Functional Specification

Title: Drone Traffic Simulation

Name: Jack Murphy

Student Number: 13436498

Course: CASE

Year: 4

Plagiarism Statement:

Table of Contents — Page Number

1. Introduction

1.1 Overview

My idea is to create a system that simulates drone traffic in a cityscape. The simulation will be a 3D Model of a city with drones flying through the air. The drones will fly between certain heights using the cover of the buildings to protect them from the elements, which is the attraction of cityscape environments. Their payloads will be deliveries, ultimately replacing DHL, FedEx, as well as other delivery companies and assisting with first responders for accidents and emergencies. With such a future, what should drone traffic look like, how many and how fast should they be able to fly? Should there be a traffic system with traffic lights, rights of way at crossings, roundabouts, etc.? These are the questions I hope to answer with this simulation.

The idea is that I will successfully be able to create a simulation of a smooth traffic flow of drones, regardless of whatever external factors may occur. Simulation will produce a series of statistics about various occurrences in the simulation e.g. time it ran for without a crash, number of drones in the world, etc. The system will also have to find a way to maximise the traffic threshold. The traffic threshold is the point at which there are so many vehicles travelling the same route that traffic begins to slow down. I will try to create a simulation to push the boundaries of this threshold.

The map will be divided into a series of cells. These cells will horizontally combine to create lanes of traffic. As all drones may not be the same size the cells must be large enough to cater for the largest drone, and also allow extra room so that the drones can react in case of an unforeseen event. The cells will also stack upward to create multiple levels of traffic flow which is what differentiates from current vehicular traffic flow. Lanes may also be assigned to a drone based on the speed of the drone. For example, the fastest drones or those responding to an emergency, will travel in the top lane.

Each individual drone will have to know its own location on the map, i.e. what cell it is in. It must also know the location of the other drones. As mentioned previously the drones will be able to react to various events that might affect other drones in their vicinity or themselves. These events include sudden loss of power, strong wind, a drone crash. Each of these events will no doubt cause a disruption in the flow of traffic so the unaffected drones will have to realise this and react accordingly. The affected drones will also have to react, in a safe manner so that they do not cause damage to property, other drones or more importantly pedestrians. My system will be designed to randomly execute events like theses to demonstrate that drones can deal with the situation with minimal disruption to traffic.

It is my view that a system like this will shape the future of legislation around drones. As it stands the laws controlling drone usage are very limiting. These stringent controls are negatively affecting the many beneficial uses of drones. These uses range from commercial use like package delivery to emergency response. The aim of my system is to create a model by which you can have hundreds or thousands of auto-piloted drones flying above the streets in a safe and continuous manner. This model will be proven to work with a set of statistics.

1.2 Business Context

The main business case for this system is to prove the benefits of drones to the regulators of aviation technology in not only Ireland but all over the world. The main reason for strict controls on drones is due to the stigma attached with their use, or rather their misuse. Many of the general public find them to be an invasion of privacy and a nuisance. It is for this reason that Irish Aviation Authority and the Civil Aviation Authority in the UK have banned their use in urban areas.

This system will hopefully change the opinions of the general public by showing them that automated drones are safe and have many benefits. By changing the general perception of drones, legislation can be changed to allow automated drones for commercial use. I will be able to present the set of statistics to back up my points that the automation of drones can be run in a safe and efficient manner.

1.3 Glossary

.Net Framework

Software framework that allows interoperability between various different programming languages.

CSharp (C#)

Programming language that incorporates multiple programming disciplines

Structured Query Language (SQL)

Used for sending and retrieving data from a database

Language Integrated Query

Adds data querying capability to a number of .Net languages.

Unity

Cross platform game-engine to develop games or simulations.

R

Language developed for use with probabilities and statistics, I will use to analyse the data output and graphically represent the data.

2. General Description

2.1 System Functions

The system that I am creating will be a simulation of drone traffic. This simulation will be run using the Unity game engine. The main functions with this system will be to control the automation of the drones and to create events to try and disrupt this automation. The drones will act in the system much like an AI would act in a regular computer game. The drones will be user generated with a destination and they will navigate to this destination. In addition to these user generated drones, there will be computer generated drones.

For the function to add user generated drones, the user will be able to choose from a number of different drone models stored in a drone database. The drone models will be stored in a database for reuse. The idea behind this is to make the system more realistic. So drones will have attributes like battery life, damage, speed, size, etc. The function to add a drone will require a start position and a destination. This information will be used to allow the drone to navigate to the destination and back. There will be a set number of start positions for drones to start from, this will imitate a delivery warehouse.

Once a drone has been added to the system, a navigate function will take in a drone object and navigate to the destination and back. The navigation will use other functions to determine the correct lane to be in and what route to take. The function to select a route will be called first, this will then call the function to choose a lane. The reason for this is so that a route can be recalculated mid route due to events occurring on the map. The route will be calculated based on the shortest distance to the destination along with, the drone traffic, the number of lanes available (note: a lane may become unavailable due to an event occurring).

Once a route has been chosen the best lane will be chosen. This function will determine the best lane to take based upon the volume of traffic in the lane, the speed of the drone and the direction the drone is travelling i.e. is it taking the next left, going straight on, etc. The route function will be called when the drone is created, it will then be called while the drone is moving. For this reason, the time of execution for this method must be faster than the time it takes for the drone to move off or else the method will have to start calculating the route from the drones next position. This way the route calculation will be as accurate as possible.

The navigation function will also include an event handler. This event handler will handle events occurring with drones in its vicinity as well as the drone itself. If the event is occurring to the drone itself then it will notify the other drones, as well as taking specified action depending on what the event is. If the drone is being notified of an event happening in its vicinity, then it will take a specified action depending on the event.

Another function that the system will perform will be to generate an event. An event can be user generated or computer generate. The computer will randomly generate events. The event generation function will randomly select an active drone and randomly select from a list of possible events. Once this method executes the active drone will react to the event.

The system will also contain a function to record the data from the simulation. It will record the number of drones currently in the system and the run time between system failure. A system failure in the case of the simulation would be if the model stops working due too much traffic. It will also record when an event is triggered and what kind of effect this had on the traffic flow in that area. When this data has all been collected I will be able to analyse it and come up with statistics using the R programming language.

2.2 User Characteristics and Objectives

As this system is a simulation, users will have a very limited interaction with the system. The main interaction with the system will be a minor element of data entry on the part of the user. The main requirements of this system would be to present the simulation in a realistic and engaging way, then at the end have some useful statistics about the running of the simulation. The user input area must be as simplistic as possible to allow for somebody unfamiliar with the system to use it. The reason for this is because our aim is to change public perception of drones for commercial use. So if the system is not usable by somebody in a demonstration it would be a difficult task to change peoples' opinions.

The user interaction element of the system will be to add a drone. The adding a drone will have most interaction as the user will be required to select a drone from a list of drones and then select a start point from a list of set start points and a destination which could be chosen on the map.

2.3 Operational Scenarios

User Generated Drones

This is the first interaction the user will have with the system. They will be required to select a drone from a list of available drones. This list will be retrieved from the database. Once a drone is selected they will select a start location on the map and once the start location is chosen they will be able to select a destination. Once this is done they must select how many drones are to be running in the simulation at one time.

Computer Generated Input

In this scenario the computer is generating the drone. The computer will select a random drone and a random location from the database and then randomly generate a destination. It will repeat this process until it has created the number of drones to be active in the system as per the users input.

Drone Navigation

This will be the drones first action in the world. The method to navigate will make use of a number of different methods. First it will calculate the best route. The best route will always be the quickest way from A to B. The quickest way from A to B is a mixture of the shortest distance between the points and the traffic flow. This scenario will make use of two methods, route calculation and lane selection. The best route will be calculated first, to do this a number of routes will be calculated and compared. The best one will be the quickest and shortest route. Once this is chosen, the best lanes to take will be next. To do this, the function will determine what height the drone should be at based on the top speed of the drone and if it is an emergency response drone. Once this has been determined, the function will select which lane to be in at that height. Based on where and when it will be turning. The drone object will continuously poll this method to make sure it can handle events.

Generate Event

This scenario will occur randomly throughout the program. It will select a random drone object and choose a random event from a stored list of events. These may include power failure, crash, etc. The purpose of this is to create a disruption for the drones to react to.

Generate Statistics

This is the final scenario of the program. When the simulation has finished running. This will gather all the data recorded in the database and run an R script to graphically represent the data and to perform some statistical functions on it. This will be the output of the program. The simulation will just show the running of the program.

2.4 Constraints

While undertaking this project I believe that I will face a number of constraints which may limit the effectiveness of the project. These include:

Time Constraints

Time may become a constraint during this project. Ideally I would like to fully flesh out all my ideas but due to time being a factor this may not be possible. Many assignments and exams will take away development time from this project. I plan on solving this issue by dividing my time efficiently and to keep the components in an organise manner, so that I can make the most out of the time I have for development.

System Constraints

System requirements may be a factor in this project. The program will have a lot of heavy processing to do, running the game while also reading and writing to a locally stored server and then to finish it will be working with a large dataset or a number of datasets. Currently this will not be a problem for my PC, but it may cause problems if it were to be run on a less powerful machine.

3. Functional Requirements

3.1 Generating a Drone

Description

Generating a drone will be a function either user triggered or system triggered. A user or the system will select a drone from a list of drones which are stored in the database. A user/the system will choose a start location from a set list and then they will choose a random destination on the map. If they try and enter a location not on the map they will be given an error message to prompt them to enter a valid location.

Criticality

This function is essential to the entire system. If there are no drones in the system, then there can be no simulation of drone traffic. This will be the first feature to be implemented as all the other features rely on drones being created.

Technical Issues

The feature to add a drone will be basic user input. I will use the set models that are available in unity to create the selectors. The data to fill the selectors will come from a table of drones in the database. These drones will have a list of attributes to them e.g. speed, size and damage. The selector will only show inactive drones. There will also be selectors for the location which will come from a list in the database. This will be a small list of set starting points. To select the destination, the user will drop a pin on the map, the coordinates of this pin will be stored in the details for that drone for use by the navigation function.

Dependencies

Creating a drone is not dependent on any other functions created yet. Although, the method does require that the database tables are created and filled with drones and locations.

3.2 Route Calculation

Description

The route calculation function will determine the quickest most efficient way to get from the drones' current position to the destination. This will be worked out as soon as the drone is created. It will also be recalculated as the drone moves to allow the drone to readjust in the case of disruptive events that may occur. The calculation method will take into account drones that are currently active on the map and their respective locations.

Criticality

This function is necessary to the running of the simulation. This is needed for the drones to move around the map as well as react to events being triggered. On a more specific level this is needed so that a lane can be selected.

Technical Issues

The function for the drone to calculate a route will require no user input. The system will retrieve the destination from the database. Once it has this it will select the shortest route to this location, it will search this route for any obstructions or events that may have occurred along the route. It will do this by checking the database to see if any events are active as well as checking the volume of drones along that route. It will then repeat the process for other routes, then compare the estimated travel times and choose the best one.

Dependencies

The calculation of the route is dependent on the generate a drone method. The drone must be created in order for the route to be calculated. The method requires active drone objects to be passed to it.

3.3 Selecting a Lane

Description

The function to select a lane will be used by the overall navigation function. This method will select the most suitable lane for a drone to take, this will contribute to the drone reaching its destination in the shortest time possible. The method will select one from a number of lanes on each street in the map. The lane will be selected based on the speed of the drone, the direction the drone is travelling and if the drone is an emergency responder. If the drone is turning left it will be in the leftmost lane, if the drone is heading straight on then it will be in the centre lane and so on. This will allow for the smoothest flow of traffic.

Criticality

This system is critical to the smooth running of traffic flow. If lanes are not assigned to each drones, then a lot more crashes will happen. It will not prevent the simulation from running but it will cause the simulation to fail as traffic will be unable to flow.

Technical Issues

This function will be passed a drone object and a route object, using this route object the function will select a lane based on the route that has been passed and the speed of the drone. This method will also need to make sure that the optimum lane is free to use and that no events have occurred in this lane.

Dependencies

In order for this method to work first a route must be calculated and the drone passed must have been created also. If they are not created, then the function will fail.

3.4 Generating an Event

Description

The generation of an event will be randomly triggered in the application. There will be two types of events. The event can effect one of the drones or it can cause an effect on a point in the map. An example of an event that may occur on a drone might be a power failure to one of its propellers. Then an example of an event that may occur to the environment would be an obstruction such as a stray tree limb. These events should cause some sort of a disruption to the traffic flow, then the drones should react accordingly.

Criticality

This feature is not critical to the system, it should still run without this functionality. The purpose of this functionality will be to add more complexity to the system. As well as this it should show that the flow of traffic will remain more or less unaffected by unforeseen events.

Technical Issues

This event trigger will be run in the main game loop. The method will trigger randomly. It will then choose a random event from a table of events. If the event type is an event that occurs to a drone, then the drone object will be randomly selected and the event will be applied to that drone. The drone will have an event handler, to keep checking if an event has occurred to it.

Dependencies

In order for this method to work the method to create a drone must first be implemented. The world model must also be in place and the table must be created and events must be added into it. If these are not in place before the method is to be coded, then it will fail in the implementation.

3.5 Generating Statistics

Description

The generation of statistics will be the output of the simulation. The statistics will record the length of time the simulation runs, the number of drones active in the system and it will record the occurrence of events. From these statistics we will be able to show that the simulation ran successfully for X amount of time, with Y number of drones and with Z number of events occurring. Using these statistics, we will be able to learn how to many drones can be active in the system at one time.

Criticality

This function is not a critical system, excluding it would not cause a system failure. Removing this function would not cause the overall system to fail but the purpose of the simulation is to create these statistics so it is critical to the output of the program. It is not critical to the running of the system.

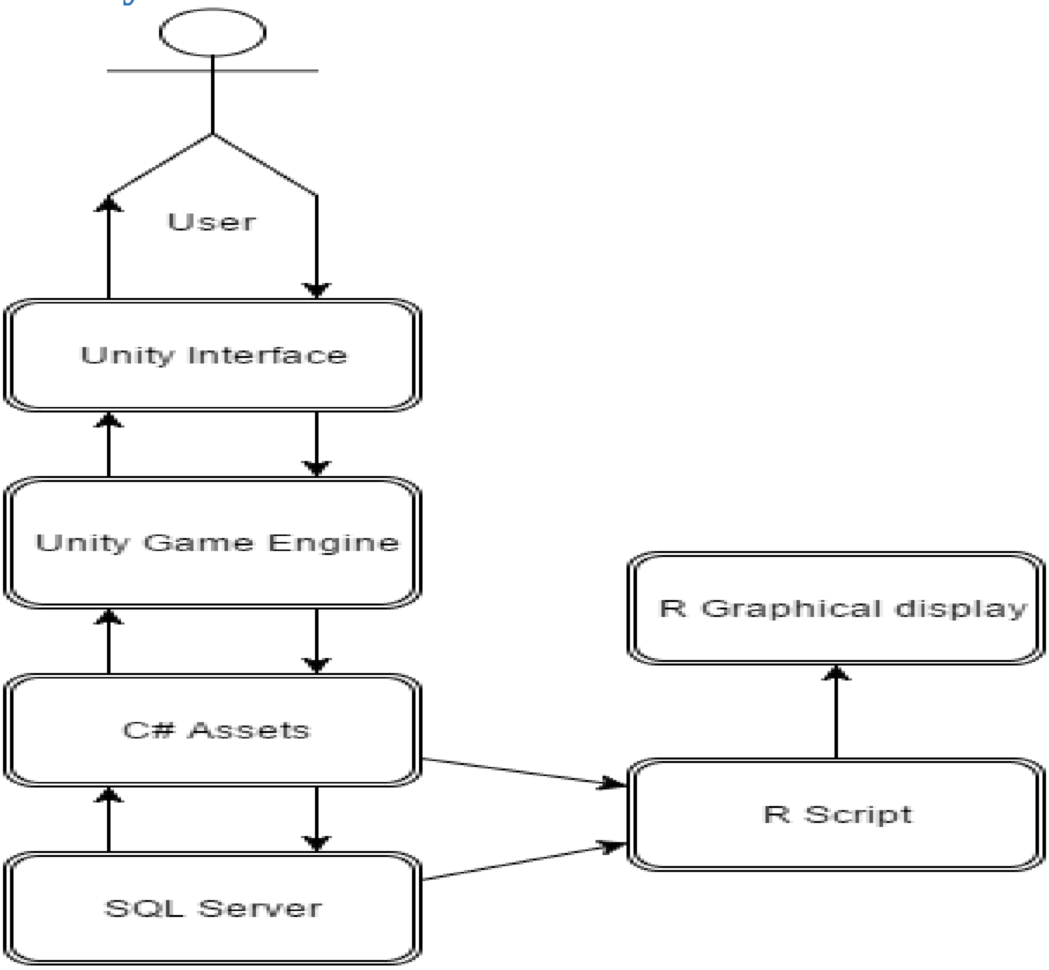
Technical Issues

The statistics for the simulation will be stored in a database table. Once the simulation finishes running a method will trigger an R script to read the data from the database and create graphical displays on the data. I will also perform statistical functions on the data.

Dependencies

In order to create statistics, the simulation must run. The statistics will require the simulation to be completed before I can run them. So, this function requires all other features to be implemented before I can begin working with the statistics.

4. System Architecture



Unity Interface

This will be the main interface of the simulation. This is where we will be able to see the drones interact and move around the world. The interface will be a combination of the landscape, which will be a 3D model and the drone objects.

Unity Game Engine

The Unity Game Engine will provide 3-Dimensional physics for the game. It also allows for the manipulation of the objects in the game. The environment is set up as a scene, in this scene the logic of how objects will act in the world will be coded. This section takes the logic from the C# assets then creates behaviour and physics to be displayed by the interface.

C# Assets

The C# Assets will be the C# end of the project. Here I will be able to edit some of the Unity generated files to change interaction with the world. This will also hold the logic for generating a drone, navigating through the map, generating an event and handling an event. The C# Assets will also handle the communication with the SQL Server.

SQL Server

The server will be a local server stored on my machine, setup using SQL express. The database will store all the information about the drones, the starting locations on the map, the list of events that can be triggered. I will also have a table to store the statistics created by the simulation.

R Script

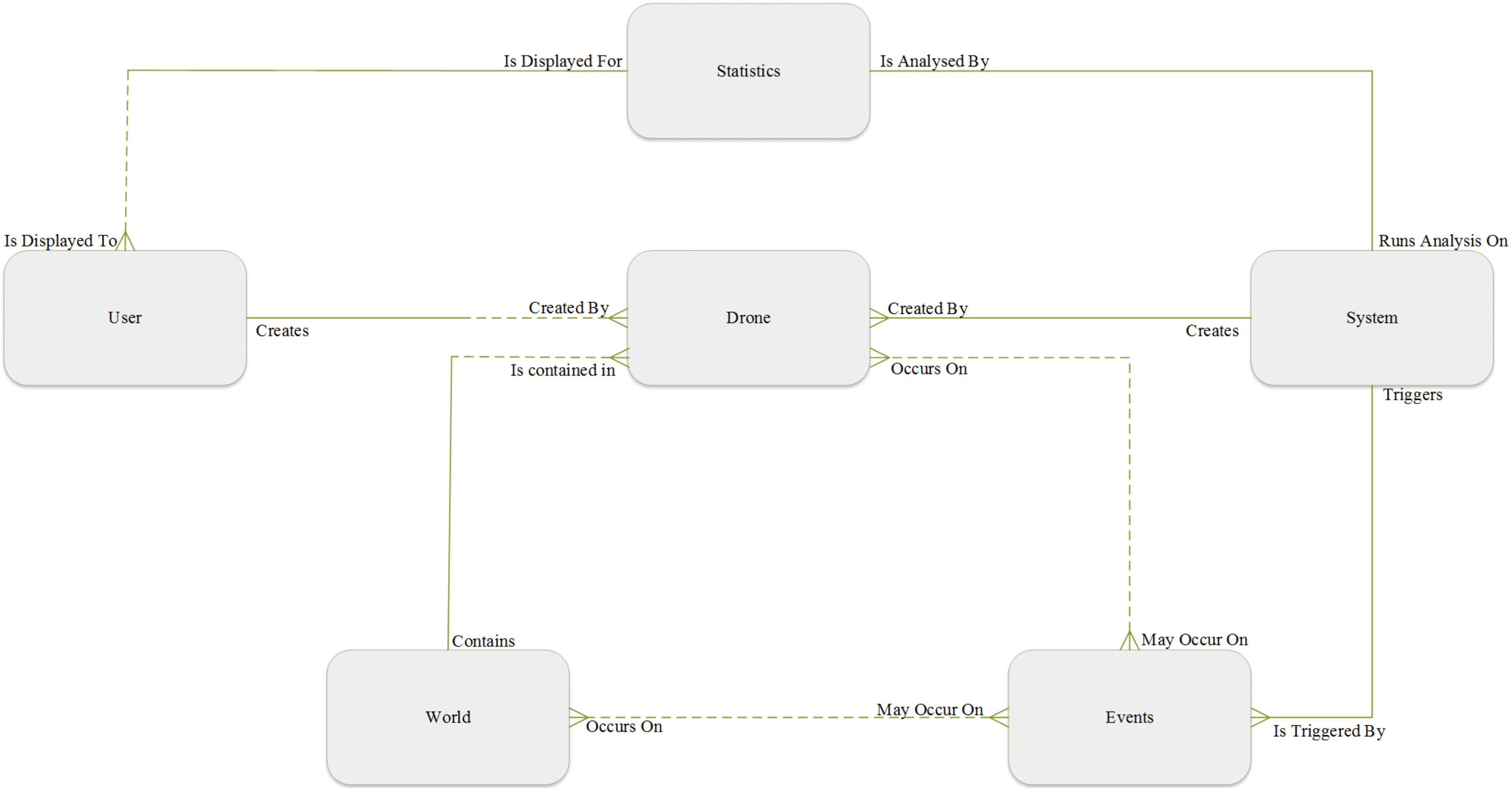
The R script will run once the simulation has finished, it will pull the data captured by the simulation from the server and run a set of statistical functions on it to display useful summarisation of the data. The script will also generate useful graphical display which will be passed to the graphical display window.

R Graphical Display

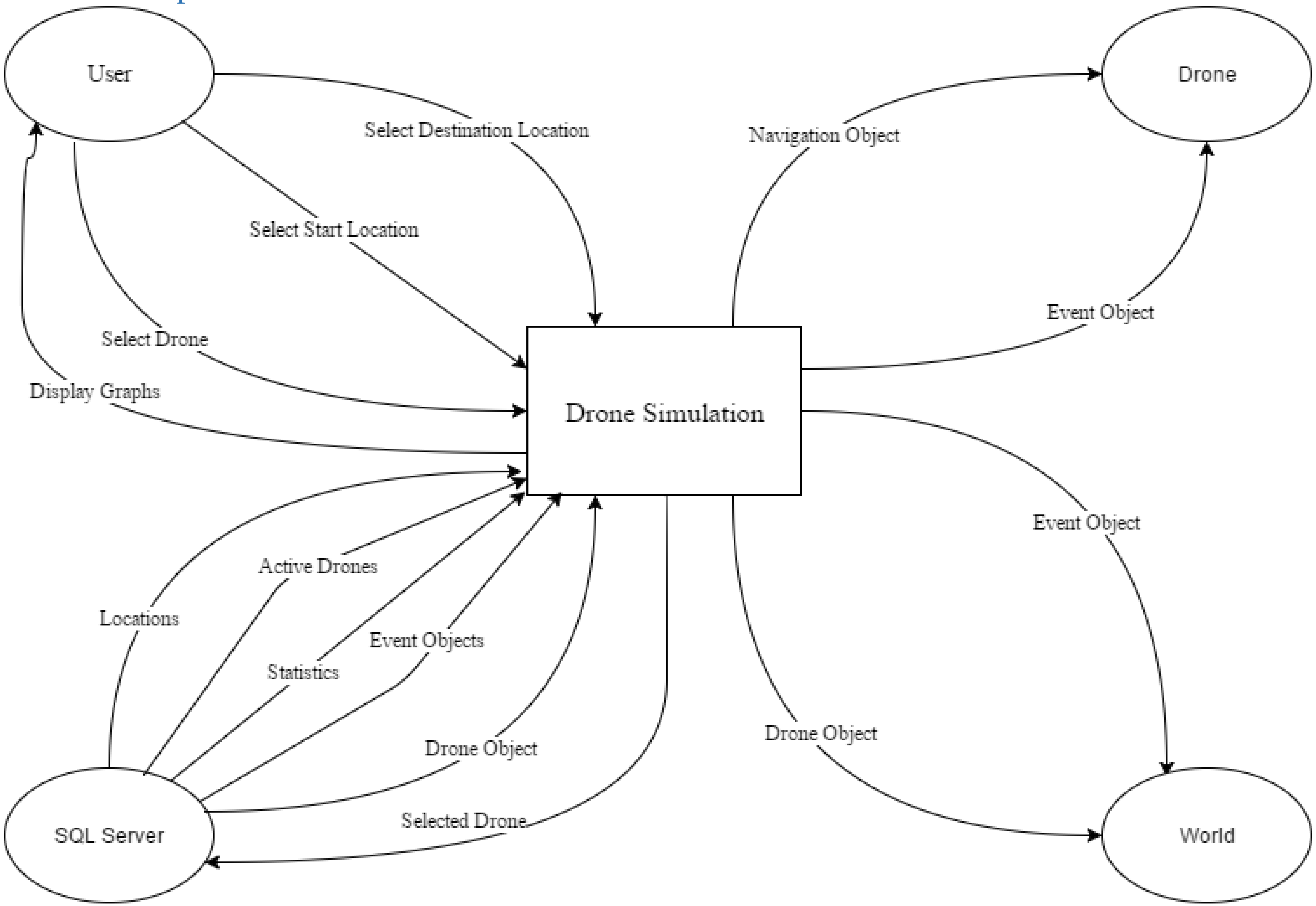
The graphical display will be generated from the statistical functions run by the R script. The graphical display window will be a series of different graphs and charts that will give a useful visualisation of the data captured in the simulation.

5. High Level Design

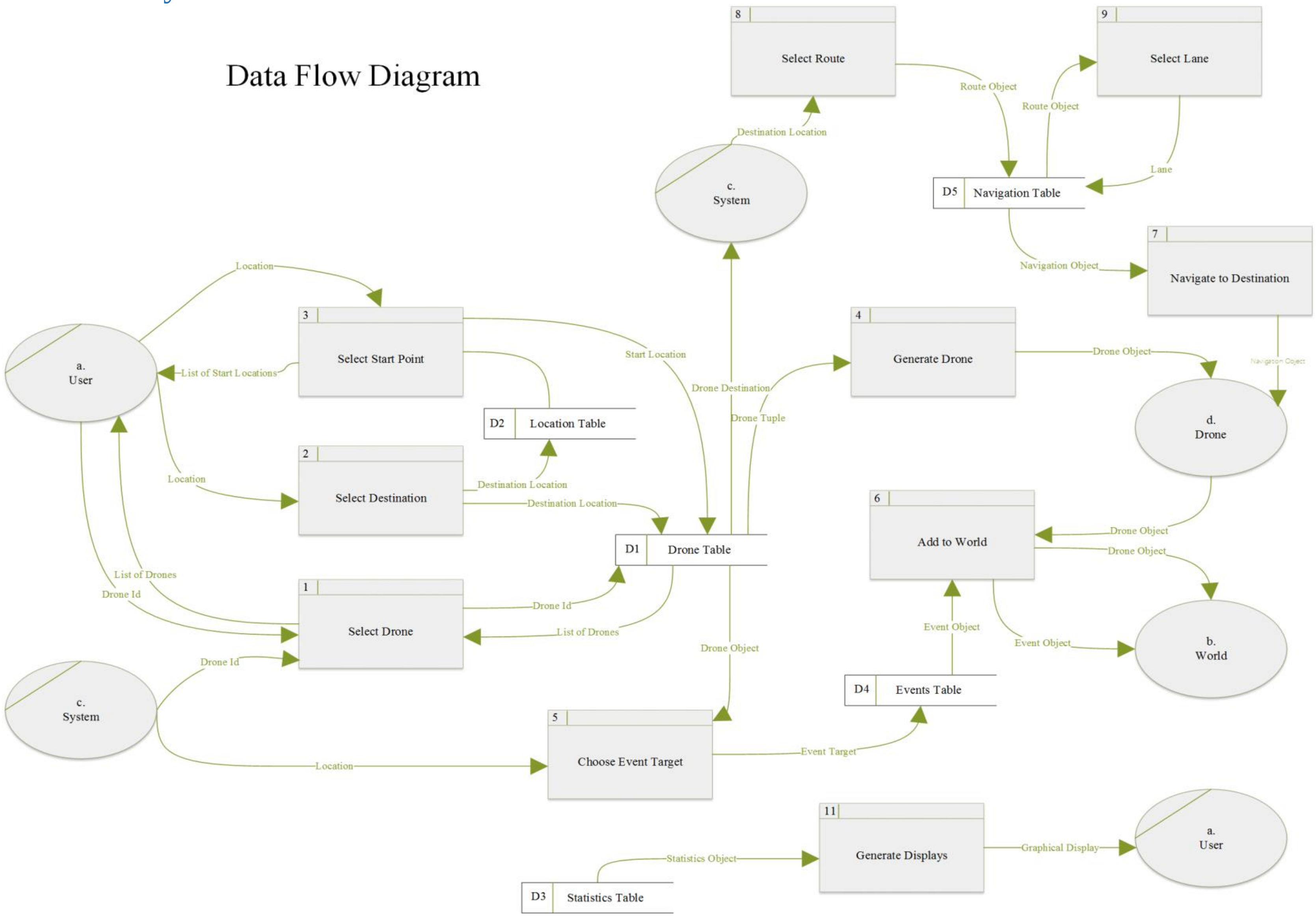
5.1 Logical Data Model



5.2 Top Level Context DFD

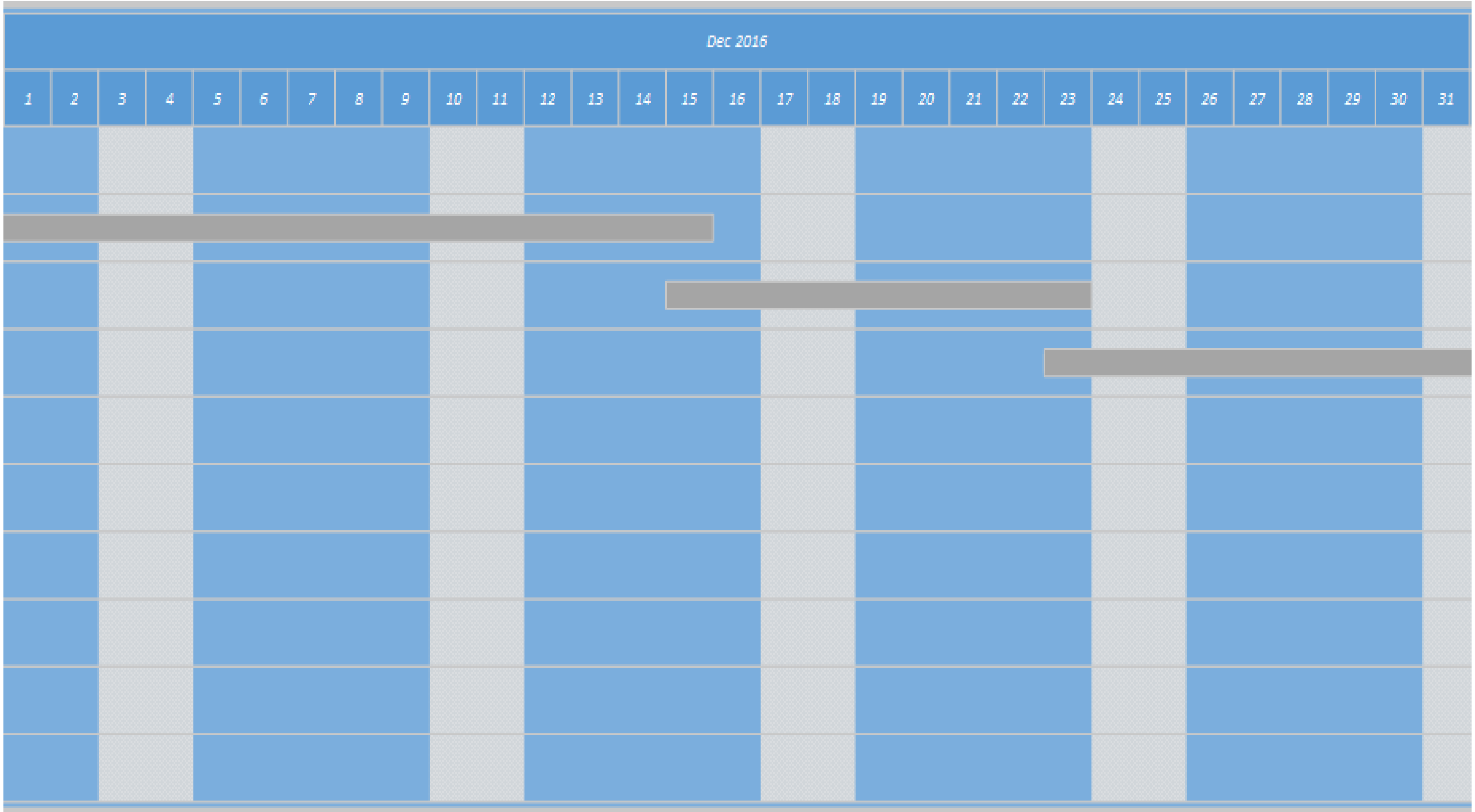


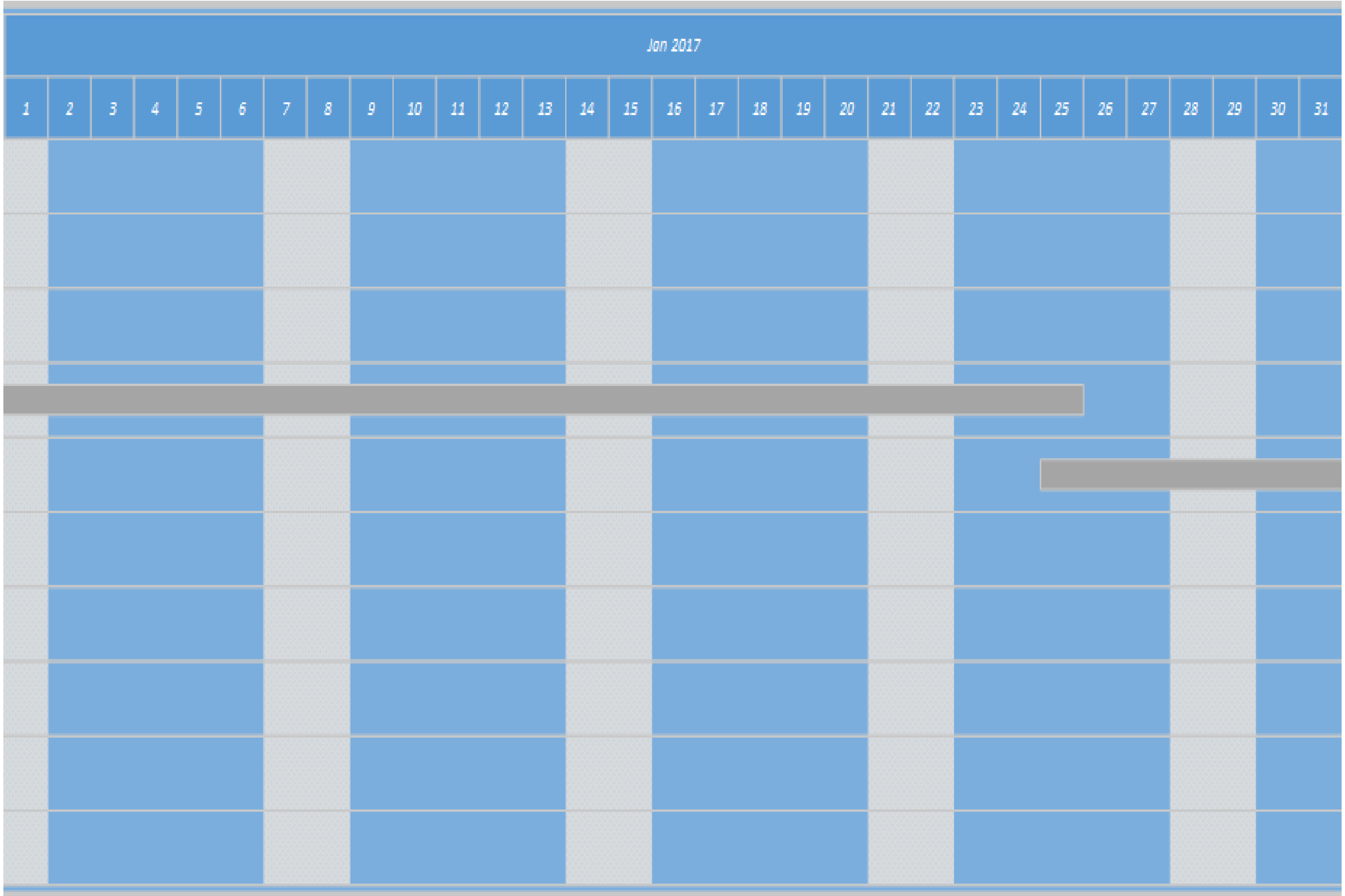
5.3 Physical Level One DFD

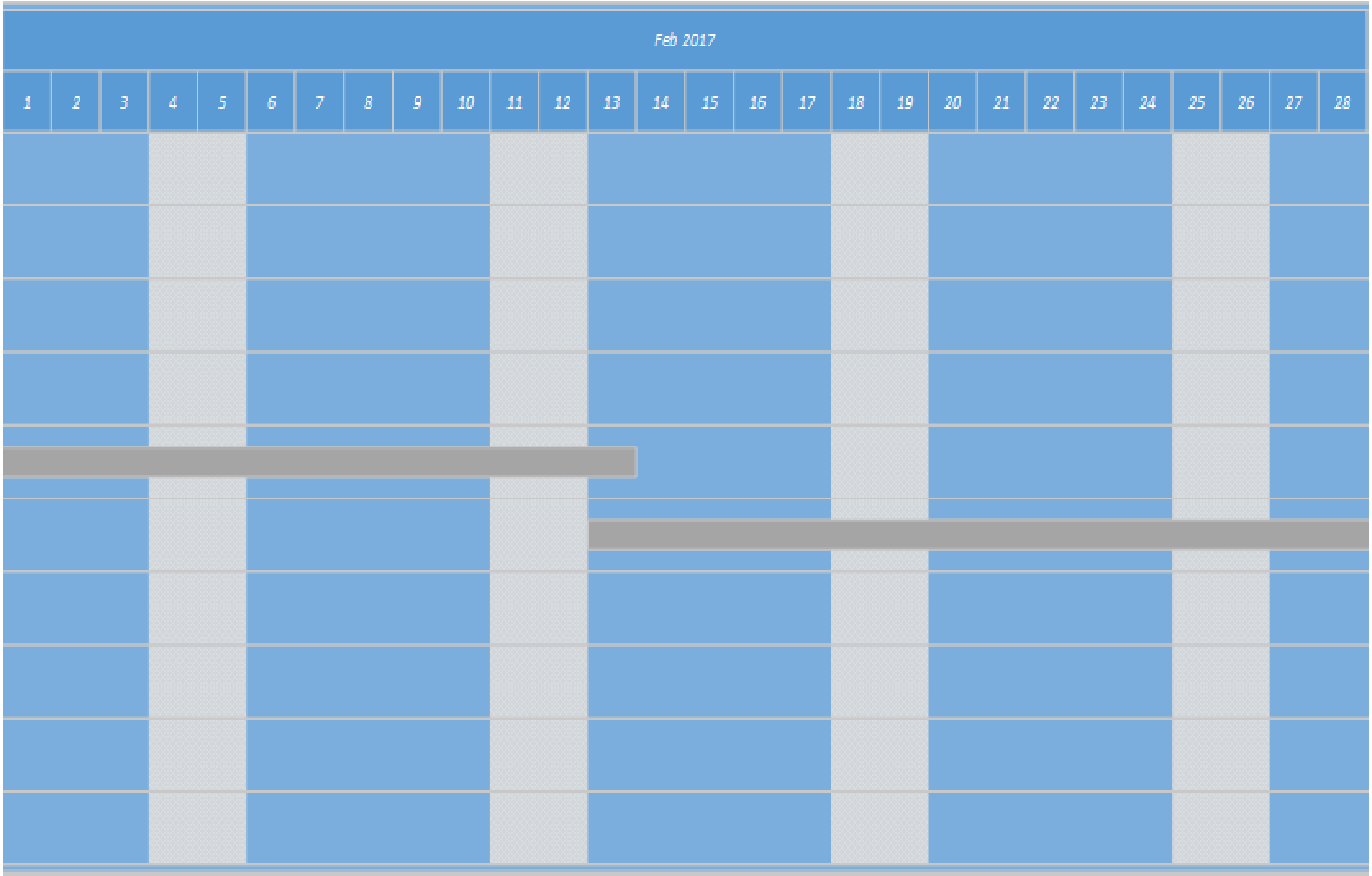


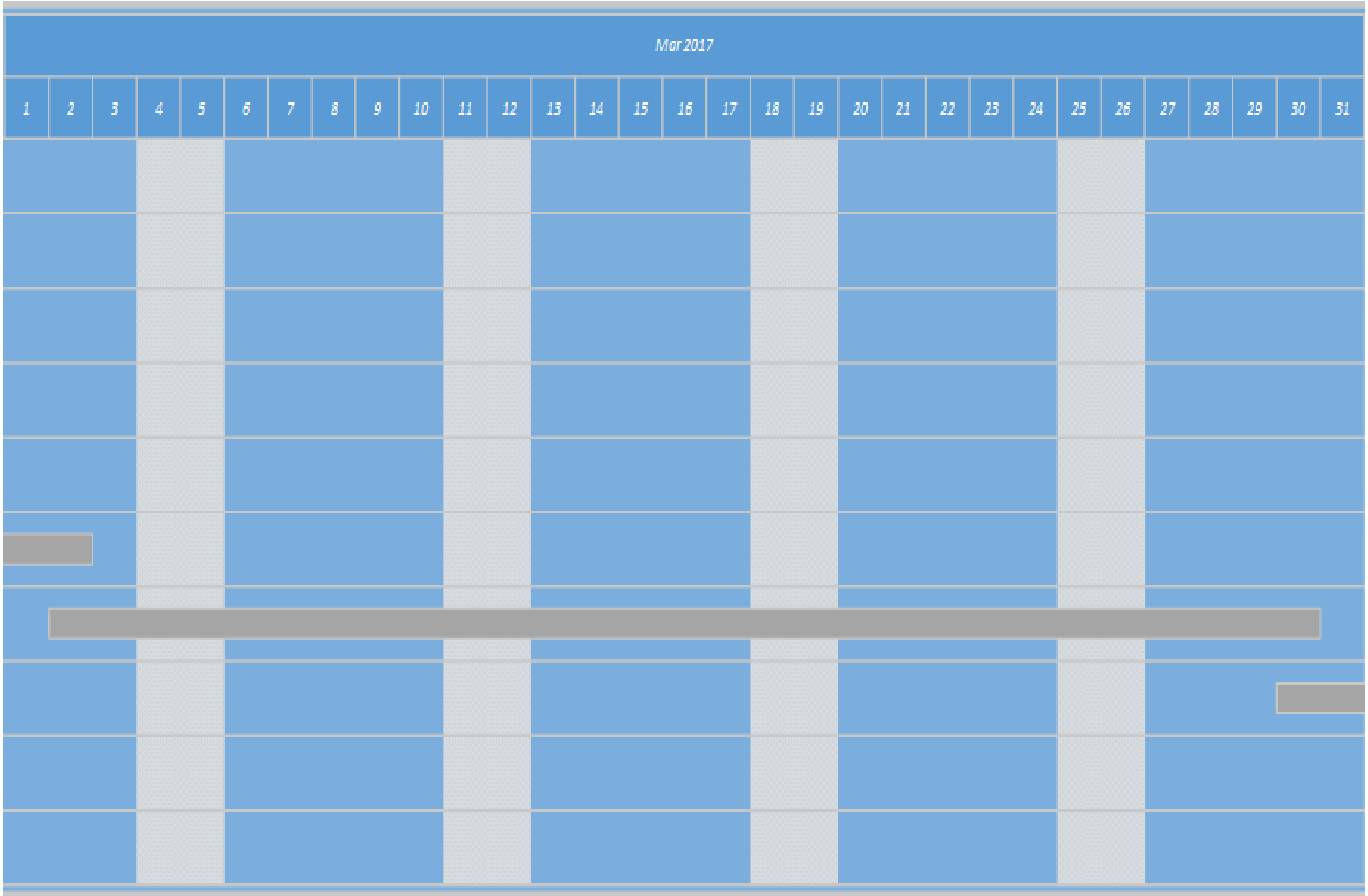
6. Preliminary Schedule

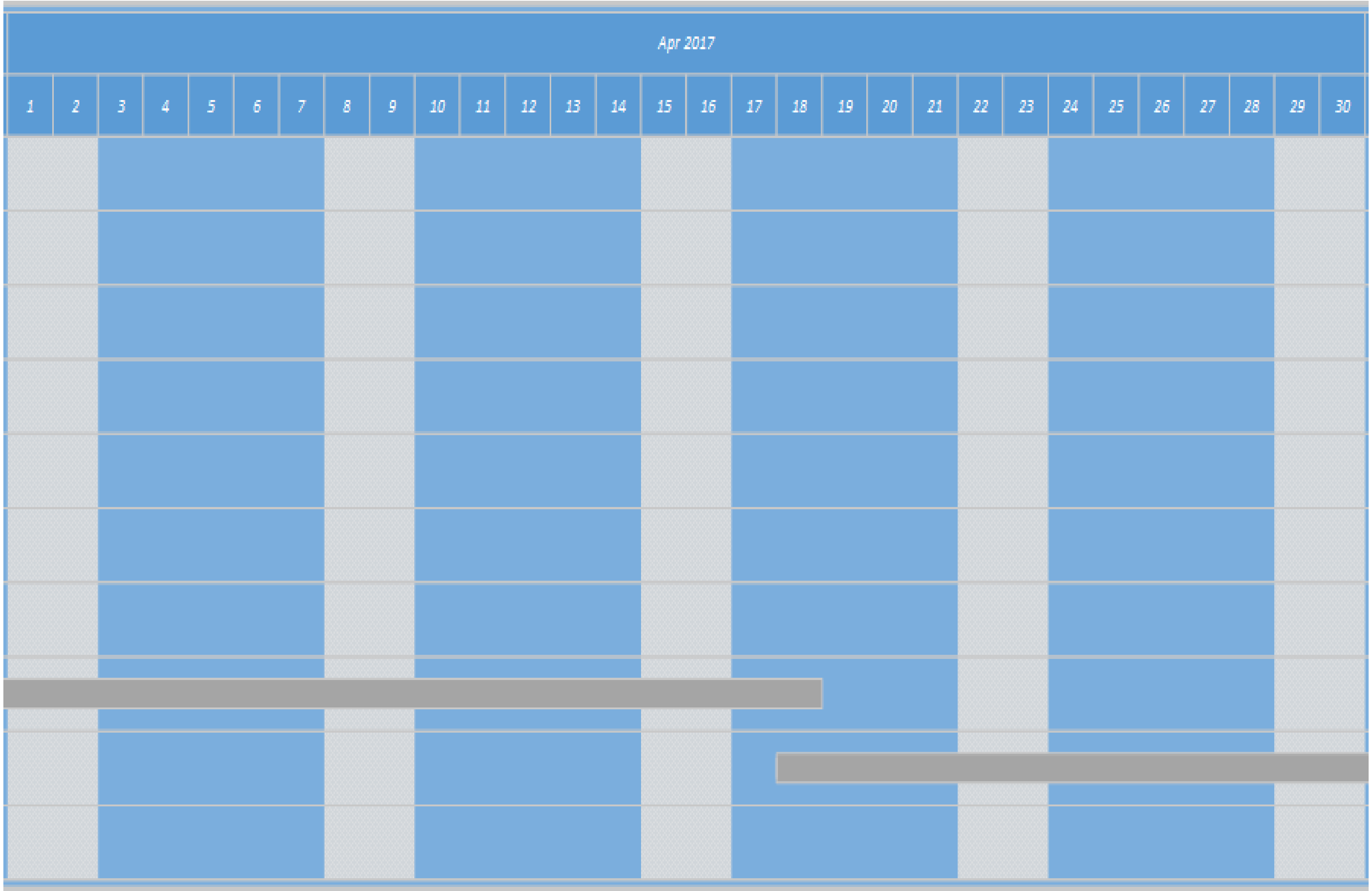
ID	Task Name	Start	Finish	Duration	Nov 2016					
					25	26	27	28	29	30
1	Import Cityscape Model	25/11/2016	28/11/2016	2d						
2	Create Basic Simulation	28/11/2016	15/12/2016	14d						
3	Add Drone To Simulation	15/12/2016	23/12/2016	7d						
4	Add Navigation	23/12/2016	25/01/2017	24d						
5	Add Events and Event Handling	25/01/2017	13/02/2017	14d						
6	Add More Drones to the System	13/02/2017	02/03/2017	14d						
7	Try and break the system	02/03/2017	30/03/2017	21d						
8	Gather Statistics	30/03/2017	18/04/2017	14d						
9	Generate R Functions and Graphics	18/04/2017	05/05/2017	14d						
10	Create Video Log and Finish Compile documentation	05/05/2017	22/05/2017	12d						

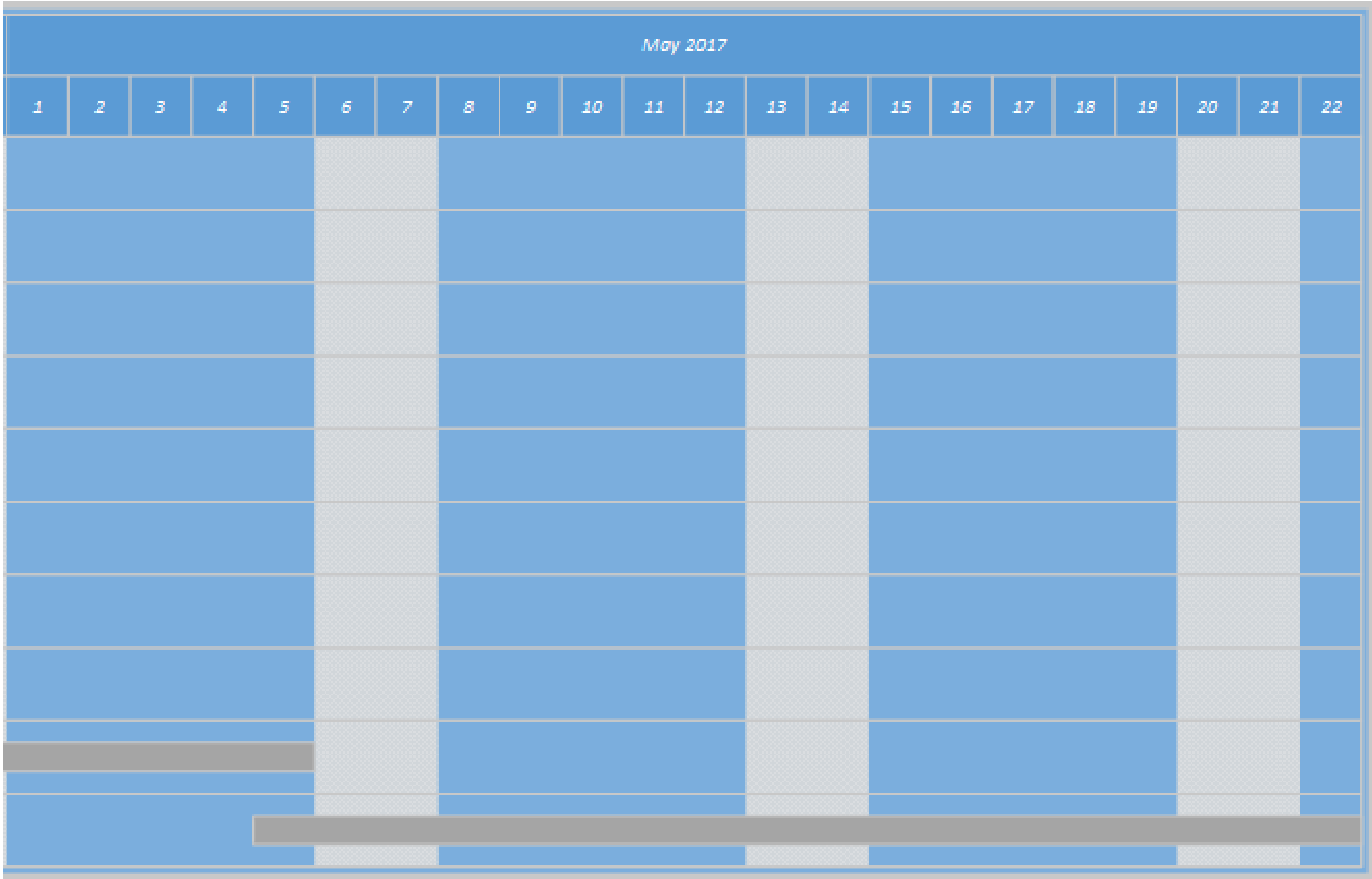

Dec 2016
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31


Jan 2017
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31


Feb 2017
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28


Mar 2017
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31


Apr 2017
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30


May 2017
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

7. Appendices

7.1 Blog Link

- https://gitlab.computing.dcu.ie/murpj238/2017-ca400-murpj238/blob/master/docs/blog/blog.md

7.2 Research References

- Irish Legislation for Drone Usage
    - https://www.iaa.ie/general-aviation/drones/drone-regulations-guidance
- 3D Models of Cityscapes
    - https://cadmapper.com/#metro
    - http://www.sketchup.com/#find-3d-models-anchor
- Article on a study of traffic threshold
    - http://www.sciencemag.org/news/2008/03/traffic-jams-happen-get-used-it
- Unity Tutorials
    - https://www.youtube.com/watch?v=Ep0rlBQRcVc&t=22s
    - https://www.youtube.com/watch?v=QUCEcAp3h28
    - https://unity3d.com/learn/tutorials/topics/interface-essentials