

Criterion B: Solution overview

Design Overview

Overall Structure

Figure #1: My client's initial visualization of the program layout, provided during first interview

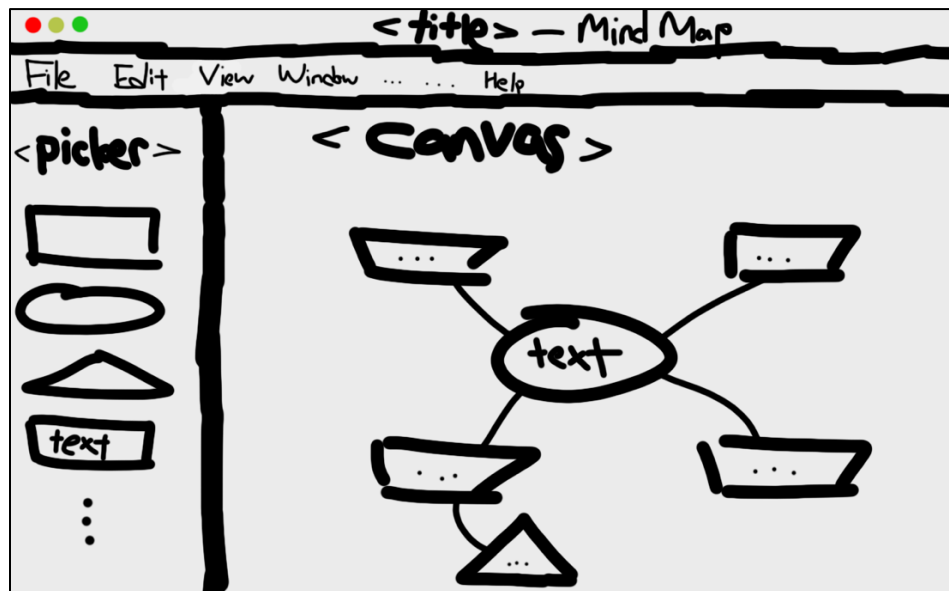
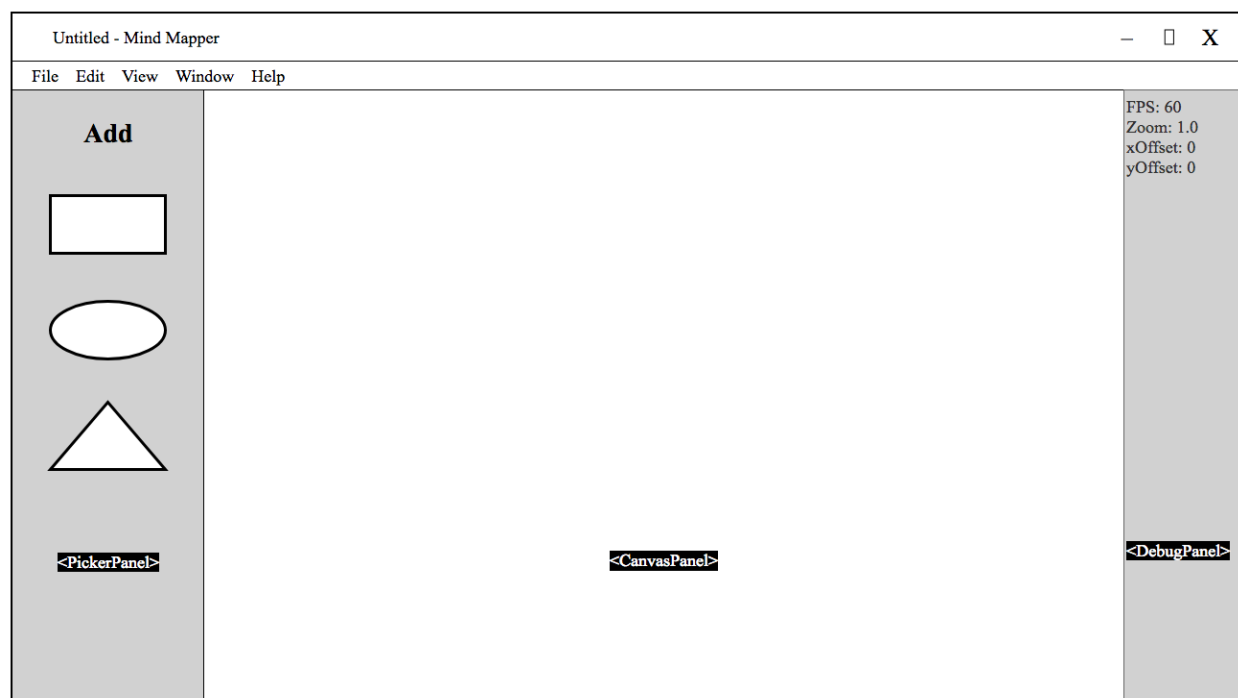


Figure #2: Formalized diagram of the program's layout



Internal Structures

Figure #3: Menus and submenus in the menu bar

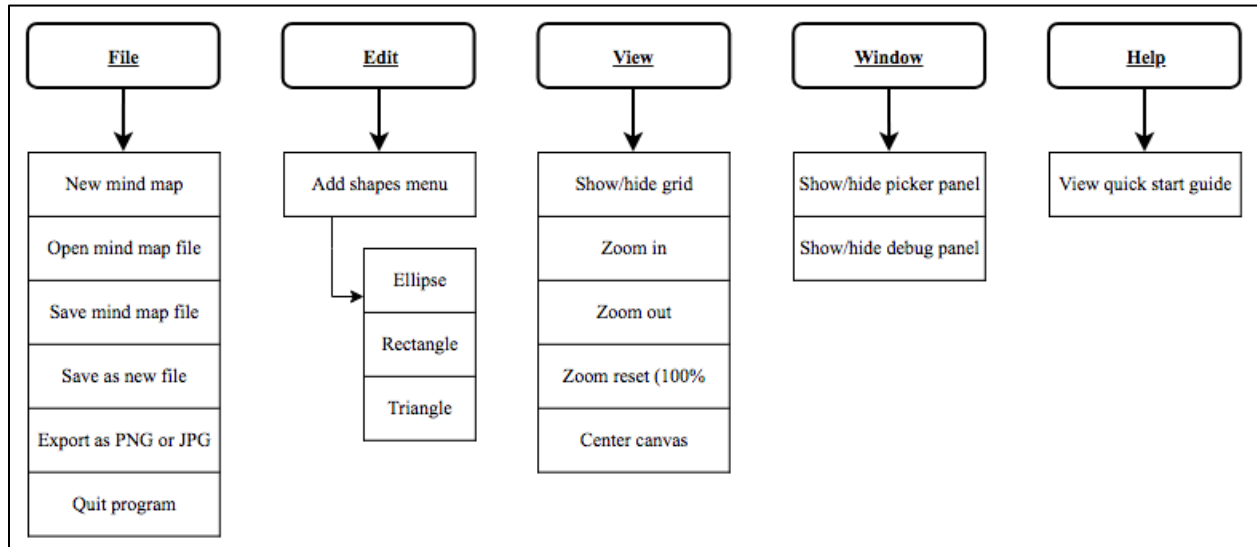


Figure #4: Submenus and actions for customization in the right-click context menu (dynamically changed depending on selected shape)

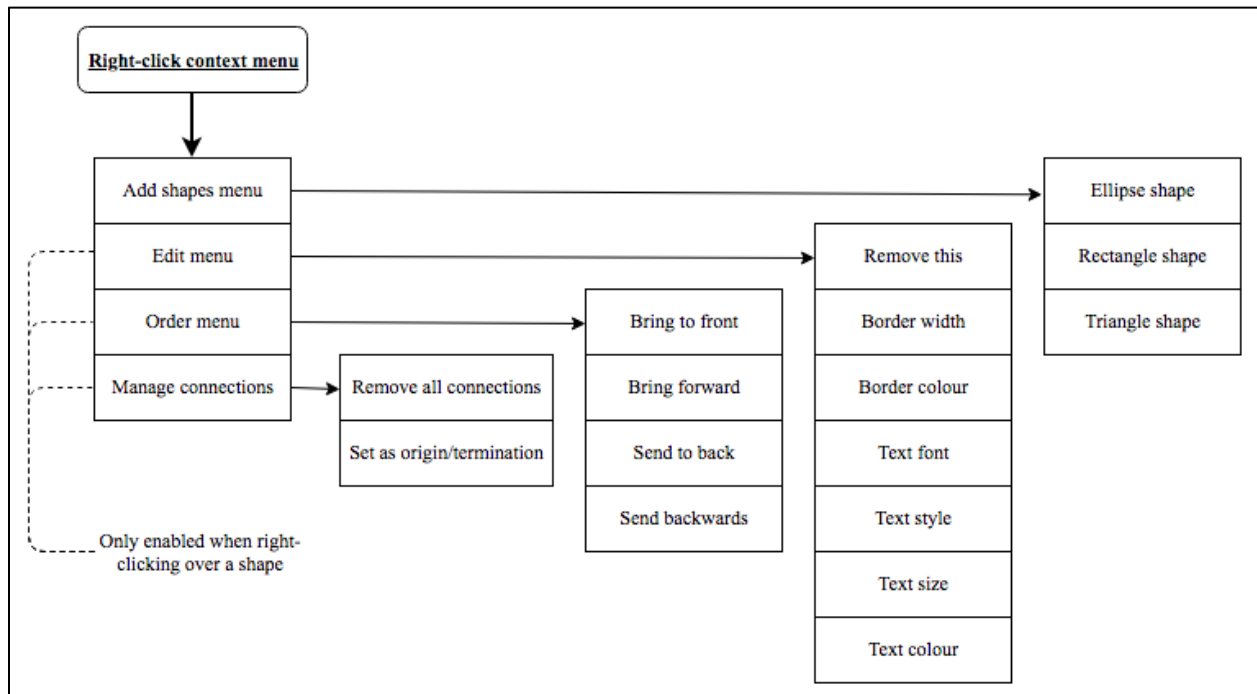


Figure #5: UML class diagram of entire program

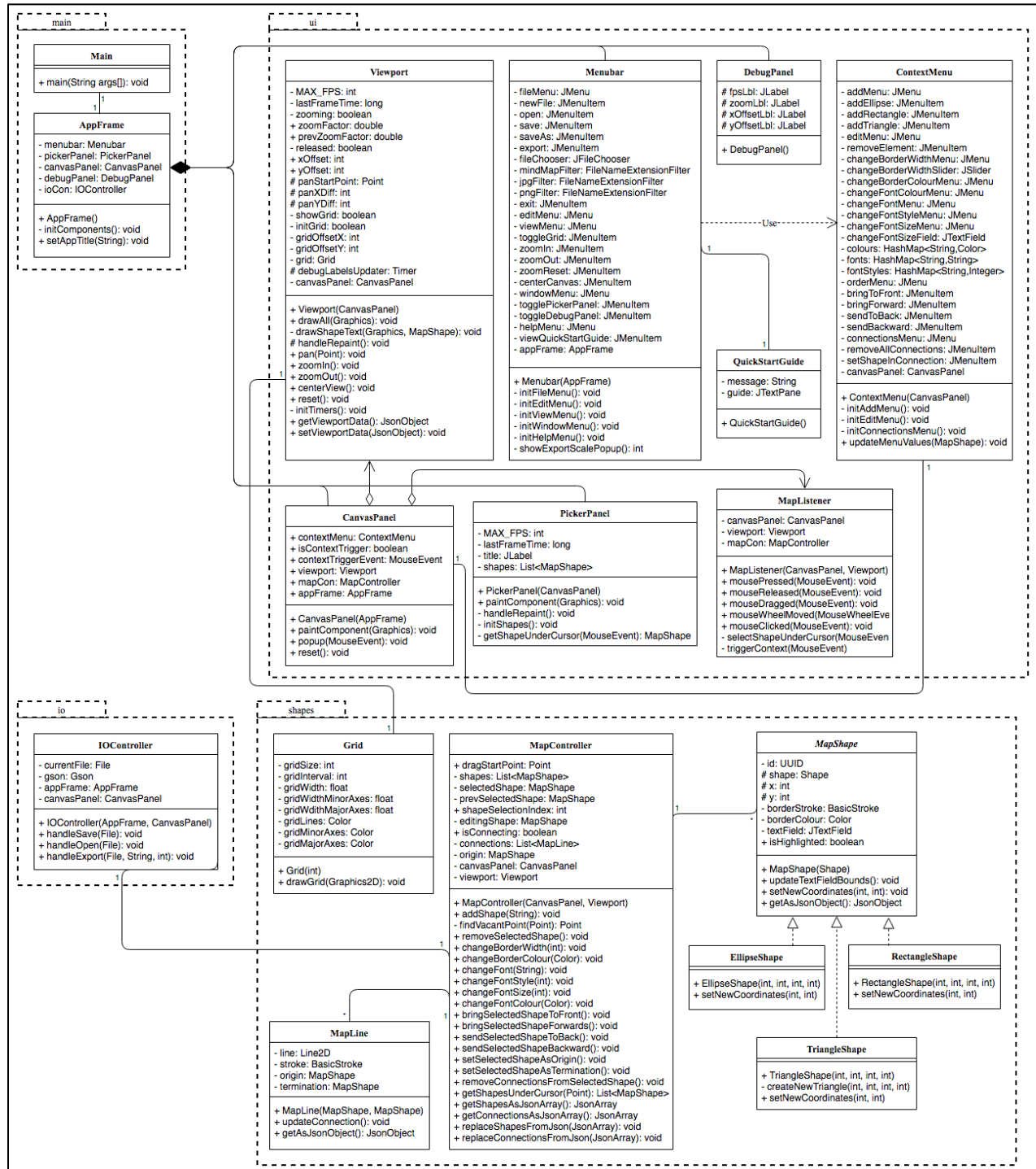


Table #1: Table of classes with descriptions of their purposes

Package	Class	Purpose
main	Main	Main entry point into program
	AppFrame	Constructs and manages the main frame
ui	Menubar	Manages the menu bar and handles actions, including the open, save and export dialogs
	CanvasPanel	Overarching panel for all mind map elements and listeners
	MouseListener	Handles clicking, dragging and scrolling in the CanvasPanel
	Viewport	Manages and handles drawing mind map elements
	ContextMenu	Manages and handles actions in the right-click menu
	PickerPanel	Allows the user to select a shape to add
	DebugPanel	Displays debug statistics if desired
	QuickStartGuide	Contains the text for the quick start guide, accessed via help
shapes	MapController	Manages and controls mind map elements
	MapShape	An abstract interface for all mind map shapes
	EllipseShape	An ellipse MapShape able to be drawn by the viewport
	RectangleShape	A rectangle MapShape able to be drawn by the viewport
	TriangleShape	A triangle MapShape able to be drawn by the viewport
	MapLine	Handles the connection between shapes
	Grid	Contains a grid able to be drawn by the viewport
io	IOController	Handles opening, saving and exporting of mind maps

Figure #6: Process of adding a shape to the canvas

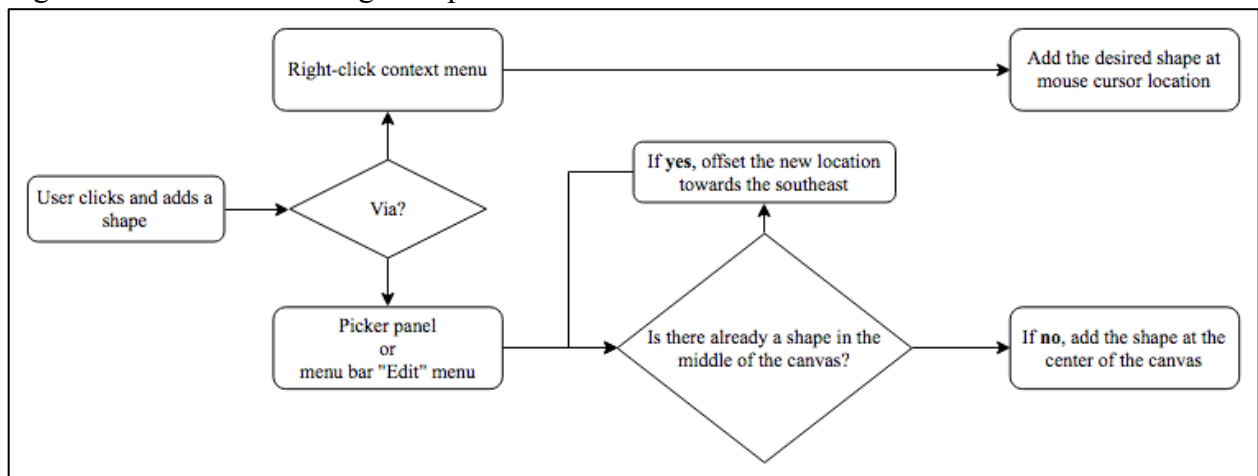
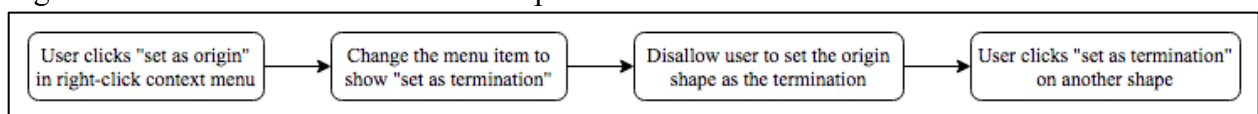


Figure #7: Process of connection two shapes with a line in the canvas



Test Plan

Feature to be tested	Test method	Example of test
Adding shapes to the canvas	<p>Add a shape to the canvas via the menu bar “Edit” menu or by double-clicking it in the picker panel. The shape should appear in the center of the canvas. Repeat immediately and the second new shape should not overlap but appear southeastward</p> <p>Add a shape via the right-click context menu anywhere in the canvas. The shape should appear centered where the cursor was when the right-click context menu was triggered</p>	<p>> User clicks on “Ellipse shape” in the menu bar “Edit” and “Add” menu</p> <p>> An ellipse appears in the center of the canvas</p> <p>> User double-clicks on the ellipse shape in the picker panel</p> <p>> Another ellipse appears southeastward of the first shape in the canvas</p> <p>> User right-clicks in the northwest region of the canvas and clicks on “Rectangle shape” in the “Add” menu</p> <p>> A rectangle appears centered where the cursor was when the right-click context menu was triggered</p>
Selecting shapes	<p>When shapes are clicked, their border should change from their original colour to cyan. After clicking on empty canvas space or on another shape, the previously selected shape’s border should restore its original colour. Multiple selections are not supported and do not need to be tested.</p>	<p>> User adds a shape</p> <p>> User left-clicks on the shape</p> <p>> The shape’s border should have changed from black to cyan</p> <p>> User clicks on empty canvas space or on another shape</p> <p>> The shape’s border should become black again</p>
Mind map elements should be able to be dragged and dropped to be placed wherever desired in the canvas	<p>Drag a pre-added shape around in the canvas. Its position should update smoothly without panning the canvas</p>	<p>> User adds a shape</p> <p>> User left-clicks and holds over the shape and moves it eastward</p> <p>> The shape moves eastward</p>
Connecting shapes with lines	<p>Add two shapes to the canvas. Right-click one shape and select “Set as origin”. Right-click the other shape and check to see the menu option has changed to “Set as termination” and click it. A line should appear between them and its location should update when the shapes are dragged around</p>	<p>> User adds two shapes</p> <p>> User right-clicks on one shape and selects “Set as origin” under the “Connections” menu</p> <p>> User right-clicks the other shape and selects “Set as termination” under the same menu</p> <p>> A line should instantly appear connecting the two shapes</p>

Customization of the mind map design, such as options for shape, font, size, colour	Change the various attributes of a pre-added shape. The appearance of the shape and text should change accordingly	<ul style="list-style-type: none"> > User adds a shape > User right-clicks the shape and changes the border colour to red > The shape's border colour becomes red
Panning and zooming in/out in the canvas	Drag on empty space in the canvas and the canvas and all mind map elements should move in the same direction. Scroll on empty space in the canvas and the canvas should zoom in the same direction.	<ul style="list-style-type: none"> > User left-clicks and holds on empty space in the canvas and moves eastward > The canvas and all mind map elements move eastward together
Centering the canvas	When clicked, the menu item "Center Canvas" under the "View" menu should pan the canvas towards the relative center of the utilized canvas space	<ul style="list-style-type: none"> > User adds three shapes in an equilateral triangle formation > User pans the canvas in a direction until the three shapes are not visible in the viewport > User clicks on "Center Canvas" in the "View" menu > The relative center of the viewport should align with the center of the equilateral triangle formation of shapes
Importing a mind map file to continue editing	Add some mind map elements to the canvas. Save the mind map to a .JSON file. Close the program or click on "New" in the "File" menu. Open the same file. The original canvas pan and zoom, position of shapes and lines, and customization of shapes should all be restored	<ul style="list-style-type: none"> > User adds some shapes and connections > User clicks "Save" in the "File" menu > User chooses a directory and filename from the popup and saves > User closes the program or clicks "New" in the "File" menu > User clicks "Open" in the "File" menu > User traverses to the same directory and opens the saved file > The exact characteristics of everything in the canvas should all be restored
Exporting to an image format	Add some mind map elements to the canvas. Export to a .PNG and .JPG image for all of the five qualities available. The exported	<ul style="list-style-type: none"> > User adds some shapes and connections > User clicks "Export" in the "File" menu

	images should all depict the working mind map, with transparency if .PNG was selected	<ul style="list-style-type: none"> > User chooses a directory and filename from the popup and clicks “Export” > User selects a desired quality from the popup > User views the exported image
Respond to user input without unintended or unresponsive behavior, and not crash	<p>Multiple ways to test:</p> <ul style="list-style-type: none"> - Press many buttons at once in an effort to cause unintended behavior, but should not happen - Pan and zoom at the same time in the canvas, which will pan and zoom smoothly and accordingly, not wildly 	<ul style="list-style-type: none"> > User adds some shapes and connections > User drags on empty space in the canvas and scrolls inward at the same time > The canvas should pan in the same direction and zoom in
Efficient and responsive to operate. Must not feel sluggish with large data sets	Stress-test the program by adding many shapes and connections to the canvas. Dragging, panning, zooming, and all other aspects of program operation should be smooth and lag-free	<ul style="list-style-type: none"> > User adds 50 shapes of varying type to the canvas at random locations > User creates many connections between the added shapes > User drags some shapes and pans and zooms the viewport > User customizes the attributes of the shapes > The perceived speed of operation and responsiveness of the program should be identical or very close to a new canvas

Word count: N/A (0)