

User Interface (View)	
Controllers	
Use Cases	
Entities	

NAME		TYPE	Parent:
			Child:
RESPONSIBILITIES			COLLABORATORS
<ul style="list-style-type: none">● Do this and that<ul style="list-style-type: none">○ In this way○ And in that way			<ul style="list-style-type: none">★ Class<ul style="list-style-type: none">○ Used method(s)



OnionShop

TYPE

Parent: [Application](#)

Child:

RESPONSIBILITIES

- Start up the initial scene (project explorer fxml)

COLLABORATORS

ProjectExplorerController

Controller

Parent:

Child:

RESPONSIBILITIES

- Outputs directories of most recently edited .onion projects, and allows users quick access to open them
- Open .onion project files from file explorer
- Sends the file to the ProjectManager, which will handle loading that file as the currentProject
- Creates new project -> open new project scene

COLLABORATORS

- ★ SceneSwitcher
- ★ ProjectManager

NewProjectController

Controller

Parent:

Child:

RESPONSIBILITIES

- Handles Textbox inputs for project name, and canvas width and height
- Allows users to choose directory they want to save the project to
- Creates a NewProjectEvent which is used by ProjectManager to create a new project
- After a new project is created, uses SceneSwitcher to switch to the Main Canvas view

COLLABORATORS

- ★ ProjectManager
- ★ NewProjectEvent
- ★ SceneSwitcher

ProjectStateController

Controller

Parent:

Child:

RESPONSIBILITIES

- Takes state from ToolStateManager
- Keeps track of tool, color, and canvas

COLLABORATORS

- ★ ToolStateManager
- ★ CanvasEvent
- ★ ToolBarEvent
- ★ ColourSelectionEvent

KeyboardEventController

Controller

Parent:

Child:

RESPONSIBILITIES

- Handles keyboard inputs
- Ctrl Z - undo
- Ctrl Y - redo
- Ctrl S - save

COLLABORATORS

- ★ ProjectManager
- ★ UndoRedoManager

<div>SceneSwitcher</div>		Controller	Parent:
			Child:
<div>RESPONSIBILITIES</div> <ul style="list-style-type: none">Switches the scenes from the current event to the given fxml file			<div>COLLABORATORS</div> <div>★</div>

<div><div>CanvasEvent</div><div>Class</div></div>		Parent:
		Child:
<div><div>RESPONSIBILITIES</div><div><ul style="list-style-type: none">• Takes coordinates from Java fx canvas in a format that is used by the DrawingManager</div></div>		<div><div>COLLABORATORS</div><div><div>★</div><div>Tool</div></div></div>

<div><div>ToolbarEvent</div><div>Class</div></div>		Parent:
		Child:
<div><div>RESPONSIBILITIES</div><div><ul style="list-style-type: none">Contains details of which tool was selected</div></div>		<div><div>COLLABORATORS</div><div><div>★</div><div>Tool</div></div></div>

<div><div>ColorSelectionEvents</div><div>Class</div></div>		Parent:
		Child:
<div>RESPONSIBILITIES</div> <ul style="list-style-type: none">Contains details of which colour was selected		<div>COLLABORATORS</div> <div>★ Color</div>

NewProjectEvent

Class

Parent:

Child:

RESPONSIBILITIES

- Holds the following new project details:
 - Filename
 - Directory
 - Width
 - height

COLLABORATORS

<div>ProjectManager</div>	Class	Parent:
		Child:
<div>RESPONSIBILITIES</div> <ul style="list-style-type: none">• Creates new project (blank)• Stores current project• Open .onion file<ul style="list-style-type: none">○ Loads Canvas instance○ Loads Project instance and passes the Canvas instance○ Buncha text that has rgb, color palette, brush state, project entity saved in the onion file text• Save function that stores project file (and keeps track of time) -> .onion• Contains instance of UndoRedoManager	<div>COLLABORATORS</div> <ul style="list-style-type: none">★ Project★ NewProjectEvent★ OnionFileLoader★ UndoRedoManager	

UndoRedoManager

Class

Parent:

Child:

RESPONSIBILITIES

- Stores 5 previous states of Drawing instance
- Stores 1 redo state until changes are made to a previous state
- Keeps track of which state canvas is in currently
- Allows the user to switch between saved states
- Get (Likely from the CanvasController) and update the 5 states as the user makes changes to the canvas

COLLABORATORS

- ★ CanvasController
- ★ ToolStateManager(?)

<div>ToolStateManager</div>		Class	Parent:
			Child:
<div><div>RESPONSIBILITIES</div><div><ul style="list-style-type: none">Manages the state of the tool that is being used on the canvasStores current tool</div><div>Definition of Tools:</div><div><ul style="list-style-type: none">Anything that modifies the canvas pixels directly.Colors modify tools directly.</div></div>			<div><div>COLLABORATORS</div><div><ul style="list-style-type: none">★ Tool★ Colour</div></div>

DrawingManager

Class

Parent:

Child:

RESPONSIBILITIES

- Looks at the tool state (current tool, size), current colour,
- Calculates the specifics using those data
 - Affected pixels

COLLABORATORS

- ★ Tool
- ★ Color

<div>OnionFileLoader</div>		Class	Parent:
			Child:
<div>RESPONSIBILITIES</div> <ul style="list-style-type: none">• Manages Save & Load• Load<ul style="list-style-type: none">◦ Returns an instance of Project based on the .onion file• Save<ul style="list-style-type: none">◦ Takes instances of project & path to where you're saving, store the project		<div>COLLABORATORS</div> <div>★ Tool</div>	



Project

Class

Parent:

Child:

RESPONSIBILITIES

- Stores current project save address
 - E.g. users/finn/art/canvas1.onion
- Stores drawing
- Stores dimensions of drawing
- Stores current project colour palette
- Can serialize itself for saving

COLLABORATORS

★ Drawing

<div>Pixel</div>		Class	Parent:
			Child:
<div><div>RESPONSIBILITIES</div><div><ul style="list-style-type: none">• Stores RGB value• Get method<ul style="list-style-type: none">○ Returns RGB value• Set method<ul style="list-style-type: none">○ Changes current colour of pixel</div></div>			<div><div>COLLABORATORS</div><div><div>★ Class</div><ul style="list-style-type: none">○ Used method(s)</div></div>



Colour

Class

Parent: Pixel

Child:

RESPONSIBILITIES

- Stores a colour (RGB)
- These can be added or removed from the palette
- Stores a name to allow a user to find specific colours again
- Modify method
 - Allows adding or subtracting from the rgb values (e.g. `this.RGB[0] += 10`)

COLLABORATORS

- ★ Class
 - Used method(s)

Tool	Interface	Parent:
		Children: ShapeTool, Brush, (insert other tools here)
<div>RESPONSIBILITIES</div> <ul style="list-style-type: none">● Includes a function that makes the tool selectable (ie loads up the tool's option panel which may include size, shape type, etc)● Draw method, on how the tool is supposed to be outputted onto the canvas● Calculate pixels the brush or tool will update based on it's current size and shape	<div>COLLABORATORS</div> <div>★ toolStateManager</div>	

<div><div>Brush</div><div>Abstract Class</div></div>		<div>Parent (interface): Tool</div> <div>Children: Pen, Shape</div>
<div>RESPONSIBILITIES</div> <ul style="list-style-type: none">• Constructs a Brush that has the properties brush shape and brush size• Getter methods:<ul style="list-style-type: none">○ Return brush shape○ Return brush size• Setter methods:<ul style="list-style-type: none">○ Set brush shape○ Set brush size		<div>COLLABORATORS</div> <div>★ Tool</div>

<div>Shape</div>		Class	<div>Parent (interface): Tool</div> <div>Children: Rectangle, Circle</div>
<div>RESPONSIBILITIES</div> <ul style="list-style-type: none">• Store shape types• Fill color• Had draw method, when mouse pressed and dragged inwards and outwards, the shape will grow and shrink, until the mouse is released, then shape is set		<div>COLLABORATORS</div>	

<div>Pen</div>		Class	Parent (interface): Brush
			Children:
<div>RESPONSIBILITIES</div> <ul style="list-style-type: none">• Constructs a Pen object, child of Brush, that has properties brush shape (round by default) and a brush size (1 by default)• Calculates pixels the Pen object will affect• Updates pixels that are affected by the Pen<ul style="list-style-type: none">○ Fill color		<div>COLLABORATORS</div> <ul style="list-style-type: none">★ Tool★ Colour	

CanvasPropertiesState	Class	Parent:
		Child:
<div>RESPONSIBILITIES</div> <div><div>○</div>currentColour</div> <div><div>○</div>currentTool</div> <div><div>○</div>currentToolSize</div>		<div>COLLABORATORS</div> <div><div>★</div>Color</div> <div><div>★</div>Tool</div>



Interface

Parent: Object

Child: ColourSlider, ColourWheel

RESPONSIBILITIES

- Interface for modifying and selecting colours
- Modify method (for existing colours)
- Select method (for new colour)

COLLABORATORS

- ★ Class
 - Used method(s)



Class

Parent: ColourMod

Child:

RESPONSIBILITIES

- Allows the modification of one aspect of {R, G, B}
 - Linear scale where min=0, max=255
- Set method
- Mod method
 - Starts slider at a given Colour's relevant RGB position

COLLABORATORS

★ Colour



Class

Parent: ColourMod

Child:

RESPONSIBILITIES

- Used to create a Color
 - This is done using some trig
- modify method
 - Starts colourwheel at the given colours RGB values and allows you to modify from there
- newColour method
 - Creates and returns a new Colour

COLLABORATORS

★ Colour

Onion file format

[dimensions]

width:<width>

height:<height>

[saved colours]

<name1>: R, G, B

<name2>: R, G, B

...

<nameN>: R, G, B

[pixels]

Pixel at 0, 0

Pixel at 0, 1

Pixel at 0, 2

...

Pixel at 1, 0

Pixel at 1, 1

...

Pixel at Width, Height

[end]