Audio Tactile Map (ATM) Prototype

Version 4.0 - August 2015

Liam O’Sullivan

Git test

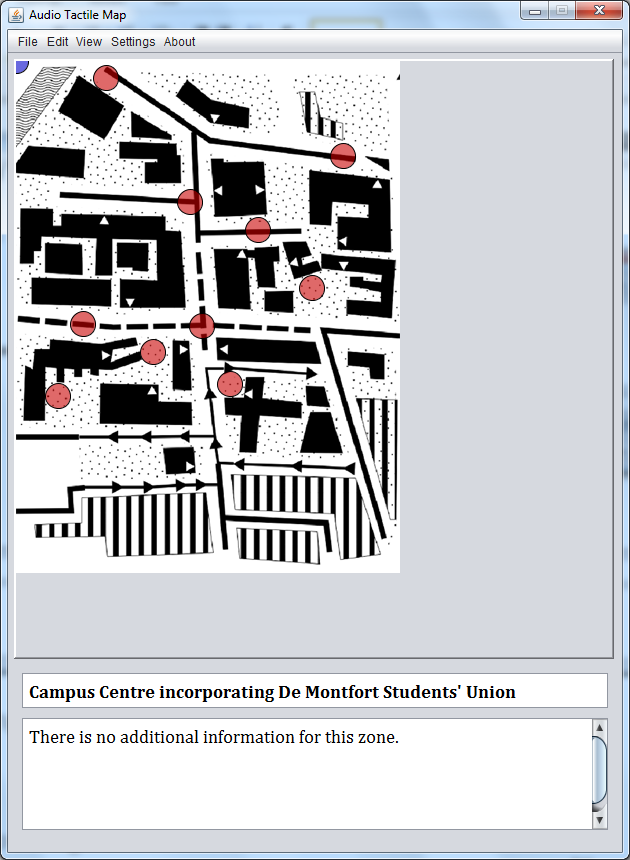


Table of Contents

[Notes to add: 1](#_Toc426646064)

[1. System Setup 2](#_Toc426646065)

[1.1 Download 2](#_Toc426646066)

[1.2 Directory Structure 2](#_Toc426646067)

[2 Running the Application 2](#_Toc426646068)

[3 Choosing an Input Source 4](#_Toc426646069)

[3.1 Arduino Configuration 5](#_Toc426646070)

[3.2 Hardware Setup Guidelines 5](#_Toc426646071)

[3.3 Leap Motion Configuration 5](#_Toc426646072)

[Digital Map Window 6](#_Toc426646073)

[3.3.1 DMU Campus Map 6](#_Toc426646074)

[3.3.2 Mouse Interaction 6](#_Toc426646075)

[3.3.3 Tactile Map Interaction 6](#_Toc426646076)

[3.3.4 Queen’s Building Map 7](#_Toc426646077)

[3.4 Keyboard Shortcuts 7](#_Toc426646078)

[3.5 Settings Window 7](#_Toc426646079)

[4 Known issues 7](#_Toc426646080)

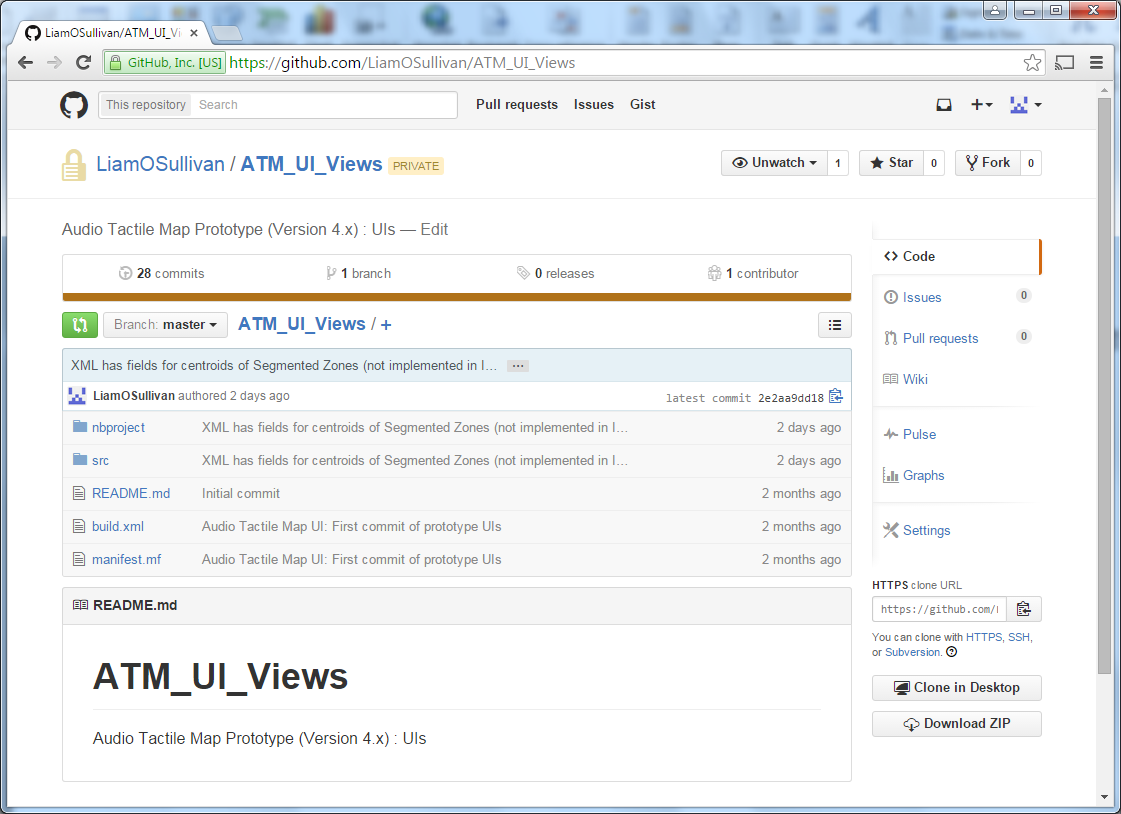
[4.1 Bugs 8](#_Toc426646081)

[4.2 Future improvements 8](#_Toc426646082)

[Appendix A 8](#_Toc426646083)

# System Setup

This section provides guidelines for the setup of the Audio Tactile Map (ATM) system prototype, with guidelines for hardware/ physical setup and details on running the software on the Windows platform.



## Download

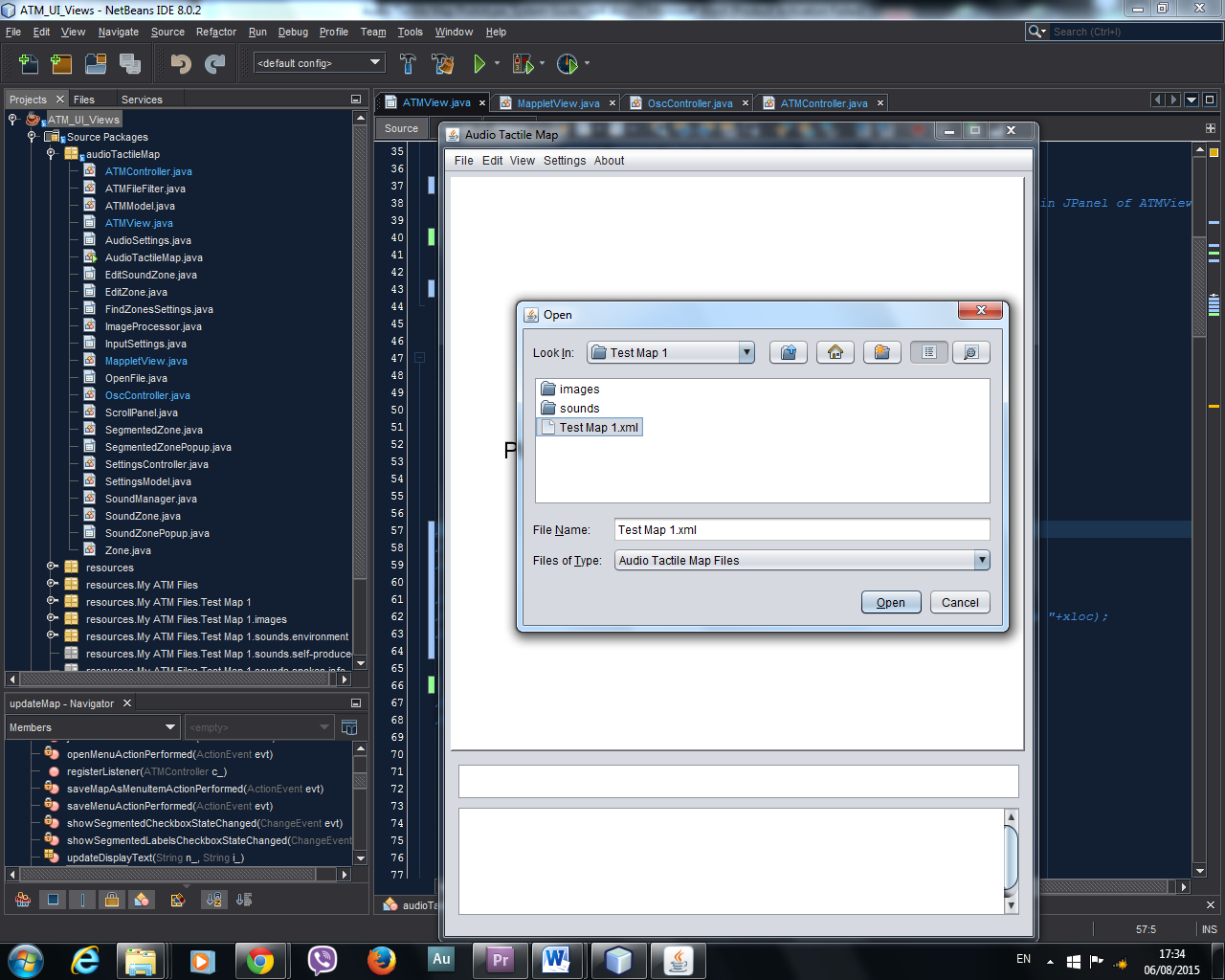
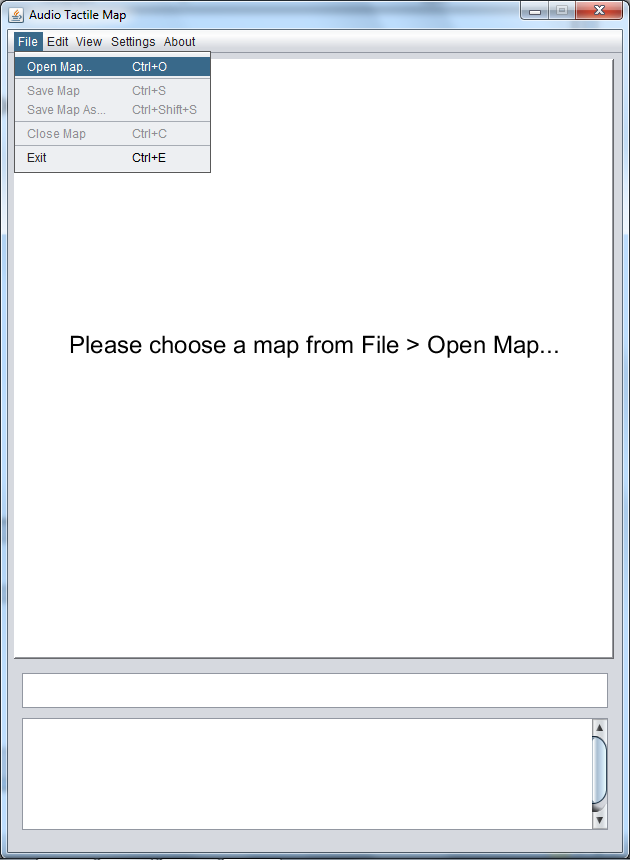
The source code for the ATM is located in a repository on GitHub at [https://github.com/LiamOSullivan/ATM\_UI\_Views as per figure 1.1](https://github.com/LiamOSullivan/ATM_UI_Views%20as%20per%20figure%201.1). Access is required to view and download the source files.

## Directory Structure

# 

# Running the Application

The ATM application may be started by clicking on the shortcut XXX



## Opening a Map

For convenience, map information and settings for a particular map are stored in a single file formatted with Extensible Markup Language (XML). As such, opening a map requires selecting the associated **.xml** file via the **Open Map…** command in the **File** menu, as shown in fig xxx. Details of the format of these files are given in section xxx. Once a map has been loaded a number of actions take place:

1. Information is loaded from the map file to identify the metadata associated with particular features on the map.
2. Processing of the map image creates a lookup table of segmented sections corresponding to buildings and other map features.

## Saving a Map

Changes made to a map (see section xxx) can be saved to the current map file (overwriting the file) or to a new file using the S**ave as…** command. When a new version of a map is created in this way, the directory structure of the original map is copied so that all assets and dependencies are available to the new map.

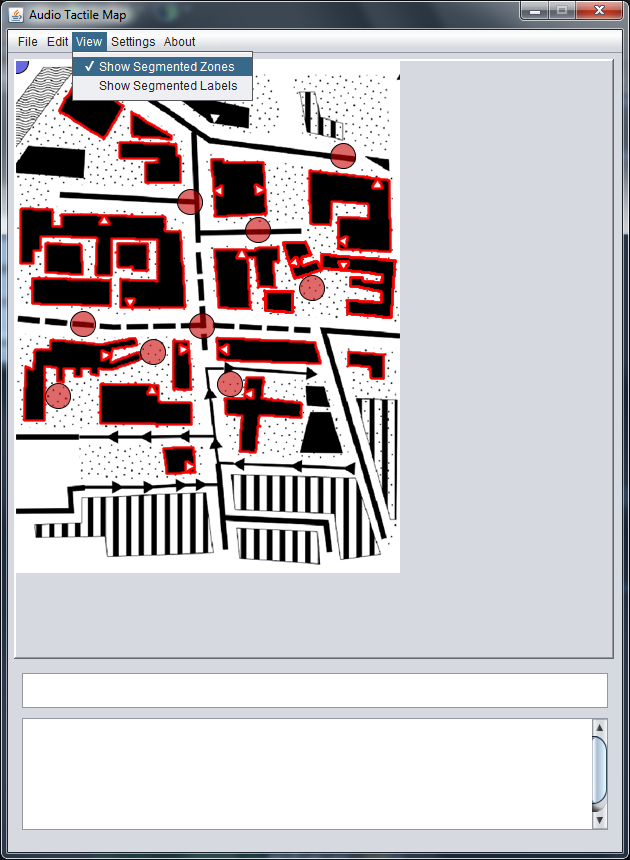
## Map Navigation

An image of the currently loaded map is shown in the main application GUI panel. The example map shown in fig xxx is of the campus of De Montfort University in Leicester.

### Types of *Zone*

The ATM system categorises two types of *Zone*  an interactive map; *Segregated Zones* and *Sound Zones*

The map returns information on the selected map location through an auditory display. The type of audio information returned will vary based on the nature of the location chosen. In this example, buildings return information such as their name and faculty association through a text-to-speech voice synthesiser or via pre-recorded audio files. Exterior spaces can also have binaurally-recorded environmental sounds associated with them, as indicated here by the red circles on the map display. The audio cues encoded in such background sounds can aid recognition of location and navigation in visually-impaired individuals.



### Mouse Interaction

The computer mouse may be used to interact with the map for testing or illustration purposes.

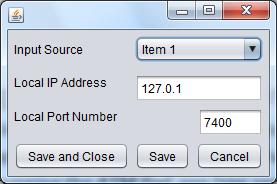
* **Left-click** on a building to return the name of the building, both as textual information in the text display areas below the map and as audio feedback rendered using a text-to-speech (TTS) synthesiser.
* **Left-click** on a *Sound* *Zone* to start playback of the binaural recording made at the corresponding map location.

## Editing a Map

# Choosing an External Input Source

Users may explore a tactile paper version of a map, identifying map features using various textures and the Braille map key provided at the bottom. A suitable input device can be used to add interactivity to the paper map; the current system has been tested with two particular input devices; (i) an *Apple iPad* running the *ATM Pad* app (see section xxx), and (ii) a *Microsoft Kinect* sensor configured with the *Ubi Displays* tool and communicating using a *node.js* webserver (section xxx) .[[1]](#footnote-1) The ATM application can take input from any device capable of sending *Open Sound Control*(OSC)messages in the appropriate format (see xxx).

Selection and configuration of the input source can be done through the **Input Settings** item in the **Settings** menu, as shown in fig xxx. Settings present in the map **.xml** file are automatically loaded when the map is opened, but these may be edited using the Input Settings dialog. Note that changes made and saved using the dialog save button will not be written to the map file until the map itself is saved via **File > Save Map** or **File > Save Map As…** .



The position of the lead finger interacting with the physical map is indicated here in real time and the map is annotated with additional features depending on the metadata available in the map file. These are illustrated using the example map below.

# Input Sources

A number of control input sources may be used with the ATM and are detailed in the following sections. These are:

1. Apple iPad tablet computer.
2. Microsoft Kinect Sensor.
3. Leap Motion controller

In order to provide robust selection detection with the Kinect and Leap Motion, a set of electronic buttons are used. This necessitates the use of an Arduino microcontroller as detailed in 4.3.

*Please note that as of this software version, Leap Motion has been disabled as a control device, but it is documented below for reference purposes.*

## Apple iPad

An Apple iPad can be used as an input source by installing the ATMPad application. This must be provided with the IP address and port number of the computer running the ATM program.

## Kinect Sensor + Ubi Displays

The Ubi Displays software[[2]](#footnote-2) turns any horizontal surface into an interactive display using a *Microsoft Kinect* sensor. It is used here to provide a robust interactive surface only- a display or projector is used to calibrate the interaction space, but it can be replaced with a tactile map after setup (fig xxx).

The system overview is shown in figure xxx.

### Dependencies

GIT

node.js

socket.io

osc-web

Download and install nodejs from [http://nodejs.org](http://nodejs.org/), then:

**Nav to**

**Getting the Socket.IO Server**

The officially maintained Socket.IO server is written in Node.JS and available on NPM. To install Node.JS and NPM, click the Install button on the [Node.JS website](http://nodejs.org/). Then run the following command from a prompt:

npm install socket.io

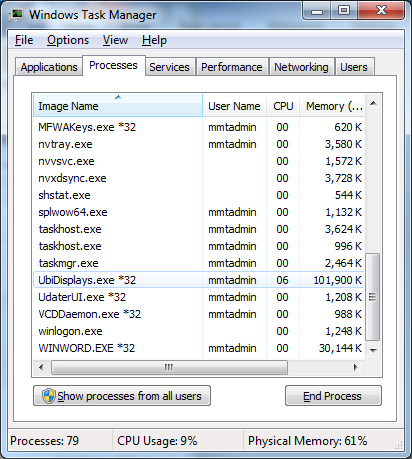
$ git clone https://github.com/automata/osc-web.git

$ cd osc-web/

$ npm install

### 

Creates folder osc-web at the current directory specified in cmd prompt.



May need to stop Ubi Displays in Windows Task Manager.

Start ubi displays beta application

Select desired screen, calibrate.

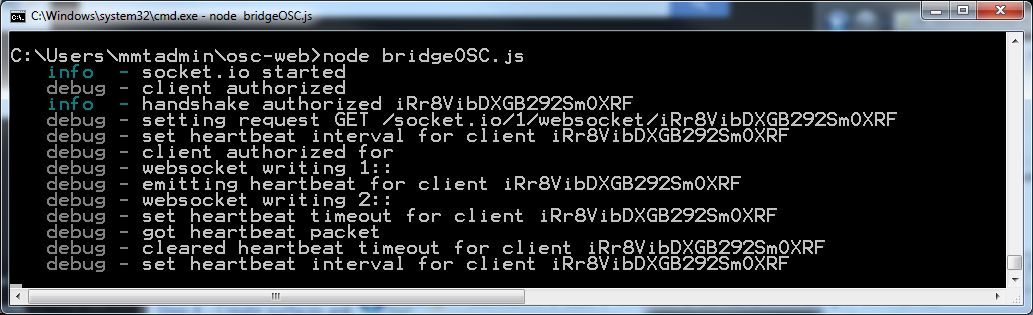
Place tactile map on screen (photo) and draw surface (screen).

Adjust boundaries to suit (photo)

Save if required

Drag the Blank Touch with OSC\_2.html to ubi displays

Start Node server by typing:



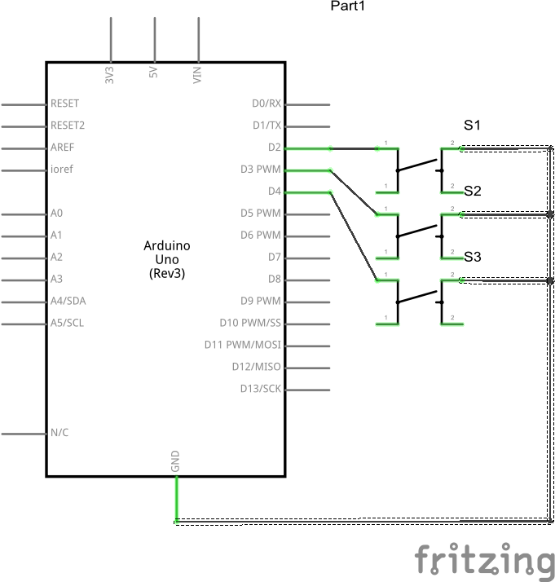
This screen will tell you if the server is still running correctly.

ubiTouchTest.pde

### Ubi Displays

### Node Server

## Arduino Configuration

When using the system with Ubi Displays, three electronic push-buttons perform various selection functions. These connect to the host computer running the system software via an Arduino microcontroller. The wiring diagram for the Arduino is shown below.

The circuit uses digital pins 2, 3 and 4 to read the states of the momentary buttons. On a push-button event, the states of the three buttons are sent to the main ATM system software over a serial USB connection.

Note that the sense pins are used with an internal pull-up resistor, and that logic values are therefore inverted before forwarding.

### Tactile Map Interaction

The Leap Motion tracks user interaction with the paper tactile map and the location of the forefinger is projected onto the digital map image. When a feature of interest such as a building is chosen, it may be selected using the large information push-button.

* **Point** to a map feature of interest on the tactile map using a distinct gesture. **Press** the large information push-button to retrieve information on that feature (Exploration mode).
* **Press** the green button to put the system into *Navigation* mode; the system responds stating that the route departure point has been set. **Press** the red button to set the route arrival point. **Press** the large information button to step through the pre-recorded sequence of navigational information (Navigation mode).

### Queen’s Building Map

Interior spaces may be explored using the ATM system prototype. The example shown below is of the Queen’s building, on DeMontfort University campus in Leicester. Additional binaurally-recorded information is also provided for the spaces associated with a building, as indicated by the yellow circles and red dots on the map display.

In this example, the background sounds associated with particular points in the building may be selected using one of the smaller selection buttons.

Self-produced sounds are often used by the visually-impaired as they navigate a space. These are also encoded in this map example and are shown as red circles on the map display. Selection is made using one of the smaller pushbuttons on the hardware interface.

This map only implements an exploration mode.

#### Mouse Interaction

* **Left-click** on a space to retrieve its information.
* Use **left-click + shift** to retrieve environmental sounds.
* Use **left-click + cntrl** to retrieve impulse sounds.

#### Tactile Map Interaction

* **Point** to a map feature of interest (e.g. a room or interior space) on the tactile map using a distinct gesture. **Press** the large information push-button to retrieve information on that feature.
* **Point** to a sound zone on the tactile map using a distinct gesture. **Press** the green button to start playback of environmental sounds recorded at the corresponding physical location.
* **Point** to a sound zone on the tactile map using a distinct gesture. **Press** the red button to start playback of self-produced impulse sounds recorded at the corresponding physical location.

## Keyboard Shortcuts

When the *Digital Map* window is selected, a number of keyboard shortcuts to system functions are available as per the table below.

|  |  |
| --- | --- |
| **Key** | **Function** |
| **S** (or **s**) | Leap Motion tracking stabilisation toggle |
| **C** (or **c**) | Calibration start |
| **B** (or **b**) | Background tracking toggle |

## Settings Window

TBC: As of this code revision, the settings dialog remains not implemented- all settings are automatically loaded to achieve the best results with the maps used in the prototype.

# Known issues

This is a list of known issues with the current system software, including both unresolved programming bugs and potential areas of improvement.

## Bugs

1. Action to recreate: On Queen’s building map, select sound zone #57.

Behaviour: Index-out-of-bounds error in audio module, source unknown.

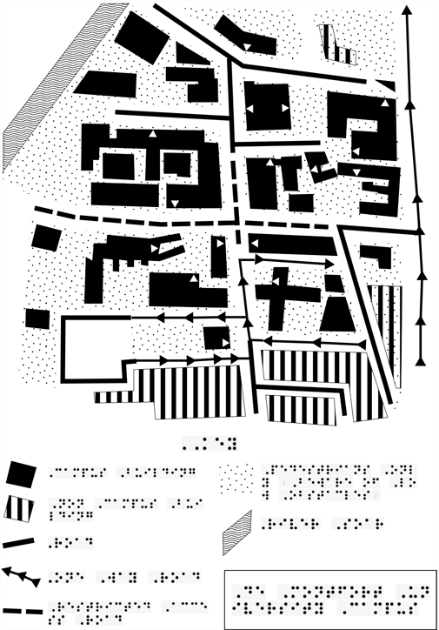
1. Action to recreate: On Queen’s building map, selecting sound zone #57.

Behaviour: no audio recording is rendered

## Future improvements

* Sound zones should use some type of bounded tessellation to fill the corresponding spaces instead of requiring that selection occur within the associated circular region. Selection should trigger the nearest available sound to the section point. This should use a labelling segmentation technique similar to the map image processing to allow fast lookup of index number.
* The actions of the push-buttons should stop playback of sounds when released, rather than triggering playback of a file which continues until its end or until another file is triggered.
* Revise tracking algorithm to implement new features of the Leap Motion API, e.g. more robust finger location and identification via skeletal tracking (beta released week 3 July 2014).
* Full implementation of a more sophisticated projection class (*ProjectionComplex*) to account for skew of paper map relative to Leap Motion detection plane.
* Change hard-coded navigation sequence to functional real-time route-following.
* Render full set of high-quality TTS replacement audio files and/or implement a more natural TTS engine. If in ‘replace TTS mode’, play notice when no pre-recorded file.
* Integrate all metadata, settings and calibration data into a single file (XML format).
* Save/load calibration data to file to reduce need to recalibrate.
* The GUI should be re-designed (with consideration of accessibility etc.) and should include the following:
  + - Select menu for choosing map to load (could include path browser).
    - Options for (i) projection type (simple or complex), (ii) show/hide rulers, (iii) LEAP configuration; front finger only, stabilization and background tracking, (iv) dummy mode etc. (see settings class), (v) use TTS/ high quality TTS/pre-recorded files for information display.
* Optimization of CPU performance & memory usage.

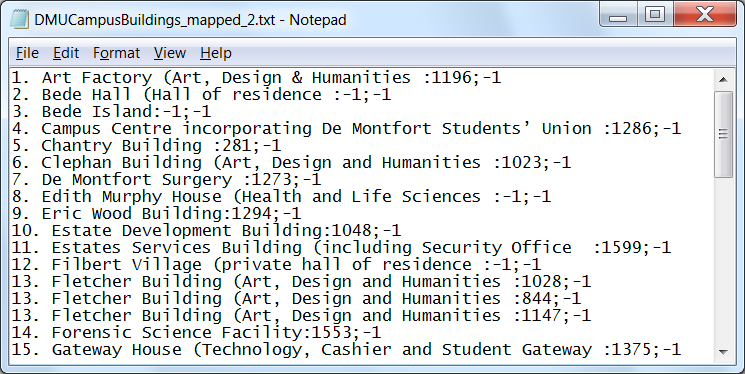
# Appendix A

The input files required to run the ATM system prototype are as listed below.

1. A digital map image file. The file shown in the figure on the right is a portable network graphics (**.png**) file containing a map of the DMU campus and associated Braille annotation.

This map is scanned by the image processing module and blobs (with a weight greater than some minimum value) are segmented, resulting in each building being labelled with an index number.

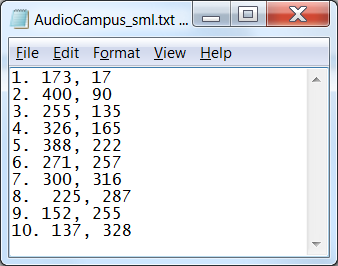
For the campus map, this image is used both by the image processing module and for display in the system GUI. In the case of the Queen’s building map, there are two images loaded, the image displayed in the GUI and a simplified and colour-inverted version used for segmentation.

1. A text file containing the names and information for all spaces on the map. In the case of the campus map, this contains a list of all campus builds as shown in the figure on the right.

For the ATM system to be able to parse this information, the text is formatted as per the table below.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Index (integer)* | *Full stop* | *Space* | *Building name* | *Space* | *Open bracket* | *Building info* | *Colon* | *Blob number* | *Semi-colon* | *Blob label* |
| **1** | **.** |  | **Art Factory** |  | **(** | **Art, Design & Humanities** | **:** | **1196** | **;** | **-1** |

The blob number is used to match the textual information to the list of segmented blobs returned by the image processing module. A blob number of -1 means the building is not included in the map.

1. A textual list of the sounds embedded in the map. As shown in the figure on the right for the campus map, this provides the index number (followed by a full stop) and coordinates (separated by a comma) of the software Sound Zone that will be created and displayed on the map. The system will attempt to load the associated sound file based on the index number e.g. for the first sound in the list shown, it will attempt to load the file ‘**1.wav**’.
2. Audio files containing environmental sounds, self-produced impulse sounds, verbalised directions for route navigation etc. These are placed in subfolders within the main **data** folder. For the campus map, the relative path is **../data**/**sounds/campus/** for the 10 wave files listed in the text file shown in 3 above. The current system requires that the files be named as per the convention ‘**1.wav’** with no spaces in the filenames. Some file paths are hardcoded into the software in this version of the prototype (e.g. route navigation information).

# Notes to add:

* When creating a new map, the map image should be scaled to a suitable onscreen size prior to adding to the map. Screen coordinates can then be set for zones etc.

1. Note that Leap Motion input functionality has been disabled in this current version of the software. [↑](#footnote-ref-1)
2. https://code.google.com/p/ubidisplays/ [↑](#footnote-ref-2)