CAB302 A2

Alexander Brimblecombe

N10009833

Group 471

Contents

[Instructions 3](#_Toc9940877)

[Drawing 3](#_Toc9940878)

[Colours 4](#_Toc9940879)

[Undo / Clear 5](#_Toc9940880)

[I/O 5](#_Toc9940881)

[Architecture 7](#_Toc9940882)

[graphicsManage 7](#_Toc9940883)

[DrawableVector 7](#_Toc9940884)

[FixedPointVector 7](#_Toc9940885)

[Ellipses 7](#_Toc9940886)

[Line 7](#_Toc9940887)

[Point 7](#_Toc9940888)

[Polygon 7](#_Toc9940889)

[Rectangle 8](#_Toc9940890)

[VecFileException 8](#_Toc9940891)

[VecFileManager 8](#_Toc9940892)

[VectorCommand 8](#_Toc9940893)

[gui 8](#_Toc9940894)

[GUI 8](#_Toc9940895)

[VecCanvas 8](#_Toc9940896)

[VecColorButtonPanel 8](#_Toc9940897)

[VecCommandButtonPanel 8](#_Toc9940898)

[VecPainterMenuBar 8](#_Toc9940899)

[Agile Methods 8](#_Toc9940900)

[Deliver Early and Deliver Often 8](#_Toc9940901)

[Working Software 8](#_Toc9940902)

[OOP 9](#_Toc9940903)

[Abstraction 9](#_Toc9940904)

[Inheritance 9](#_Toc9940905)

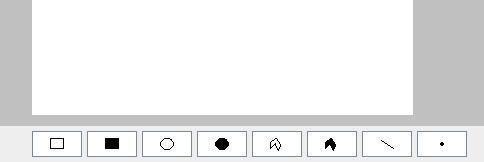
[Encapsulation 9](#_Toc9940906)

[Polymorphism 9](#_Toc9940907)

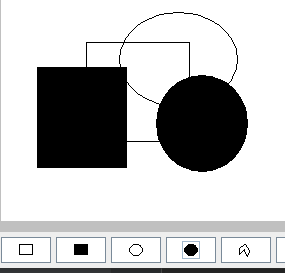
# Instructions

## Drawing

Select drawing commands from the bottom panel. Solid icons denote a filled shaped.



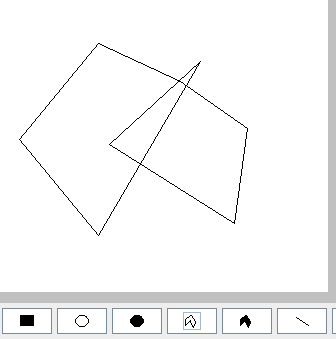
For rectangles, ellipses and lines **left click** to start drawing. **Drag** until the point you desire, then **Release** to commit.



For plots (single point) **left click** the location to immediately commit the instruction.



For polygonuse **left click** to start drawing the shape, add new points by **left clicking**. After at least one point is added **right click** to finish drawing.

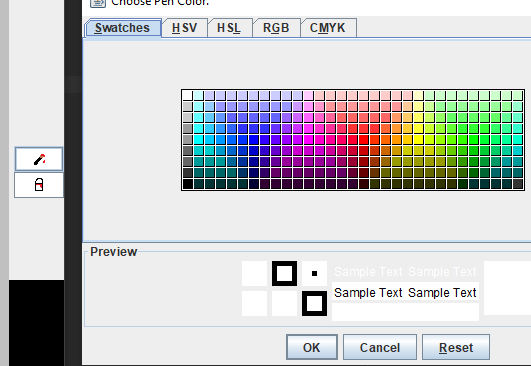


## Colours

To select both **pen** colour and the **fill** colour use the buttons on the right-hand panel. These will open a colour chooser.

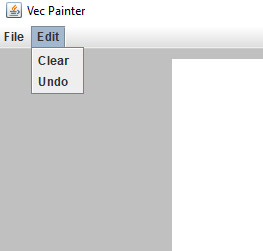
Use the paint brush icon to open a chooser for the **pen** color.

Use the paint bucket icon open a chooser for the **fill** color.

****

## Undo / Clear

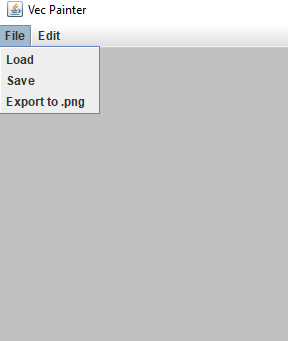
To undo the last command press **ctrl z** on the keyboard or navigate to the **edit** menu on the top panel.



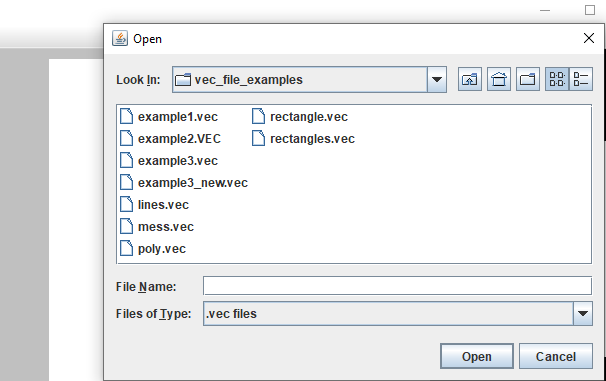
The menu also provides the option to clear the screen. This will remove everything from the canvas.

## I/O

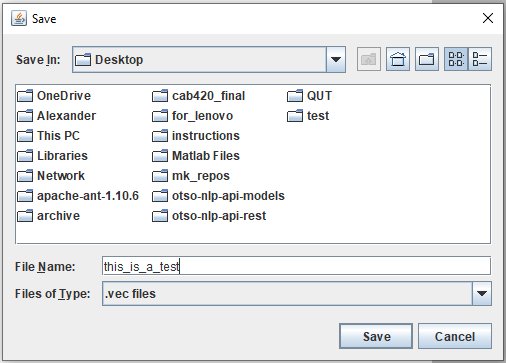
To **load, save** and **export** used the file menu.



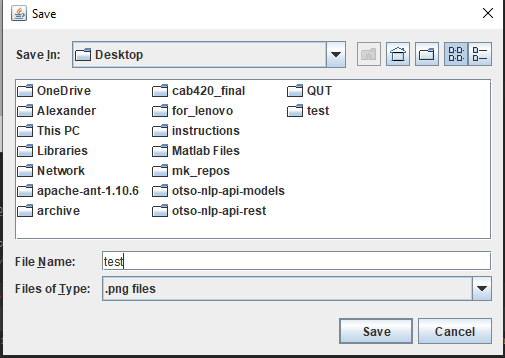
**Load** will open the file chooser letting you navigate your system’s file system while showing all files with the .vec (or .VEC) extension.



**Save** will open the file chooser letting you navigate to a directory to save the current state of the canvas in .vec (or .VEC) format. The file name can be entered with or without *.*vec (or .VEC) in the name (if without, it will be added automatically).



**Export to .png** will open the file chooser, letting you select the path and name of the file to export the current state of the canvas to **png**. Similarly to saving .vec the filename can be entered with or without png (if without, it will be added automatically).



# Architecture

The program is split into two main packages, **graphicsManage** and **gui. graphicsManage** is where the shape classes are implemented, as well as file parsing for reading and writing .vec files. **gui** is where the graphical user interface is implemented. The following section outlines the classes used in each package and their purpose.

## graphicsManage

### DrawableVector

An interface containing abstract methods all the custom shape/graphic classes will need to implement. Including the draw command, the get and set methods of colors (pen and fill), the get function for command type (VectorCommand), and the filled method which returns a boolean determining whether a shape is filled.

### FixedPointVector

This is an abstract class which extends drawable vector. It has been created for shapes which have the common property of having a fixed number of vertices. This is basically everything except the polygon. In this case it’s a shape which is defined by two x points and two y points.

### Ellipses

### Line

### Point

### Polygon

### Rectangle

### VecFileException

### VecFileManager

### VectorCommand

## gui

### GUI

### VecCanvas

### VecColorButtonPanel

### VecCommandButtonPanel

### VecPainterMenuBar

#### MenuListener

#### VecFileChooser

# Agile Methods

This program followed agile principles in its development. The following section outlines a couple of the principles that were followed during the development. As there were no specific client interactions, and the group only consisted of one person a full agile approach wasn’t exactly necessary or reasonable.

## Deliver Early and Deliver Often

The software was updated frequently with new features.

## Working Software

It was ensured that the software that was committed was always functional, as to ensure the product on the repository was always in a working state. In case something were to go wrong there would always be something to submit.

# OOP

This section outlines how the four principles of object-oriented programming were followed in the development of this program.

## Abstraction

## Inheritance

## Encapsulation

## Polymorphism