

Virtual Tabletop

Final Report and Technical Specifications

A Senior Project in Computer Science at Yale University

for Michaela Papallo

Advisor: Benedict Brown

Apr 28, 2020

I. Abstract

In the interest of preservation, various 3D scans and high resolution images of ancient fresco fragments have been collected into an existing database. This data is grouped into “tabletops” based on the locations where the fragments were discovered, which sometimes but not always corresponds to fragments that fit next to each other in a completed reconstruction of the fresco. An existing program is able to provide a “virtual tabletop” view of the fragments and allow the user to virtually manipulate and reorganize these models, which are grouped by confirmed matches. However, the existing program depends on outdated frameworks that are not portable on every OS or device.

This senior project is designed to ultimately have many of the features of the existing program, but be especially tailored for portability and efficiency. It is a web application based on a Vue.js frontend and Node.js backend, but built on the Quasar framework which allows for flexible deployment. This project allows the user to browse the existing workspaces in the database which are stored as “.xml” files. Once a workspace is selected, it uses Leaflet plugins to display a map view of the fragments in the workspace, as well as allow for user interaction. Currently the functionality that is supported is to drag or rotate single or multiple fragments, and reset all changes – but there is no support yet for the user to save their changes. The map is meant to be responsive to touch as well, but testing on a tablet was not able to be performed due to the extenuating circumstances of this semester. There are certainly areas for improvement in extending the functionality of the visual interface, as outlined in this report.

II. Download and Run Code

The source code is available for download at <https://github.com/mpapallo/VirtualTabletop>.

Mask Images

The first step that need only be done once is to run the **mask-images** shell script (can be found in the *other/* folder of the Git repository) in your *tongeren_vrijthof_db/* directory. This uses an ImageMagick command to generate an image for each fragment of just the fragment’s top surface with no background.

Image Server

Start a server for your database that will serve static images from the fragment directory and serve “.xml” files from the workspace directory as JSON. See **image-server.js** (can be found in the *other/* folder of the Git repository).

Run Application

From the top-level *VirtualTabletop/* directory, run:

```
npm install
```

To start the app in development mode with full server functionality run:

```
quasar dev -m SSR
```

For production:

```
quasar build
```

III. Features and UI

Welcome Page

- The icon at the left of the header routes back to this home page.
- Use the navigation links at the right of the header to go to “Select a Workspace”.

Selection Page

- One section displays cards for all of the workspace (“.xml”) files in the *workspace/crates/* directory of the database; Another section displays the workspaces from the *workspace/matches/* directory. Select “View” for the desired workspace.

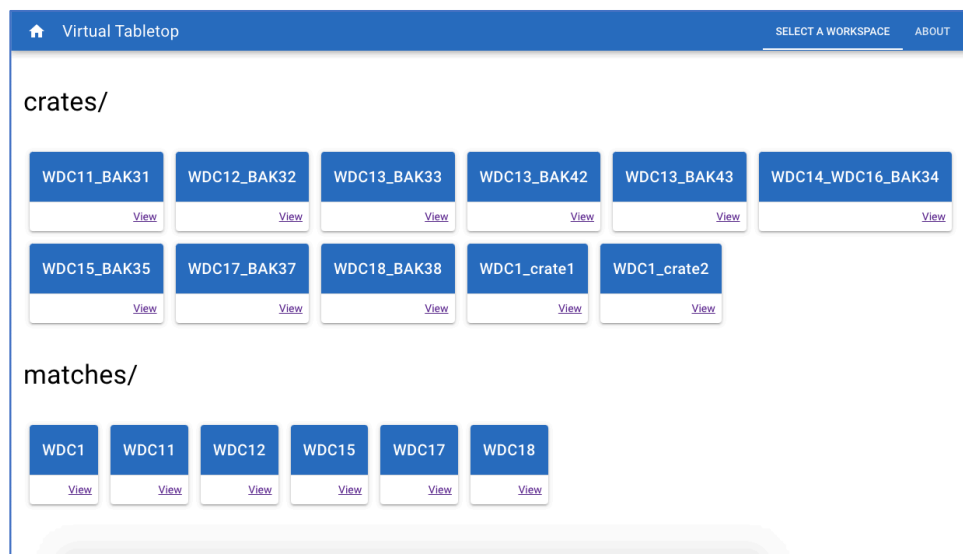


Figure 1: Selection page interface.

Workspace View

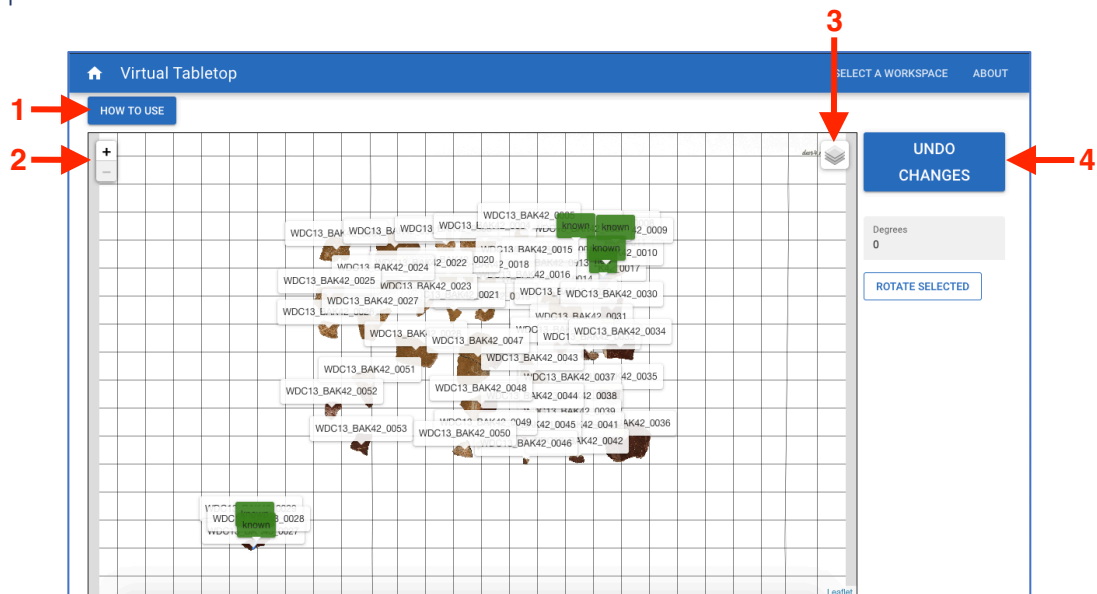


Figure 2: The full workspace interface.

1. **How to Use button:** Opens a dialog box with some tips explaining the workspace interface, click OK to exit.

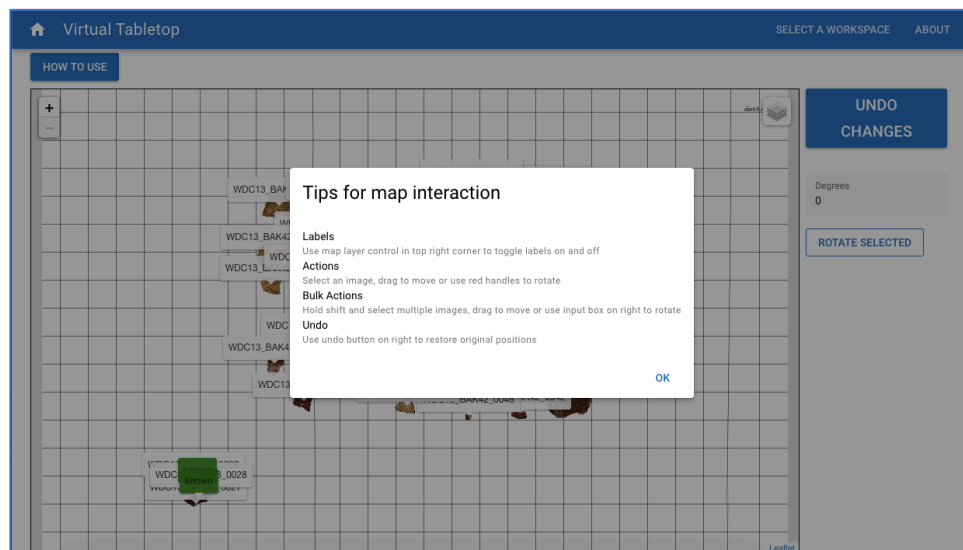


Figure 3: "How To Use" dialog box.

2. **Leaflet Map Zoom Control:** Used to zoom in or out on the workspace, alternatively the user can scroll within the Leaflet map container, presumably the user can pinch to zoom on a touch screen.
3. **Leaflet Map Layer Control:** Hover over the top right icon to reveal the layer control, where the labels and match annotations can be toggled on and off.

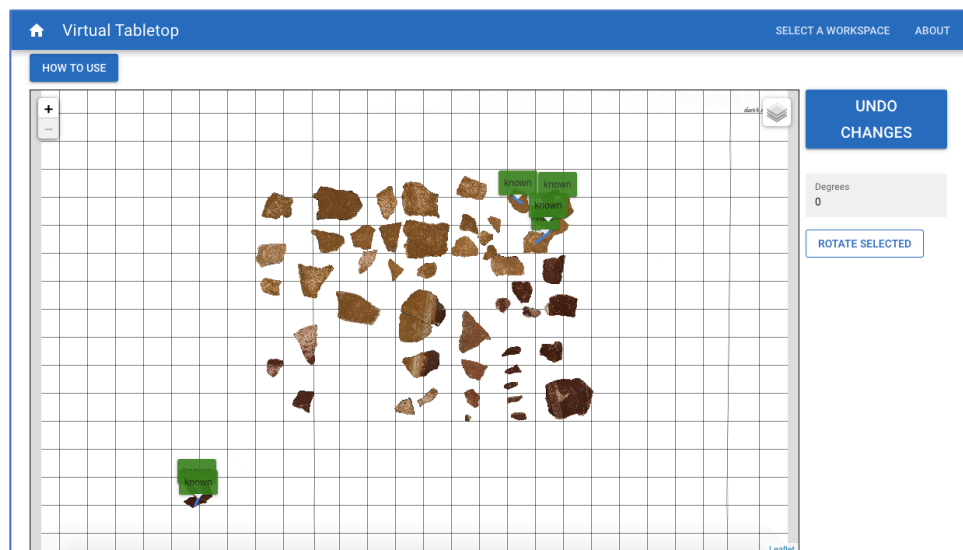


Figure 4: Labels toggled off for clearer zoomed out view.

4. **Undo Changes Button:** Resets all fragment images to their original positions.

Fragment Interaction

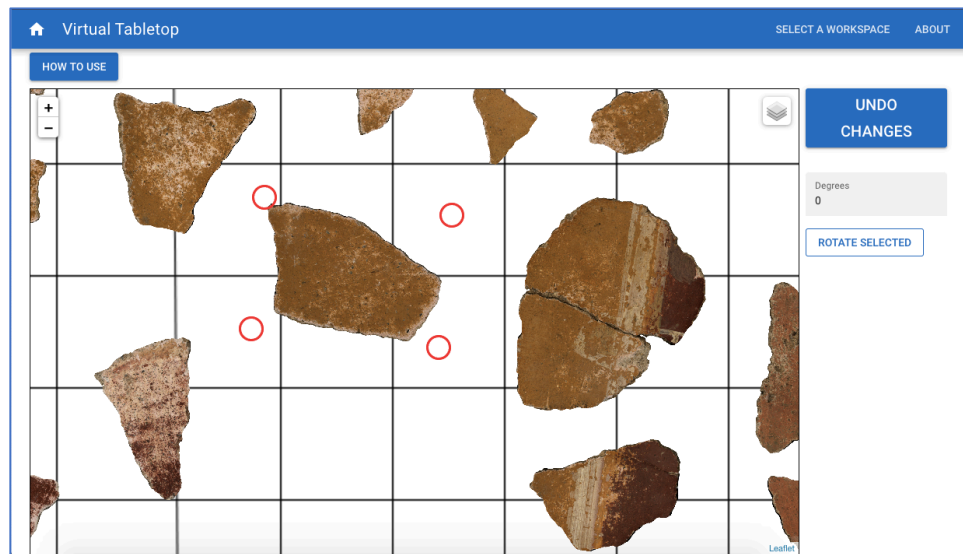


Figure 5: Click to select a fragment.

- Click to select a fragment, drag to move it around, or use the red handles that appear to rotate it in place. Presumably dragging is touch screen friendly as well. (With labels toggled on, label positions will update automatically when image positions update.)

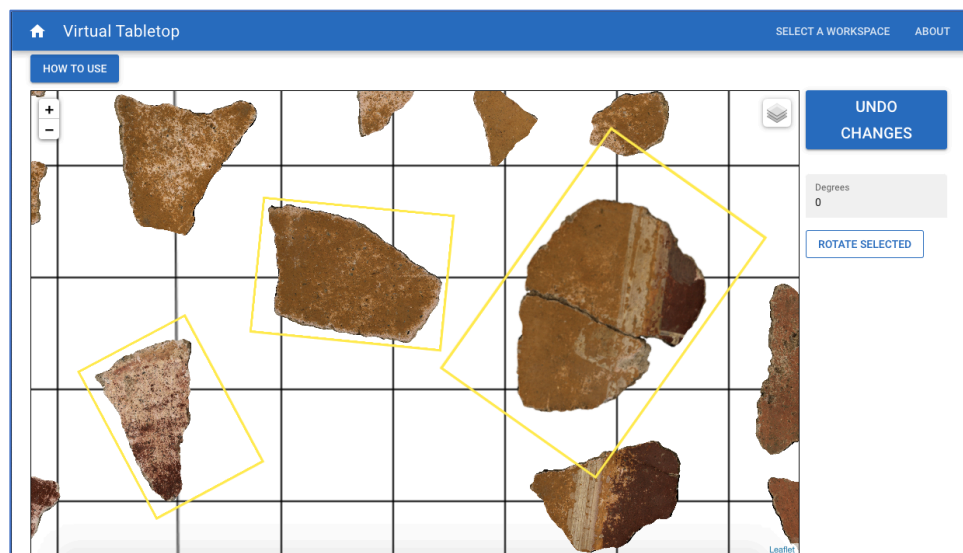
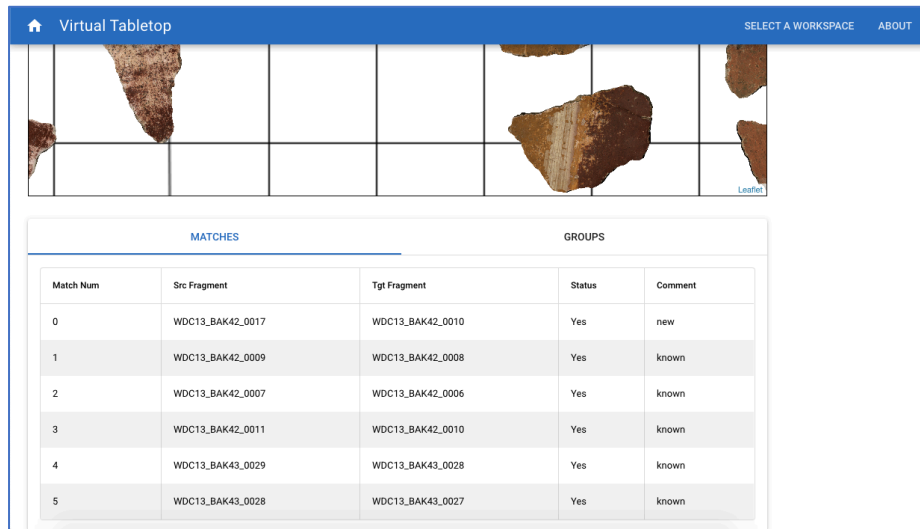


Figure 6: Hold Shift to select multiple fragments.

- Hold shift and click on multiple fragments to select them, drag to move them simultaneously. Use the input to the right of the map to specify a rotation angle around a common origin. Alternatively, the user can hold shift and click+drag the mouse to create a selection box. Presumably the user can press and hold each fragment on a touch screen.

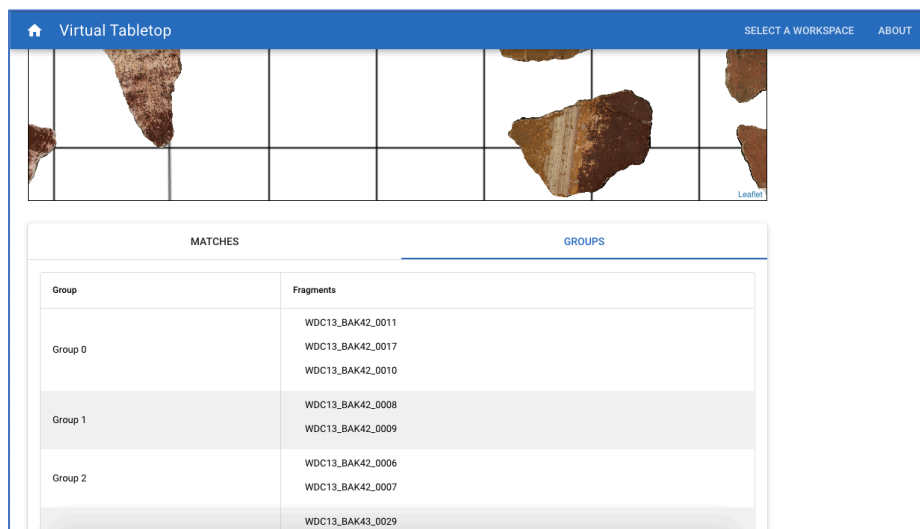
More Workspace Information



The screenshot shows the 'Virtual Tabletop' interface. At the top, there's a blue header with a home icon, the text 'Virtual Tabletop', and links for 'SELECT A WORKSPACE' and 'ABOUT'. Below the header is a grid of fragment images. Underneath the grid is a table with two tabs: 'MATCHES' (selected) and 'GROUPS'. The 'MATCHES' table has five columns: 'Match Num', 'Src Fragment', 'Tgt Fragment', 'Status', and 'Comment'. It contains six rows of match data.

Match Num	Src Fragment	Tgt Fragment	Status	Comment
0	WDC13_BAK42_0017	WDC13_BAK42_0010	Yes	new
1	WDC13_BAK42_0009	WDC13_BAK42_0008	Yes	known
2	WDC13_BAK42_0007	WDC13_BAK42_0006	Yes	known
3	WDC13_BAK42_0011	WDC13_BAK42_0010	Yes	known
4	WDC13_BAK43_0029	WDC13_BAK43_0028	Yes	known
5	WDC13_BAK43_0028	WDC13_BAK43_0027	Yes	known

Figure 7: Scroll down further for match information table.



The screenshot shows the 'Virtual Tabletop' interface with the 'GROUPS' tab selected. The table has two columns: 'Group' and 'Fragments'. It lists three groups of fragments.

Group	Fragments
Group 0	WDC13_BAK42_0011 WDC13_BAK42_0017 WDC13_BAK42_0010
Group 1	WDC13_BAK42_0008 WDC13_BAK42_0009
Group 2	WDC13_BAK42_0006 WDC13_BAK42_0007 WDC13_BAK43_0029

Figure 8: Switch tabs for fragment group information.

IV. Code Organization

Database

- `image_server.js`
- `public/`
 - `tongeren_vrijthof_db/`
 - `mask_images.sh`
 - `fragments/`
 - `workspace/`
 - `crates/`
 - `matches/`

Application

VirtualTabletop/

- **src-ssr/**
 - index.js
 - extension.js
- **src/**
 - **components/**
 - GroupTable.vue
 - MatchTable.vue
 - **layouts/**
 - basic.vue
 - **pages/**
 - about.vue
 - Index.vue
 - selection.vue
 - workspace.vue
 - **router/**
 - routes.js

Backend

- index.js
 - Creates express server for application on default port localhost:8080
- extension.js
 - Implements CAS authentication for the express app
 - */get-workspaces* endpoint queries the db server for a list of valid workspaces
 - */workspace* endpoint queries the db server for the JSON info for a specific workspace (given its folder and ID), and formats it in a useful way for the frontend to handle (a list of groups, ungrouped fragments, and matches)

Frontend

- components/GroupTable
 - Component for listing groups and ungrouped fragments
- components/MatchTable
 - Component for displaying match pair information (comments may be useful)
- pages/selection.vue
 - Interface for selecting a workspace, queries the */get-workspaces* endpoint
- pages/workspace.vue
 - Interface for display and interaction with workspace, queries the */workspace* endpoint with a specific workspace ID
 - Leaflet map initialized in the mounted() lifecycle hook, images and labels are populated immediately

V. Further Improvements

- **Rotation of multiple fragments** is currently only supported by specifying the number of degrees by which to rotate a group of fragments. This is because the Leaflet plugin

DistortableImage (specifically DistortableCollection) does not yet support manual rotation of a group of selected images in the way that users can rotate one image. Perhaps the functionality will be added in a future update, or perhaps there is a way around this by creating some custom control and applying a transformation matrix to each image individually (which is what the current rotation button does).

- **Portability** was a big factor in this project proposal. The plugins used should have much flexibility and touch support out of the box, but the project needs more testing (specifically on a tablet or other device) to confirm and fine-tune this functionality.
- **More visual features** such as using L.Polygon to draw borders around groups of fragments could be helpful to improve the user interface.
- **Further database interaction** including the ability to save changes to workspaces or create new workspaces.
- **Support for 3D views** was one of the stretch goals of the original proposal, and unfortunately was not started this semester. The database contains all of the files necessary for viewing 3D meshes with full texture and shading. In the future this project can be extended by using some Vue plugin with Three.js functionality to display a 3D view of each individual fragment.

VI. Resources

Quasar documentation

<https://quasar.dev/start/quasar-cli>

Node modules

<https://www.npmjs.com/package/cas-authentication>

<https://www.npmjs.com/package/xml2json>

Leaflet and Plugins

<https://leafletjs.com/reference-1.6.0.html>

<https://github.com/publiclab/Leaflet.DistortableImage>