# Log Book

25/10/2018

Worked on data structure to store information about building layouts.

* Created “Wall” class
* Created “Building” class

Encountered issues with using JavaFX to display the GUI for the desktop application- the JDK is not set up correctly to work with JavaFX. In version 11 of the JDK JavaFX has been moved outside of the code Java libraries.

26/10/2018

Continued to look into the issue with JavaFX.

Investigating using Maven to manage dependencies.

28/10/2018

Figured out how to use JavaFX version 11 with version 11 of the JDK.

Updated the project to use Maven, and applied the fix for JavaFX to the project:

* Add JavaFX as a dependency to Maven pom.xml
* Edit the way IntelliJ builds the project to make sure it compiles the pom.xml- otherwise it will cannot find the correct runtime

Video on setting up Maven and JavaFX: <https://www.youtube.com/watch?v=r_tdK8zWr_w>

IntelliJ Java Version Setup: [https://stackoverflow.com/questions/21006136/intellij-idea-13-uses-java-1-5-despite-setting-to-1-7#](https://stackoverflow.com/questions/21006136/intellij-idea-13-uses-java-1-5-despite-setting-to-1-7)

JavaFX Scene Graph: <https://docs.oracle.com/javafx/2/scenegraph/jfxpub-scenegraph.htm>

29/10/2018

Created a product backlog. Added user stories for the prototype to the backlog.

30/10/2018

Researched using a BorderPane for the layout of the GUI.

<https://examples.javacodegeeks.com/desktop-java/javafx/javafx-borderpane-example/>

<https://docs.oracle.com/javafx/2/layout/builtin_layouts.htm>

<https://docs.oracle.com/javase/8/javafx/api/javafx/scene/Scene.html>

Worked on rendering the layout of the building to the screen:

* Created a Canvas and a Graphics Context which can be used to draw to the canvas.
* Created a static method on the Building class that creates mock building layout for testing and prototyping purposes.

01/11/2018

Created an AnimationTimer which allows the application to display moving images.

Fixed an issue where the animation was very slow, and became slower the longer the application was open; the problem was that the method drawWalls() contained a memory leak and would continue to add the same data to the data structure over and over, rather than clear it in between method calls. The fix was to use beginPath() on the graphics context to clear the data structure.

02/11/2018

Created an array to store information about the positions of the people, and a mock testing method which will allow the GUI to be tested that it is reading the data structure properly. Wrote a method to draw the locations of the people to the GUI; this will be extended in the future. Currently I am planning the next steps to be:

1. Add a schedule to people that they will follow, now that they can be observed on the GUI
2. Write unit tests using Junit for the current code that I have, to check that the methods return what is expected in a variety of test cases.

07/11/2018

Started work on implementing an A\* pathfinding algorithm to allow the people to move around the building. Created a new class to store information about coordinates which is used in the pathfinding algorithm.

08/11/2018

Finished work on implementing the pathfinding algorithm. The algorithm can currently find a path if one exists; if no path exists, the person will simply not move- this behaviour may need to be adjusted in future.

14/11/2018

Added an edit mode to the program. The program uses a pane to display rectangles and circles in the edit mode, which will be made to be clickable in the future if the user wants to edit their position.

15/11/2018

Added a toolbar with pause/play button, speed up and slow down buttons. Added the ability for the user to speed up or slow down the animation- this currently has some bugs which occur when the animation plays too fast.

16/11/2018

Fixed bugs which occur by running the animation too fast. A limit was imposed on the speed the animation could be run at to prevent the bugs that occur when it runs too fast. A possible solution if the user wants to run a very fast simulation without watching it would be to add functionality that instantly simulates a day (without animating it) and reports statistics about the day to the user- for example, how much certain facilities are used.

22/11/2018

Started work on a menu to add a person. The menu currently does not feature all the necessary GUI components to work properly. I am currently investigating how to best display the person’s schedule to the user- tables in JavaFX are not editable by default.

23/11/2018

Started work on the loading/saving files functionality. The file format used will be .xml as this is a standard format that has been proven to work for storing a variety of data, and Java has good functionality for reading xml files.

24/11/2018

Finished off the loading/saving files. The application features a window where the user can choose which file to open and where to save their file. The application features both a save and save as method. There are some minor bugs with the functionality, for example the file extension does not always set correctly as .xml.

29/11/2018

Added a color picker to the add person menu. The button to add the person to the data structure now works, but it will probably have some bugs when this menu is used to edit the person- the id of the person who is being edited may not be saved properly so the program won’t know if this is an existing or new person. Added a unique UI field to the person class, which is currently initialised when any new person object is created.

30/11/2018

Fixed a bug with the .xml file where it could not read the people stored in it. Fixed another bug- when loading a file with 0 walls and 1 person the method would throw an exception due to trying to iterate through the (wrong) empty array. This was fixed changing the method to iterate through the correct array that is populated.

09/12/2018

Added a re-sizeable table class for the GUI. This can be used by the user to schedule their person’s tasks. They can add as many rows as they want to the table and delete rows as necessary. An improvement to this table would be to allow the user to click and drag rows to re-order them, however this is an advanced optional functionality that there probably won’t be time to implement.

12/12/2018

Fixed a bug where the .xml file would not save the schedule correctly due to the method not adding the array of activity elements to the .xml file. Added the ability for the player to set a user-friendly “name” for each person so that they can remember who they are. Added this name field to the .xml file so that it would be saved.

14/12/2018

Fixed a bug where, upon adding a new person to the data structure, they would not be shown in the building until the animation was played. Fixed a bug where the dot representing a person was twice the size in edit mode as in animation mode. Discovered a bug where the displayed x and y position of a person varies in edit mode and animation mode (this is a GUI layer bug not a back-end logic bug). Started investigating how to click on a person’s circle to edit their information- this will most likely require an object wrapper which associates each circle with a person’s id. This could also be achieved by checking the mouse position against which people are closest in the data structure. Added an edit menu to the top menu, added an “add person” functionality to this top menu.

Current plans for future development:

* Prototype version is now complete. Write unit tests for back-end logic
* Add in instrumentation. Before doing this a “room” object that stores information about rooms should be created- then measuring who is inside a room’s bounds becomes a simple task. The rooms should have information about the walls that make up the room. Rooms could also have different coloured flooring that represents the type of room they are.
* Add the ability for users to click on people and walls to edit them.