AUTONOMOUS ROBOTIC VEHICLE (ARV) / WEARABLE APP PROJECT

Team Android Optimizers

Jose Franco Baquera

Atiya Kailany

Jared Peterson

CS 448

Statement of Work

Introduction / Background

The Department of Computer Science at New Mexico State University (NMSU) has many research projects that are currently underway. One of these research projects focuses on designing and developing a wearable interface to control a team of virtual drones. The research project also aims to test different configurations of user interfaces for wearable devices. In order to accomplish both goals, a wearable interface was designed to implement a mixed reality game in which users are required to wear the device (see Figures 1 and 5).

Figure 1. Using wearable devices to implement a mixed reality game.

The purpose of the game is simple: Players will need to find hidden objects by controlling a team of virtual drones while making sure to avoid any danger zones discovered along the way. It is important to note that danger zones and hidden objects can be either human-based or drone-based (e.g. there can exist human danger zones, as well as hidden drone objects). In order to manipulate the team of drones, the player will have to interact with a map user interface being displayed on the wearable device (see Figure 2).

Ultimately, the goal for this senior project is to aid researchers within NMSU's Department of Computer Science by designing and implementing an Android application capable of displaying an alternative map user interface using Google Maps. More specifically, the new map user interface should represent the drones and other game components in a more elegant, efficient, and simple manner. The Android application will also connect to an NMSU server named *Spitfire* in order to continuously get information about the team of drones, hidden objects, and danger zones (e.g. locations within the map, drones' simulated battery life, etc.).

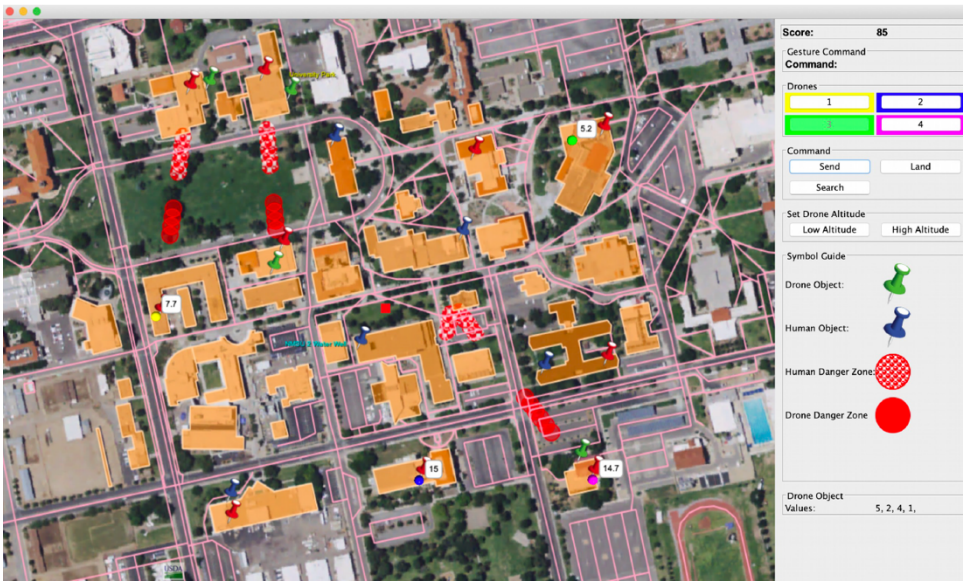

Figure 2. Current map user interface being used to control the virtual drones.

The successful completion of the project would have extensive and important real-world applications. For example, it would provide a foundation for future projects that aim to use wearable devices to control a *physical* team of drones in order to aid humans during both man-made and natural disasters, as well as other emergencies such as a search and rescue. Such aid to humans would not only help save lives, but also time and resources (e.g. a team of drones could be used by park rangers to conduct a search and rescue mission for missing campers or hikers). It is worth noting that such technologies are yet to be developed, meaning that researching, prototyping, and testing different configurations of user interfaces on wearable devices is still needed until more sophisticated drones emerge on the market.

*(*Note: To see a real-world example of how drones can be used to aid humans in emergencies, please click on the following link: *https://www.youtube.com/watch?v=OIUSVDUq0mo)*

Scope of Work to Be Performed

For this project, we will be developing a more usable, simple, and effective map user interface. It is important to note that such interface will be implemented by an Android application.

- Map User Interface Design:

  Quick Overview – The first objective of the project is to design a new map interface using Google Maps' API (i.e. the current interface uses NASA's WorldWind, which is outdated). The developed user interface will be the main interface for the game and will be displayed at all times while players operate the team of virtual drones. The map interface should, ultimately, enhance a player's experience and display relevant information about the game onto the screen.

User Experience – The new map interface will be designed with various qualitative traits in mind. That is, the map interface should:

- Be easy to use
- Not have significant time lag
- Be implemented by well-established user interface and game design practices
- Display the current status of the game in an accurate manner
- Allow users to quickly respond to changing events
- Be simple yet effective
- Display all relevant information without looking too cluttered
- Allow users to find specific commands quickly
- Have well-recognizable icons

Information to Display – The new map interface will display several important types of information about the game, such as the following:

- Aerial view of NMSU's campus
- Drones' locations within the map
- Score
- Drone objects
- Human objects
- Human danger zones
- Drone danger zones
- Drone commands such as *send*, *land*, *search*, *low altitude*, and *high altitude*

Drone Search Modification – A new functionality that modifies how drones search for hidden objects will be implemented. This new functionally will allow users to pick two coordinates on the map and will send the selected drone to search an "imaginary" square area (see Figure 3).

Figure 3. Screenshot demonstrating how the "imaginary" square area will be selected by using two user-chosen coordinate points. In this example, the user would have selected the purple and orange coordinates. The selected drone would then search on this new square area by following the horizontal green lines.
Source: Andrews, A. [Aeroworks Productions]. (2016, February 27). *DJI Phantom 3 Pro | Search and Rescue | Drone Deploy | S.W.A.R.M.* [Video File]. Retrieved from https://www.youtube.com/watch?v=0l4MLWJxxy0

User Stories – During the final phases of the project, user stories that highlight user experiences with the new map interface will be collected, reviewed, and used to make improvements.

- Implementation of Map Interface Using an Android Application:

  Quick Overview – The second objective of the project is to create an Android application that can display the newly designed map interface. The Android application should, ultimately, allow players to control the team of virtual drones using the touch screen capabilities of an Android tablet.

  Map Interface – The application will successfully display the new map user interface and will adhere to all the required qualitative traits/display information that was discussed in the previous bullet point (e.g. danger zones, hidden objects, etc.).

  Server / API Connections – The application will successfully connect to NMSU's *Spitfire* server, which is responsible for providing important game information such as drones' height and danger zones. Furthermore, the application will use Google Maps' API to display an aerial view of NMSU's campus.

  Drone Commands – The Android application will use user input to manipulate the following drone actions:

  - Send
  - Land
  - Search
  - Low altitude
  - High altitude

  It is worth noting that the application will have to connect to NMSU's *Spitfire* server in order to issue drone commands.

  Hardware – The Android application will be developed for tablets only. That is, the application will be tested with Android tablets and will not be implemented with smartphone usage in mind.

- Future Consideration:

  If time permits, we plan to create a unified interface that combines the map and data interfaces together (see Figure 4). This would allow players to stay more focused on one interface instead of getting distracted by having to open different tabs to check something as simple as battery level or altitude height in meters.

Figure 4. Data interface that is currently being used. If time permits, we plan to create a unified interface that combines both the map and data interfaces together.

- Functionality Outside the Scope:

  Functionalities that lie outside the scope of work include the following:

  - Developing more/multiple user interfaces
  - Developing an additional smartphone application
  - Implementing the algorithm that is needed to make the "imaginary" square area (i.e. *Spitfire* will be responsible for this)
  - Reconfiguring/Reprogramming NMSU's *Spitfire* or any other server that is responsible for providing important game information (see Figure 5)
  - Implementing more than four drones into the map interface
  - Implementing software for other wearable devices

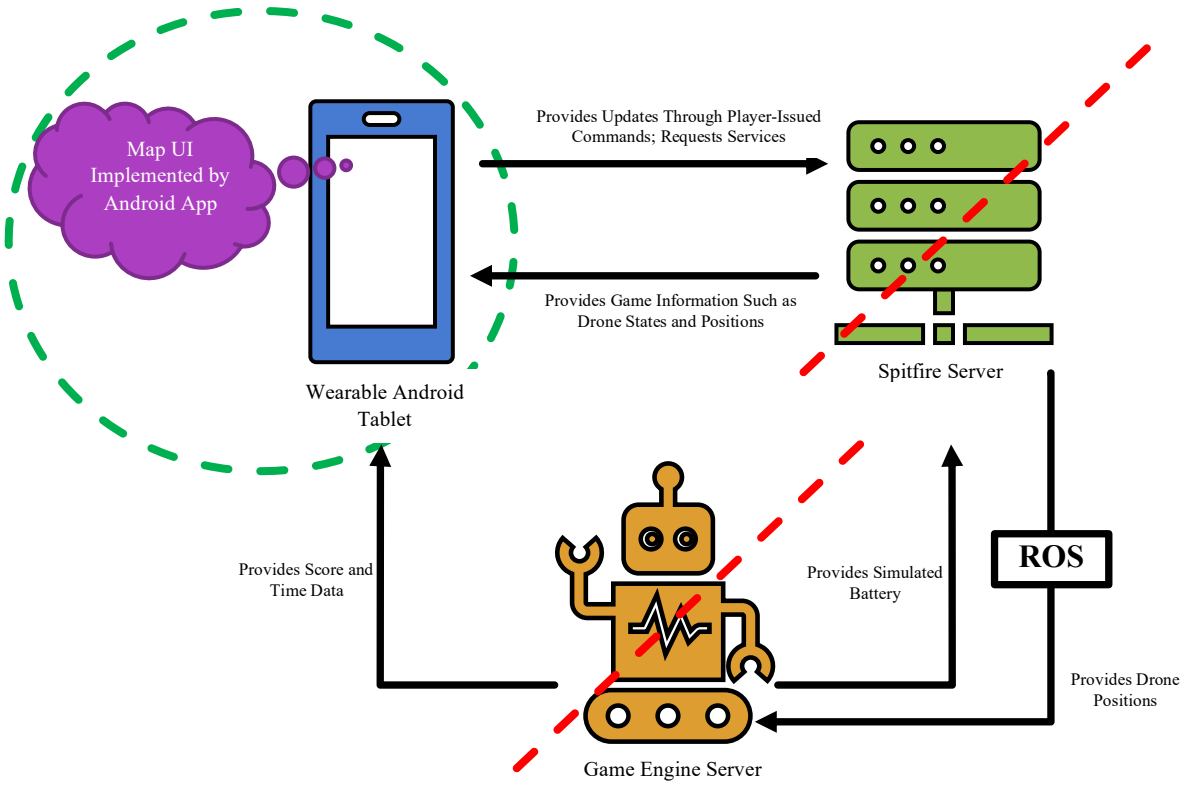

Figure 5. Diagram illustrating the client-server architecture between the Android application and NMSU's *Spitfire*/game engine servers. We note that reconfiguring or reprogramming the server and game engine is outside the scope of work (represented by a red dashed line). In contrast, everything related to the Android application is inside the scope of work (represented by a green dashed circle).

Location of the Work / Meetings

All meetings with project sponsors shall be conducted inside NMSU's Science Hall building.

Period of Performance

August 21, 2019 – December 6, 2019

Work Requirements / Tasks

The project will require the completion of all of the following essential lists of tasks, milestones, and phases:

Phase I – (Sprint 0 / Lightweight Prototype Presentation)

Research, Brainstorming, Android Application Foundation / Google Maps Implementation

1.1 Research

- 1.1.1 **Task:** Research how to use Android Studio and Google Maps' API
- 1.1.2 **Task:** Research potential drone user interfaces that are already implemented

1.1.3 **Task:** Research current user interface game design principles that are applicable to the project

1.2 Brainstorm Map Interfaces

1.2.1 **Task:** Hand-draw sketches of what the map interface can potentially look like

1.3 Android Application Foundation / Implement Google Maps

1.3.1 **Task:** Remove the old NASA WorldWind map on the new Android application while making sure that all pre-existing functions that communicate with *Spitfire* are not affected

1.3.2 **Task:** Implement and test the new map using Google Maps' API

Phase II – (Sprint 1 / Functional Prototype Demo 1)

Create New Game Icons, Redesign Commands Interface Section, Send Drones Implementation

2.1 Create New Game Icons

2.1.1 **Task:** Create new icons that represent the hidden objects, danger zones, and drones

2.2 Redesign the Commands Interface Section

2.2.1 **Task:** Redesign the drone commands interface section (i.e. the *send*, *land*, and *search* commands) by using the hand-drawn sketches and research done in phase I

2.3 Send Drones to a Location

2.3.1 **Task:** Allow users to send a drone to a specific location within the map

2.3.2 **Task:** Test that the end locations of all four drones are correct by comparing both the old and new interfaces concurrently

Phase III – (Sprint 2 / Functional Prototype Demo 2)

Drone Search Implementation, Drone / Human Hidden Objects Implementation

3.1 Drone Search Implementation

3.1.1 **Task:** Allow users to search a square area on the map by using two coordinate points

3.1.2 **Task:** Test that the drone moves in a linear pattern throughout the selected square area

3.2 Drone / Human Hidden Objects Implementation

3.2.1 **Task:** Correctly display both human and drone hidden objects once the drone finds them. The location of such hidden objects is provided by NMSU's *Spitfire* server

3.2.2 **Task:** Test that the end locations of hidden objects are correct by comparing both the old and new interfaces concurrently

Phase IV – (Sprint 3 / Functional Prototype Demo 3)

Drone Altitude Implementation, Drone Land Implementation, Danger Zones Implementation, Street and Building Overlay

4.1 Drone Altitude Implementation
- 4.1.1 **Task:** Allow users to set the altitude of drones (i.e. either *high altitude* or *low* altitude)
- 4.1.2 **Task:** Test that the altitudes of all four drones are correct by comparing both the old and new interfaces concurrently

4.2 Drone Land Implementation
- 4.2.1 **Task:** Allow users to land drones on specific locations within the map
- 4.2.2 **Task:** Test that the end locations of all four landed drones are correct by comparing both the old and new interfaces concurrently

4.3 Danger Zones Implementation
- 4.3.1 **Task:** Display both human and drone danger zones that are discovered along the way
- 4.3.2 **Task:** Test that the danger zones discovered along the way are correct by comparing both the old and new interfaces concurrently

4.4 Street and Building Overlay
- 4.4.1 **Task:** Display the street and building overlay on top of the map interface

Phase VI – (Sprint 4 / Final Functional Prototype Demo 4)

Relevant Game Information, User Stories, Optimizing the User Interface, Testing

5.1 Other Relevant Game Information Implementation
- 5.1.1 **Task:** Display all other relevant information such as user scores, drone object values, and building heights
- 5.1.2 **Task:** Test that all other relevant information is correct by comparing both the old and new interfaces concurrently

5.2 User Stories
- 5.2.1 **Task:** Allow other students to play the game and gather user stories

5.3 Optimizing the User Interface
- 5.3.1 **Task:** Review and use the collected user stories that highlight user experiences with the new map interface to make improvements.

5.4 Testing
- 5.4.1 **Task:** Conduct bottom-up integration testing to assess the accuracy and precision of the product

Optional Tasks

(Note: The following list of tasks/milestones will be completed only if time permits.)

- Modify the new user interface so that it also displays other information such as mission status, battery life, and drones' altitude in meters (see figure 4)
- Warn the player that a drone is too low to land into a particular building whose height exceeds the drone's altitude

Project Schedule / Schedule of Milestones / Deliverables

Start and End Dates for Scheduled Tasks		
Scheduled Tasks	***Start Date***	***End Date***
1.1.1	September 9	September 11
1.1.2	September 10	September 12
1.1.3	September 11	September 13
1.2.1	September 12	September 15
1.3.1	September 16	September 21
1.3.2	September 22	September 27
2.1.1	September 28	October 2
2.2.1	October 1	October 9
2.3.1	October 5	October 11
2.3.2	October 5	October 11
3.1.1	October 12	October 19
3.1.2	October 12	October 19
3.2.1	October 17	October 25
3.2.2	October 17	October 25
4.1.1	October 26	November 1
4.1.2	October 26	November 1
4.2.1	October 30	November 6
4.2.2	October 30	November 6
4.3.1	November 1	November 8
4.3.2	November 1	November 8
4.4.1	November 5	November 8
5.1.1	November 9	November 13
5.1.2	November 9	November 13
5.2.1	November 14	November 16
5.3.1	November 15	November 19
5.4.1	November 20	November 22

Sprint Legend
Sprint 0
Sprint 1
Sprint 2
Sprint 3
Sprint 4

(Note: The scheduled tasks are described in more detail in the work requirements/tasks section of the report.)

Start and End / Due Dates for Scheduled Deliverables, Milestones and Documentation		
Scheduled Deliverables / Milestones / Documentation	***Start Date***	***End / Due Date***
Meeting with Project Sponsor	-	*August 30*
Statement of Work Draft	August 21	September 6
Meeting with Project Sponsor	-	*September 6*
Statement of Work	September 7	September 10
Meeting with Project Sponsor	-	*September 13*
Needs and Requirements Draft	September 11	September 16
Meeting with Project Sponsor	-	*September 18*
Meeting with Project Sponsor	-	*September 25*
Requirements Specifications / Lightweight Prototype Presentation	September 17	September 27
Meeting with Project Sponsor	-	*October 2*
Meeting with Project Sponsor	-	*October 9*
Requirements Update and Design Specification Draft 1	September 28	October 10
Functional Prototype Demo 1	September 28	October 11
Meeting with Project Sponsor	-	*October 16*
Meeting with Project Sponsor	-	*October 23*
Functional Prototype Demo 2	October 12	October 25
Meeting with Project Sponsor	-	*October 30*
Meeting with Project Sponsor	-	*November 6*
Requirements and Design Specification Updates	October 26	November 7
Functional Prototype Demo 3	October 26	November 8
Meeting with Project Sponsor	-	*November 13*
Meeting with Project Sponsor	-	*November 20*
Final Functional Prototype Demo 4	November 9	November 22
Meeting with Project Sponsor	-	*November 27*
Video Screening / Uploaded Video	November 15	November 22
Meeting with Project Sponsor	-	*December 4*
Final Presentations / Final Report, Code, and Documentation	November 22	December 6

Phase 1 Gantt Chart

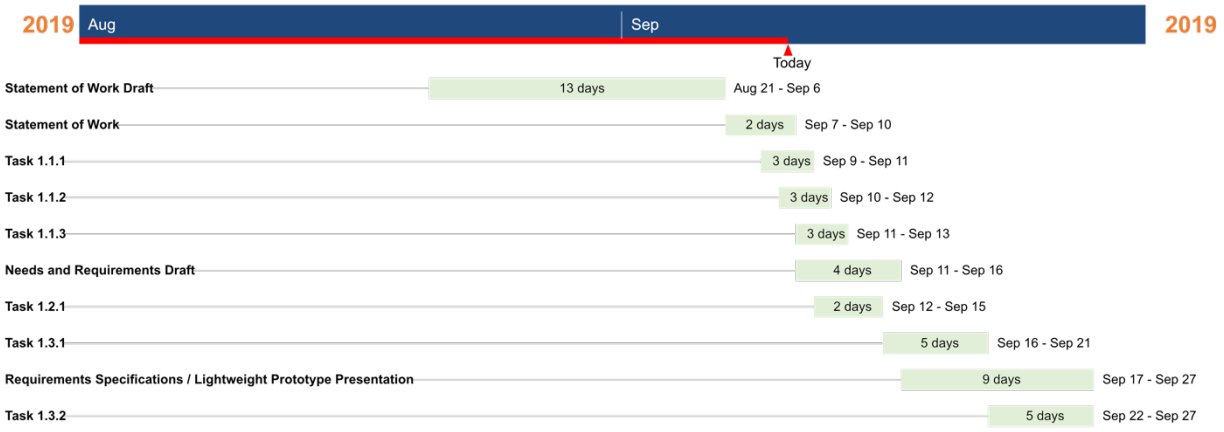

Figure 6. Gantt chart listing the start and end dates for scheduled tasks, deliverables, and milestones for phase 1.

Phase 2 Gantt Chart

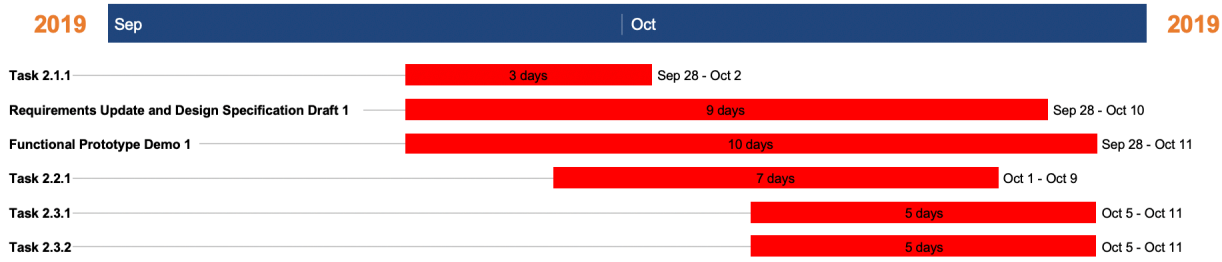

Figure 7. Gantt chart listing the start and end dates for scheduled tasks, deliverables, and milestones for phase 2.

Phase 3 Gantt Chart

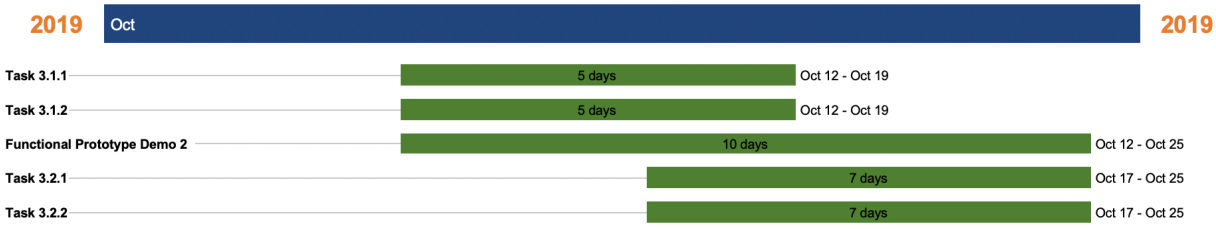

Figure 8. Gantt chart listing the start and end dates for scheduled tasks, deliverables, and milestones for phase 3.

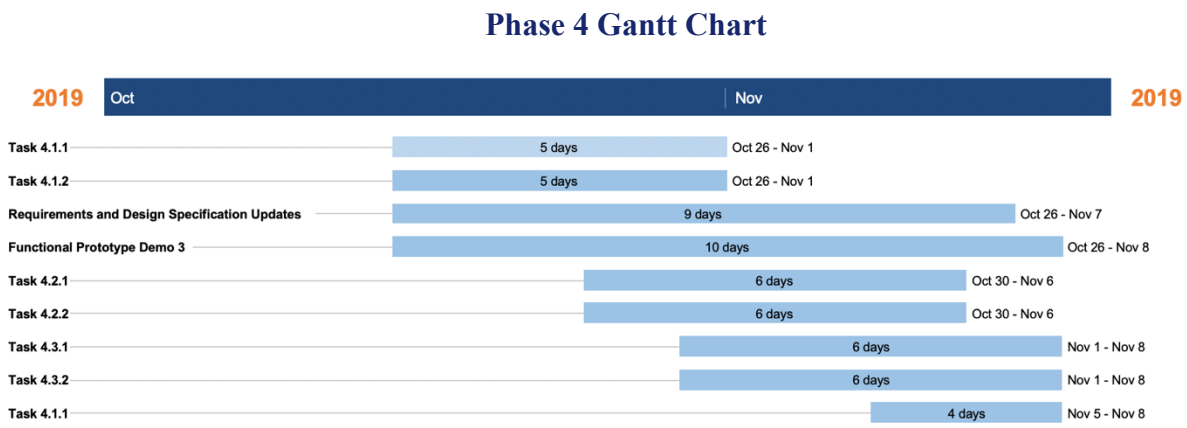

Figure 9. Gantt chart listing the start and end dates for scheduled tasks, deliverables, and milestones for phase 4.

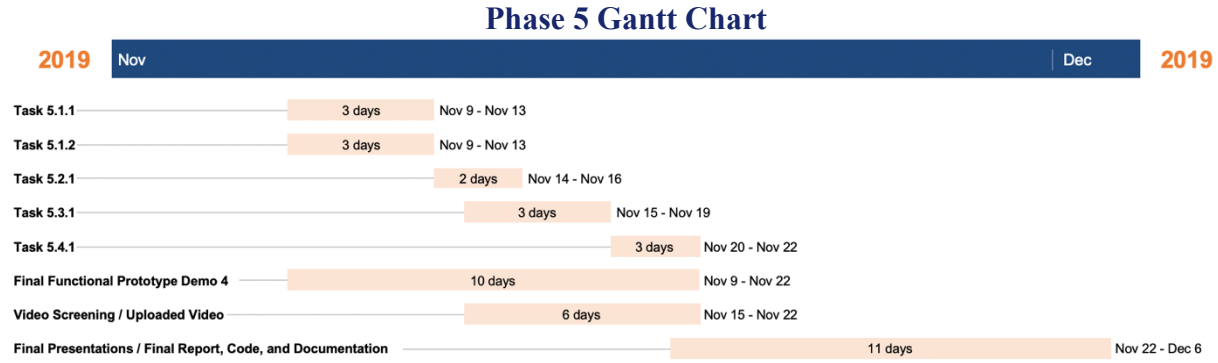

Figure 10. Gantt chart listing the start and end dates for scheduled tasks, deliverables, and milestones for phase 5.

Special Requirements

Special requirements that will be needed to successfully complete this project include the following:

Special Skills / Knowledge

- Extensive knowledge of user interface and game design
- Basic understanding of client-server and publish-subscribe computer architectures
- Understanding and evaluating code written by others
- Knowledge on how to program in Java

Software

- Android Studio for developing the Android application
- How to implement Google Maps' API with Android Studio
- Using ROSBridge as a tool for server communication

Hardware

- Access to NMSU's *Spitfire* server

Acceptance Criteria

An acceptable product will be an Android application that connects to NMSU's *Spitfire* server, is able to send drone commands/updates to the server, and displays the drone movements, hidden objects, danger zones, and other relevant information (see Scope of Work section) on a Google Map with a new and intuitive user interface. Furthermore, an acceptable application will prove to be useable by not creating a lot overhead, being implemented with modular software that can be easily modified and maintained, and having detailed documentation on how the product works, either by code comments or technical documents. Both parties will know when the final product is acceptable once we present and demo our final project to Ahmed Khalaf *and* Dr. Toups, and it is agreed upon that our project fulfills their initial requests. However, either Ahmed Khalaf or Dr. Toups is authorized to accept the deliverables throughout the course of the project (i.e. only one individual is needed to sign off deliverables). In addition, both parties will know when the work is acceptable when we demo functional prototypes throughout the semester. The product will be accepted through our group giving them full access to the project that was developed so that they can incorporate it into their research.

Reporting and Communication

Project sponsors and developers shall meet at least once a week, or twice every half a month.

Change to Statement of Work Agreement

Any minor changes in the statement of work by either party shall be submitted for approval at least one week in advance. Examples of minor changes include the following:

- Meeting dates
- Order in which tasks should be completed in a given phase
- Color schemes within the user interface

It is important to note that no major changes to the statement of work will be allowed after Ahmed Khalaf or Dr. Toups sign it. Examples of major changes that are not allowed include the following:

- Changing from Google Maps' API to another map API
- Changing the implementation to an OS application instead of an Android one
- Creating a smartphone application instead of a tablet application
- Changing the order in which the phases are completed

Intellectual Property Agreement

Any software that is developed, including its documentation, presentations, and other works alike, can be used however the project sponsors wish, but they shall give proper credit to all three creators mentioned on the title page.