets (by hand, in crayon, scanned then meticulously edited on computer to give our comics a nice theme). Explained algorithms to Owain, being better at maths in general. Came up with great idea of creating stick man body parts which could be combined to create huge variety of stick men (in [XKCD](#) style)

### What we all did:

- Brainstormed ideas
- Gave each other constant feedback on the work

### How to run the program:

Extract everything from the zip file, double-click comicbookmaker.jar in extracted directory.

‘Test.png’ is an example image we created with our program.

The JAR will only run if the ‘assets’ folder exists, which is full of subfolders which are in turn full of images – so it will run in our project’s root directory, but not outside this folder. This is intended, and the external dependency on images means new images/categories can be added very easily.

There is an example project saved in the saves/ folder, it can be viewed by using File > Open.

## Problem Definition

A user wants a simple way to create comic strips (for example, a webcomic). A popular webcomic artist could draw hundreds of strips in their comic's lifetime. Using conventional graphics editing packages usually means drawing out characters and props by hand every comic.

Therefore, we should provide a program tailored towards these people, which allows them to quickly create comic strips (or other types of picture/diagram) using already created props, with a nice, intuitive drag and drop interface.

## How we worked together (and our design & discussion)

How we worked together quite well in some ways, but there was bound to be some conflicts both Dean and Mathew are last minute workers and Owain is someone who starts to work early on their projects, so when Both Dean and Mathew finally got their game Faces on Owain had already done a lot of the work, this meant they had to play catch up to make the work load a bit more even. We had attended some group meeting where we discussed Plans for are project and had a Mind map while eating Chinese and having a laugh we did this twice for about 3 to 5 hours both times, it was not really concentrated work but we had some really great plans that we have managed and implemented in to the program we also enjoy the time and it brought the group together well.

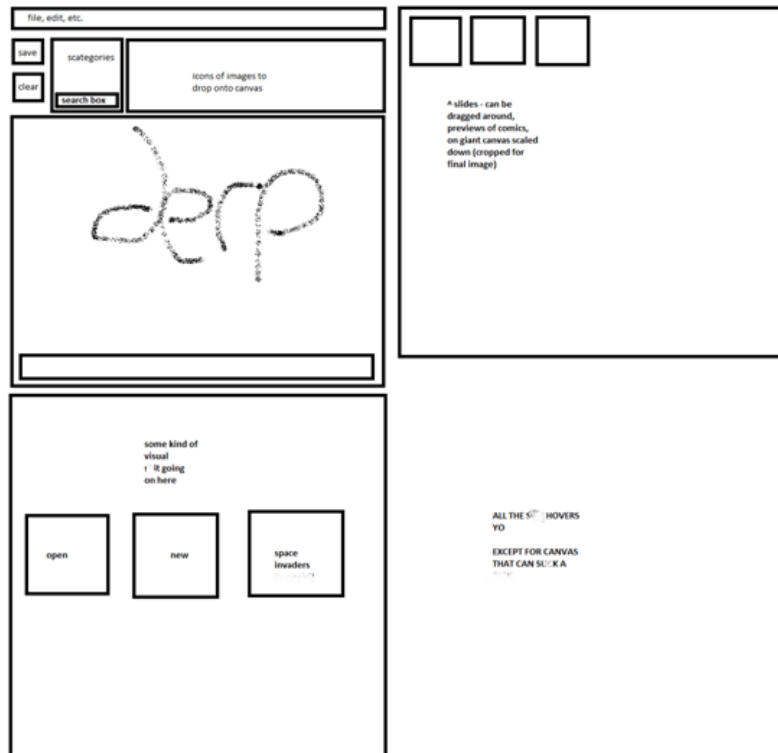
In conclusion I think are group worked very well and was a very close knitted Group

**Project notes**, from the beginning of our project, are below. We started with a bullet-point list, which we saved as a Google Document so we could all view it and add to it later. Important features we added to a todo list at the top of Run.java in the sourcecode.

- Drag and drop
- Order – to the front, back etc (z-index)
- List box to change category
- Different ways to view canvas (observer-observable)
- Pick a background colour
- Do shapes like SQUARES AND CIRCLES
- AND SPEECH BUBBLES is a must
- Animation - key frame (google: interpolation)
- ORGANIZE STRIP using other view - previews of panels - preview is giant canvas you can drag the panels around on, final image crops to top left + bottom right end
- Saving and loading (serializable)
- MAIN MENU PAGE
- what images to get/draw for program -
  - crayon everything
  - in transparent .png
- search for files
- be able to write text and position it

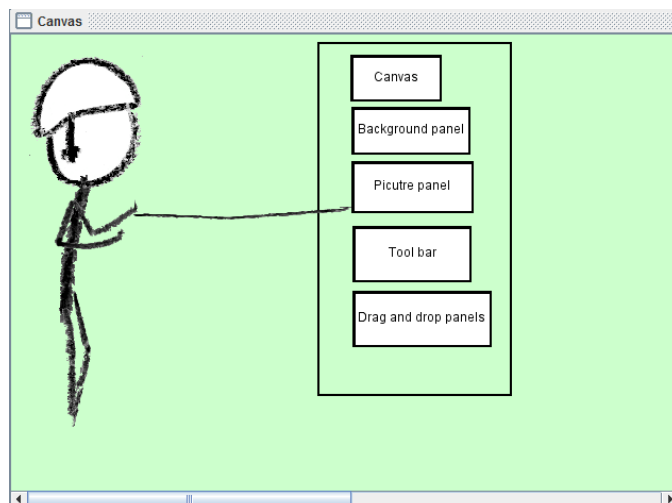
- rotate, scale on fly
- button for paint blob, draw on background

**user interface – what we originally thought it should look like, before we started writing any code:**



We have blanked out the obscenities used (as 3 students brain storming there was improper language used)

Dean did a use-case diagram, done in our program afterwards, to show what ours might look like had we done a use-case diagram before writing all of our code:



## Testing strategy

Are strategy for testing was running the program after trying to add every new action we would run then modify it, eventually it would run and we would note down the bugs we came across as we made more actions to perform

We then documented the bugs and fixed them in the testing when program hit a milestone before continuing on with the programming of the new actions

## How to use different Aspects of our program

### Getting started

To start are program double click “comicbookmaker.jar”, once the program starts the best thing to do is maximize the Main pane and arrange the other panes where you want while resizing them, once you are satisfied with their arrangement just start dragging images from the panes or changing colours,

### Nifty top tips

- Any objects on the canvas can be right clicked and flip vertically and horizontally including Text bubble and u can get backwards text ☺.
- Resizing object is a simple task thanks to are amazing program that spawn a black box in the bottom corner of the object when selected you can even select to keep in aspect ratio with a simple tick of a box.
- To Move an already placed object on the canvas just reselect it and drag it to its new desired designation.
- Want to make your canvas like a comic, well you can with are Frames just click the frame button drop one of the canvas and resize BAM! instant comic frames.
- There is a button to make your image back to its original size if you accidentally make it a bit big/small and can't get it back; you do this by right clicking the image needed.
- Texts have never been more fun than with the fantastic capabilities with our program, not only can you choice between 3 amazing different text images consisting of speech, taught or text box but it auto resizes the text for ease of your use.
- Not going to add any more text or change the background colour? Then why not minimise that pane by clicking the minimise button at the top right of it and have more screen to make art on.
- You made a big mistake? Don't sweat it this program has a undo and redo button as a drop down menu form 'edit' on the Tool bar.
- Having trouble with inspiration why not save your work then come back later and open that file and carry on thanks to are amazing system.
- In case the thumbnail isn't a clear picture, you can hover your mouse over it for a tooltip which will show the filename (which should hopefully be a descriptive one).

## **What we found hard**

We found making the images thumbnails keeping aspect ratio while still on the pane hard as it kept making them look squished or stretched.

Lots of unexpected bugs appeared throughout the process, since Swing uses interfaces and extensions so heavily, sometimes these were hard to track even whilst debugging in Eclipse (NullPointerExceptions everywhere etc).

It was hard to delegate work between people, this is probably because we started writing code before properly designing anything, but it seems to have worked out well since our program has lots of features.

## **Our group evaluation**

We think that, whilst our group didn't go through the design process very well (complete lack of use-case/UML diagrams), we made up for it by thinking on our feet and working as a team – we didn't really argument and got on well together, and our program benefited from this, as we believe we've implemented all the required features, plus plenty of extra features to make making comics even easier.

Expected mark: roughly 70%, losing marks for design process, and lack of web applet.

**Our testing is all on the following pages (which exceed the 15-page limit, but most of it is screenshots to go with the testing table).**

| ID   | Requirement | Description   | Inputs  | Expected outputs  | Pass/<br>Fail | Comments  |
|------|-------------|---|---|---|---------------|---|
| A1.1 | N/A         | Testing that the canvas is formed on the main window                  | N/A   | A swing main pane with a canvas pane also formed in it                | P             | The Canvas was lovely and white   |
| A1.2 | Assets file | Testing the image pane form with all assets                           | Files for the different kind of pictures and pictures in them files | A pane with the different files in assets as buttons                  | p             | N/A   |
|      |             |   |   | The different buttons to have pictures load in the pane               | P             | Scroll bars where already automatically made from the coding used   |
| A1.3 | A1.2        | Testing the images can be drag and dropped on to the canvas           | Mouse interactions with swing                                       | Images to be dragged from the image pane when clicked and dragged     | P             | The pictures flicker while moving but moves.<br>The other problem he have is if the main window is moved round the screen |
|      |             |   |   | When dragged over the canvas and the dropped they stick to canvas     | P             | The pictures where dropped there the mouse is even when the main window was moved   |
| A1.4 | A1.2+A1.1   | Testing whether we can drag and move images once placed on the canvas | Mouse interactions with canvas                                      | Items selected will form Boxes around them showing they were selected | p             | N/A   |
|      |             |   |   | Items would follow the mouse once clicked till the mouse is released  | p             | First attempt caused multiple items to be caught on the mouse as it moved but success later on                            |

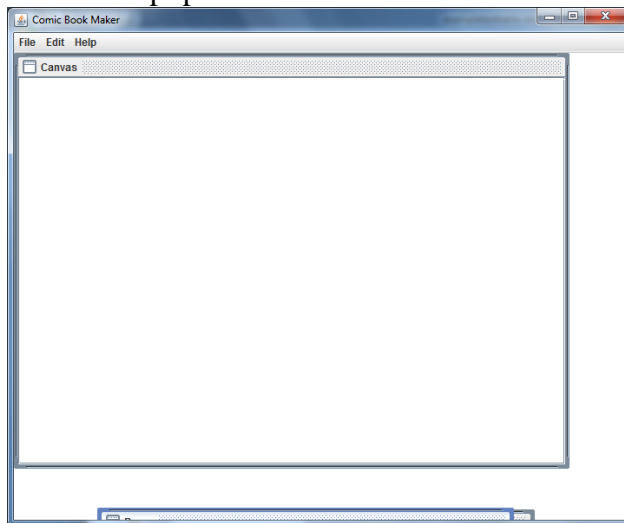
|      |                            |                                     |  |   |           |   |
|------|----------------------------|-------------------------------------|--|---|-----------|---|
| A1.5 | A1.1                       | Resizing placed images              | Mouse interactions with swing                      | Images could be resized as needed on canvas   | P         | This works well but there is flaw of being able to flip the image upside down and back to front making it unable to be used again if released at that point ~now fixed~ |
| A1.6 | History class + menu class | Undo and Redo                       | Button mouse interaction via menu                  | The canvas would either revert to a previous version when Undo is click or convert back to a later version when Redo is clicked | F ~now p~ | They used to work but later on after coding It unfortunately stopped and now does nothing this will hopefully be rectified ~now fixed~                                  |
| A1.7 | Canvas                     | Background colour changing facility | Mouse interaction on the Background changing panel | The background will change to the selected colour from the background changing panel  | P         | The background changes with all the colour choices  |
| A1.8 | N/A                        | Saving                              | Serialisation Mouse interaction                    | The canvas should be saved in to a file   | P         | N/A   |
| A1.9 | N/A                        | Open                                | Serialisation Mouse interaction                    | It calls up one of the saved canvas from the file   | P         | N/A   |
| A2.1 | N/A                        | New                                 | Mouse interaction                                  | Clears canvas so a new beginning can be used  | P         | N/A   |
| A2.2 | N/A                        | Exit                                | Mouse interaction                                  | Closes the program  | P         | N/A   |
| A2.3 | N/A                        | Frames                              | Mouse interaction on the canvas                    | Allows a frame to be places and then resized or moves   | P         | N/A   |
| A2.4 | A1.7+A1.2                  | Minimising the panels               | Mouse interaction                                  | The panels should minimise to the bottom of the Main panel  | P         | N/A   |
| A2.5 | N/A                        | Text                                | Keyboard class                                     | When a text box is drop it should be able to a typed in   | P         | Text can be inverted  |

|      |     |                   |                   |   |   |  |
|------|-----|-------------------|-------------------|---|---|--|
| A2.6 | N/A | Keep aspect ratio | Mouse Interaction | The image should keep its high and with an equal amount apart | P |  |
|------|-----|-------------------|-------------------|---|---|--|



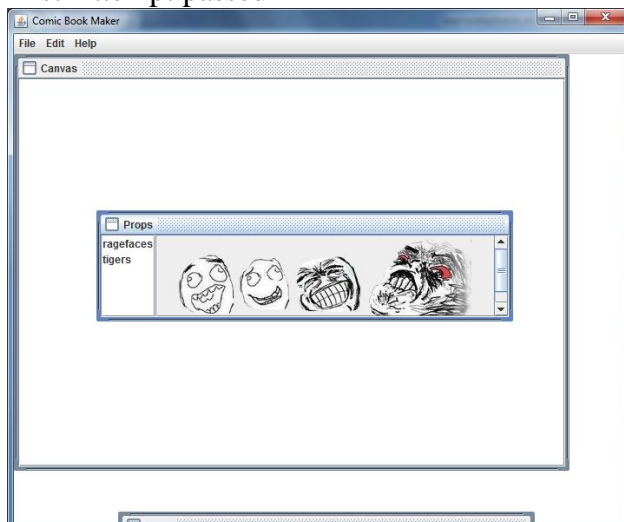
## A1.1 pictures

First Attempt passed



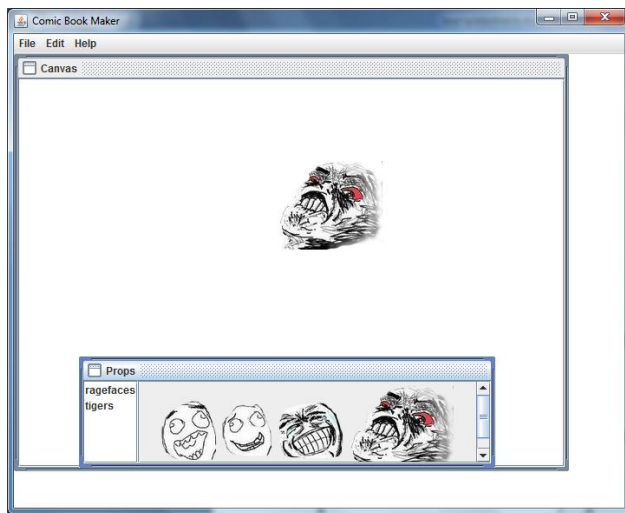
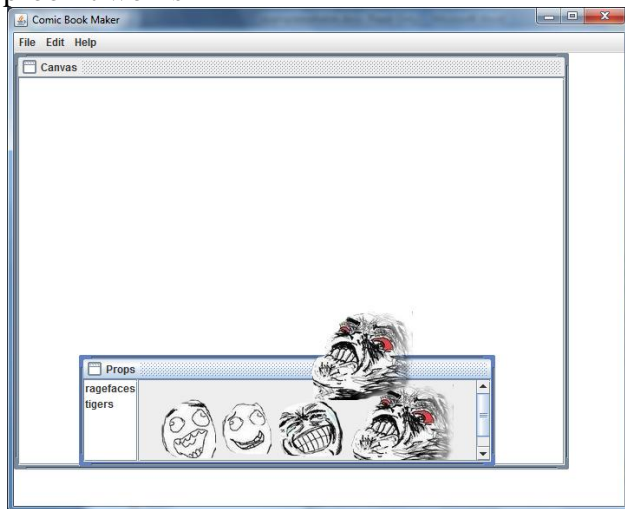
## A1.2

First Attempt passed



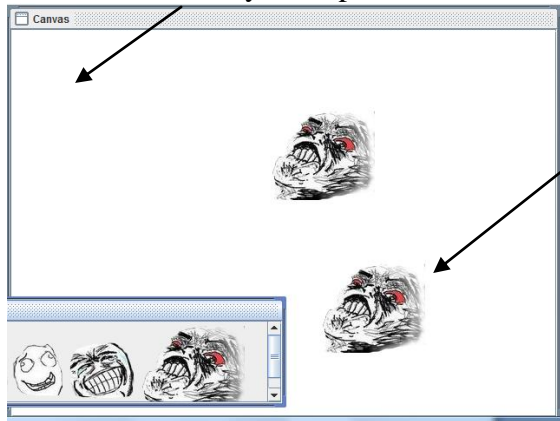
### A1.3

Minor unsolved problems but overall performance is unaffected  
proof it works

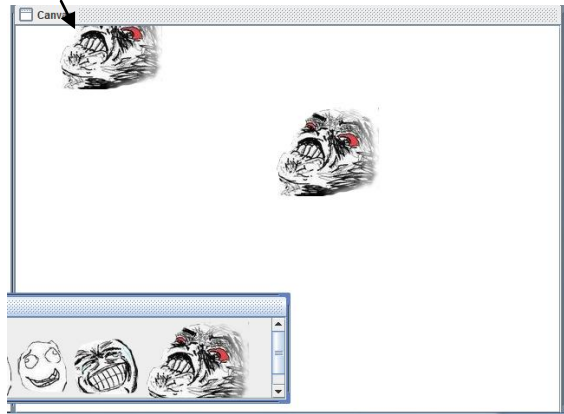


The minor problem

The mouse is here yet the picture is still following down there

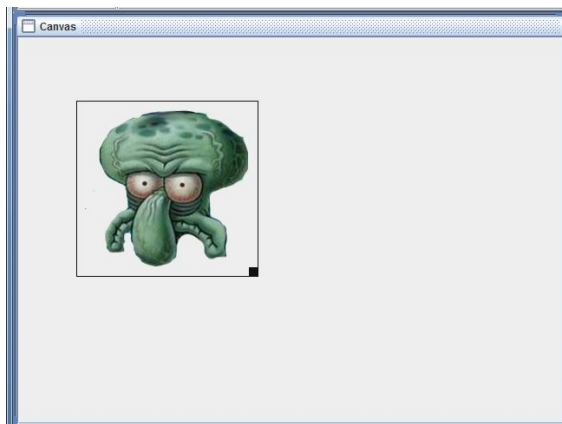
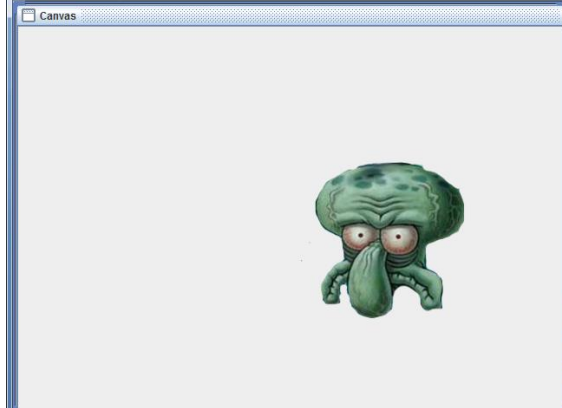


Once the button is released on the mouse the image then jumps to where the mouse was



A1.4

After Bug fix of picking up multiple Images

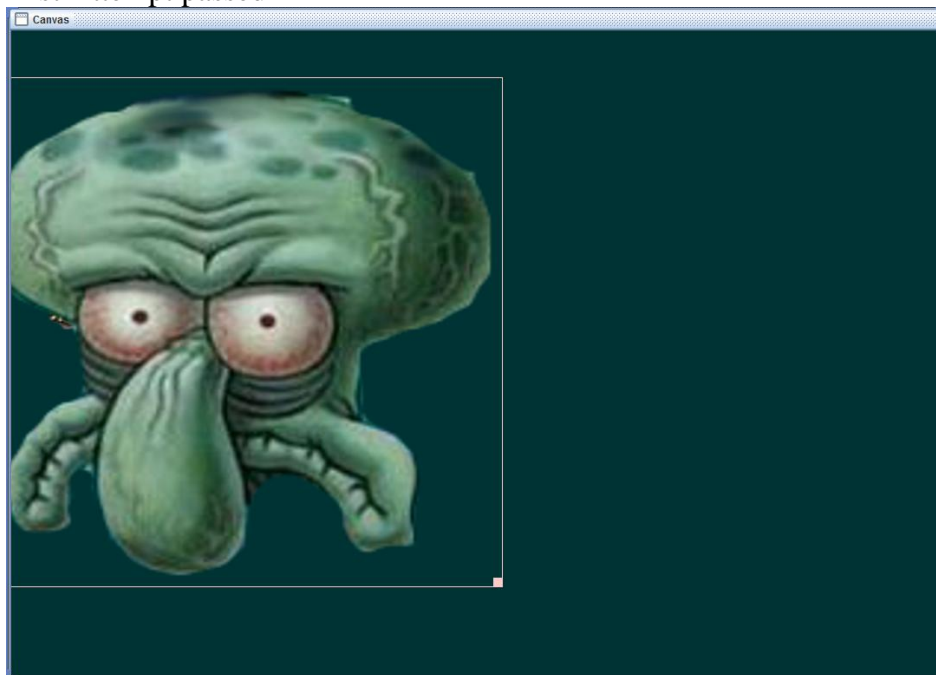


Box has appeared and the image moves with the mouse

A1.5

I will shrink the face with the resize tool in our program

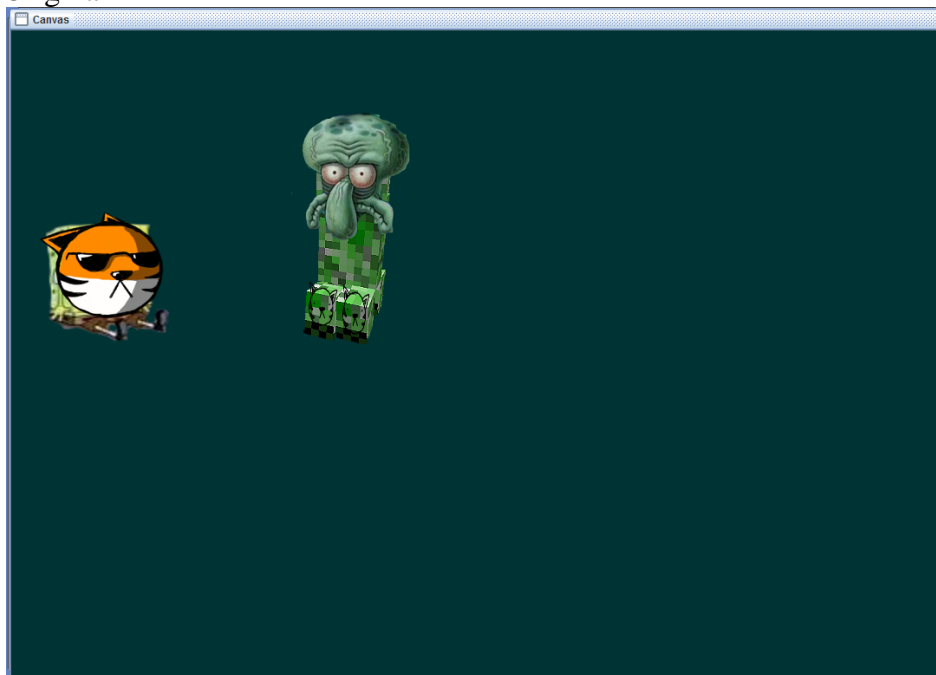
First Attempt passed



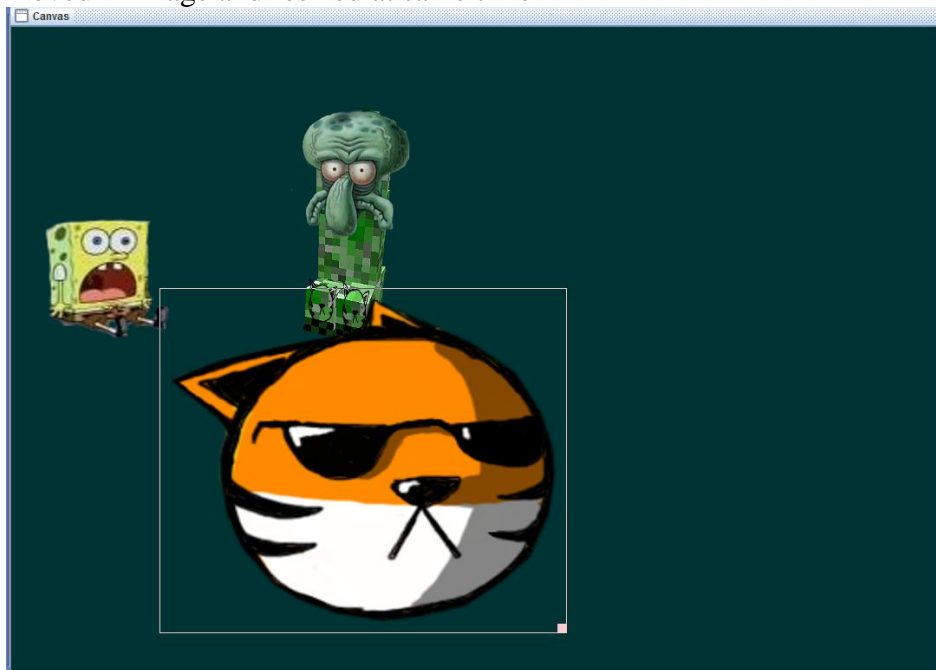
A1.6

First attempt passed

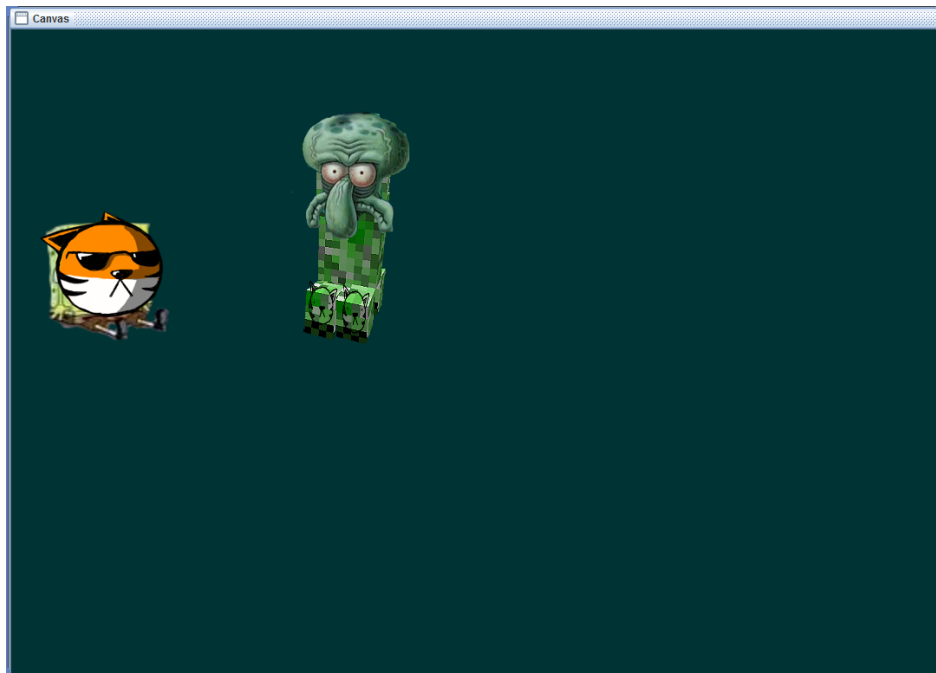
Original



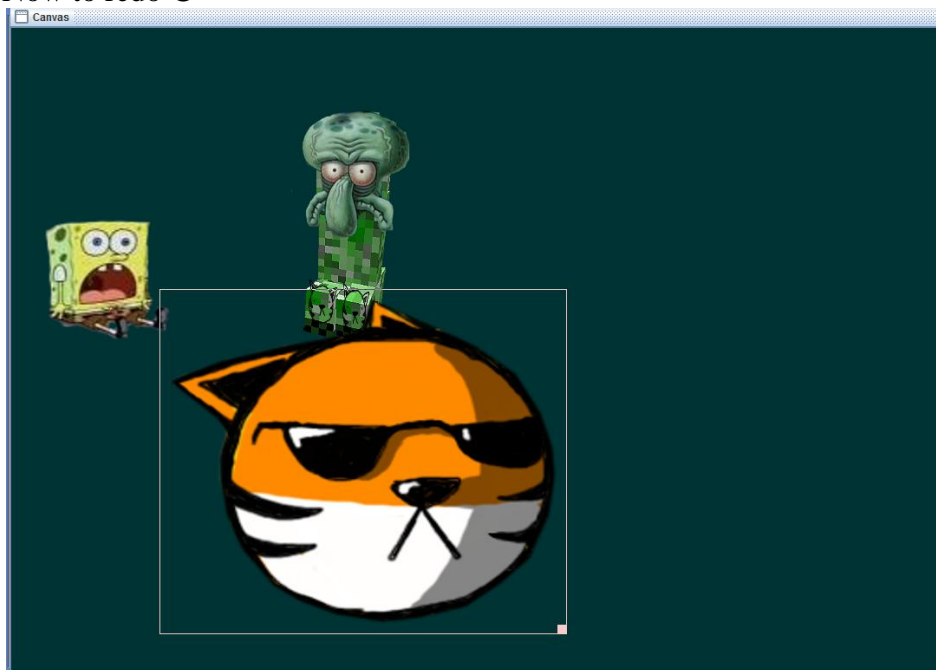
Moved A image and resized at same time



Time to Undo ☺

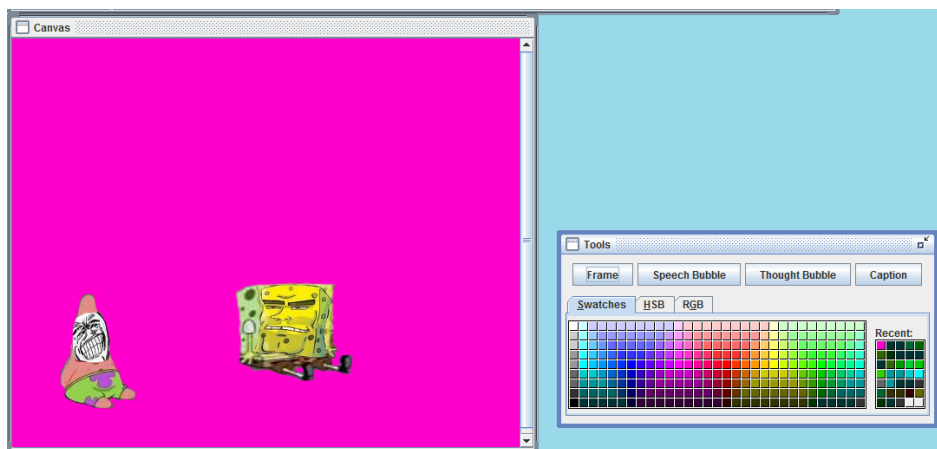
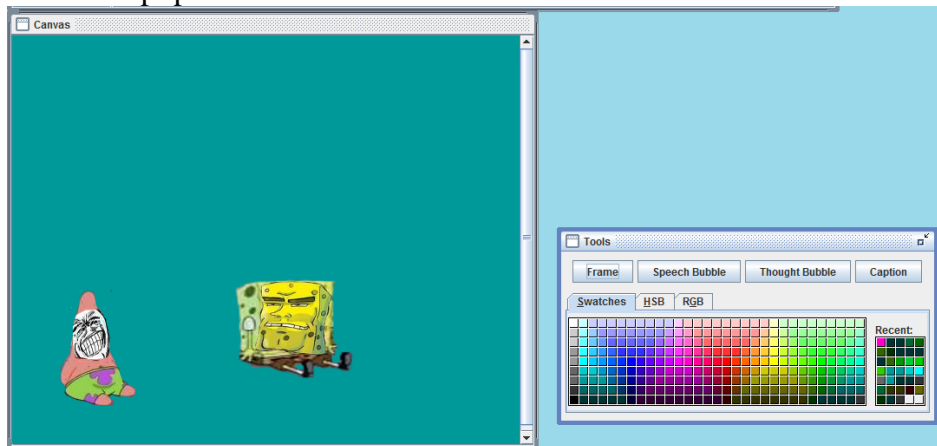


Now to redo ☺



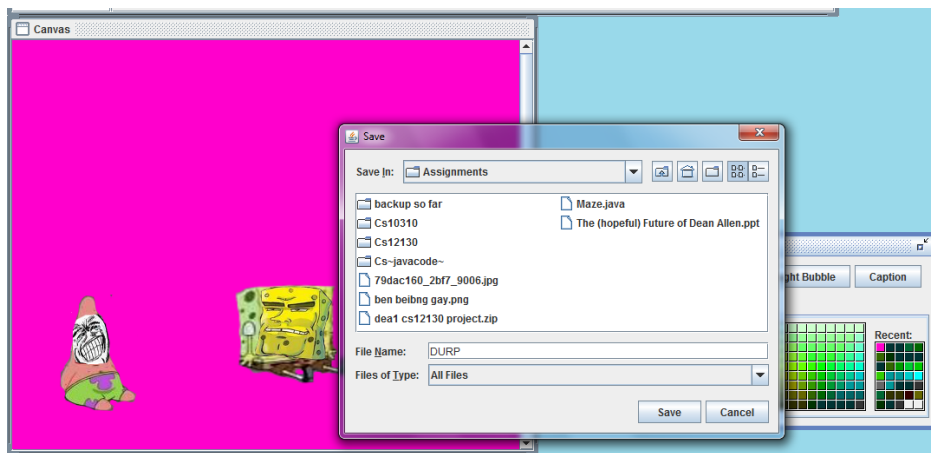
A1.7

First attempt passed

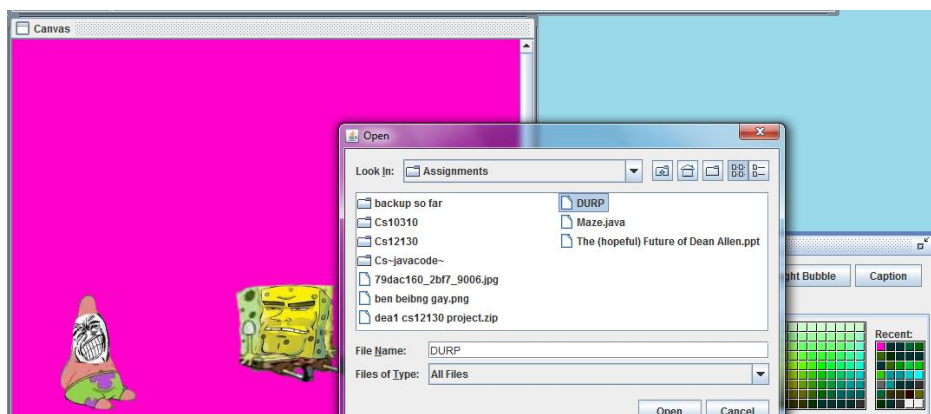




A1.8  
First attempt passed

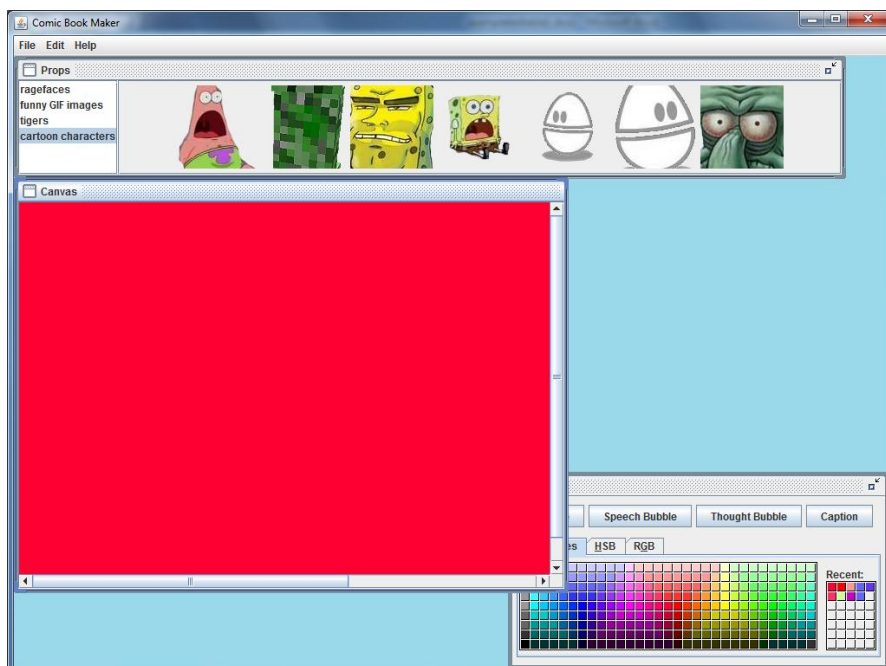
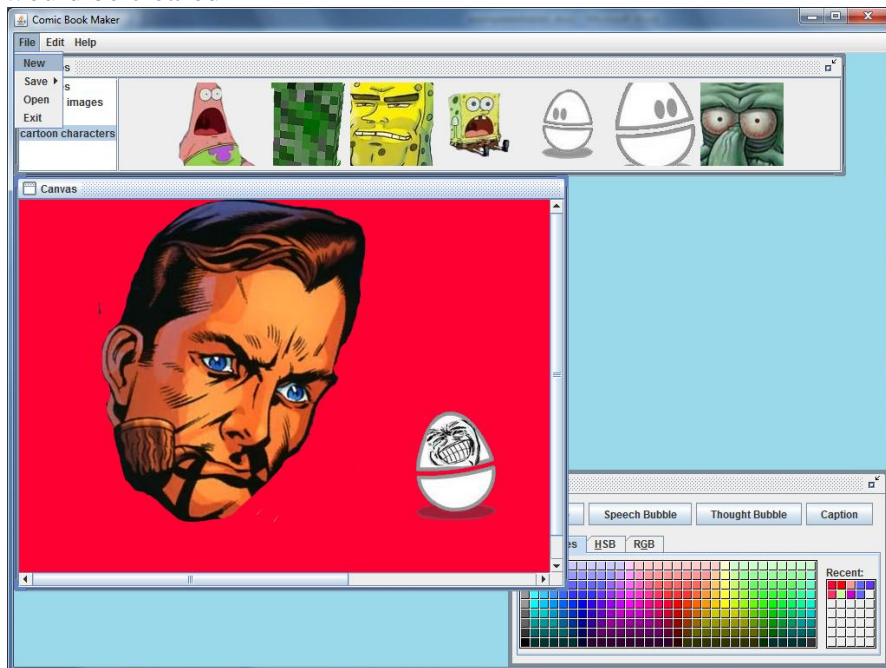


A1.9  
First attempt passed



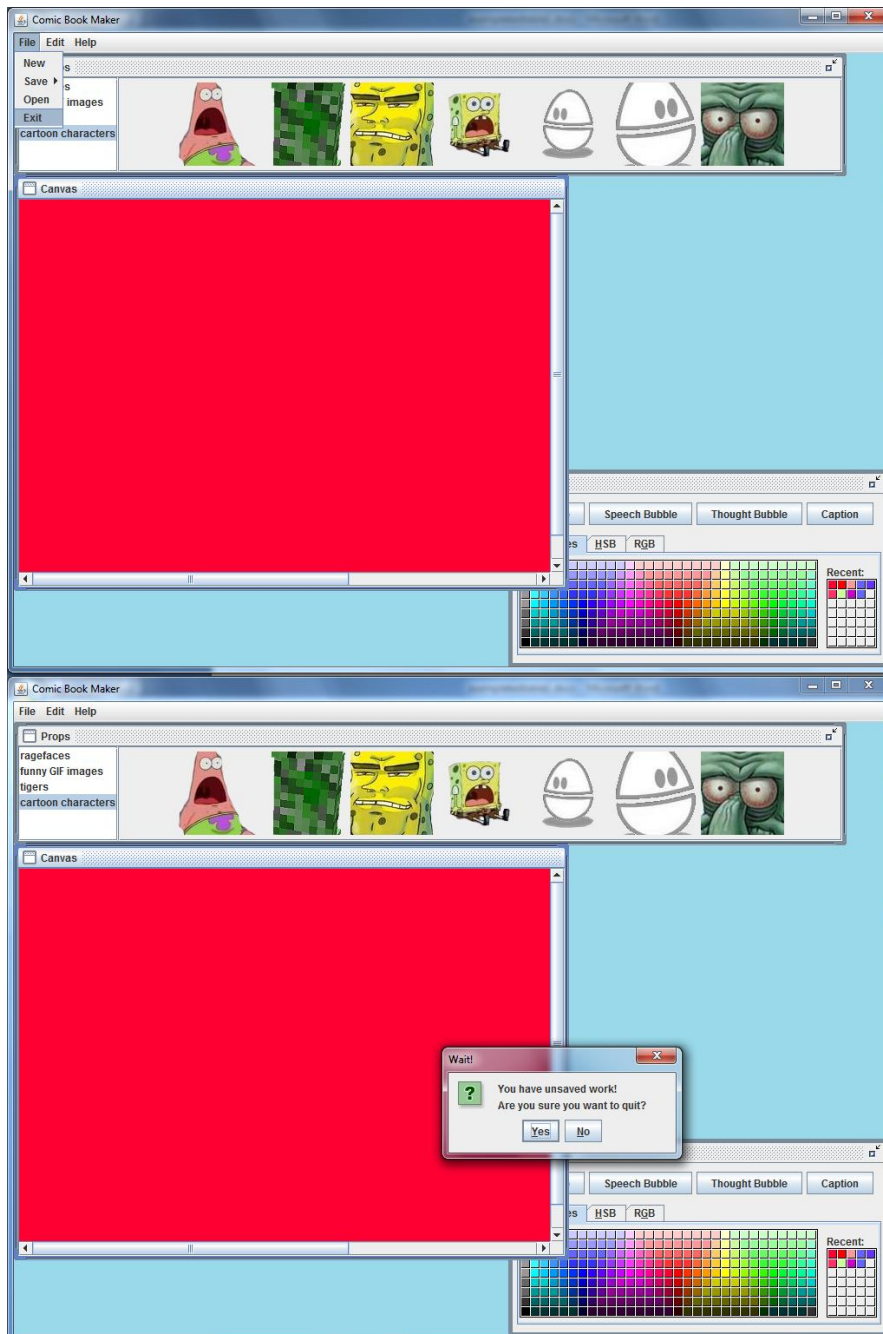
## A2.1

A small bug where the colour does not change but the main thing was the images would be cleared



## A2.2

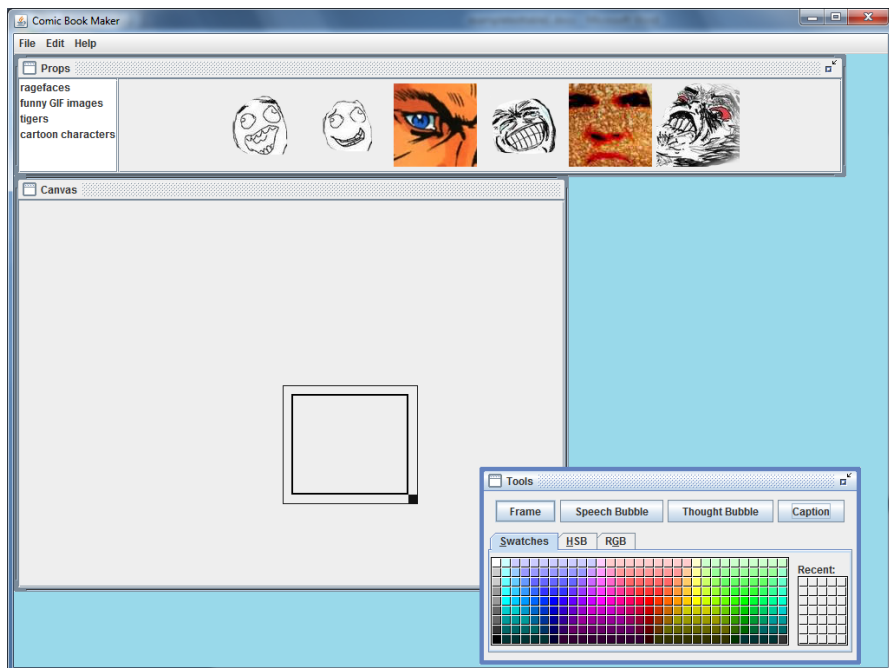
First attempt passed



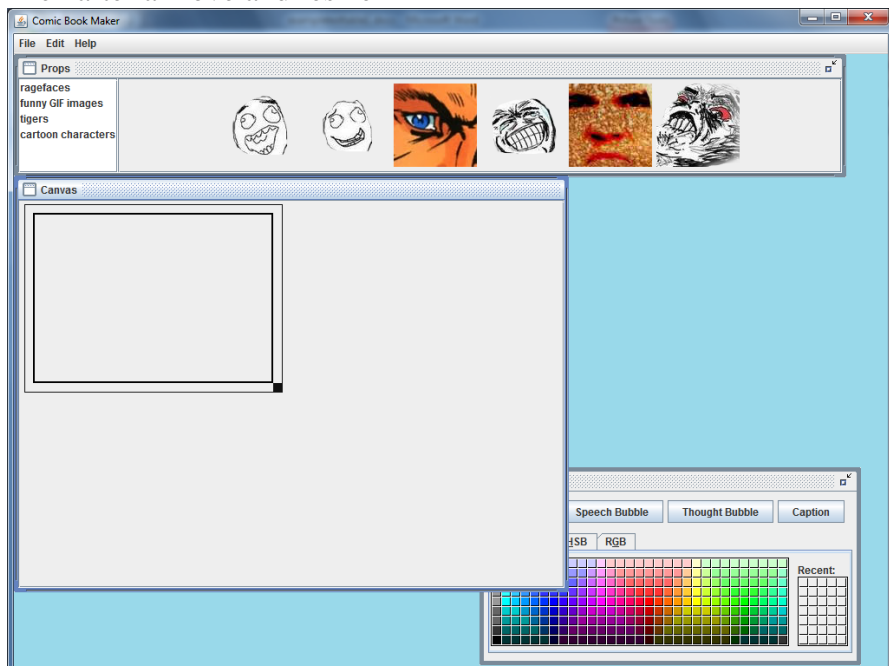
Then it closed 😊

## A2.3

First attempt passed

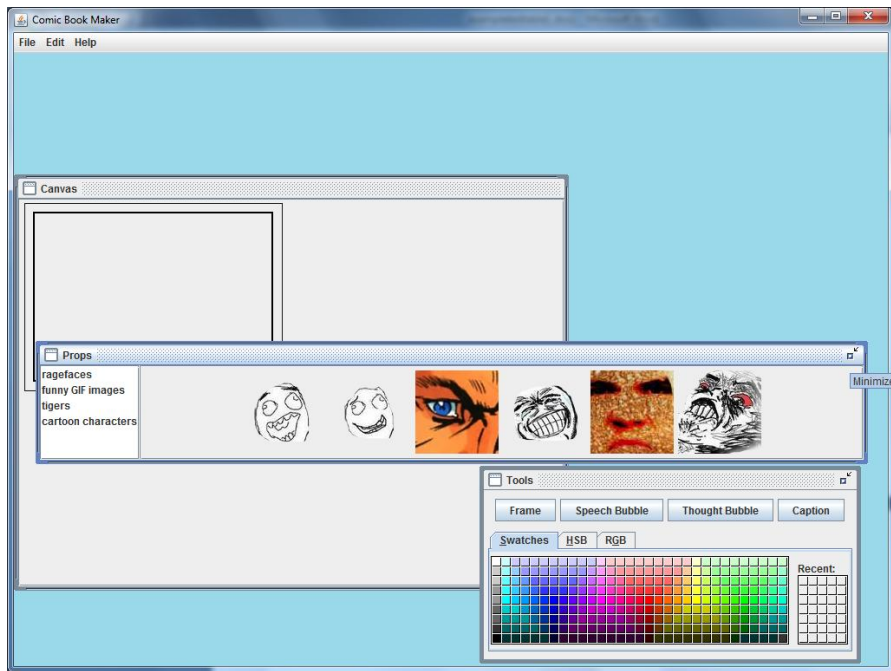


Then after a move and resize

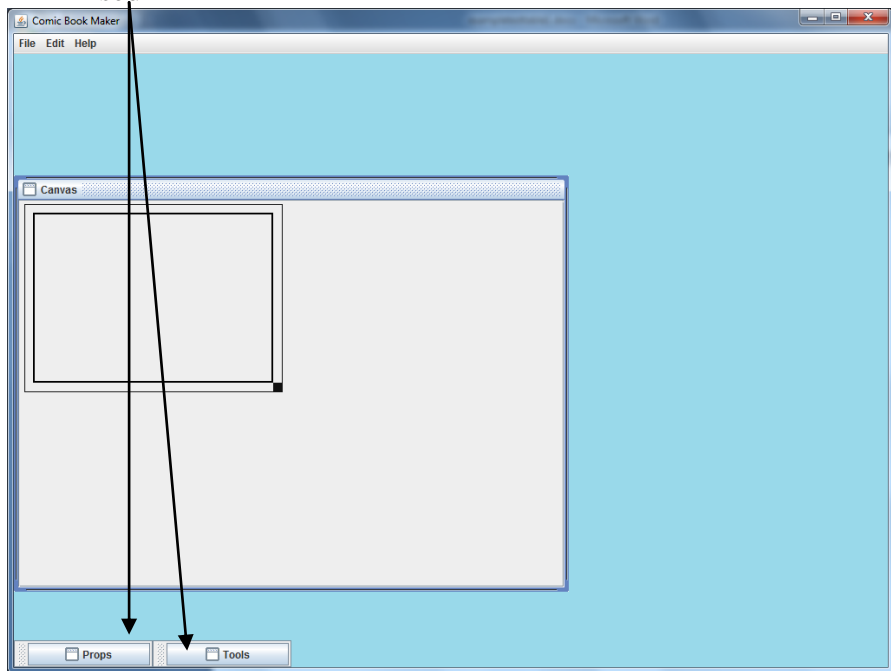


A2.4

First attempt passed

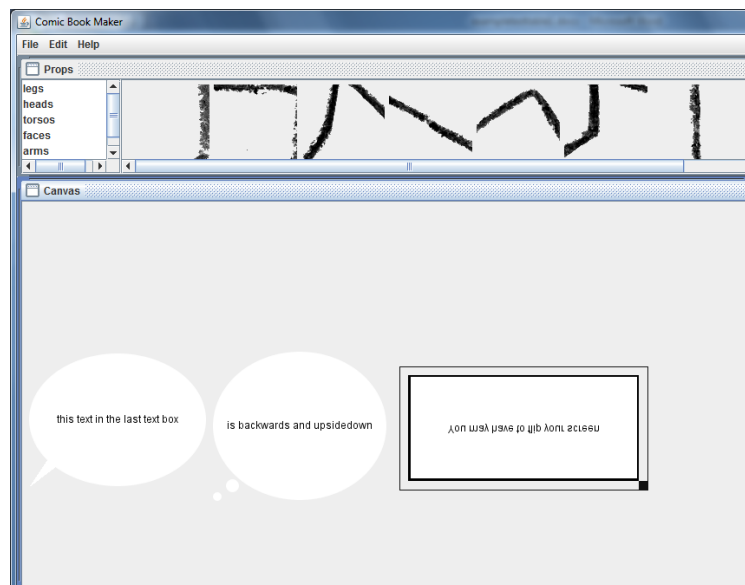
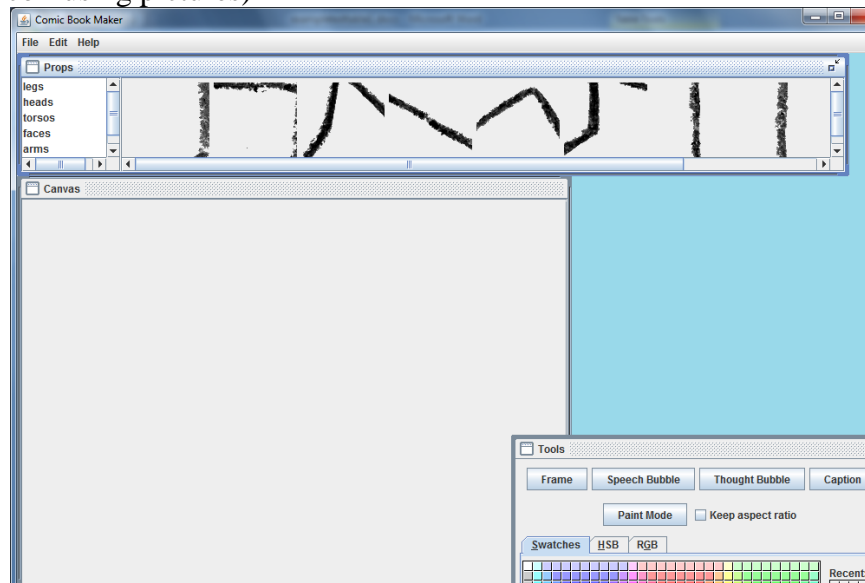


Minimised



## A2.5

First attempt passed with a minor bug as demonstrated (could be used for funny and confusing pictures)



A2.6

First attempt passed

