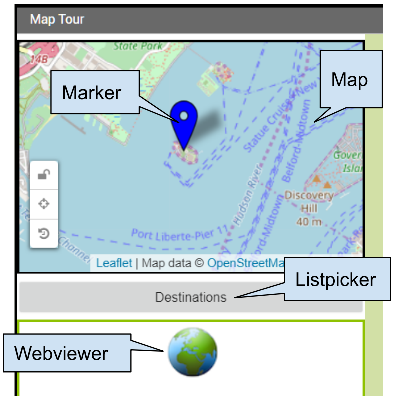
Map Tour Overview

I can:

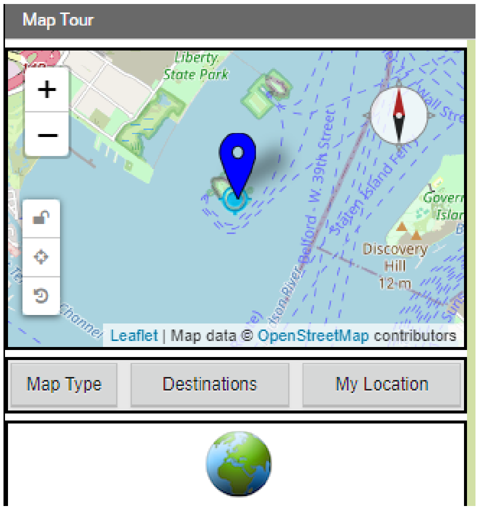
- Use the Map, ListPicker, and WebViewer UI components in MIT App Inventor
- Use lists to store and access destinations on the map and use an API (Application Programming Interface) to display Wikipedia pages of destinations

Getting Started

Open App Inventor and start a new project called Map Tour. Decide on 3 destinations/landmarks for your Map Tour. Choose well-known locations that appear in Wikipedia.



After Enhancements:



User Interface

UI Component	Name	Properties
Screen	Screen1	- Background Color of your choice - Title - Map Tour
Maps/Map	Map1	- Height - 50 percent - Width - fill parent - Check Show Compass, Show User, Show Zoom, try different Map Types - Center from String: set to a latitude, longitude using https://www.latlong.net/ or https://gps-coordinates.org/ or maps.google.com (right click on location and choose what's here).
Maps/Marker	Marker1, Marker2, Marker3 (rename with name of your locations)	- Check Enable InfoBox - Title - Name of your location - Set latitude and longitude from https://www.latlong.net/ if you do not see its location initially on the map.
ListPicker	ListPicker1 - Rename DestinationListPicker	- Text - Destinations - Width - Fill Parent
User Interface/ WebViewer	WebViewer1	- Width - Fill Parent

Coding the App

Abstraction: List Variables	Values
destinations	A list of destinations created using a make a list block, for example "Statue of Liberty", "Chichen Itza"
destinationsLatLong	A list of latitude, longitude strings corresponding to the destinations in the destinations list, for example ["40.689249, -74.0445", "20.684285,-88.567783"] which correspond to the latitude, longitude strings for the Statue of Liberty and Chichen Itza.

Event Handlers	Algorithms
DestinationsListPicker. BeforePicking	Set DestinationsListPicker.Elements to the destinations list.

DestinationsListPicker. AfterPicking	-Set Map1.CenterFromString to select a list item from destinationsLatLong list using DestinationsListPicker.SelectionIndex. -Call WebViewer1.GotoURL and join the text "https://en.wikipedia.org/wiki/" and DestinationsListPicker.Selection (you may need to add a replace text block to this to replace spaces with underscores (_).

Testing the App

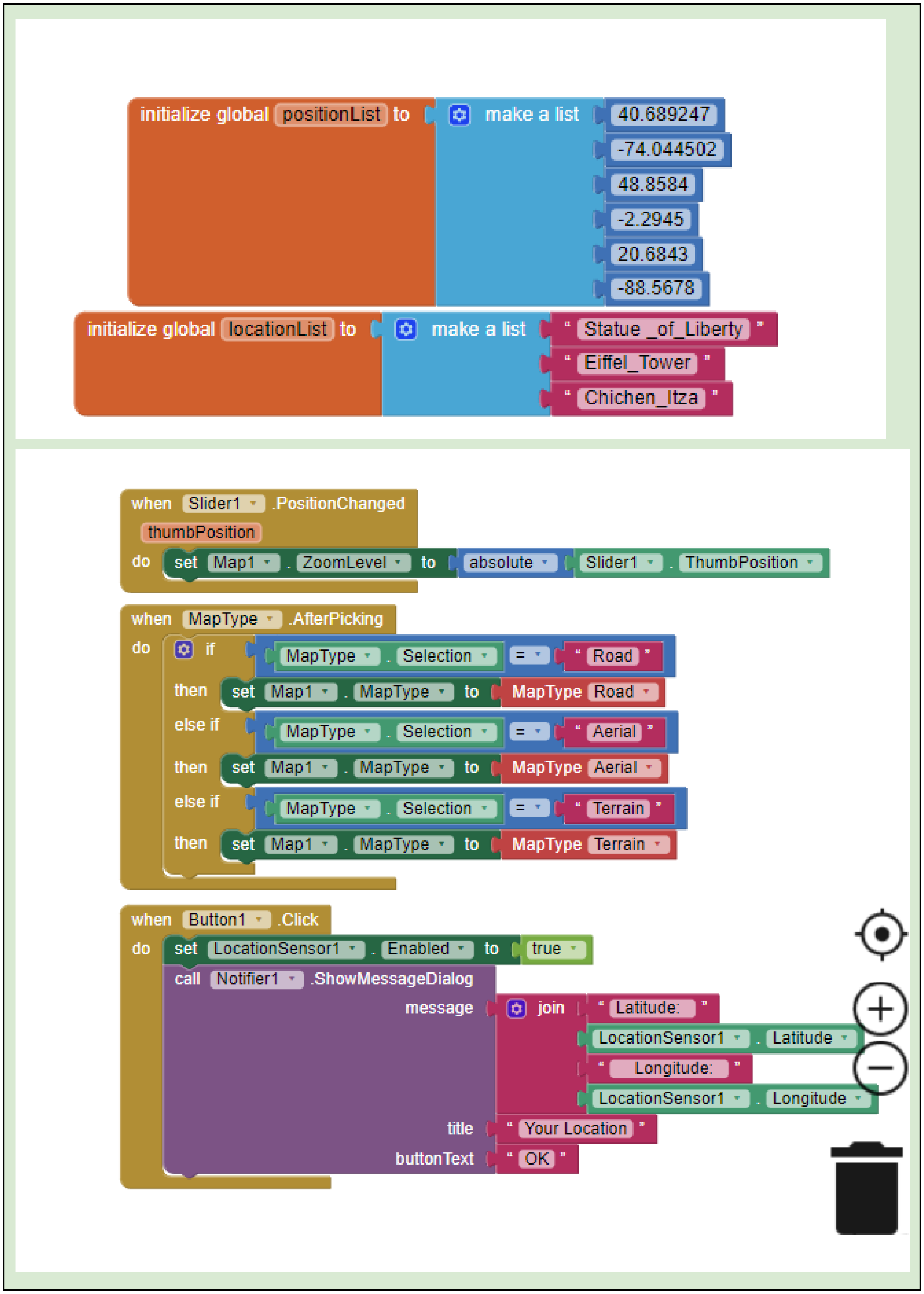
Inputs	Expected Outputs	Actual Outputs
Pick a destination	The map should change to the picked destination and the Web Viewer should display the wikipedia page for the destination if it exists. Make sure you try all on your list.	location's longitude and latitude, and as it loads it loads wiki

Required Enhancements

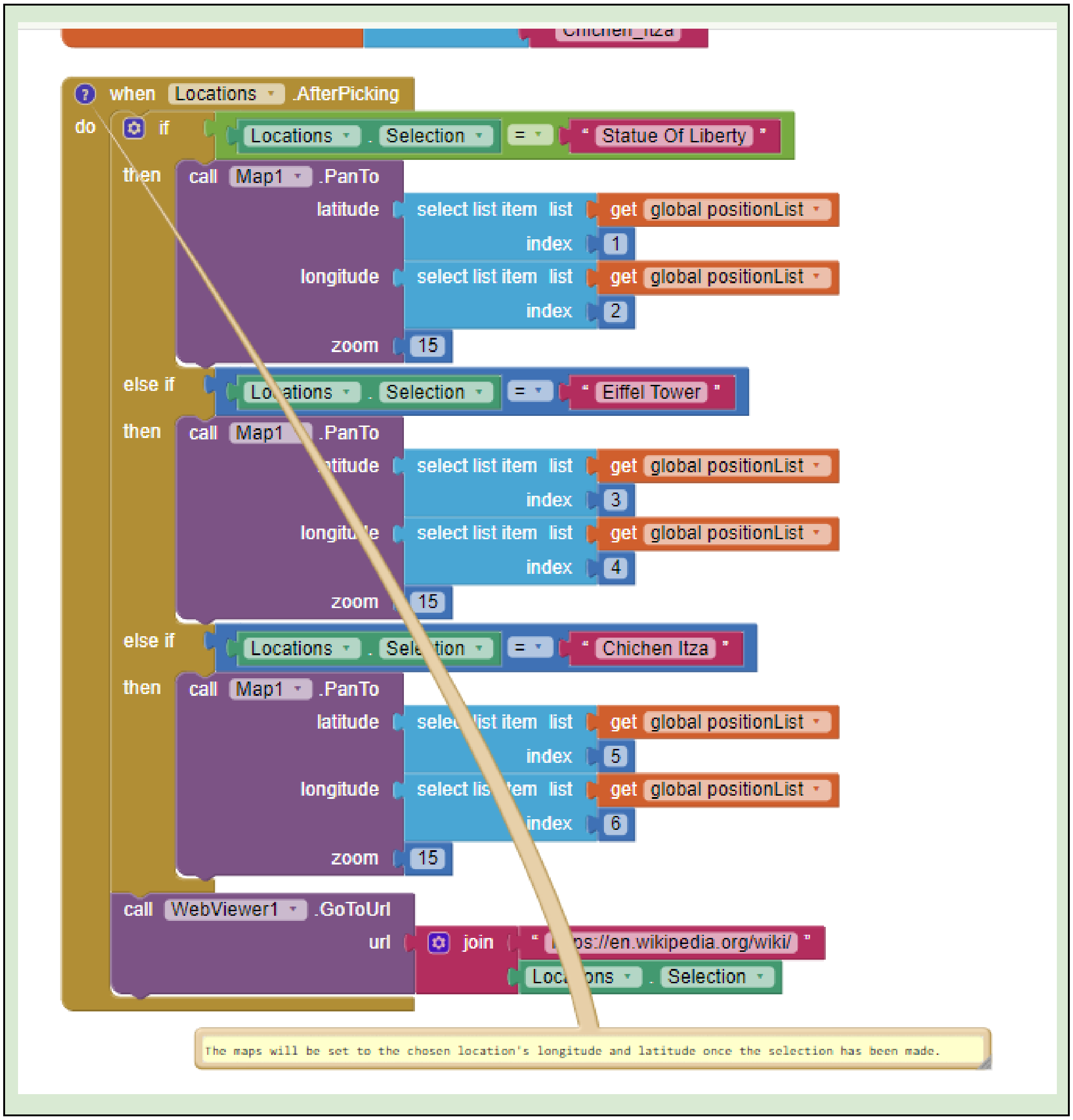
Be creative! We will add a database for the next iteration.

1. Add more destinations to your map tour. Make sure you have at least 3 destinations.
2. **MapType ListPicker:** Add a ListPicker to choose the Map Type with the Elements Roads, Aerial, and Terrain. These elements can be set in the UI or in the code in the BeforePicking event handler. After picking, use the user's Selection to set the Map.MapType to 1 for Roads, 2 for Aerial, and 3 for Terrain. You could do this with an if block using the blue mutator button to add if/elseif/else parts to make a 3 way choice.
3. **Zoom Slider:** Add a slider to your UI to control the zoom level on the map. You may want a horizontal arrangement to arrange these new controls. In the slider's properties, set the MaxValue to 20, MinValue to 1, and ThumbPosition to 13. The slider has a When Slider Position Changed event handler that is called when the user slides the slider. Inside this event, you can change the Map1's Zoom property to value in the Slider's ThumbPosition.
4. **My Location button and GPS:** OpenStreetMap keeps track of the user's location using GPS. The Map's properties UserLatitude and UserLongitude will give the latitude and longitude of the device currently running your app if the device has GPS capabilities. Add a button called My Location. When it is clicked, use the Map.PanTo procedure to go the the Map's UserLatitude, UserLongitude, Map.ZoomLevel. **NOTE:** This enhancement may not work with all devices and indoors.

Screenshots of Blocks 5 pts. (see rubric below)



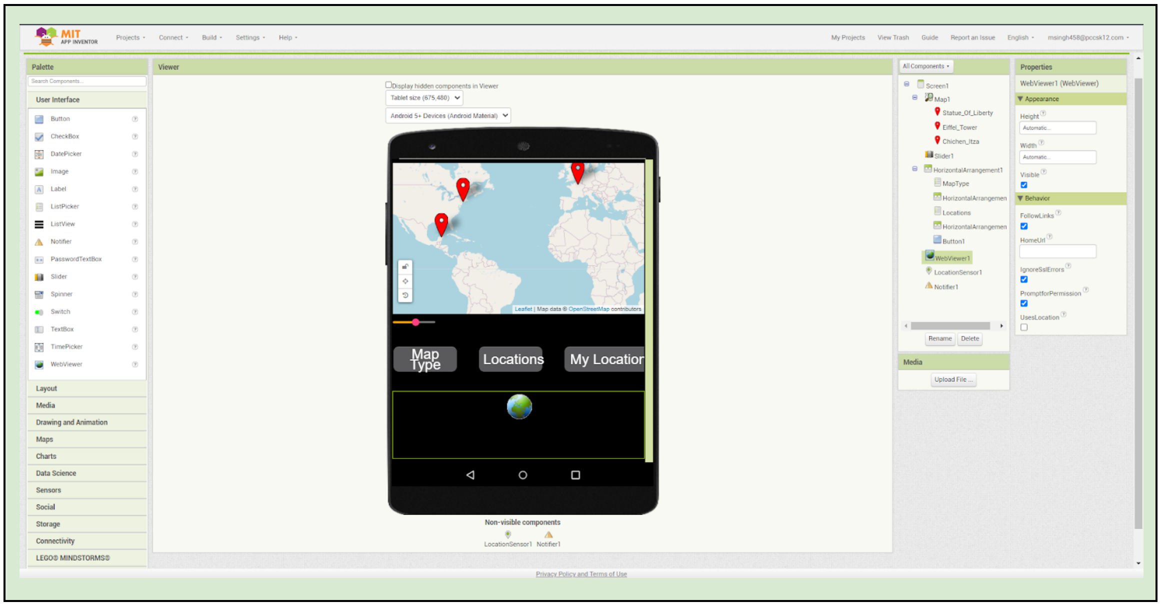
```
initialize global positionList to  make a list  40.689247
                                                -74.044502
                                                48.8584
                                                -2.2945
                                                20.6843
                                                -88.5678
initialize global locationList to  make a list  " Statue _of_Liberty "
                                                " Eiffel_Tower "
                                                " Chichen_Itza "

when Slider1 .PositionChanged
  thumbPosition
do  set Map1 . ZoomLevel to  absolute  Slider1 . ThumbPosition

when MapType .AfterPicking
do  if      MapType . Selection  =  " Road "
    then    set Map1 . MapType to  MapType Road
    else if MapType . Selection  =  " Aerial "
    then    set Map1 . MapType to  MapType Aerial
    else if MapType . Selection  =  " Terrain "
    then    set Map1 . MapType to  MapType Terrain

when Button1 .Click
do  set LocationSensor1 . Enabled to  true
    call Notifier1 .ShowMessageDialog
                          message  join  " Latitude: "
                                         LocationSensor1 . Latitude
                                         " Longitude: "
                                         LocationSensor1 . Longitude
                            title  " Your Location "
                       buttonText  " OK "
```



```
Chichen_Itza
when Locations .AfterPicking
do  if    Locations . Selection = " Statue Of Liberty "
    then  call Map1 .PanTo
                  latitude   select list item  list   get global positionList
                                               index  1
                  longitude  select list item  list   get global positionList
                                               index  2
                  zoom       15
    else if  Locations . Selection = " Eiffel Tower "
    then  call Map1 .PanTo
                  latitude   select list item  list   get global positionList
                                               index  3
                  longitude  select list item  list   get global positionList
                                               index  4
                  zoom       15
    else if  Locations . Selection = " Chichen Itza "
    then  call Map1 .PanTo
                  latitude   select list item  list   get global positionList
                                               index  5
                  longitude  select list item  list   get global positionList
                                               index  6
                  zoom       15
    call WebViewer1 .GoToUrl
                    url  join  " https://en.wikipedia.org/wiki/ "
                               Locations . Selection
```

The maps will be set to the chosen location's longitude and latitude once the selection has been made.

Screenshot of Designer View



Rubric

Criteria	Exemplary/Outstanding	Proficient	Does not meet expectations
User Interface Design	GUI is very appealing; layout is clear. It is very easy for the user to navigate. Evidence of your effort towards considering the user. Demo working app to Mrs. O’Neil.	GUI meets specifications and is easy to navigate.	Basic or incomplete and/or no demo to Mrs. O’Neil.
Naming Components	All variable names have been changed and given names that are descriptive of their content and type.	Most names have been changed to descriptive names.	Few or no names were changed from their defaults.
Code Organization	Code is well organized and easy to read AND code contains no dangling, useless blocks. Appropriate comments for code.	Code is arranged and code contains no dangling and useless blocks. Some comments or too many.	Code has little or no organization and is difficult to read. No comments explaining code.

Feature Implemented	All features are fully implemented to specifications AND code compiles and runs with no errors. Required enhancements implemented and you used creativity to improve the app.	Most features are fully implemented to specifications AND code compiles and runs with no errors.	Missing two or more features and/or code does not compile without errors.
Use of appropriate algorithms.	Code demonstrates use of appropriate algorithms.	Most of the code uses appropriate algorithms.	Algorithm does not fully function.