Deliverable F

Group 12: Brayden Baker, Shailen Mann, Nicholas Pighin, Jaron Roy

GNG1103

1. Abstract

This document outlines the first stage of Group 12's prototyping cycle, detailing all the initial pieces of software which we have developed to make progress towards our global design concept. It also explains the tests which were run on each of these prototypes, displaying the areas in which they succeeded and those in which they failed. The feedback of potential users is then included, which will be used in future prototyping cycles to help refine the prototypes in this first stage. An updated bill of materials (BOM), based on the discoveries during this prototyping cycle, is then discussed. Finally, the projected tests of the next prototyping cycle are then outlined, all of which are set to iterate upon Group 12's current prototypes using the knowledge we have gained from user feedback and our testing.

2. Defining the prototype

Our first prototype is split into 3 sections that all function independently from each other.

2.1. Boundary Box System:

The Boundary Box prototype uses Python code to determine whether a point is contained inside the boundary defined by specific vertices, as well as the distance between the point and each wall of that boundary. The goal of this prototype is to develop a system that can properly identify the location of a point within a set boundary. In the future, this code can be modified to accommodate Shabodi's location retrieval API in order to track the location of a UE relative to its boundary in real time.

2.2. Excel Alert/Documentation system:

This alert system takes set input values and inserts them into an Excel file, noting down the date and time. The goal is to develop a system that can take a set of variables and save/document them somewhere. In the future, this code can be modified to take the information returned by the Boundry Box System and document. In future prototypes, we intend to add the ability to send real-time notifications to individuals and expand the documented notifications.

2.3. Primitive GUI:

Currently, the GUI has a button to send an alert manually. The alert pops up in the center of the screen with an option to clear it. After 10 seconds the alert will disappear by itself. In future there will be a dedicated alert box on half of the screen where all the alerts will pop

up. Modable buttons will be added to increase functionality. These will be able to clear all the alerts, turn on and off the system, and save all the alerts to a dedicated email and/or file. The other buttons will be the choice of the clients. The final addition to our GUI will be the addition of a visual zone which displays a chosen zone and all the UE's inside of it.

3. Critical Components

GUI	A functional browser GUI which uses HTML and CSS, and which allows the modification of variables; these are then interpreted and used by back-end Python code to manipulate the UE's boundary, for instance.
Documentation System	Using python code with the openpyxl library is used to upload information from python into an excel spreadsheet. The time function is used to upload set information at varying intervals.
Alert System	Using simple mail transfer protocol (SMTP), specified users and system operators are alerted by email whenever the Python code determines that a UE has left its specified area.
UE-Boundary Tracking	Python code, using the math library, is used to evaluate a system of equations, returning the point on the boundary nearest the UE; this is used in the 2D distance formula to determine the distance between the UE and that point.

4. Prototype 1 Test Plan:

4.1. Boundary Box System:

Test Description	Within Boundary Test: code will evaluate whether a point sits within an area defined by other points	Zone Proximity Test: code will evaluate the distances between a point and each boundary around it
Reason for Prototype	Analysing critical subsystems --- Reducing risk and uncertainty	Analysing critical subsystems --- Verifying feasibility
Evaluation Criteria	Accuracy of the boundary checking portion of the software	Accuracy of the proximity calculation portion of the software
Level of Prototype	Middle fidelity --- Focused	Middle fidelity --- Focused
Kind of Prototype	Analytical	Analytical

Metrics	Correct? (boolean)	ΔDistance (units)
Analysis Method	Visual comparison with the Desmos graphing tool	Comparison to manual calculation with the same values
Stopping Criterion	3 randomly generated points, each tested on a random triangle, quadrilateral, and pentagon	3 randomly generated points, each tested on a random triangle, quadrilateral, and pentagon

Results

Point	Vertices	Code	Desmos
(5,4)	(4,9) (10,7) (2,3)	Out	Out
(5,4)	(3,5) (8,2) (9,4) (1,10)	In	In
(5,4)	(6,5) (2,4) (3,9) (10,7) (8,1)	Out	Out
(6,8)	(4,9) (10,7) (2,3)	In	In
(6,8)	(3,5) (8,2) (9,4) (1,10)	Out	Out
(6,8)	(6,5) (2,4) (3,9) (10,7) (8,1)	In	In
(10,1)	(4,9) (10,7) (2,3)	Out	Out
(10,1)	(3,5) (8,2) (9,4) (1,10)	Out	Out
(10,1)	(6,5) (2,4) (3,9) (10,7)	Out	Out

Point	Vertices	Code	Manual
(5,4)	(4,9) (10,7) (2,3)	4.427 0.447 2.530	4.427 0.447 2.530
(5,4)	(3,5) (8,2) (9,4) (1,10)	0.171 3.578 2.400 2.236	0.171 3.578 2.400 2.236
(5,4)	(6,5) (2,4) (3,9) (10,7) (8,1)	0.728 2.942 4.258 3.795 1.342	0.728 2.942 4.258 3.795 1.342
(6,8)	(4,9) (10,7) (2,3)	0.316 2.683 2.214	0.316 2.683 2.214
(6,8)	(3,5) (8,2) (9,4) (1,10)	4.116 5.000 1.400 3.900	4.116 5.000 1.400 3.900
(6,8)	(6,5) (2,4) (3,9) (10,7) (8,1)	3.000 3.138 0.138 4.111 3.000	3.000 3.138 0.138 4.111 3.000
(10,1)	(4,9) (10,7) (2,3)	6.000 5.367 8.222	6.000 5.367 8.222
(10,1)	(3,5) (8,2) (9,4) (1,10)	2.236 2.236 3.162 8.062	2.236 2.236 3.162 8.062
(10,1)	(6,5) (2,4) (3,9) (10,7)	5.657 8.544 6.000 1.897	5.657 8.544 6.000 1.897

		(8,1)				(8,1)	2.000	2.000
Interpretation	Pass: All 9 tests matched the result expected from the Desmos graph				Pass: All distances from all 9 tests were equal			

4.2. Excel Alert/Documentation system:

Test Descriptions	Code will upload information into an excel spreadsheet 10 times at a 5 second intervals between each update
Reason for Prototype	Analyzing critical subsystems --- Verifying Feasibility
Evaluation Criteria	Number of alerts updated into an excel spreadsheet at a correct interval.
Level of Prototype	Low Fidelity --- Focused
Kind of Prototype	Analytical
Metrics	Correct (Number of Alerts sent)
Analysis Method	Comparing time of alerts sent out in python to the time of excel file being updated.
Stopping Criterion	10 separate alerts updated into an excel spreadsheet
Results	
Interpretation	Pass: 10 updates were sent at 5 second time intervals

Print displayed in python terminal	Time excel Spreadsheet updated
Data written to output.xlsx at row 2 at 2024-11-03 19:59:23	19:59:23
Data written to output.xlsx at row 3 at 2024-11-03 19:59:28	19:59:28
Data written to output.xlsx at row 4 at 2024-11-03 19:59:33	19:59:33
Data written to output.xlsx at row 5 at 2024-11-03 19:59:39	19:59:39
Data written to output.xlsx at row 6 at 2024-11-03 19:59:44	19:59:44
Data written to output.xlsx at row 7 at 2024-11-03 19:59:49	19:59:49
Data written to output.xlsx at row 8 at 2024-11-03 19:59:54	19:59:54
Data written to output.xlsx at row 9 at 2024-11-03 19:59:59	19:59:59
Data written to output.xlsx at row 10 at 2024-11-03 20:00:04	20:00:04
Data written to output.xlsx at row 11 at 2024-11-03 20:00:09	20:00:09

4.3. Primitive GUI:

The first test on the GUI was confirmation that the alert remained hidden on page load, ensuring that the default “display: none” setting works as intended. The next test was the send alert button, confirming that the message appears with the correct orientation and position. The automatic 10-second fade-out was also tested and fixed on that same test as

it did not work for the first attempt. The final test was the clear alert button. There were some issues with function calling, however it also passed the test.

5. Feedback Collection

From Client:

- Immediate Alerts are a top priority
- Ability to sort through alerts so as not to overwhelm the manager
- Alert priority should vary based on the importance of the UE. Ie A human is more important than a machine

From other potential users:

- "The location detection needs to be very accurate, within a few feet if possible."
- "Immediate alerts are essential. If someone crosses into a restricted area, I want to know right away"
- "A log of all alerts and movements within zones would help us analyze patterns and address potential risks."

6. BOM update:

Component	Quantity	Unit	Unit Cost	Total Cost	Link
Visual Studio Code	4	EA	0.00	0.00	https://code.visualstudio.com/
Arduino IDE	4	EA	0.00	0.00	https://www.arduino.cc/en/software
Shabodi Workshop	1	EA	0.00	0.00	
Python 3.13	4	EA	0.00	0.00	https://www.python.org/downloads/
Microsoft Excel	1	EA	0.00	0.00	
				Total	
			Added Total	0.00	
			Final	0.00	

7. Prototype 2 Test Plan:

Test Number	1	2	3	4
Test Description	Zone Change Test: a laptop will be made to cross a specified boundary	Zone Proximity Test: a laptop will write to the console its distance to a specified boundary, which will be compared to a tape measure	Alerts Test: a computer will send an email to a specified address	GUI Functionality Test: a user will be asked to get to a certain menu in the GUI
Reason for Prototype	Analysing critical subsystems --- Reducing risk and uncertainty	Analysing critical subsystems --- Verifying feasibility	Analysing critical subsystems --- Verifying feasibility	Analysing critical subsystems --- Communicating and getting feedback on ideas
Evaluation Criteria	Proximity to boundary when change detected	Difference between code and tape measure distances	Proportion of successful emails --- Delay between send and receive	Time needed to get from the main menu to the specified location
Level of Prototype	Middle fidelity --- Focused	Low fidelity --- Focused	Middle fidelity --- Focused	Middle fidelity --- Focused
Kind of Prototype	Physical	Physical	Analytical	Physical
Metrics	Distance (cm)	ΔDistance (cm)	Proportion (%) --- Delay (s)	Time (s)
Analysis Method	The laptop will write to console if it detects that it is in a certain zone	The distance between the laptop and boundary will be compared to that measured by a tape measure	The specified email inbox will be checked to determine whether the correct message was sent	Someone unfamiliar with the software will be asked to get to a specified screen in the GUI

		between the same points		
Stopping Criterion	10 total tests accomplished	10 evenly spaced intervals between 0-10m are tested	5 different emails sent to 5 addresses each	3 users are timed getting to 3 screens each

8. Conclusion

During this stage of the prototyping cycle, we realized that our initial prototyping test plan was very ambitious, and relied heavily on the functionality of Shabodi's APIs, which we have not yet been able to incorporate. As such, we pivoted to accomplish much more realistic goals during this first prototyping stage, and we plan to iterate towards our initial goals much more slowly. Moreover, we realized that using an Arduino overcomplicates our project, which, in conjunction with Shabodi's recommendation to keep our project simple, has led us to remove it from our BOM. Going forwards, we will mainly focus on the alerts and GUI subsystems, as the location tracking subsystem is practically done, save for incorporating Shabodi's API, and the zone definition subsystem relies on the GUI to be reasonably developed before its development can begin.