Royce Mou
CS 352
Data Analysis and Prototyping Plan
Robot Visualization Algorithm

## 1. Honor Code

I pledge on my honor that I have neither given nor received unauthorized aid on this assignment.

Note: Large portions of this initial description are copied and adapted from NASA Grant #NNX13AL65H; this project is a portion of and described extensively in the grant.

## 2. Data Collection and Analysis Techniques

To collect more detailed information regarding the requirements, the relevant stakeholders were interviewed with open-ended questions. Examples of such questions include "How is the team of a first responder structured?" and "What inputs does the RVA algorithm use to gauge visual state?" The interview yielded raw data in the form of the interviewer's notes, which yielded qualitative data in the form of the interviewee's responses to open-ended questions. The data was initially processed by reviewing the interviewer's notes and expanding bulleted points.

To analyze the qualitative data, we categorized the responses from the interviewees into two distinct types, informational data and functional data. Feedback and suggestions that did not directly impact the interface, but might give a broader picture of the intended use case as a whole, would be classified as informational data. Feedback and suggestions that would directly impact the interface would be classified as functional data. For example, data about the team structures of the first responders would be classified as informational and data about the specific inputs to the RVA algorithm, such as neglect time, would be classified as functional.

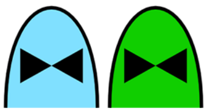
## 3. Prototyping Plan

Throughout this development process, "implementation" and "look and feel" prototypes will be developed. Low fidelity prototypes will first be presented in the form of computer-drawn renderings of images and scenarios, and the feedback will be used to guide the development of high fidelity prototypes. The low fidelity prototypes will present distinct visual states for the unmanned vehicles and scenarios in which the visual states of the unmanned vehicles will change. The low fidelity prototypes will be used to inform the flow of visual state transition for unmanned vehicles in the high fidelity prototype.

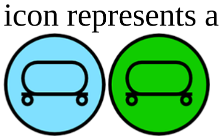
The high fidelity prototypes will demonstrate how the RVA algorithm changes the visual states of robots between the "residue", "normal", and "detailed" states, as well as how it visually differentiates between the friendly (on one team) unmanned vehicles being controlled by the unmanned vehicle supervisor (on the same team) and the unmanned vehicles being controlled by unmanned vehicle supervisors from other teams. The high fidelity prototypes will be presented to users and stakeholders to gauge the "implementation" and "look and feel" of the prototypes. Users and stakeholders will especially be asked for feedback on how the visual states transition between one another. For example, unmanned vehicles should transition to detailed view when the neglect time of the unmanned vehicle ends. If the detailed view of the robot is too large, then it might become overwhelming if several robots reach the neglect time threshold at once. Prototypes are necessary for users and stakeholders to give feedback on the comfort level of visual state transitions in the context of a chemical, biological, radiological, nuclear, or explosive (CBRNE) event. This feedback will then be used to evolve the visual state transitions in higher fidelity prototypes or the product so that the visual state transition of the unmanned vehicles will be fully intuitive and comfortable.

The scenarios and low fidelity prototypes are described and shown.

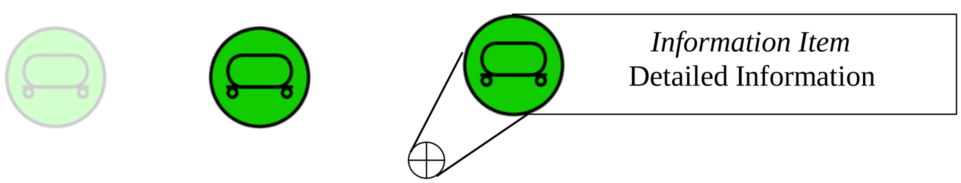
The icons for unmanned aerial vehicles (UAVs) in "normal" state are shown below. A green icon represents a UAV that is controlled by the friendly unmanned vehicle specialist/supervisor (UVS), i.e. a friendly UAV, and a blue icon represents a UAV that is controlled by another team's UVS.



The icons for unmanned ground vehicles (UGVs) in "normal" state are shown below. A green icon represents a UGV that is controlled by the friendly UVS, i.e. a friendly UGV, and a blue icon represents a UGV that is controlled by another team's UVS.

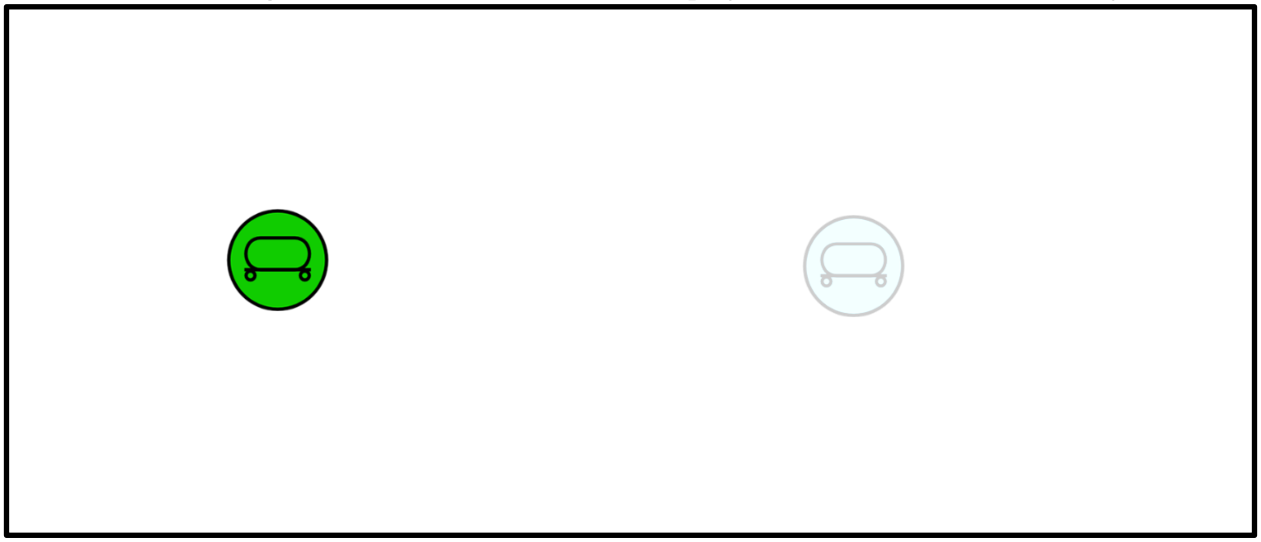


The "residue", "normal", and "detailed" visual states are shown for a friendly UGV below.

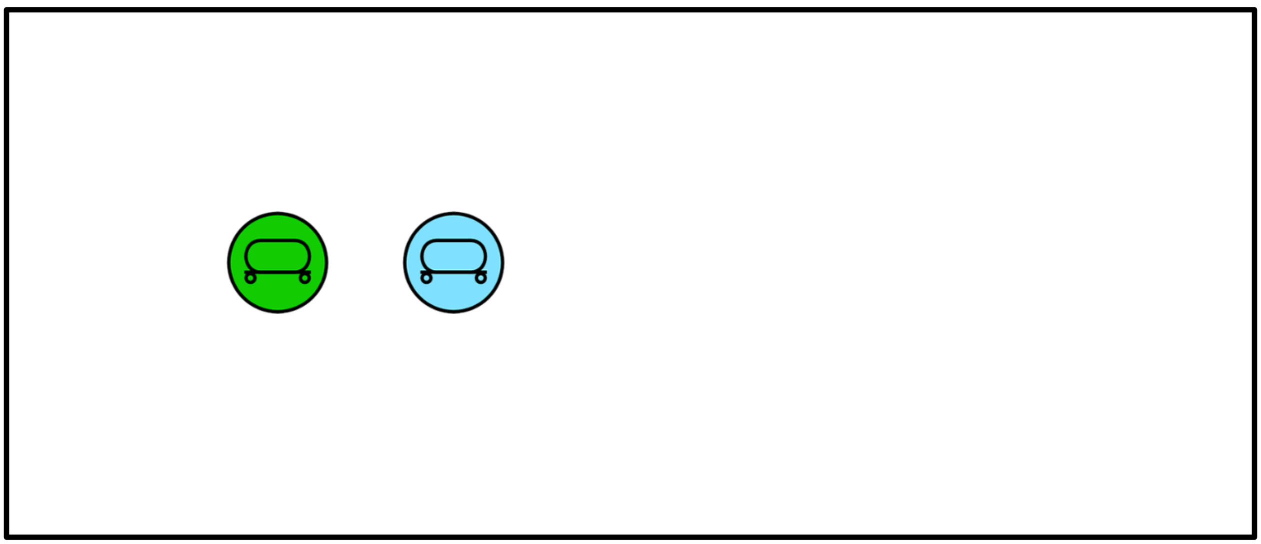



We construct an initial scenario to demonstrate the visual state differences. The background will eventually be a map of the incident area, but for demonstration purposes, the background will be omitted.

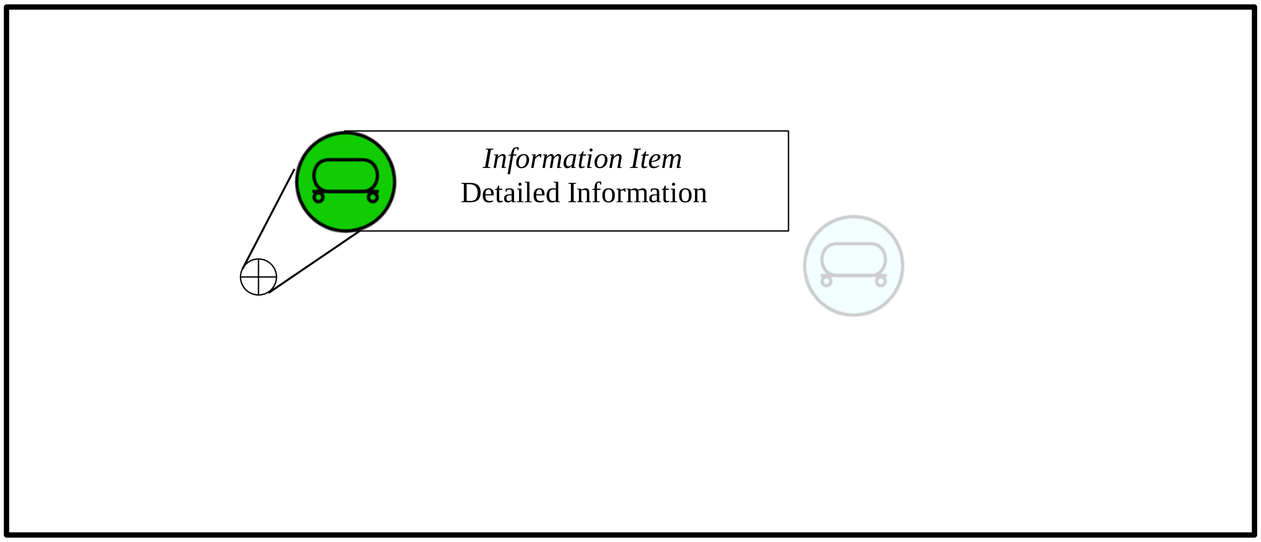
In the base state, there are no incidents. We will call the friendly UGV Green and the other team's UGV Blue. Only Green is supervised by the friendly UVS. Blue will be supervised by a UVS on another team. Green will be in "normal" visual state when awaiting a task or performing a task and within neglect time limits. Blue will be displayed in "residue" visual state by default.



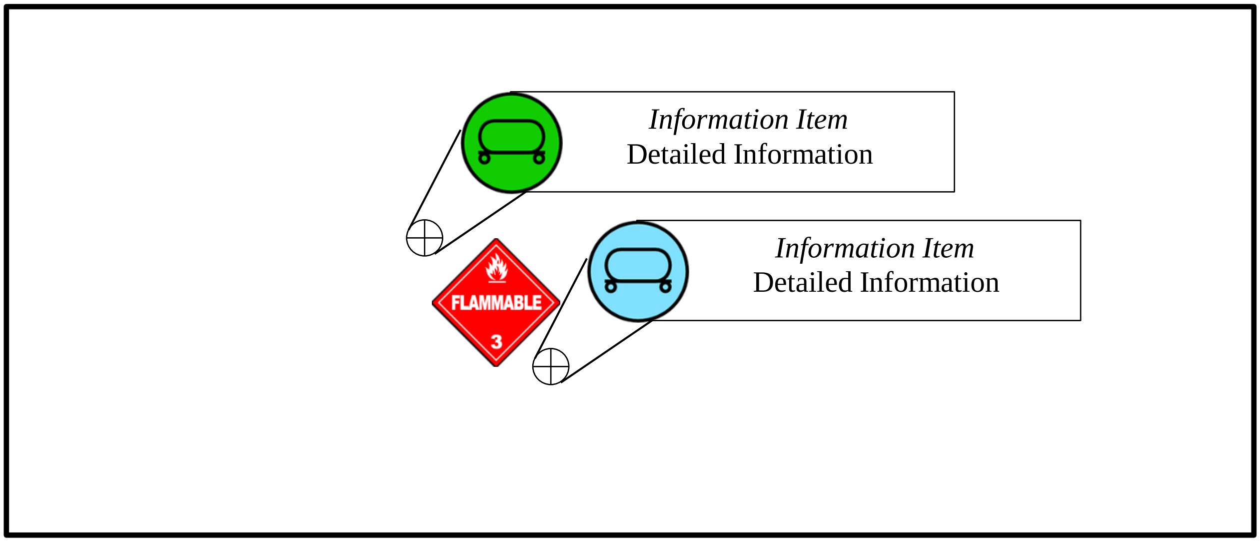
When Blue moves closer to Green, its visual score increases. When Blue's visual score passes a threshold, it will be displayed in "normal" visual state.



When a user moves their pointer over Green, Green is displayed in the “detailed” visual state. Green is displayed in a “detailed” visual state when a user moves their pointer over the UGV, when neglect time ends, or when an incident occurs in a surrounding area. Blue will be displayed in a “detailed” visual state when a user movers their pointer over the UGV.




A hazardous incident has occurred. Green is in close proximity to the incident. Green’s visual state will be elevated to the “detailed” state. Even though Blue is on another team and is controlled by another team’s UVS, it is in close proximity to an incident and to Green. Its visual state will also be elevated to the “detailed” state.




Visual state transitions are difficult to depict on static images, so visual state transitions will be tested in higher fidelity prototypes. The low fidelity prototype demonstrates the differences between unmanned vehicles controlled by the friendly UVS and other unmanned vehicles controlled by UVSs on other teams. The low fidelity prototype also demonstrates the different visual states and how they change in different scenarios.

## 4. Schedule

| | |
|---|---|
| Sept 24 | - Due: Requirements documents<br>- Design RVA integration: Design how the RVA will be integrated with the existing Generalized Visualization and Abstraction (GVA) framework. Appearance (e.g. icons, etc.) and implementation (e.g. relevance rankings, simulation, etc.) will be considered for defined metrics (neglect time, cognitive workload, time to completion, etc.) |
| Oct 1 | - Due: Data Analysis Results; Prototyping Plan<br>- Prototype design will be drafted and refined. Relevant metrics will be evaluated in greater detail. |
| Oct 13 | - Due: Prototype Demonstrations<br>- Complete RVA evaluation plan: An evaluation plan specifying simulation and testing will be drafted. Implementation will be targeted at defined metrics.<br>- Complete implementation of RVA: Prototype will be finished and ready to be demonstrated. Full integration with the GVA framework is expected. |
| Oct 22 | - Due: User Test plan |
| Oct 27 | - User Testing begins |
| Nov 17 | - User Testing Ends<br>- Complete user evaluation of RVA: User evaluations will be aggregated and accounted for. Changes to the RVA will be made as necessary and the existing implementation may be refined. |
| Nov 30 | - Due: Project and Supporting Documentation |