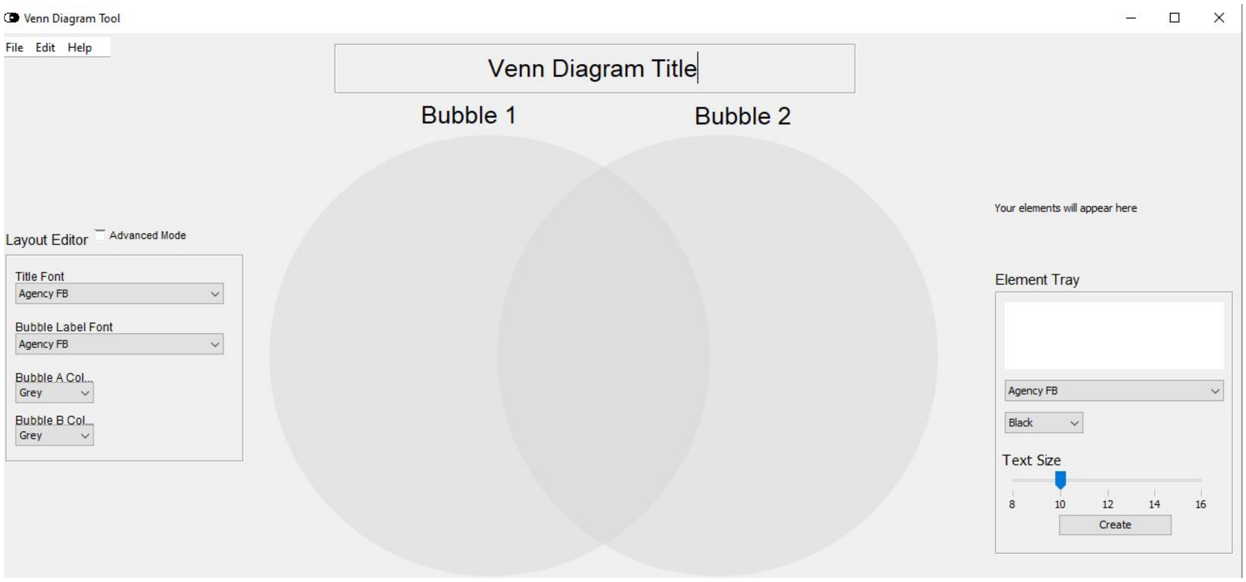
Final Submission

Venn Diagram Tool



Title Font: Agency FB

Bubble Label Font: Agency FB

Bubble A Col...: Grey

Bubble B Col...: Grey

Agency FB

Black

Group 1:

Hassan Mukhtar
Ahmd Faruqi

Professor:

Dr Vassilios Tzerpos

Course:

EECS 2311

TABLE OF CONTENTS

Requirements

What the System does for the Clients

To satisfy a customer's requirements best, there must be options for complete customizability, anything that is on the screen should be able to be changed. The base application consists of a simple non-customizable Venn diagram that drags and drops pre-loaded words with no customizability features. This base Venn diagram is a terrible design because it is limited by its size and customizability. The main customizable feature required to have a usable Venn diagram would be the resizability of elements. With this feature added the customer can add as many words into the diagram, and the Venn diagram would accomplish the base job. However, this is just the bare minimum as our application will provide everything that could possibly be needed in terms of customizability. Our application has various customizable features, such as; resizability and colour of the Venn diagram circles, optional font styles, colours, and sizes. And to neatly wrap up these features and keep them from clustering the screen, a pop-out toolbar was added to encapsulate all tools used to adjust the Venn diagram. Additionally, the application also has features that allow a file to be read and sorted into a Venn diagram, while also having the ability to manually input words for the Venn diagram. Furthermore, the application has a save and load state which allows the client to save and load their previous work to progress on at a later time. Lastly, once the Venn diagram is completed by the client, they will have an option to export the final work to PDF for viewing.

Use Cases

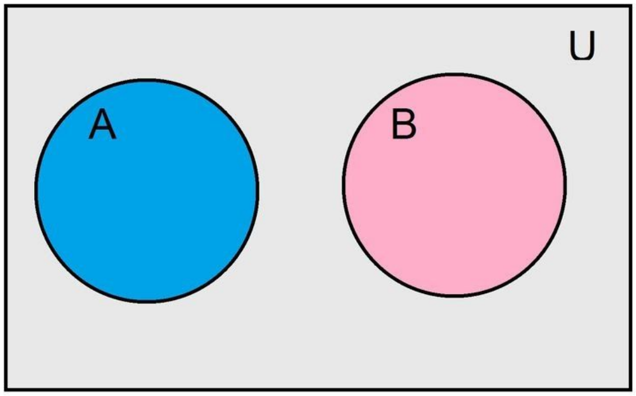
1. Given a word file containing certain numbers of values that each belong to 3 different categories: A, A U B, B.
    a. A and B are mutually exclusive, meaning A U B is empty
    b. A and B contain the same value, but they both have unique value, meaning A U B is not empty
    c. A and B are the same set, meaning A U B is A or B
    d. A includes B, meaning A U B is B (B includes A, meaning A U B is A)
2. Inputting values in the text field and using the drag-and-drop function
    a. A and B are mutually exclusive, meaning A U B is empty
    b. A and B contain the same value, but they both have unique value, meaning A U B is not empty
    c. A and B are the same set, meaning A U B is A or B
    d. A includes B, meaning A U B is B (B includes A, meaning A U B is A)

Acceptance Test Cases

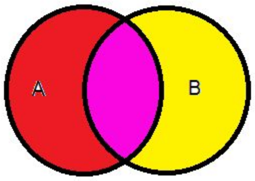
The following figures show how the Venn Diagram should appear position-wise. They do not reflect the colour, size or label. Those factors will be customizable.

1. User Case 1

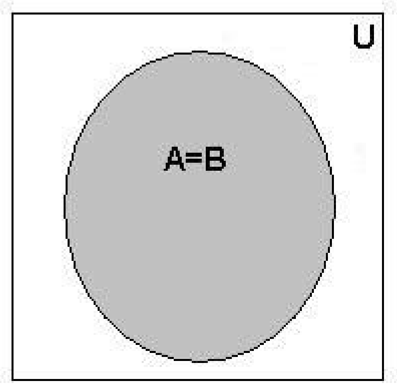
a.

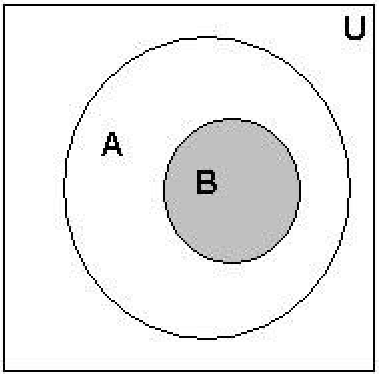


b.



c.



d. 

2. User Case 2 should have the same result position wise.

   Venn diagrams for all 8 cases should all have customizable labels, sizes, and labels. These options need to be available to users in the most convenient way.

Design Document

There are four test cases that we have implemented. Test case *a* is the situation in which neither of the Venn diagrams share any similarities and are their own two entities in the universe (figure 1a). Test case *b* is the scenario in which there are two venn diagrams which share features as well as having their own features (figure 1b). Test case *c* is the scenario where the two Venn diagram circles are completely identical (figure 1c). Lastly, test case *d* is the scenario where one Venn diagram circle is encapsulated within another Venn diagram circle (figure 1d).

These test cases were derived through the reality of outcomes. Assuming that we have two Venn diagram circles, the only possible arrangements are the test cases explained above. Even if additional Venn diagram circles were added, the number of base scenarios would remain the same, however, there could be a conjunction of test cases. For instance, if there were three Venn diagram circles there could be a layout combining figures *a* and *b,* where two circles would share similarities and the third would be unique to both.

The test cases listed above are sufficient because they are the only possible outcomes of the base case of having two Venn diagram circles. Any additional circles being added would result in a conjunction of the test cases.

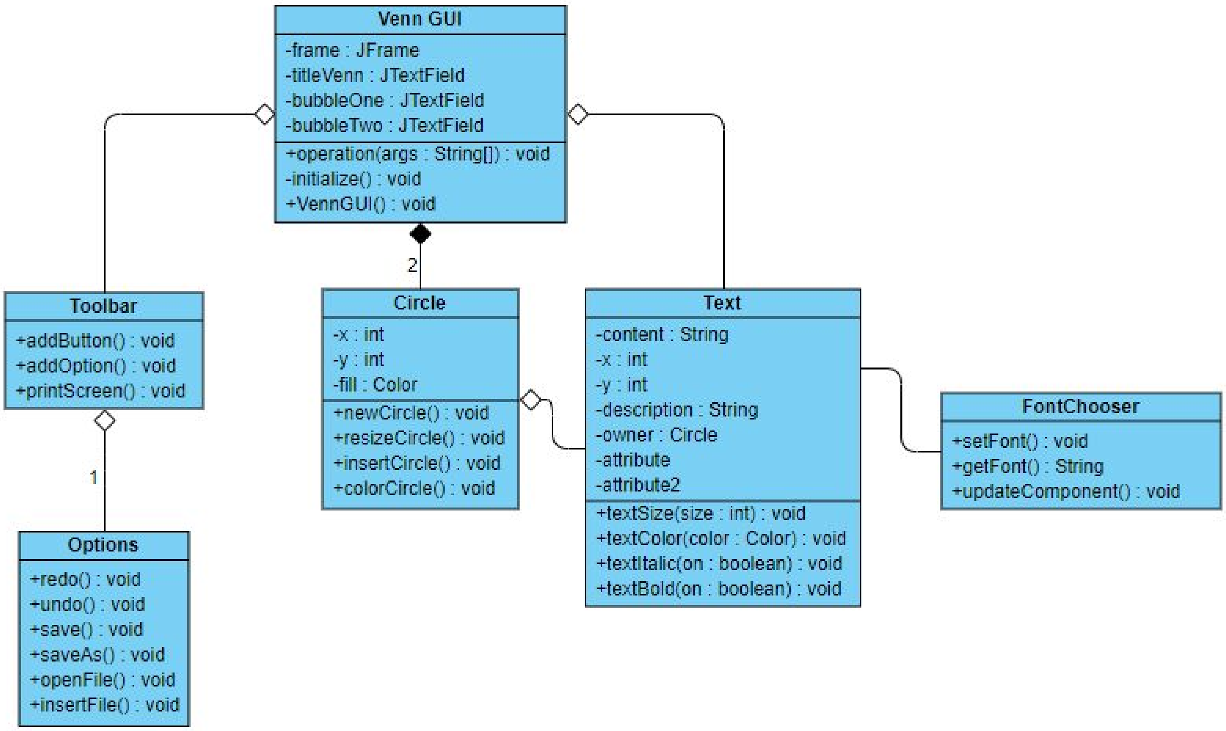
Class Diagrams

VennGUI Interactions

The application centres around interfacing with the VennGUI class. The order of operations are independent of one another, however a typical scenario will be depicted.

It is assumed that the user is interested in a Venn Diagram with 2 circles. Furthermore, the circles have fixed sizes and locations. Lastly, it is assumed the user will not generate text endlesley and cause various memory allocation errors.

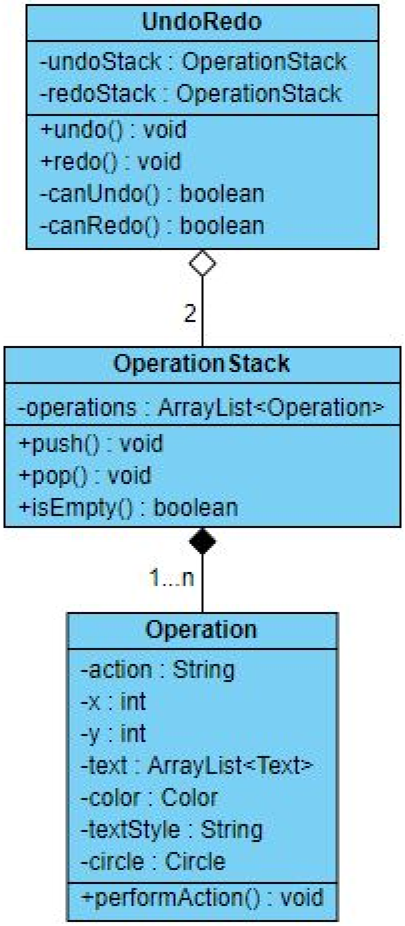
The software's main purpose is to provide drag and drop functionality for text into circles. From there, users are able to customize the visual style of the text and circles at any time. Finally, the user can alter their work so far with the toolbar by undoing unwanted changes and saving the project for future use.



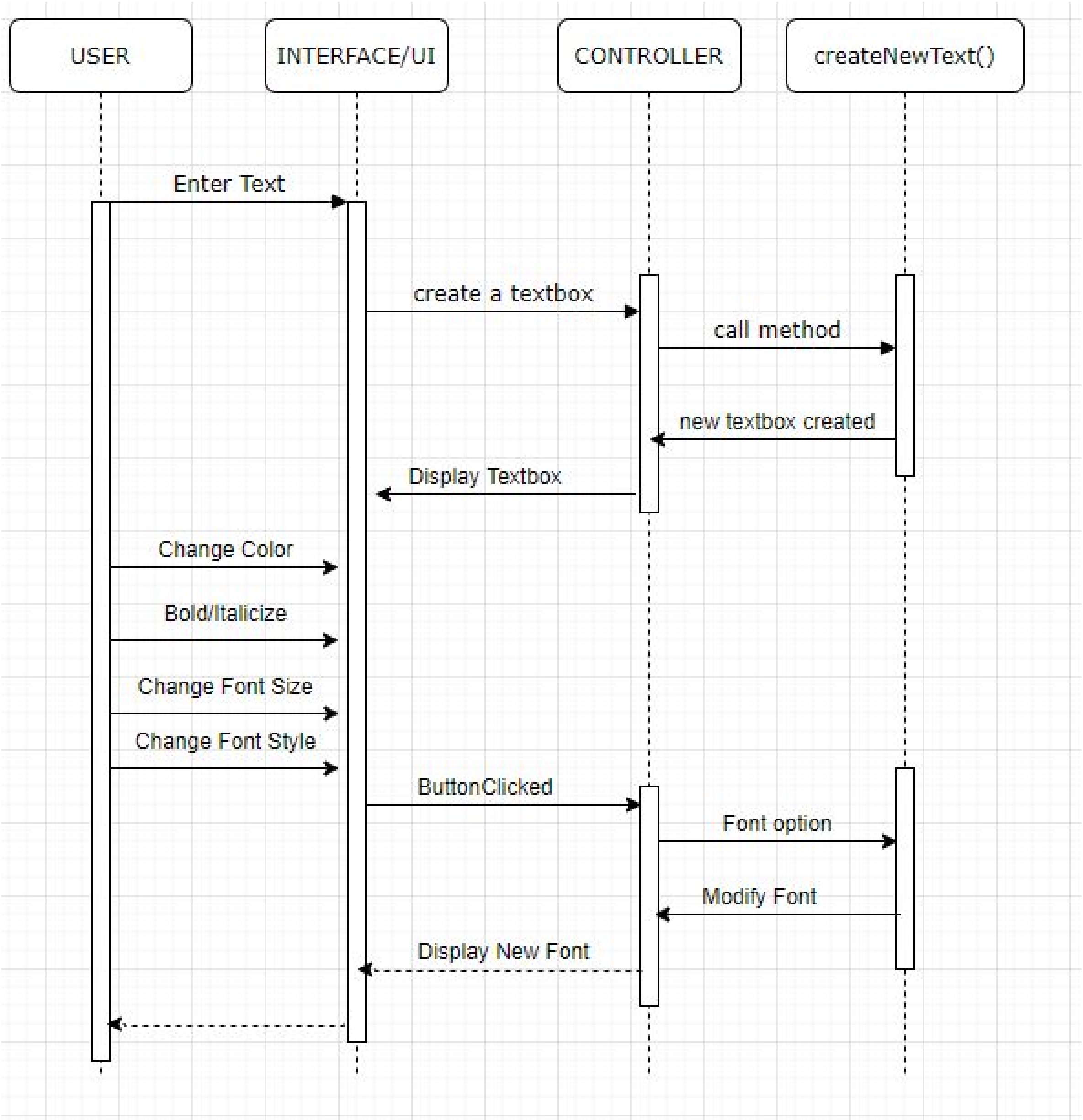
Undo and Redo

The implementation of undo/redo functionality was implemented via 2 stacks and instances of operations. In addition, the custom class OperationStack was implemented in place of the importable Stack class provided in Java.

As the user creates text, moves text, and changes visual styles, each action is pushed to a stack for undoing. Upon undoing, the popped operations from the undo stack are performed and pushed into the redo stack. From there a redo can be performed or the user can continue making changes. After a project is closed, both stacks are cleared.



Sequence Diagrams



Maintenance Scenarios

Overview

The following scenarios account for planned improvements to the software. Although not every improvement is noted, the most urgent and noticeable changes are highlighted. Each point will consist of a flaw, the improvement, and plausible code implementation.

Focus

The user is capable of performing actions on text and titles but it is unclear whether or not a specific object is currently being acted upon.

A simple and effective solution would be to highlight the selected text by giving it a visible shadow. This would be quick to implement and solves the problem of highlighting by a specific colour.

Adding a mouseClickedListener to all text objects would allow for selection and deselection. The display method of all text objects can be edited to redraw the text with black transparency to the lower right.

Multiple Selection

The user is capable of generating their desired Venn Diagram through various insertions, colour changes, and possible deletions midway through the process. However, tedious operations like dragging multiple text objects must be done one text at a time.

The user should have the option to apply the same operation to multiple text objects at once. This should be achieved by holding the ctrl key and performing the operation as usual. Because of the nature of multiple selection, the only operations that can be applied are: moving, colour changing, and deletion.

An ArrayList<Text> would need to be added to the VennGUI to keep track of the selected text objects. Furthermore, the operation class would need to be edited to account for an array of actions. This will likely result in a new operation for multiMove, multiDelete, multiInsert, and multiColourChange would be needed.

Passive Interaction

Although the user can perform actions on text, it is unclear until the user clicks on the text for the first time. Essentially, it is unclear whether a user can perform certain actions on text objects without prior testing.

The user can be encouraged to interact by having text objects in range hover. This will serve as a visual cue that the text is interactable. Only text objects directly under the cursor should hover and the hover should be slight but smooth.

This can be achieved by editing the display method of all text objects and the already implemented mousePosition() checker. If the coordinates of the cursor overlap the text's location, the position of the text should gradually increase to a set maximum "height". Once the cursor is no longer above the text, the text gradually returns to its original position.

Three Circles

The current Venn Diagram is the typical 2-circle model but other models exist. For example having 3 circles is less likely but still a significant user case.

At the moment, complete customization of the Venn Diagram's circles is unnecessary. So, a hard-coded fix would be the most pragmatic solution. An additional circle can be added by toggling a switch in the toolbar. From there, the Venn Diagram will instantly change to a 3-circle model with 3 different colours.

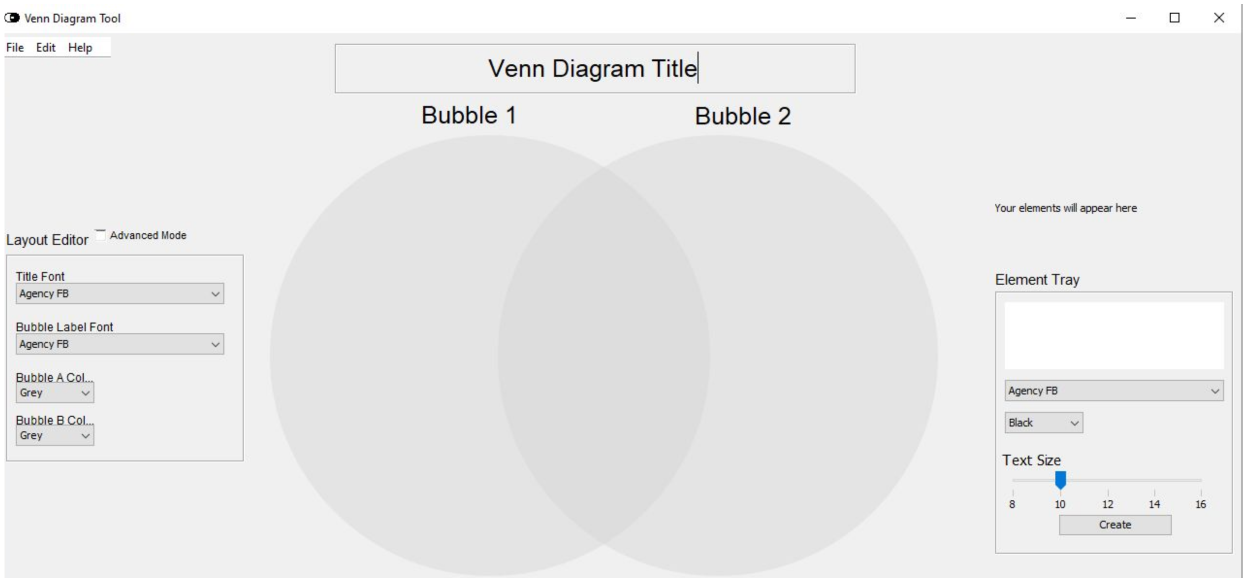
The toolbar will need one additional option to toggle the visibility of the two Venn Diagram options. Regarding the 3-circle model, a third JPanel representing the extra circle can be added to the existing model. This will allow for the reimplementation of all existing operations in the initial 2-circle model.

CHUTNEYSLURPEE

Installation and User Instructions

Venn Diagram Tool

The Complete Version



Title Font: Agency FB

Bubble Label Font: Agency FB

Bubble A Col...: Grey

Bubble B Col...: Grey

Agency FB

Black

April 2020
Version 4.0

TABLE OF CONTENTS

2. Interface Instructions

3. Video Demonstrations

GETTING STARTED

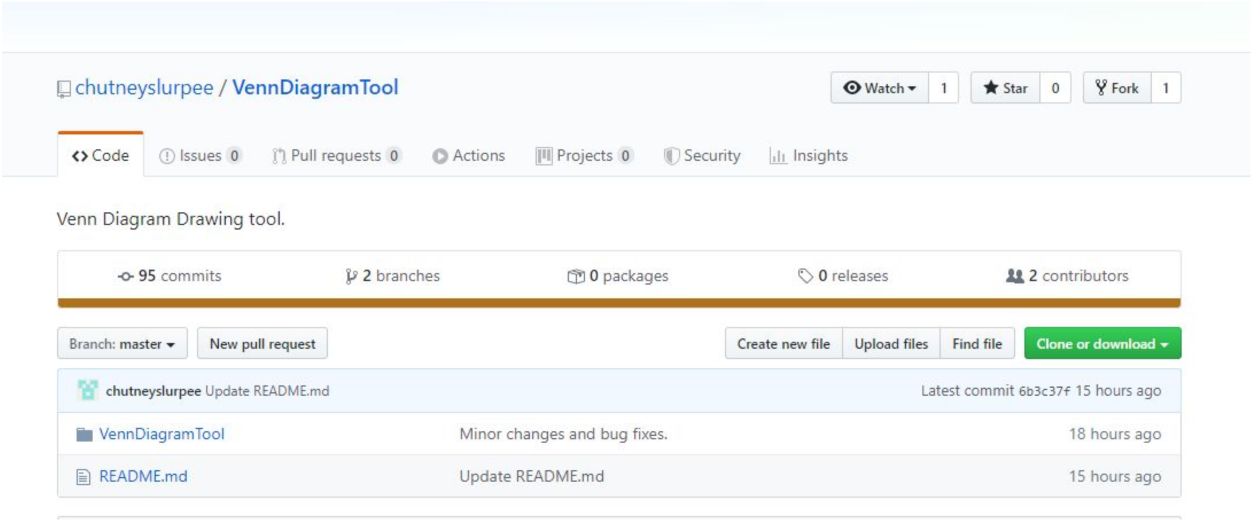
About: Powerful Venn Diagram Rendering tool which allows the user to create text based labels that can then be intuitively dragged and dropped.

Software Requirements:

- Windows XP or newer.
- MAC OS Mountain Lion or newer.

Installation Instructions:

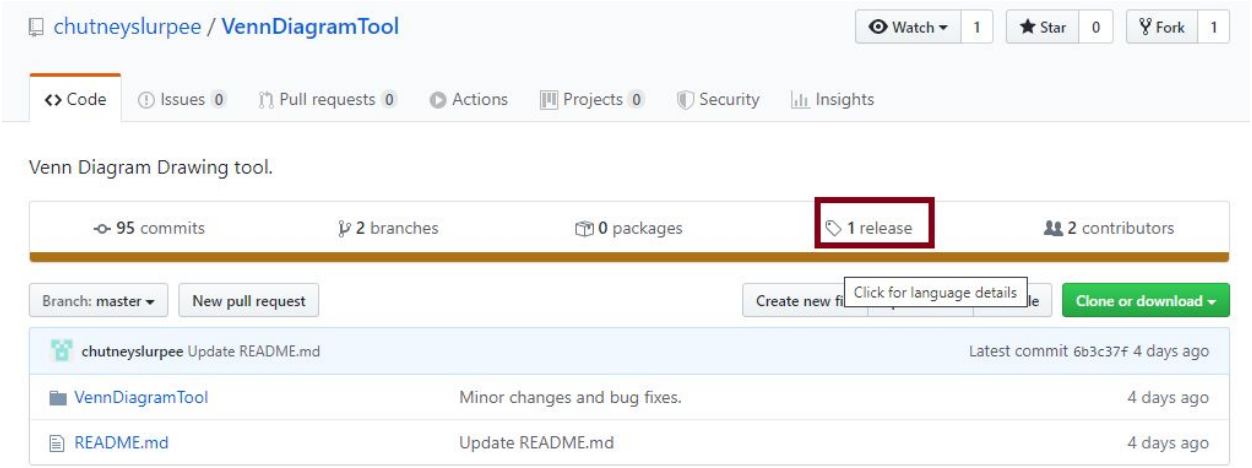
1) Open the following Github Link: https://github.com/chutneyslurpee/VennDiagramTool
   You should now see the following in your browser.



Venn Diagram Drawing tool.

chutneyslurpee Update README.md		Latest commit 6b3c37f 15 hours ago
VennDiagramTool	Minor changes and bug fixes.	18 hours ago
README.md	Update README.md	15 hours ago

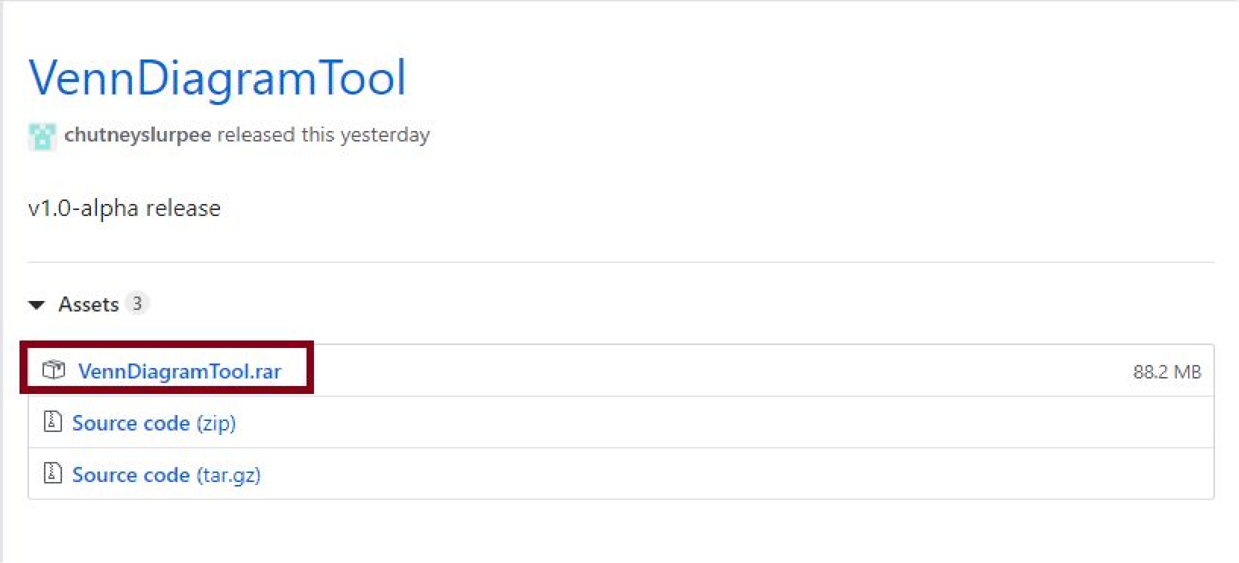
2) Next click on the “releases”.



Venn Diagram Drawing tool.

chutneyslurpee Update README.md		Latest commit 6b3c37f 4 days ago
VennDiagramTool	Minor changes and bug fixes.	4 days ago
README.md	Update README.md	4 days ago

You will then get a screen as:



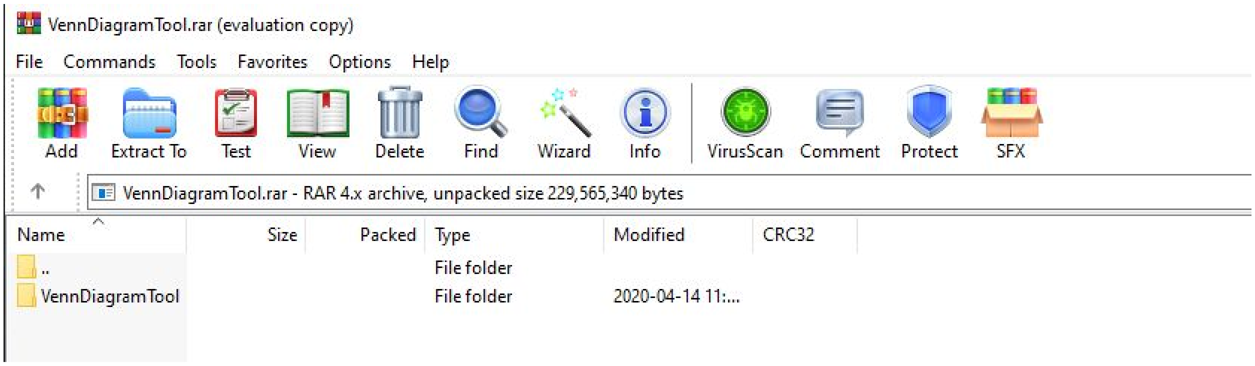
VennDiagramTool

chutneyslurpee released this yesterday

v1.0-alpha release

VennDiagramTool.rar	88.2 MB
Source code (zip)	
Source code (tar.gz)	

After clicking the VennDiagramTool.rar file, download it then open it with WinRAR or any application that can open .rar files.

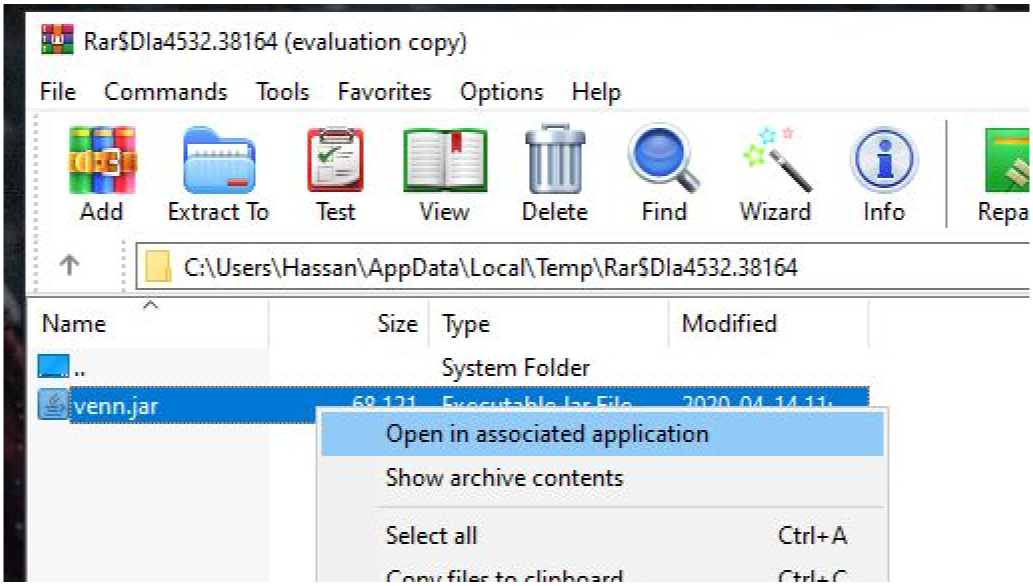


VennDiagramTool.rar (evaluation copy)

VennDiagramTool.rar - RAR 4.x archive, unpacked size 229,565,340 bytes

Name	Size	Packed	Type	Modified	CRC32
..			File folder		
VennDiagramTool			File folder	2020-04-14 11:...	

3) Click on the VennDiagramTool folder and then run the executable “venn.jar” file.



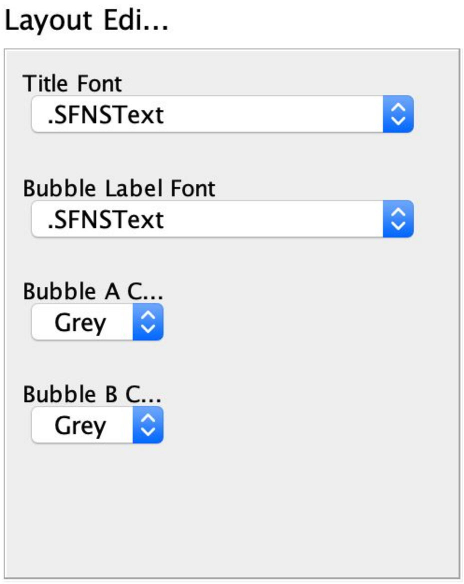
Rar$DIa4532.38164 (evaluation copy)

C:\Users\Hassan\AppData\Local\Temp\Rar$DIa4532.38164

Name	Size	Type	Modified
..		System Folder	
venn.jar	68,121	Executable Jar File	2020-04-14 11:...

INTERFACE INSTRUCTIONS

2.1: Layout Editor



Layout Edi...

Title Font: .SFNSText

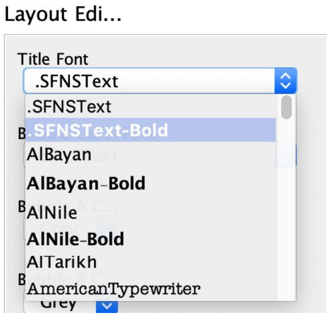
Bubble Label Font: .SFNSText

Bubble A C...: Grey

Bubble B C...: Grey

The layout editor has four buttons which can be used to customize the Venn Diagram

a) Title Font: Changes the font of the title



Layout Edi...

Title Font: .SFNSText

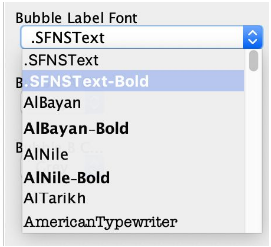
Before:

Venn Diagram Title

After:

Venn Diagram Title

b) Bubble Label Font: Changes the font of the bubble labels



Bubble Label Font: .SFNSText

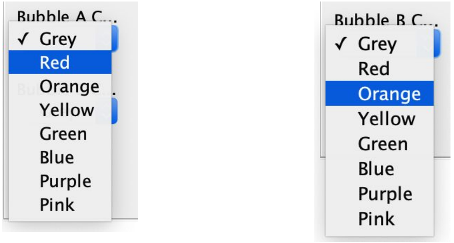
Before:

Bubble 1 Bubble 2

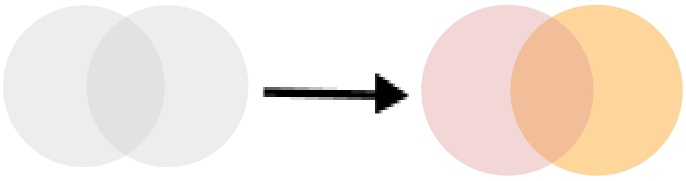
After:

Bubble 1 **Bubble 2**

c) Bubble A/B Colour w/ Advanced Mode Setting: Change the colour of either circle

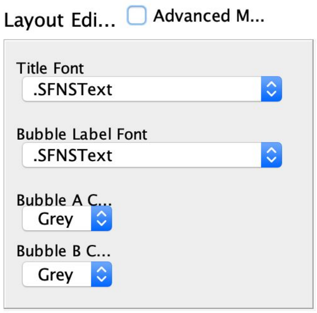


Before: After:



Additionally the "Advanced Mode" button may be clicked to show an advanced colour editor

Before:



Layout Edi... □ Advanced M...

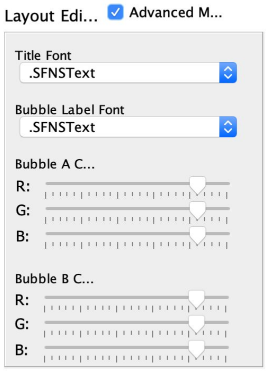
Title Font: .SFNSText

Bubble Label Font: .SFNSText

Bubble A C...: Grey

Bubble B C...: Grey

After:

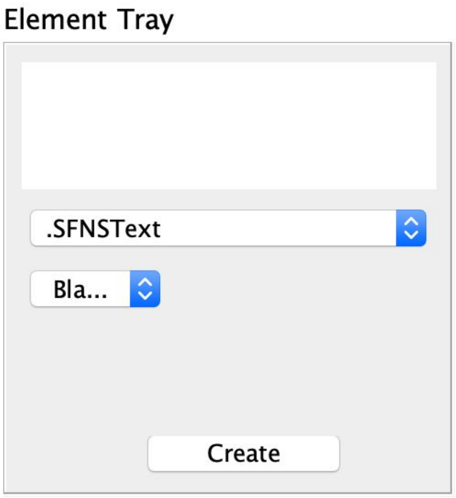


Layout Edi... ☑ Advanced M...

Title Font: .SFNSText

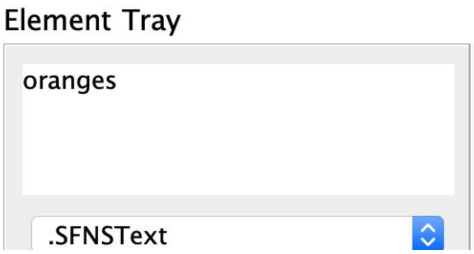
Bubble Label Font: .SFNSText

2.2: Element Tray



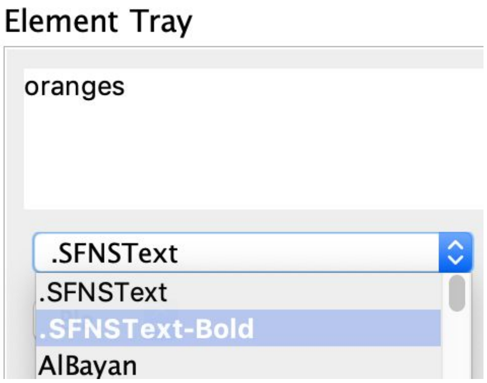
The element tray has a textbox where text can be written and 3 buttons that edit and create the text

a) The TextBox: The text that one wants to insert into the Venn Diagram is written here



oranges

b) Text Font: Changes the font of the text

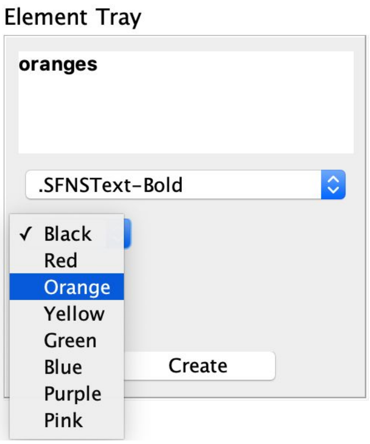


oranges

Before: oranges

After: oranges

c) Text Colour: Changes the colour of the text

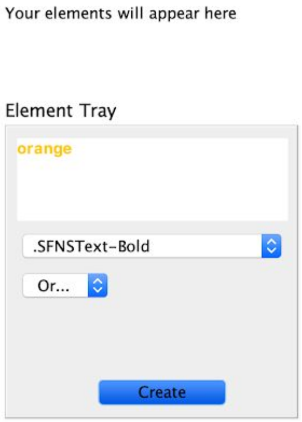


oranges

Before: oranges

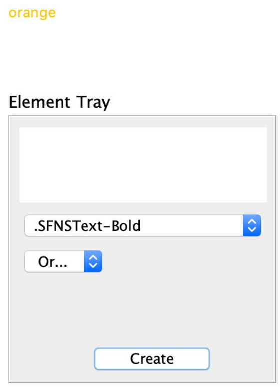
After: oranges

d) The "Create" button: creates the text made in the textbox to be put into either Venn Circle

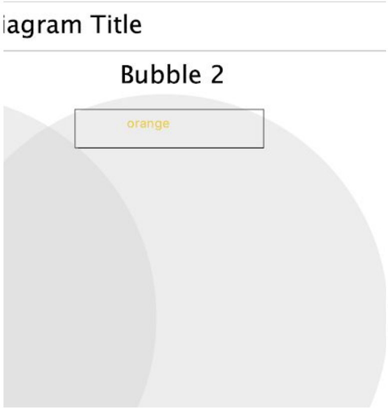


orange

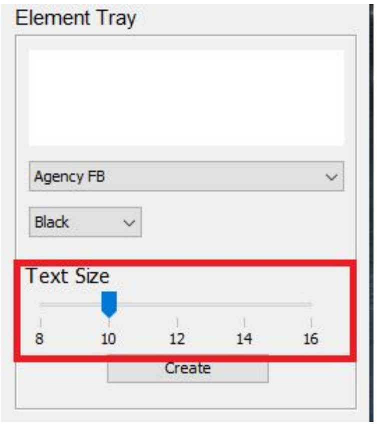
After clicking the "Create" button the text element will appear where it is written "Your elements will appear here."



Now, the created element can be dragged into either circle

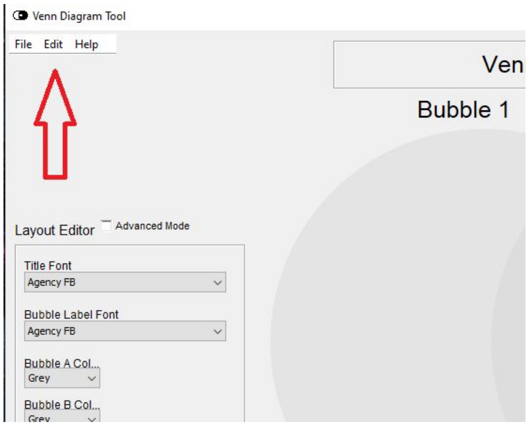


e) Text Size: Adjust the font size of the text. The red rectangle indicates where the text slider is located, the font can be adjusted anywhere between 8px to 16px.

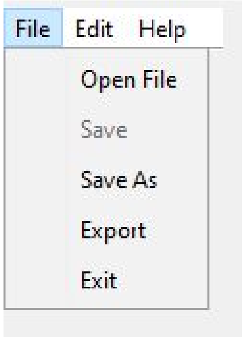


2.3: The Toolbar

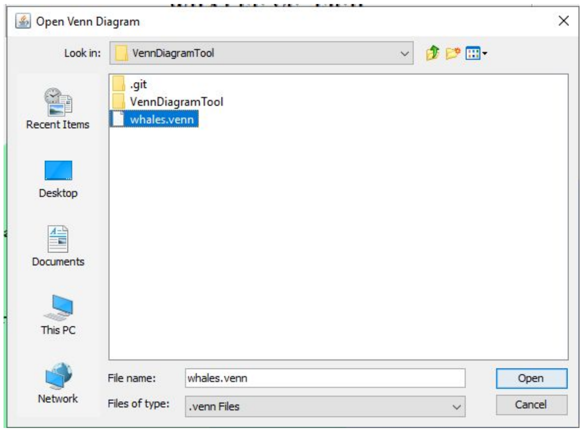
The toolbar can be found on the top left corner of the screen as indicated below:



i) File: Upon clicking “File”, a dropdown menu of five options opens up:



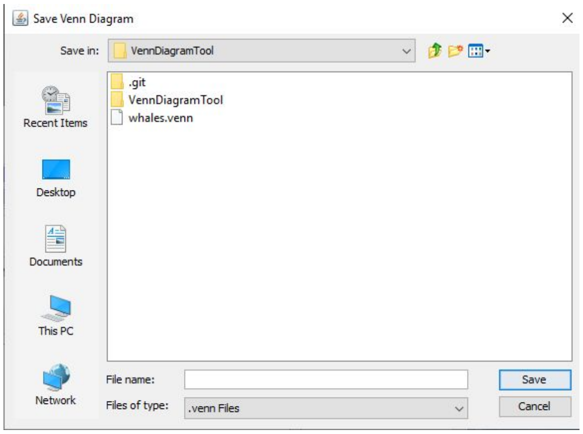
a) Open File: Used to open a pre-existing file. When clicked, a menu will appear which allows you to navigate through your PC to find .venn files



File name: whales.venn

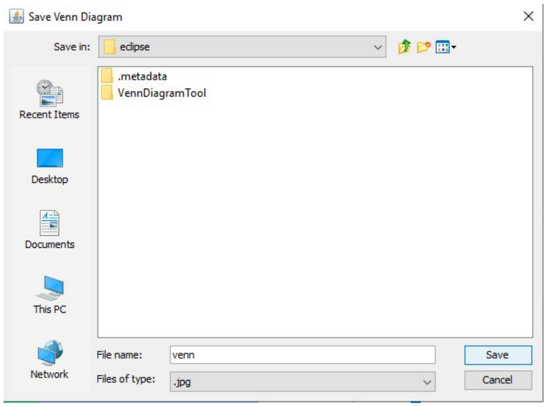
Files of type: .venn Files

b) Save/Save As: The Save function is used to save current work, and the Save As function is used to save the file into a specific location



Files of type: .venn Files

c) Export: When clicked, you get an option to export your project to a PDF.



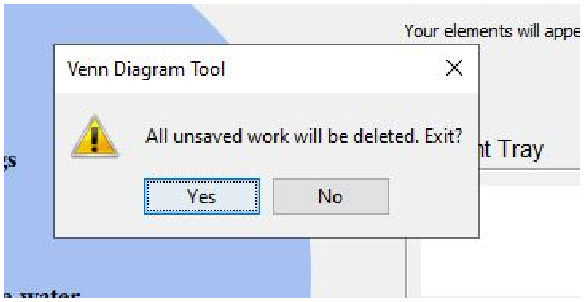
Save in: eclipse

.metadata
VennDiagramTool

File name: venn

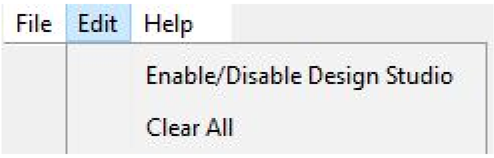
Files of type: .jpg

d) Exit: When clicked, an option to the application is given.



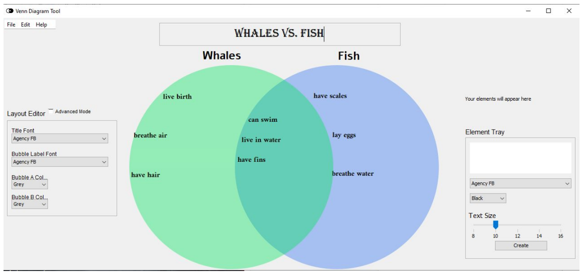
All unsaved work will be deleted. Exit?

ii) Edit: Upon clicking "Edit", a dropdown menu of two options opens up:



a) Enable/Disable Design Studio: The option lets you select between keeping the interface open or closed. When exporting to PDF the interface will always be invisible!

Before:



Layout Editor ☐ Advanced Mode

Title Font: Agency FB

Bubble Label Font: Agency FB

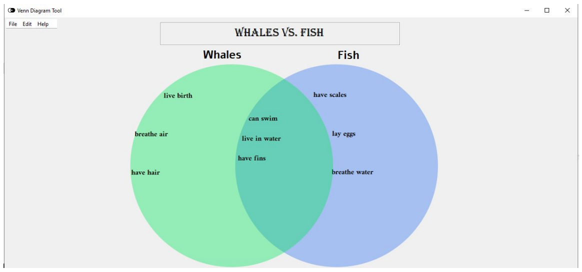
Bubble A Col...: Grey

Bubble B Col...: Grey

Agency FB

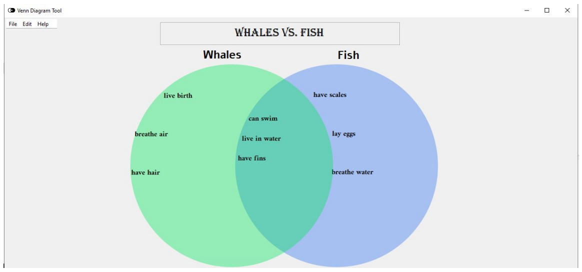
Black

After:

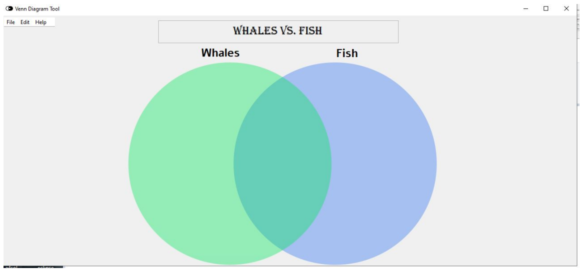


b) Clear All: Clears all the elements in the Venn circles.

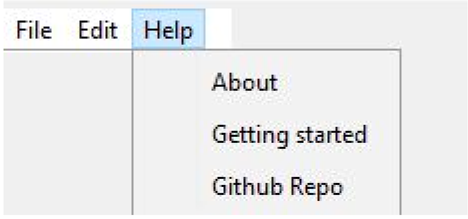
Before:



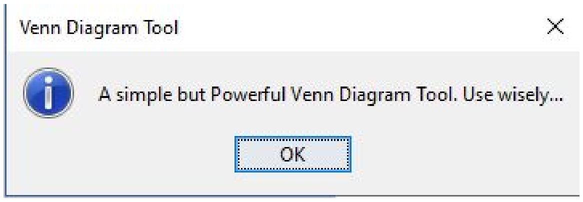
After:



iii) Help: Upon clicking "Help", a dropdown menu of three options opens up:

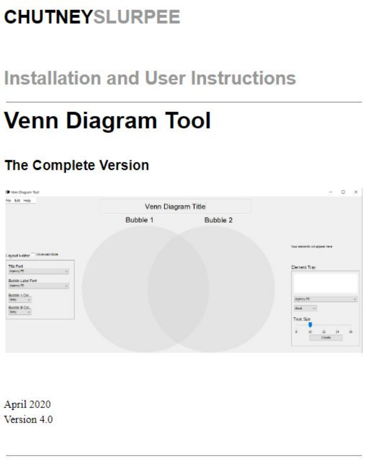


a) About: Opens up a prompt telling you what the Venn Diagram Tool is all about:


Venn Diagram Tool

A simple but Powerful Venn Diagram Tool. Use wisely...

b) Getting started: Opens up a PDF of the User Manual:


CHUTNEYSLURPEE

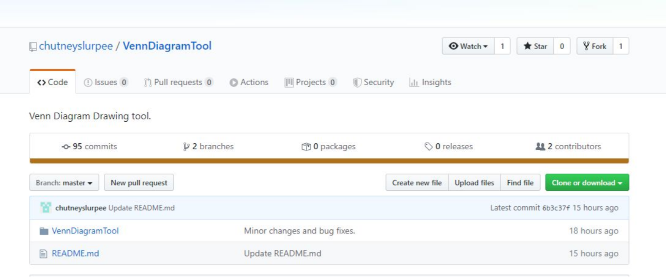
Installation and User Instructions

Venn Diagram Tool

The Complete Version

April 2020
Version 4.0

c) Github Repo: Opens the Github repository of the Venn Diagram Tool application:


chutneyslurpee / VennDiagramTool

Venn Diagram Drawing tool.

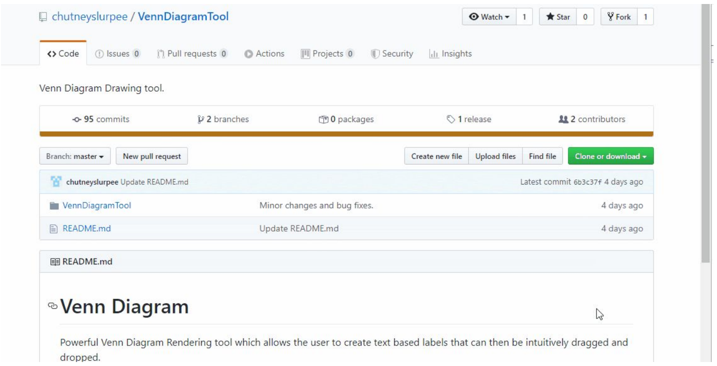
95 commits · 2 branches · 0 packages · 0 releases · 2 contributors

chutneyslurpee Update README.md		Latest commit 6b3c37f 15 hours ago
VennDiagramTool	Minor changes and bug fixes.	18 hours ago
README.md	Update README.md	15 hours ago

VIDEO DEMONSTRATIONS

3.2: Getting Started Demonstration

a) Installation:


chutneyslurpee / VennDiagramTool

Venn Diagram Drawing tool.

95 commits · 2 branches · 0 packages · 1 release · 2 contributors

chutneyslurpee Update README.md		Latest commit 6b3c37f 4 days ago
VennDiagramTool	Minor changes and bug fixes.	4 days ago
README.md	Update README.md	4 days ago

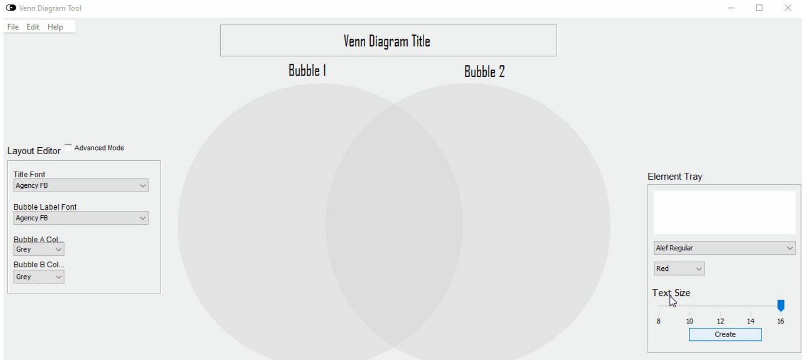
README.md

Venn Diagram

Powerful Venn Diagram Rendering tool which allows the user to create text based labels that can then be intuitively dragged and dropped.

3.2: Interface Demonstrations

a) Layout Editor:



Title Font: Agency FB

Bubble Label Font: Agency FB

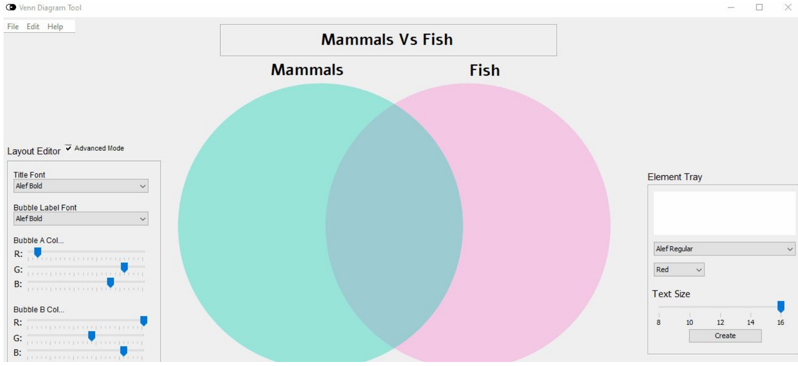
Bubble A Col...: Grey

Bubble B Col...: Grey

Alef Regular

Red

b) Element Tray:



☑ Advanced Mode

Title Font: Alef Bold

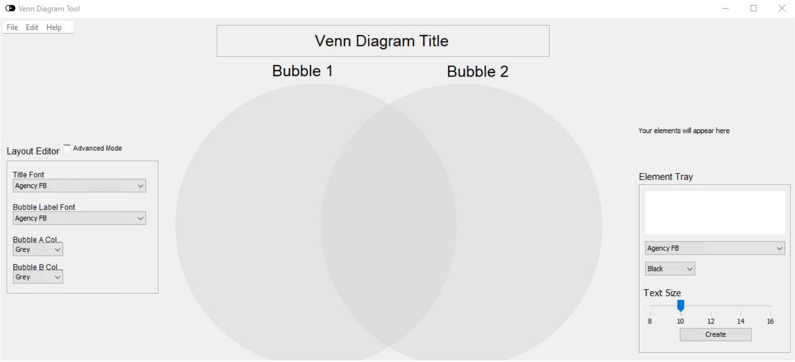
Bubble Label Font: Alef Bold

Alef Regular

Red

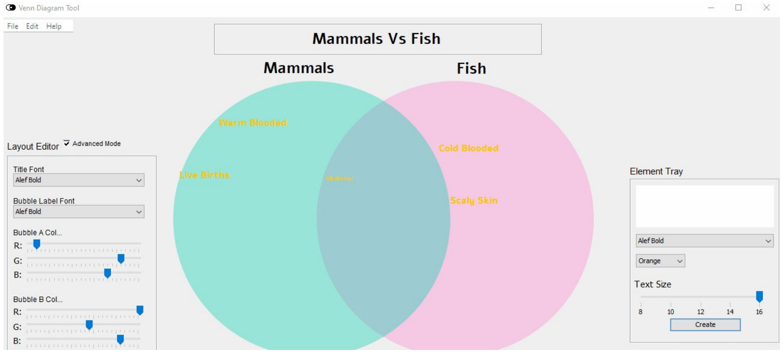
3.3: Interface Demonstrations

a) Open File:



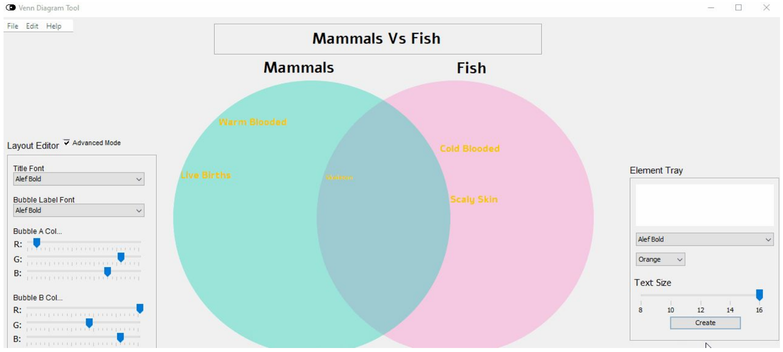
Title Font: Agency FB
Bubble Label Font: Agency FB
Bubble A Col...: Grey
Bubble B Col...: Grey
Agency FB
Black

b) Save/ Save As:



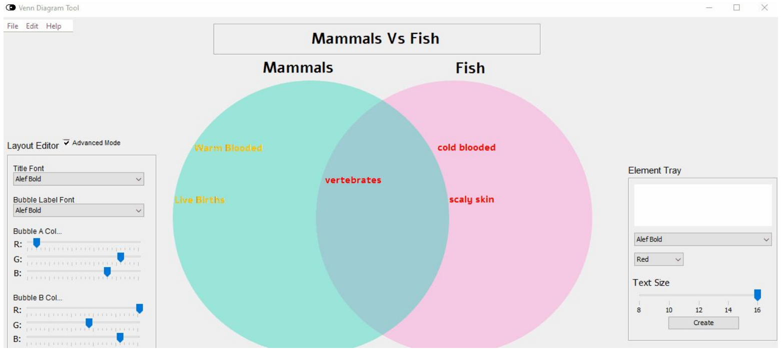
Title Font: Alef Bold
Bubble Label Font: Alef Bold
Alef Bold
Orange

c) Export:



Title Font: Alef Bold
Bubble Label Font: Alef Bold
Alef Bold
Orange

d) Edit Options:



Title Font: Alef Bold
Bubble Label Font: Alef Bold
Alef Bold
Red