**Airport Ground Vehicle Accident Management**

Patrick Mawhinney, Matthew Brooks, Christian Heimlich, Kevin Murphy, Nicholas Parisi

*Command and Control under Dr. Tod Schuck*

*Rowan University. Glassboro, NJ 08028*

***Abstract*  - The amount of accidents within ground movement in an airport causes a large number of injuries which calls for a more robust system for the movement of each vehicle. This research looks at the current command and control structure for ground movement management and outlines a solution which involves a flexible simulation for automated vehicle movement within an airport.**

***Keywords - Ground vehicle, Simulation, Airport, Command and Control, Safety***

1. INTRODUCTION

At an airport, there are safety concerns in many aspects of the operation. To most passengers, safety is viewed in the context of departure and landing. This is not always the case, as ground movement on vehicles within the airport is a large source of accidents. According to the International Air Transport Association, there are an estimated 27,000 ramp accidents and incidents each year [1]. With an injury rate of 9 people per 1,000 departures, there is cause for concern. Financially, the costs for damages and incidents reaches billions of dollars per year around the world. Stated by the Flight Safety Foundation Chairman Ed Stimpson, “we realized that the initial estimate of $3 to $4 billion lost per year due to ground accidents was low, as it did not include the indirect cost caused by injuries and death. We estimate the total number may be closer to $7 billion” [2].

In response to the large amount of incidents, the IATA launched the Ground Accident Prevention Program in 2003. This program develops products and methods in order to reduce the incidents during ground movement in airports. For example, safety procedures have been developed in the ACRP Synthesis 29 [3]. However, this practice is only implemented for ramp (aprons) safety practices. The procedure neglects the many areas of ground movement, including the Runway Safety Area (RSA), the taxiways, runways, etc.

Consideration needs to be put into these areas working together as a complex and fluid system of systems, while maintaining each system to the standards necessary for desired safety. There are many moving parts to the system, including the Aircraft Rescue and Fire Fighting (ARFF), and Foreign Object Debris (FOD) removal [4]. Environmental factors such as nighttime or bad weather are also variables for concern. Research may also be put into non-towered airports (as the pilots are not required to communicate their position). The slope or angle of a runway may also cause blind-spots for some vehicles, therefore requiring vehicles to travel in certain sections to avoid risk of accidents.

Standard patterns are not always maintained for ground movement, which calls for a more robust command and control system over every movement within an airport. This shall reduce the number of accidents and incidents at airports, saving money, reducing the number of injuries to the persons present, and making the overall operation more adaptive and efficient. Discussed within this report are the domains of interaction and the functions of command and control for the system. After, the stakeholders and cultural dimensions within the system are observed. A simulation of the system has been created and is reviewed. Located at the end of the report are future considerations for the system and conclusions of the project.

1. COMMAND AND CONTROL DOMAIN INTERACTIONS

*Physical*

The Physical Domain is where the testing of the proposed system and where the drivers of the ground vehicles safely maneuvering around the ground of an airport occurs. The proposed system is a user interface that simulates various ground vehicles moving around a designed sample airport grid. The system must be tested to ensure the controlled vehicle moves into successfully open spaces on the grid and does not move into an occupied space. The movement into an occupied space would represent a crash. The final test in the physical domain would be to see if the drivers of the ground vehicles successfully executed the shortest, safest path through the airport.

*Information*

The Information Domain is the most critical command and control domain because this is where the source of most airport ground vehicle accidents occur. Most accidents are caused because of a lack communication between the ground vehicles. Since there are no clearly defined pathways specific for each ground vehicle or any standard patterns, the vehicles need to be in constant communication in order to prevent accidents.

The proposed system seeks to amend those previously stated issues by creating a constant communication link between the user interface team and the grounded airport vehicle drivers. The user interface maps out the quickest path for one vehicles by processing the environment in terms of where the vehicle can and cannot proceed. Open space on the road is what qualifies for where the vehicle the proceed. Grass, other ground vehicles, or a solid object is what qualifies as where the vehicle cannot proceed. For this system to be successfully implemented, the quickest path needs to be relayed to the driver in order to prevent collisions. Having the information about the other paths of every other possible vehicle will drastically reduce the large number of ground vehicles accidents at airports. The determining roles, responsibilities, and relationships between the team using the proposed system and drivers will be further explored in section three of the paper.

*Cognitive*

The Cognitive Domain is a key aspect to the ground vehicles ability to avoid accidents. Utilizing any knowledge to avoid crashes, or being able to make quick decisions based on the situation at hand can be the determining factor for avoiding a disaster. This domain is heavily influenced by the Information Domain because all quick decisions made will be a result of the communication of vehicle's path and the surrounding environment. In addition, the ability of the driver to react accordingly will also determine the likelihood of a crash. As a result, decision making comes down to the driver’s ability to process the given information at that moment of time, and to make an appropriate decision. Training and experience will be a large factor for how accurately a driver can make a decision.

*Social*

The key elements of the Social Domain will be important to determining whether or not the ground vehicle will be able to follow the safest, quickest path or crash into another vehicle, person, etc. All of the information being communicated in the Information Domain and rapid decisions being made in the Cognitive Domain must reach all of its intended destinations and be clearly expressed so the drivers can utilize the information on which path to choose. If the information on which path to take or which paths to avoid due to the possibility of an accident is not clearly expressed, then a collision is just as likely. Poorly communicated information is just as useful as information not being shared.

The reach and quality of the interactions in the Social Domain is important for not only testing the proposed system but also for future iterations of this system. If this system is to be successful and become a model to be used or be a future design base, the system must be fully comprehensible and allow for a rich set of interactions. One potential iteration of this design could include the implementation of voice instructions. The control groups need to direct the drivers on what path to follow or avoid in the clearest way possible. Audio communications over an unreliable system often leads to failure in both the Social and the Information Domain because of problems in the patterns of interaction.

1. COMMAND AND CONTROL FUNCTIONS

*Establishing intent*

The overall objective of this project is to reduce the number of accidents caused by ground vehicles at an airport by implementing the proposed system. The proposed system will allow for every airport ground vehicle to be monitored in real time. This will allow the team that is monitoring carts to be able to foresee any possible collisions and communicate to the driver to alter his course of direction.

*Determining roles, responsibilities, and relationships*

For all of those involved in ground movement within the airport, there needs to be flexible collaboration. Each group or individual is to be tasked with a specific purpose, including tasks such as refueling or debris removal. There shall be a line of communication which will strengthen the structure of the organization.

*Establishing rules and constraints*

Within the system, the members need to operate by certain governance. There shall be rules in which everyone operates to maintain the system integrity. The rules will typically stay fixed, however a random variable such as weather may call for a change in operation in order to maintain a certain level of safety.

*Monitoring and assessing the situation and progress*

Awareness of the operations is crucial in determining if the system is working as intended. Metrics such as accident rate will allow for minimization goals. The monitoring will also allow for the realization of system agility to rapidly changing scenarios for ground movement.

*Inspiring, motivating, and engendering trust*

All members involved with the ground movement must be willing to trust the system as a whole. If the participants do not follow the system rules, the system will not operate as intended. All members will need to understand the strength of the system so they may be trusting of where to move in order to not have any accidents. The motivation for the new system will be brought by the potential decrease in injuries and damages.

*Training and Education*

To implement the system, every person involved in ground operations shall be trained to ensure everyone understands the extent of the system. The drivers will have to be trained to trust the system’s route and those that monitor the system will be trained to find errors or issues before they occur. At a higher level, those in charge of updating and fixing the system need to be trained and educated to take in feedback from those on the ground in order to iterate the system. There will always be new requirements for the system or a better way to implement certain function. Therefore, those in charge need to be trained to create robust updates.

*Provisioning*

Within the system for ground movement, resources may need to be allocated in order to maintain the upkeep of the new movement system. In the future, there may be a changes to the number or type of vehicles. This will call for an update to the system which will require resources. Furthermore, the system within each vehicle for tracking may require updates as technology advances. The cost of this can be determined when the tracking system is designed.

1. STAKEHOLDERS AND CULTURAL DIMENSIONS

For an airport (excluding vendors and activities within the airport building), the stakeholders are as follows: government (local/federal), employees, airport suppliers, service providers (aircraft rescue, fire fighting, FOD removal, etc.), passengers, air carriers, airport organization(s), investors, and bondholders. When observing the ground movement in the airport, it becomes clear that the number of incidents is cause for concern for all stakeholders. For example, the injury rate of 9 people per 1,000 departures is concern for the passengers and employees in the airport. Furthermore, the estimate of $4 to $7 billion lost per year due to ground incidents is a large issue for investors and bond-holders. The effort that can be put into ground movement management is beneficial to all stakeholders in the system.

The ground movement is not autonomous. There are many people working together to create a large system with many subsystems within. Due to the fact that many groups are working together, there is a culture around this organization which can be broken up into ten primary characteristics. The following are the characteristics and the point of concern within airport ground vehicle movement and

management:

*Member Identity*

The lack of connection between the groups on the ground is a reason for the number of incidents that occur. The ARFF (Aircraft Rescue and Fire Fighting) does not have expertise in air vehicle runway movement, so accidents may occur when standard paths are not taken.

*Team Emphasis*

Although all particular groups (ARFF, FOD removal, etc.) work in the same area, there needs to be more focus on working together, perhaps managed by a section of the control tower so every employee has full situational awareness.

*Management Focus*

The control towers have most control over ground movement. Although not all decisions are made within the towers, the team creates a hierarchy of decision making for all employees and personnel.

*Unit Integration*

The unit integration of airport ground movement is perhaps one of the stronger areas. The effectiveness of ARFF to move to an area and operate efficiently is key to the safety of many individuals.

*Control*

There are many rules created by the IATA for the Ground Accident Prevention Program, but there are no standards implemented for all units to work together as a whole. The control needs to be fully developed, considering the lack of supervision from control towers.

*Risk Tolerance*

The degree of risk seeking an innovation for this organization needs to be kept to a minimum. Without a closely followed standard, a lot can go wrong. Although innovation is needed to create a unique standard for movement that is effective, there should be no risk taking in how the ground movement operates on a daily basis.

*Reward Criteria*

The employees within the ground movement and control of airports shall be rewarded based on performance. If an employee performs well, follows standards, and works together with other units in a unified manner, then they will be rewarded accordingly.

*Conflict Tolerance*

Due to the lack of knowledge between units and systems, it is recommended that any personnel that has a conflict or criticism convey the issue. As an example, an employee working in the control tower does not know exactly how to operate within the ARFF, and may be underestimating the personnel needed for a specific task.

*Means vs. End Orientation*

To observe the structuring of the airport ground safety management, a view must be placed to the means. The end orientation of the this management does not capitalize on the already existing systems. Focus must be placed on adapting the units to respond in more organized and standardized ways.

*Open-Systems Focus*

The ground movement management is required to strongly respond in a flexible nature to changes within the external environment. A drastic change in weather, such as fog or heavy rains, may prove to be a large issue for management. There can be unexpected fog that reduces visibility of all employees to the point where communications need to adapt to make sure no crashes or incidents occur.

1. SIMULATION

*Summary*

In order to break down the environment of a typical airport ground layout, a ground vehicle simulation was implemented. This was done by creating a GUI in an environment called QT. QT is an IDE that provides libraries of code in C++ for creating complex GUI systems. These libraries were utilized to model an airport ground vehicle simulation. The goal of this simulation is to get a visual representation of the dynamic system to be analyzed in pseudo-real time. Once any observations are made, different procedures for each ground vehicle can be enacted in the case of a crash. In order to simplify the scenario, only of few ground vehicles are modeled with generic purposes for the aircraft. This is done to focus solely on the paths of movement of each vehicle to observe their interactions. Different scenarios of ground vehicles assisting the planes were reenacted to see the effects compared to what was discovered through research. If there are a notable amount of accidents, then certain procedures for the ground vehicles can be enacted. In the end, the program serves as a tool to determine the most beneficial parameters to apply to the ground vehicle systems, whether they be instructions to drivers, self-driving cars with multiple sensors and a data-fusion model, or other scenarios.

*Outline of Typical Scenario*

The plane’s arrival and departure times are hard coded as that is how airports work in reality. Planes are given time slots to take off and time slots to land. When the first plane comes in, it automatically travels down the runway. Once the plane gets to the end of the runway, it checks all terminals to see if they’re open and selects the first available. The plane receives the coordinates for that terminal and starts travelling towards it. The luggage and fuel cars receive instructions to travel to their respective resource blocks to pick up the fuel, luggage, etc. Once they get to their resource blocks, the coordinates update to the location of the terminal that the plan is located. The plane docks at the terminal and the fuel and luggage carts arrive behind the plane to unload the resources. After a set time the fuel and luggage cars are instructed to go back to their parking spaces. Following, the plane is given the coordinates to the start of the runway and then travels up the runway to ‘takeoff’. The difficulty of the simulation becomes apparent when multiple planes and multiple resource cars are trying to travel to their destinations simultaneously.

*Model*

1. Vehicles

The vehicles are represented in the simulation by a car graphic with a given color. Each color can be related to representing the following type of vehicle: Jet refueler, baggage cart, mobile passenger stairs, catering truck, tow tractor, toilet truck, etc.

B. Vehicle Movement Areas

The movement areas can be represented by the airport layout image as seen in Figure 1, but are controlled by a 2-D array of boolean variables that determine if a grid space is an acceptable space to move.

C. System Variables

The most significant variable in the simulation can be found in the interactions between multiple vehicles driving. This calls for rerouting logic to determine the correct path of a given vehicle.

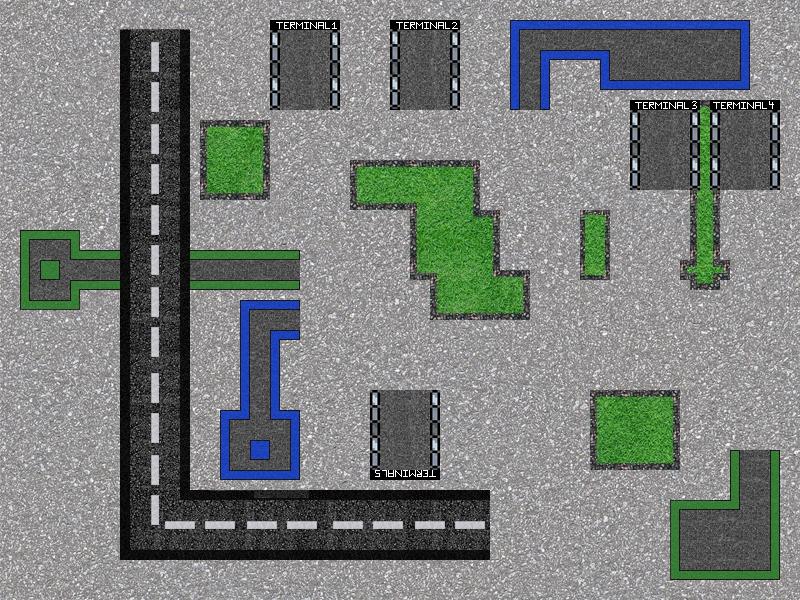


Fig. 1 View of the graphical model

*Pathing Algorithm*

The simulation is based on a pathing algorithm that determines the shortest path a vehicle has to take to get from a start position to an end position. The algorithm is able to identify a position it can not move to by a 2-D grid of booleans. It utilizes two arrays, a main array to contain possible moves and a supplementary array. The algorithm begins by adding the end coordinate to the main array. This main array is looped through until the start position has been reached. Each iteration in the main array will add the four adjacent coordinates of a given coordinate to the supplementary array. This supplementary array is then looped through and is stripped of any coordinates that the vehicle cannot move to or has already been added to the main array. The leftover coordinates in the supplementary array are then added to the main array. This continues until the start position has been reached. The algorithm then chooses the shortest path from the start to the end based on the coordinates added to the main list. When implementing the algorithm in the GUI, further logic was added to recalculate a path if another vehicle was within a certain radius of a given vehicle.

*Graphical User Interface*

The application itself is encapsulated in an interactive UI that allows the user to control the simulation in real time, which enables the user to quickly find potential issues and tweak system parameters to achieve a personalized collision detection, and pathing algorithm that fits the needs of the individual user/company. In particular, the GUI enables one to change the speed of the simulation as well as pause/resume it at anytime, with frame by frame stepping available. Additionally, there is an output log that notes down vehicles successfully reaching their destinations, among other information. The UI is a key component for ensuring a case-by-case custom fit application of the vehicle management system. An example of the GUI can be seen below in Figure 2.

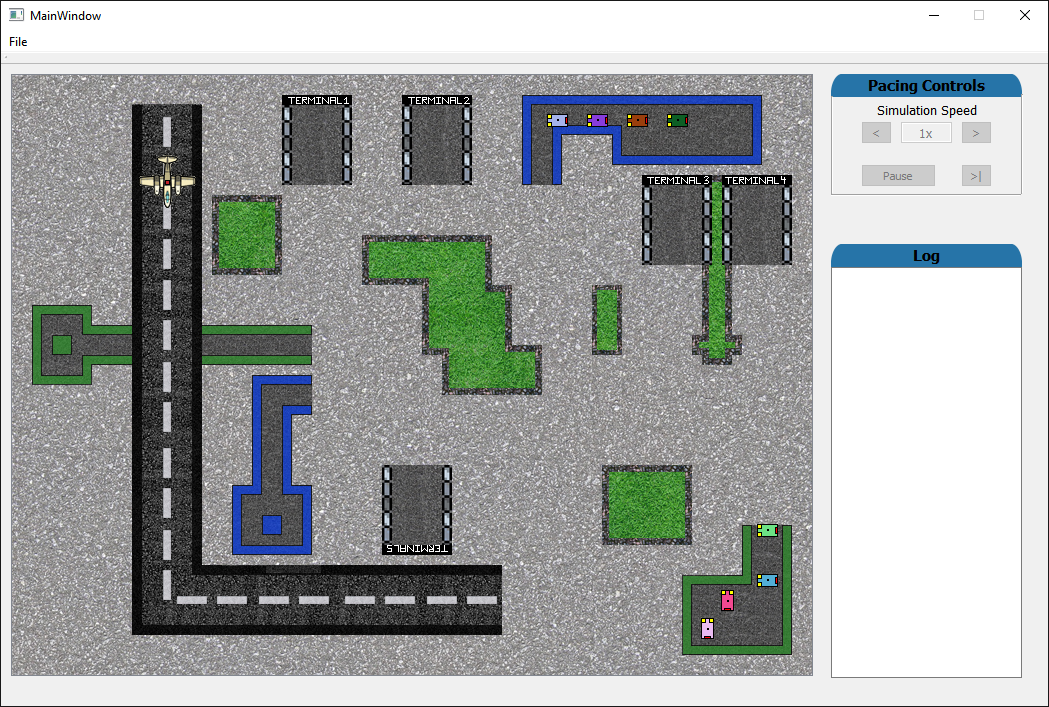


Fig. 2 Example state of the simulation GUI

*Inherent Issues*

When abstracting a real world scenario, there is an inherent loss of much of the real world error present in any given situation. It is certain that many if not all crashes in the world can be attributed to human error. Therefore, it is hard to present or replicate this real world error in a simulated world. This can be counteracted by introducing a random variable to the system to see the effects of an unpredictable aspect to the simulation, or by adding multiple variables to the scenario to see the interactions of each variable. In addition, this scenario is simplified by creating a theoretical airport layout, whereas real world airport layouts are much more complex with many more ground vehicles and aircraft in a typical day.

1. FUTURE ADAPTATIONS

The main limitation of the simulation is the lack of randomness. The real world is extremely random and humans are the main cause. By implementing our system, we eliminate the human error and randomness from the routing and driving, yet there are other sources of randomness. Because this system has to be as robust as possible, future iterations of the system will have to be able to react to these scenarios. The random variables that should be considered in future iterations are debris, weather, potholes, path adjustment, and mechanical issues. Different layouts would also have to be tested as different airports would have to be implemented slightly differently. The system would also have to be scaled as some airports have a vastly larger layout and have a greater reliance on ground vehicle. To make the system more comprehensive, the user interface would also have to be created and tested with extreme diligence. The device that actually conveys instructions to the driver would have to be connected to the main system at all times to receive periodic updates. Additionally, more capabilities could be implemented into the tool to allow the user more specific analysis of any component (i.e. number of collisions, time spent driving for each car, etc.) and let users tune the system by setting conditions/limitations on those parameters directly for a perfectly tailored system. Finally, the ultimate goal of further development would be to implement some degree of machine learning so that once the simulator understands enough of the user's intentions it can complete the rest of the refinement on its on. This simulation stands as a concept for nearly autonomous ground vehicle movement in an airport in order to maximize safety.

1. CONCLUSIONS

The current ground vehicle implementation at airports is, to some extent, a modern day command and control failure. To remedy this collapse of C2, the system proposed would attempt to remove as much human control from the system as possible to try and eliminate human error. To see how the system would behave, a simulated airport was created and the underlying logic that controls the ground vehicles was generated. The arrival and departure of the planes is hard-coded as to act as planned flights, however, the ground vehicles were controlled solely by the pathing system. As with every model based on a real world phenomenon, the simulation is not perfect. There are sources of error beyond humans which comes from the randomness that can’t be planned for. To further push the simulation future iterations would have to be able to compensate for such randomness. Also, real world airports feature much more complex layouts and have a much larger force of ground vehicles. Because this problem space scales up in difficulty, the proposed solution would have to be able to be scaled up as well. The simulation conveys how quickly the airport system becomes complex and how thorough the application of C2 must be. The simulation shows the potential for completely autonomous ground movement in an airport which shall substantially reduce the amount of accidents, incidents, and injuries.

APPENDIX

Code for boolean matrix:

|  |
| --- |
| #include "iostream"  using namespace std;  int main()  {  bool grid [80][60];  for (int x = 0; x < 80; x++) {  for (int y = 0; y < 60; y++) {  grid[x][y] = true;  }  }    for (int x = 0; x <= 79 ; x++) {  grid[x][0] = false;  }  for (int y = 0; y <= 59; y++) {  grid[0][y] = false;  }  for (int x = 0; x <= 79; x++) {  grid[x][59] = false;  } |

Code for pathing algorithm:

|  |
| --- |
| vector<Coordinates> fnFindPath(int startX, int startY, int endX, int endY){  vector<Coordinates> list1;  vector<Coordinates> list2;  //Vector initilized with the end coordinate  Coordinates end;  end.x = endX;  end.y = endY;  end.count = 0;  list1.push\_back(end);  for(int i = 0; i < list1.size(); i++){  Coordinates end1;  Coordinates end2;  Coordinates end3;  Coordinates end4;  if(list1.at(i).x > -1){  end1.x = list1.at(i).x - 1;  end1.y = list1.at(i).y;  end1.count = list1.at(i).count + 1;  list2.push\_back(end1);  }  if(list1.at(i).y < 60){  end2.x = list1.at(i).x;  end2.y = list1.at(i).y + 1;  end2.count = list1.at(i).count + 1;  list2.push\_back(end2);  }  if(list1.at(i).x < 80){  end3.x = list1.at(i).x + 1;  end3.y = list1.at(i).y;  end3.count = list1.at(i).count + 1;  list2.push\_back(end3);  }  if(list1.at(i).y > -1){  end4.x = list1.at(i).x;  end4.y = list1.at(i).y - 1;  end4.count = list1.at(i).count + 1;  list2.push\_back(end4);  }  //Removes coordinates from list2 that are walls  for(int i = 0; i < list2.size(); i++){  Coordinates temp = list2.at(i);  if(!grid[temp.x][temp.y]){  list2.erase(list2.begin() + i);  i--;  }  }  //Removes coordinates from list2 that are already in the main list.  for(int j = 0; j < list1.size(); j++){  for(int i = 0; i < list2.size(); i++){  Coordinates temp1 = list1.at(j);  Coordinates temp2 = list2.at(i);  if((temp1.x == temp2.x) && (temp1.y == temp2.y) &&((temp1.count == temp2.count)||(temp2.count > temp1.count))){  list2.erase(list2.begin() + i);  i--;  }  }  }  //Add all the elements from list2 to list 1  for(int i = 0; i < list2.size(); i++){  Coordinates temp = list2.at(i);  list1.push\_back(temp);  }  //Delete all elements from list2  list2.clear();  //Check list1, if any are the starting position  for(int i = 0; i < list1.size(); i++){  if((list1.at(i).x == startX) && (list1.at(i).y == startY)){  return selectPath(list1, startX, startY, list1.back().count);  }  }  }  return list1;  }  vector<Coordinates> selectPath(vector<Coordinates> list, int startX, int startY, int startCount){  vector<Coordinates> temp;  vector<Coordinates> choices;  Coordinates temp2;  temp2.x = startX;  temp2.y = startY;  temp2.count = startCount;  temp.push\_back(temp2);  for(int i = startCount; i > -1; i--){  for(int j = 0; j < list.size(); j++){  if((i - 1) == list.at(j).count){  choices.push\_back(list.at(j));  }  }  Coordinates back = temp.back();  for(int k = 0; k < choices.size(); k++){  if((back.x == choices.at(k).x && back.y == choices.at(k).y - 1) || (back.x == choices.at(k).x && back.y == choices.at(k).y + 1) || (back.y == choices.at(k).y && back.x == choices.at(k).x - 1) || (back.y == choices.at(k).y && back.x == choices.at(k).x + 1)){  temp.push\_back(choices.at(k));  break;  }  }  choices.clear();  }  temp.erase(temp.begin());  return temp;  } |

ACKNOWLEDGEMENT

We would like to acknowledge and thank Dr. Tod Schuck of Rowan University and Lockheed Martin for his advice and assistance throughout the Command and Control class at Rowan University during the Spring of 2016.

REFERENCES

[1] Flight Safety Foundation. *Ground Accident Prevention (GAP)*. Available from:

<http://flightsafety.org/archives-and-resources/ground-accident-prevention-gap>

[2] Vandel, R. *FSF: Ground Accident Prevention*. Aviation Pros. April 2, 2006. Available from:

<http://www.aviationpros.com/article/10382392/fsf-ground-accident-prevention>

[3] Landry, J. Ingolia, S. *Ramp Safety Practices - A Synthesis of Airport Practice*. Transportation

Research Board, Washington D.C. 2011. Available from:

<http://onlinepubs.trb.org/onlinepubs/acrp/acrp_syn_029.pdf>

[4] Federal Aviation Administration. *FAA guide to Ground Vehicle Operations.* Available from:

<https://www.faa.gov/airports/runway_safety/media/Ground_Vehicle_Guide_Proof_Final.pdf>

[5] Alberts, D. S., and Hayes, R. E. *Understanding Command and Control*. Command and

Control Research Program.

AUTHOR INFORMATION

Patrick Mawhinney

Rowan University CGCE

[mawhin48@students.rowan.edu](mailto:mawhin48@students.rowan.edu)

609-304-4696

Matthew Brooks

Rowan University

[brooks24@students.rowan.edu](mailto:brooks24@students.rowan.edu)

908-399-6053

Christian Heimlich

Rowan University

[heimlichc0@students.rowan.edu](mailto:heimlichc0@students.rowan.edu)

856-669-9133

Kevin Murphy

Rowan University

[murphyk0@students.rowan.edu](mailto:murphyk0@students.rowan.edu)

856-649-3655

Nicholas Parisi

Rowan University

[parisi86@students.rowan.edu](mailto:parisi86@students.rowan.edu)

732-429-3586