# Progress Report

## Abstract

This project aims to find an efficient solution to cave mapping using multiple drones all initially unaware of their environment. The project will tackle various problems in the field of robotics such as localisation, frontier searching, teamwork and collision avoidance of both static and moving objects. The drones will gradually build an occupancy grid of its environment using built in sensors then proceed to map the entire cave by continually identifying and searching frontier cells which are boundaries between unoccupied areas and unknown areas.

## Contents

# 1. Introduction and Motivation

Cave exploration is high-effort, time-consuming and potentially dangerous to humans. This is due to factors such as flooding, hypothermia, tight spaces, falling rocks, harmful gases and particulates such as histoplasmosis from animal droppings [1]. Many caves are too extensive to fully map and are constantly being further explored. This is certainly the case for the Mammoth cave in Kentucky, USA the largest cave network on Earth which contains over 400 miles of cave [2].

This project aims to propose a safer and risk-free method of cave exploration and mapping by simulating a fleet of autonomous flying drones. Each drone will be equipped with capabilities to move in all directions, rotate around its y-axis, avoid collisions, sense its immediate environment, plan its search path and communicate information with other nearby drones. The project will initially be implemented for the 2D case where the environment is viewed from the top-down assuming altitude is constant. Expansion into the 3D case will be an extension to the project. The initial cave environment and each drone's path will be trackable and viewed in real-time on the same GUI to allow for testing and review of the search algorithms employed.

The project will be split into three main stages during implementation:

### i) Generation and visualisation of the cave environment

Cave simulations will be created by initially starting with random noise then passing a few iterations of a cellular automaton over it to create realistic cave looking environments. Certain parameters will be changed to allow customisability of the cave environment to ensure testing is more thorough. More complex and realistic caves will be able to be generated using Perlin noise.

### ii) Exploration by a single drone

At each time step the drone will search its local environment using the built-in sensor and add newly discovered information to its locally stored data structure mapping the cave. The drone will then have to calculate the configuration space from the work space by identifying areas near the cave wall which will cause a collision. Additionally, new frontier cells are calculated, and the best cell is chosen to be searched next.

### iii) Exploration by multiple drones working together

With multiple drones the collision avoidance algorithms will be updated to account for non-static entities such as other drones but could also be used for avoidance of cave fauna such as bats. The drones will possess the ability to communicate what they have learnt to other nearby drones, so they can decide on where to search to maximise exploration efficiency. Communication protocols will be employed to exchange exploration information to minimise mapping duplication and help decide where to explore.

Finally, this project is not solely limited to cave exploration, it may be modified to have applications with other real-world problems involving the use of multiple drones in an unknown environment such as: mass deliveries, rescue operations, cleaning and crop harvesting.

# 2. Objectives

The overall objective of the project is to find a good model for cave exploration with multiple drones such that there is a significant increased efficiency with utilising more than one drone. Each objective has been given a label noting: i) if it is a core or stretch goal, ii) the priority and iii) the dependencies.

## 2.1 Functional

1 To be able to create realistic cave environments with a cell resolution of 10cm by 10cm and maximum size of at least 250m by 250m (i.e. max cave dimensions of 2500x2500). The representation of the cave should have different values to represent different types of cells (e.g. unoccupied and occupied) to be used by the visualiser and for statistical analysis.
*(Core, High, Independent)*

2 Allow cave creation customisability for the following parameters: Maximum environment size in all dimensions, amount of open space, jaggedness of cavern walls, size of tunnel areas, length of cavern *sections and variability/variety of cave*.
(Core, Medium, Dependant on 1)

3 Ability to view the generated cave environment on a GUI.
*(Core, High, Dependant on 1)*

4 Drones should be able to move in all directions when given a request and possess the ability to search its local environment up to a specified range.
*(Core, High, Independent)*

5 Drones should be able to detect and avoid collisions with the cavern walls.
*(Core, High, Dependant on 4)*

6 Drone should be able to locally store known areas of the cave environment. The drone object will use a quadtree data structure to store occupied cells.
*(Core, High, Dependant on 4)*

7 The drone will be able to calculate frontier cells and use an algorithm to decide the optimal frontier cell to explore next.
*(Core, High, Dependant on 4,6)*

8 The drone should be able to track its path from its starting location, so it can be used when exploration is complete and shown visually.
*(Stretch, Low, Dependant on 4,6)*

9 In instances with multiple drones, collision avoidance techniques should be used to avoid drone-to-drone collisions.
*(Core, High, Dependant on 4)*

10 When multiple drones are nearby they will communicate mapping information to avoid search repetition, increasing efficiency. Both drones will use the received communication to update their locally stored map.
*(Core, High, Dependant on 4)*

11 Display search statistics in real-time and at simulation completion such as: Total time taken to explore the cave, percentage/area explored by each individual drone and ratio of explored areas to unexplored areas mapped over time. *(Stretch, Low, Dependant on 3, 4, 8)*

12 Extend the project to be able to create environments and simulate drone behaviour in 3D.
*(Stretch, Medium, Dependant on all previous objectives)*

13 Assume the drones cannot localise themselves with 100% accuracy and so use localisation techniques with the sensed data to localise themselves in the environment accurately.
*(Stretch, Low, Dependant on 4, 6, 8)*

## 2.2 Non-Functional

1 To be able to create realistic cave environments in less than 10 seconds.
2 The searching simulation should run close to real-time speed.
3 To be able to view informative data on the environment and drones within the GUI.
4 GUI should be responsive, usable and aesthetically pleasing.

# 3. Project Overview

The project will be written in the C++ language using Microsoft Visual Studio Community 2017 [3] on Windows machines and Atom [4] as the text editor for Linux. Visualisation will be achieved using GLUT [5]: an OpenGL utility toolkit which contains libraries written in C++ and will allow the creation of a GUI to showcase the cave and drones. Git [6] will be used for source-code management and version controlling. GitHub [7] will allow the Git repository to be stored remotely allowing work to be completed and transferred at home and at university, whilst also ensuring the project is backed up.

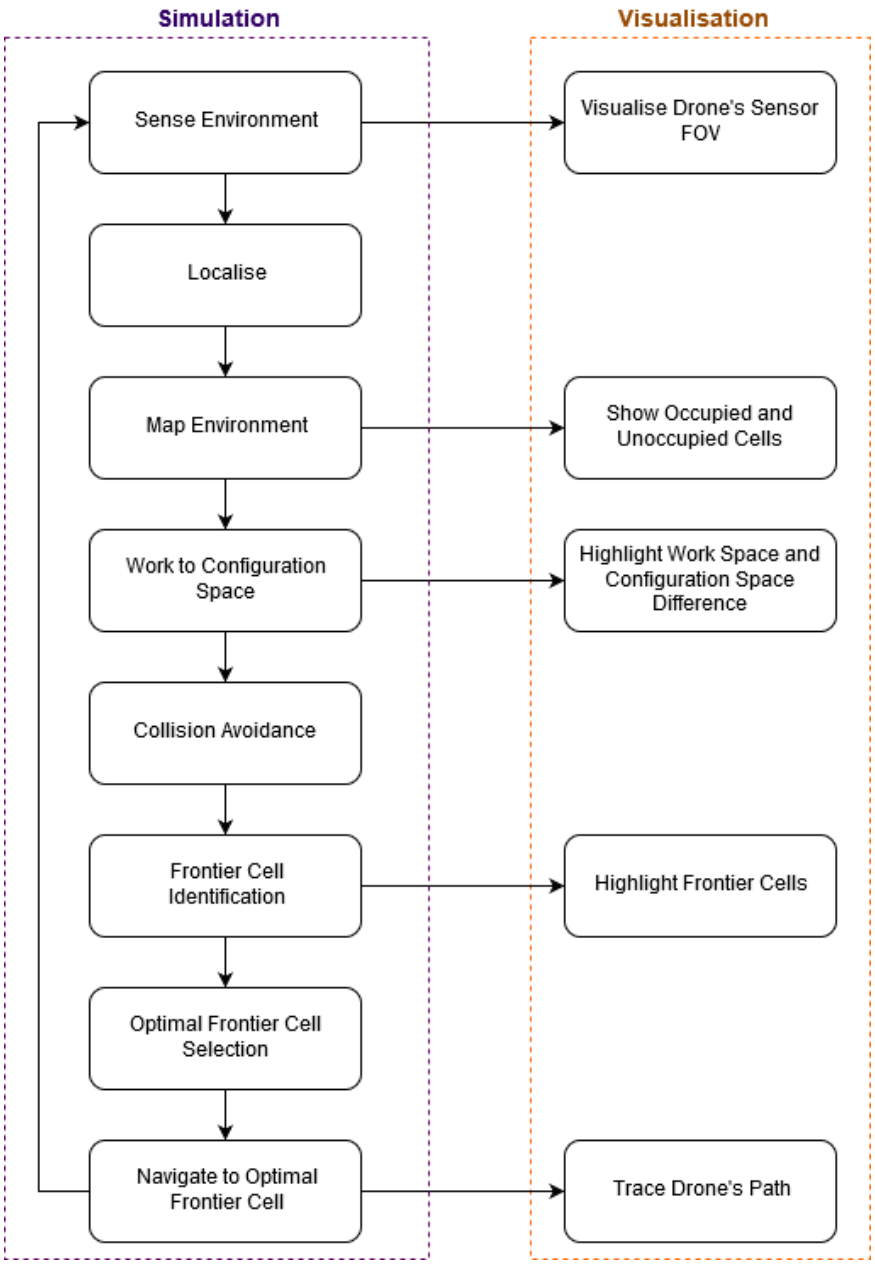



*Figure 1 - Single Drone - Searching Process with visualisation processes.*

# 4. Project State

## 4.1 Completed/Ongoing objectives.

Partial progress has been made on the cave generation and visualisation which encompasses objectives 1 and 3. As you can see from the screenshot, the cave network is well connected, but does lack variety in its structure. All paths are of similar length, width and size. Some progress has been made for objective 2 to alter some cave parameters however additional research will be required for this as the complexity was initially not fully anticipated. Perlin/Simplex noise is a possibility for adding complexity to the cave structure and making it look more realistic [8].

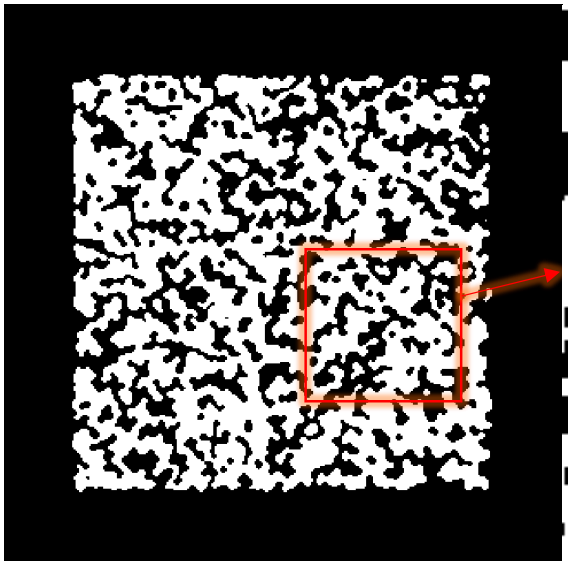


*Figure 3 - Cave generation prototype, unoccupied cells are represented in white. 250x250 array.*

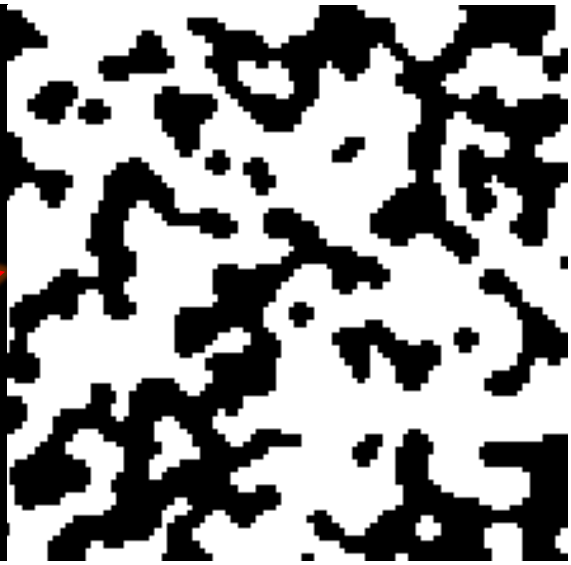


*Figure 2 - Cave generation prototype zoomed in.*

Currently cave generation uses a naïve approach by generating a random array of values and thresholding each by a set amount known as the “fill percentage”. Then a cellular automata rule is applied across the entire cave a few times to smooth and generate the eventual “cave” texture.

The visualisation window has functionality for zooming in, zooming out and moving in both the positive and negative x and y directions achieved by binding keyboard key detections with GLUT.

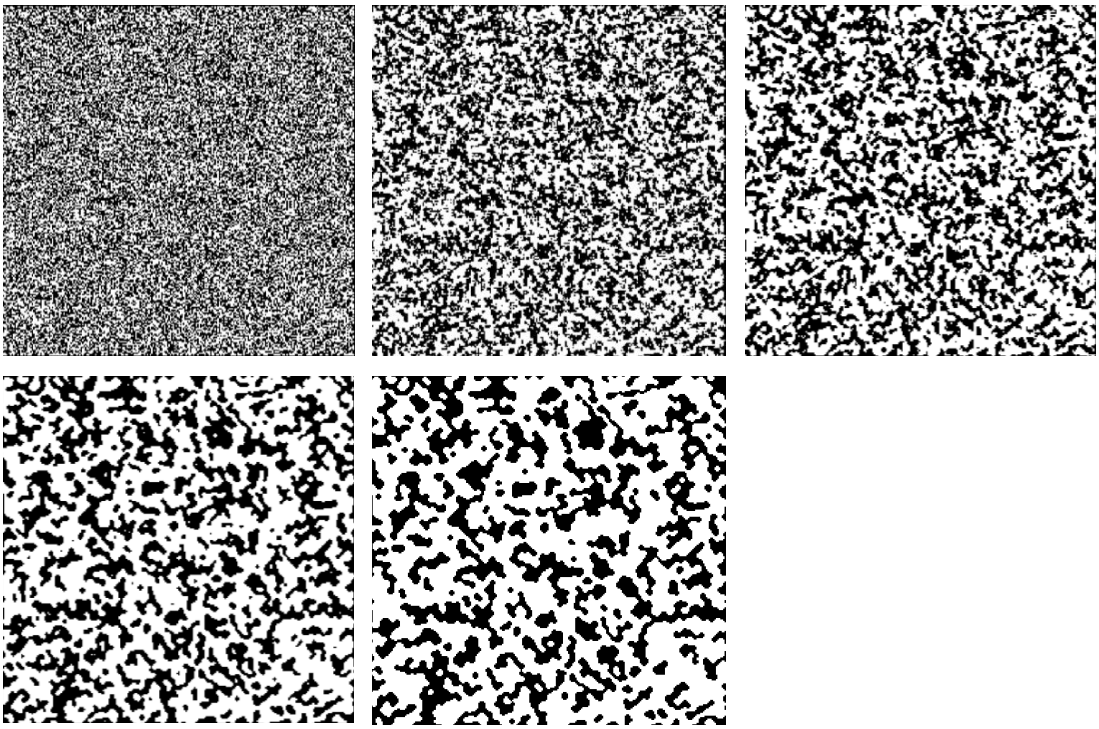


*Figure 4 - From left to right: Initial Noise, 1 Smooth pass, 2 Smooth passes, 5 Smooth passes, 20 Smooth passes. All for a 250x250 cave.*

For each smooth step: If a cell had more than 4 occupied neighbours there was a 100% chance to become occupied. On the other hand, if a cell had less than 4 occupied neighbours there was a 75% chance to become unoccupied.

## 4.2 Mapping

Various methods and data structures have been researched and compared to be used for each drone to map the environment. These include: 2D array, Quadtree, Spatial hashing [9] and Binary space partitioning [10]. Quadtrees have been selected to store occupied cells and frontier cells for each drone for the following reasons:

i) Reduces memory allocation at the start.
ii) Expandable at runtime.
iii) Combination of sparse and dense regions will not affect the data structure's performance. Particularly good for cave environments where there will be large open areas (sparse) and cavern walls (dense).
iv) Collision detection is more efficient as a small subset of nodes are checked.
v) Allows extensibility into 3D visualisation as mesh generation from quadtrees has been researched. [11]

# 5. Research

There are various aspects of the project which I have identified that require further research including but not limited to:

- Cave generation using Perlin/Simplex noise:
  Perlin noise is a well-researched and documented method for creating procedurally generated environments [8]. It can be used to build upon the naïve random noise method currently in use to add sparsity and realism.
- Collision avoidance:
  Avoidance of cave walls can be calculated by defining a secondary border between the drone and the wall with a distance equal to the size of the drone. Starting with work space and assuming the drone is a point, configuration space can indicate where the drone can navigate to without collisions [12].
- Visualisation optimisations:
  For example, drawing less polygons by combining groups of dense polygons to reduce the number of vertices.
- Tools to analyse efficiency and effectiveness of solution:
  Using flood fill to calculate the maximum area a drone could explore.
- Localisation.
- Frontier searching.
- Optimal searching with multiple drones.
- Representations of the cave.

# 6. Project Management

## 6.1 Timetable

The first version of the timetable can be found in the project specification. An improved timetable has been created, expanding stages and better accounting for free time during the Christmas break and term 2.

More time was dedicated for the cave generation stage (green), as well as for the single drone stage (blue) and teamwork stage (red). Each of these stages were also further split up into sub-stages to further improve time-management. The improved timetable can be found as Figure 5 in the appendices section.

## 6.2 Risks

In addition to the risks highlighted in the project specification the following additional risk has been identified. Cave generation has a possibility to slow down dramatically as the cave size increases and as increasingly complex features are integrated into the cave structure. For example, on a computer running an Intel i7-6700k CPU and NVIDIA GeForce GTX 970 GPU, cave generation runs with the following usages when repeatedly zooming in and out of the visualisation to force redraws.

| **Cave Size** | **CPU Usage** (i7-6700k) | **GPU Usage** (GTX 970) |
|---|---|---|
| 50x50 | 17% | 0.8% |
| 250x250 | 18% | 9% |
| 1000x1000 | 30% | 35% |

Generation of caves sizes 50x50 and 250x250 ran smoothly on this particular setup, however a cave of size 1000x1000 ran noticeably slower with lag. This is not sufficient as for a 500mx500m map: 250,000 cells are generated and will need to run fast to satisfy objective 1. A cave of dimensions 1000x1000 will have a maximum of 4,000,000 vertices drawn to the screen, this number should be kept as low as possible as it could serve as a GPU bottleneck [13].

To compensate the risk, I will explore various methods and algorithms to optimise drawing cells onto the screen by grouping them together and store the vertices instead of recalculating on each draw command.

## 6.3 Methodology

An Agile approach will allow easier re-organising of certain sub-stages of the project. For instance, if cave generation takes too long, the individual drone searching stage which occurs next chronologically can still go ahead on time using a simpler than excepted generated cave. Then complex cave generation can be implemented later in development and this should not affect code which should have occurred chronologically afterwards.

At the end of each sub-stage in development which can be found on the improved timetable (Figure 5) a small amount of time will be dedicated to testing that the sub-stage works with the previous stages and as intending in the project objectives. This is to ensure problems do not cascade unintentionally to the end of the project development and that there is a continual working product at the end of each stage.

## 6.4 Testing

Due to the graphical nature of the project and the visualisation, many aspects of the project will be easily tested by using the GUI. For example, errors in cave generation can easily be identified from viewing the displayed cave on the GUI. An example of this has already occurred when the orthographic view volume was set too small and no shapes were shown. Using the GUI, the view volume was set correctly so the cave could be viewed.

In the cave generation stage a set of unique cave environments will be used for testing purposes. The parameters of the generator and seed value will be stored to be able to reproduce them later.

# Glossary

| | |
|---|---|
| *Frontier cells* | Unoccupied cells that are adjacent to unknown cells. |
| *Localise* | Process of a robot identifying where it is in an environment. In the context of this project the robot will localise using multiple scans of the environment at different positions. |

# References


[1] "Histoplasmosis," [Online]. Available: https://www.healthline.com/health/histoplasmosis.

[2] "Mammoth Cave National Park," [Online]. Available: https://www.nationalgeographic.com/travel/national-parks/mammoth-cave-national-park/.

[3] Microsoft, [Online]. Available: https://visualstudio.microsoft.com/. [Accessed 20 November 2018].

[4] "Atom," [Online]. Available: https://atom.io/. [Accessed 10 November 2018].

[5] "GLUT - The OpenGL Utility Toolkit," [Online]. Available: https://www.opengl.org/resources/libraries/glut/. [Accessed 20 November 2018].

[6] [Online]. Available: https://git-scm.com/. [Accessed 10 November 2018].

[7] [Online]. Available: https://github.com/. [Accessed 10 November 2018].

[8] Wikipedia, "Perlin Noise," [Online]. Available: https://en.wikipedia.org/wiki/Perlin_noise. [Accessed 20 November 2018].

[9] E. J. Hastings, J. Mesit and R. K. Guha, "Optimization of Large-Scale, Real-Time Simulations by Spatial Hashing," [Online]. Available: http://www.cs.ucf.edu/~jmesit/publications/scsc%202005.pdf. [Accessed 22 November 2018].

[10] Wikipedia, "Binary space partitioning," [Online]. Available: https://en.wikipedia.org/wiki/Binary_space_partitioning. [Accessed 23 November 2018].

[11] M. de Berg, O. Cheong, M. van Kreveld and M. Overmars, "Quadtrees - Non-Uniform Mesh Generation," in *Computational Geometry Algorithms and Applications (3rd Edition)*, 2008, pp. 307-315.

[12] M. de Berg, O. Cheong, M. van Kreveld and M. Overmars, "Robot Motion Planning - Getting Where You Want to Be," in *Computational Geometry Algorithms and Applications (3rd Edition)*, 2008, pp. 284-290.

[13] "Unity - Manual: Optimizing graphics performance," Unity Technologies, [Online]. Available: https://docs.unity3d.com/Manual/OptimizingGraphicsPerformance.html. [Accessed 21 November 2018].

# Appendices

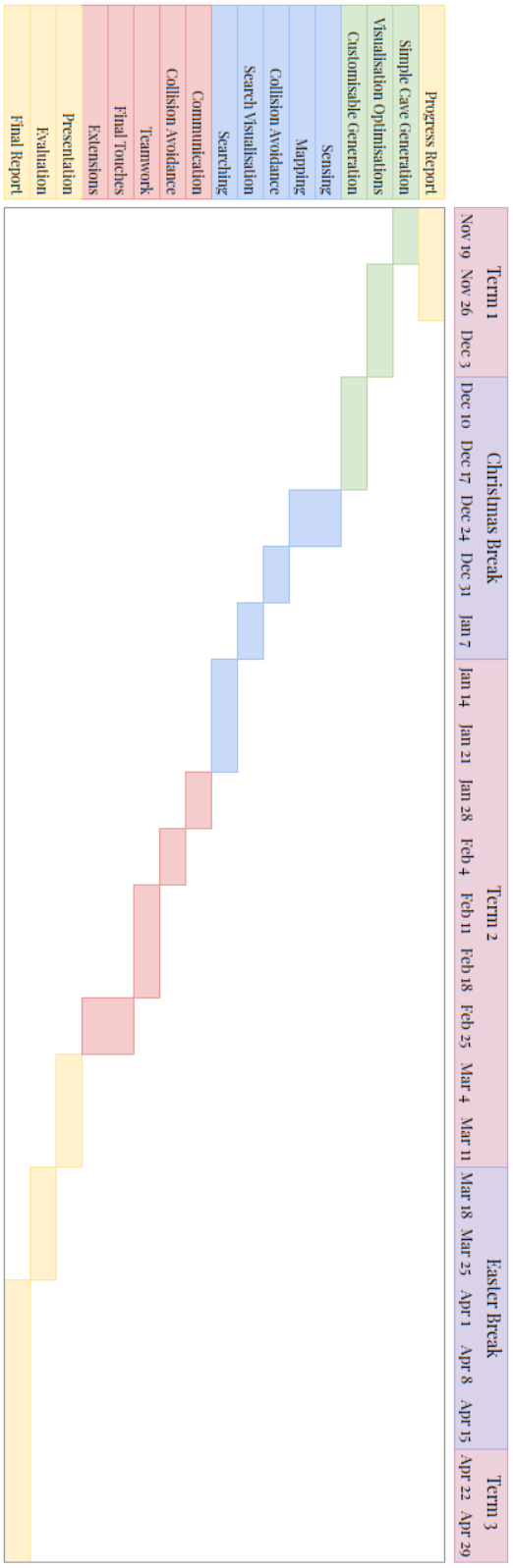



*Figure 5 - Improved Timetable*

# CS310 Dissertation
# Project Specification

Kyle Gough

October 2018

## 1 Introduction and Motivation

Cave exploration is high-effort, time-consuming and potentially dangerous to humans. This is due to such factors as flooding, hypothermia, tight spaces, falling rocks, or harmful gases and particulates such as histoplasmosis from animal droppings [2]. Some caves in particular are too extensive to fully map and are constantly being further explored which is the case for the Mammoth cave in Kentucky, USA the largest cave network on Earth which contains over 400 miles of cave [1]. This project aims to propose a safer and risk-free method of cave exploration and mapping by employing a fleet of autonomous flying drones. The drones will use swarm AI to avoid contact with each other and the cavern walls whilst simultaneously identifying potential unexplored paths. The drones will avoid crowding by splitting at junctions to increase exploration efficiency. Communication protocols will be employed to exchange exploration information to minimise mapping duplication and help decide where to explore. This project will generate realistic cave environments and simulate drone behaviour on them for them to eventually be used in the real world. Additionally this project is not solely limited to cave exploration, it may be modified to have applications with other real world problems such as: mass deliveries, rescue operations, and for the agriculture sector in planning the harvesting of crops.

## 2 Current Hardware and Software

### Zebedee - CSIRO

Zebedee is a hand-held device which contains a laser scanner in constant rotation. A human operator holds the device whilst traversing an area and the signals received from the device can be used to create a map of the surrounding area. It allows the creation of a complex and detailed map of the area, however it is restricted by the speed and reachability of the human operator in the environment [4]. The human operator is at risk of all the dangers of cave exploration.

### Hovermap - CSIRO

Hovermap is an attached to hover drones which uses LIDAR technology to map the entire surrounding environment. It is similar to Zebedee but is not required to be held and is not restricted by human speed or reachability of the environment. The hovermap is not yet capable of interacting with other similar drones to increase exploration efficiency [3].

# 3 Objectives

The objects of the project can be split into three main sections.

(i) Cave environment generator.

(ii) Independent pathing, collision avoidance and mapping of a single drone.

(iii) Teamwork with other drones, unexplored area predictions.

1. Create a 2D cave environment generator that is customisable, reproducible and realistic. The produced environment should be suitable for simulations to utilise. Some factors of the cave which may wish to be customisable include:
   - Size of open areas.
   - Size of tunnel areas.
   - Frequency of open areas.
   - Total area/volume of open space.
   - Length of cavern sections.
   - Stalagmites, stalactites and pillars where stalagmites and stalactites have joined together.
   - Cavern wall textures.

   Properties of the cave should abide by real life cave properties such as having stalactites and stalagmites occur in pairs, where stalagmites are generally larger and wider than their corresponding stalactite.

2. Create a simulation of a drone which can:
   - Sense the surrounding environment up to a specific range.
   - Track its current location and path relative to the starting point.
   - Locally map and store the currently explored environment.
   - Sense nearby drones to avoid collisions.
   - Communicate with nearby drones and exchange exploration information.
   - Identify possible unexplored areas.
   - Decide the best path to take to explore an unexplored section.

3. Drones should aim to scan the entire cave network. In instances of multiple drones they should work together to map the cave efficiently by:
    - Communicating map information when two drones are in communication distance.
    - Use exchanged information update the drone's local data.
    - Explore areas not yet explored by any known drone. Plot an efficient path to these areas.
4. Drones should aim to follow some rules to avoid unnecessary costs, increase efficiency and improve health and safety. These rules are inspired by a ruleset employed in the 2D simulation BOIDS to steer birds in a flock.
    - Avoid contact with the cave walls and other hazardous environment.
    - Avoid contact with other drones.
    - Prioritise areas where other drones are not actively exploring.
    - Explore areas not yet explored by any known drone.
5. The final simulation should convey useful information and statistics such as:
    - Total time taken to explore all areas of the cave.
    - Path taken by each individual drone.
    - Ratio of explored areas to unexplored areas mapped over time.
    - Area/Volume of unexplored areas identified by each individual drone.

# 4 Possible Extensions

- Be able to generate visualisations and simulate solutions in 3D space.
- Use navigation techniques to find the best/safest routes to a specific point in the cave, or from any point in a cave to an exit.
- Use machine learning to predict where other drones may have explored in the absence of other drones being nearby to exchange information.

# 5 Methodology

An agile approach will allow a simple cave generator to first be implemented, with multiple iterations each including a new feature to the caves such as stalagmites and stalactites. This allows sufficient time to create the drone, swarm and communication behaviours and making it work on a simple cave, then with multiple iterations making it work and be efficient in more complex and realistic cave simulations.

# 6 Testing

In conjunction with an agile development methodology, test driven development will be used. A test case for each objective will be constructed and then tested to ensure existing code still operates correctly when new features are implemented.

The graphical nature of the project will assist in debugging. It will particularly assist in identifying errors in the cave generation algorithms.

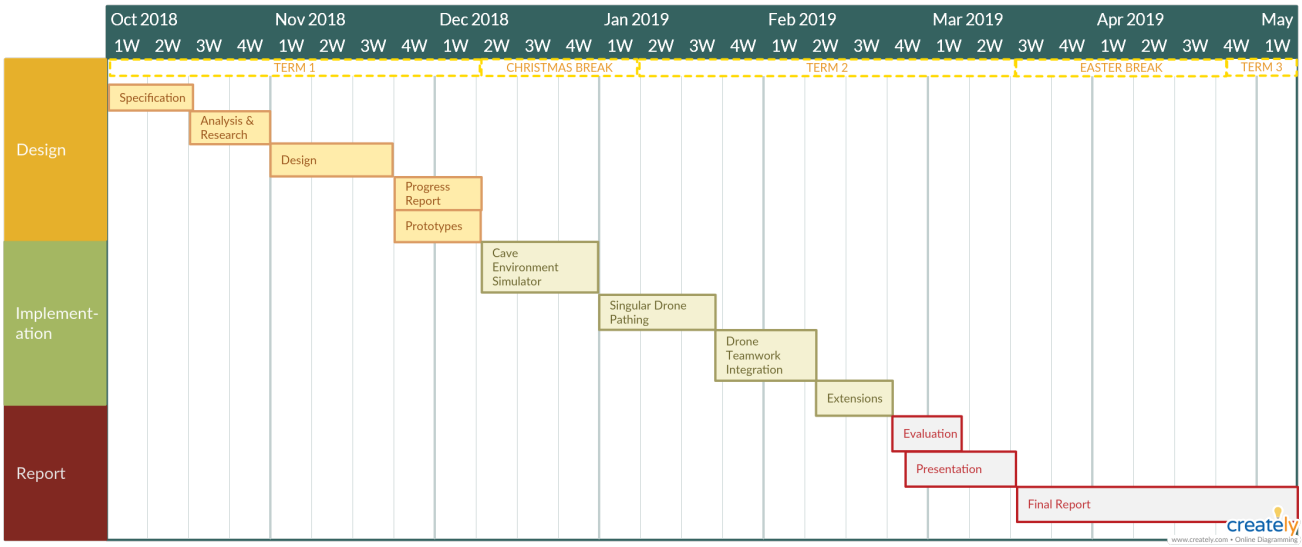
The effectiveness of the drones' behaviour as a group will be tested against: a single drone working independently, multiple drones working independently without teamwork behaviours and finally different numbers of drones with teamwork behaviour. This will identify how effective the teamwork behaviours are and how the efficiency of cave exploration scales with the number of drones present.

# 7 Timetable

Below is the schedule for the hours per week dedicated to the project. Currently 345 hours have been allocated to the project, exceeding the recommended 300 hours. This is to accommodate any unforeseen circumstances that may occur such as: illness or holidays. A larger workload per week has been allocated to the Christmas break (120 hours) and term 2 (120 hours) because I have 4 modules in term 1 and only 2 modules in term 2. Additionally Easter break is being partially reserved for exam revision.

- **Term 1 (1st Oct - 7th Dec)** - *Hours/week: 8, Total Hours: 80*
- **Christmas Break (8th Dec - 6th Jan)** - *Hours/week: 25, Total Hours: 100*
- **Term 2 (7th Jan - 16th Mar)** - *Hours/week: 12, Total Hours: 120*
- **Easter Break (17th Mar - 23rd Apr)** - *Hours/week: 15, Total Hours: 45*

The following Gantt chart showcases the different stages of the project and a guideline for when certain deadlines should be met including the design phase, implementation phase which also includes some time for extension work, and the report phase.




# 8 Risk Assessment

Here is a list of identified risks and proposed solutions to mitigate them if required.

- **Risk 1**: Illness is a highly potential risk that could occur at any stage of production and can severely slow down progress. Solution 1: Currently the schedule offers more dedicated hours to the project than the recommended amount to account for this risk.
- **Risk 2**: The project is solely dependant on software libraries, and the chosen graphics engine. The risks include the lack of software updates provided by the creator. Using the libraries and engines at a lower version if required will mitigate this risk.
- **Risk 3**: This will be my first project using C++, therefore I may run into errors not yet experienced and will have a slower production rate than other languages I am more familiar with. If the use of C++ becomes too troublesome I can switch to C# with Unity which I have more familiarity with.

# 9 Legal, Social and Ethical and Professional Issues

This project will require no dependencies with other people, therefore there are no issues to consider.

# 10 Resources

- **Git** - For version control of the software and resources. To track updates and provide rollback if a recent software version encounters an error.
- **Github** - For storing the git repository remotely. Provides online version control and backups of the project.
- **Unity** - Game engine with strong graphics capabilities for simulations. For visualisation of the cavern environment and simulating the navigating drones.
- **C#** - Object-orientated programming language used in parallel with Unity to create a simulation.
- **OpenGL** - API for 2D and 3D vector graphics. For visualisation of the cavern environment.
- **C++** - Object-orientated programming language used in parallel with OpenGL to create a simulation.

# 11 Glossary

- **LIDAR** - records the distance to an object by sending a pulsed laser light and measuring the reflected pulses with a sensor.
- **CSIRO** - Commonwealth Scientific and Industrial Research Organisation
- **BOIDS** - Program developed by Craig Reynolds which simulates the flocking behaviour of birds.
- **Stalagmite** - Rock formation that occurs on the ground and rises up.
- **Stalactite** - Rock formation that occurs on the cave ceiling.

# References

[1] National Geographic. *A Guide to Kentucky's Mammoth Cave National Park*. `https://www.nationalgeographic.com/travel/national-parks/mammoth-cave-national-park/`. Accessed: 08-10-2018.

[2] Healthline.com. *Histoplasmosis*. `https://www.healthline.com/health/histoplasmosis`. Accessed: 08-10-2018.

[3] Commonwealth Scientific and Industrial Research Organisation. *Hovermap*. `https://research.csiro.au/robotics/hovermap/`. Accessed: 01-10-2018.

[4] Commonwealth Scientific and Industrial Research Organisation. *Zebedee*. `https://research.csiro.au/robotics/zebedee/`. Accessed: 01-10-2018.