Research Proposal

Jake Morey

BSc(Hons) Computer Games Technology
University of Abertay Dundee
School of Arts, Media and Computer Games
May 2014

Abstract

Contents

Abstract **i**

List of Figures **ii**

List of Tables **ii**

1 Introduction **1**

1.1 Weather: 1

1.2 Research Question: 2

2 Literature **4**

2.1 Background: 4

2.2 Cloud Generation: 4

2.2.1 Artist Created: 5

2.2.2 Cellular Automata (CA): 5

2.2.3 Fluid Dynamics: 6

2.3 Cloud Rendering: 9

2.3.1 Single Scattering: 9

2.3.2 Multiple scattering: 10

2.3.3 Photon Mapping: 10

2.4 Rain Rendering: 10

3 Methodology: **12**

4 Results: **12**

5 Discussion: **12**

6 Conclusions: **12**

References **13**

Appendix **16**

List of Figures

Figure 1 Left: Ouranos! (1980), Right: Dear Ester (2012) 1
Figure 2 Left: rain (2013), Right: Heavy Rain (2010) 2
Figure 3 A screenshot from Tomb Raider (2013) 4
Figure 4 Harris *et al.* (2003) 16
Figure 5 Miyazaki *et al.* (2001) 16
Figure 6 Fedkiw, Stam and Jensen (2001) 17
Figure 7 Harris *et al.* (2003) 17
Figure 8 Project Plan 18

List of Tables

1 Introduction

1.1 Weather:

Since the early 1980's a major part of games have been there use of weather. The weather used in games generally has one of two different uses. The first use of weather in games is using the weather to affect gameplay such as Ouranos! (1980) which uses it as the main game mechanism . More recently Dear Ester (2012) used the weather for the second reason which is as a way to add to the immersion the player feels when playing the game. Figure 1 contains screenshots from both games and showcases how far the weather has come in games. There are some games that utilise both types of weather in games, including Microsoft Flight Simulator X (2003).

Figure 1: Left: Ouranos! (1980), Right: Dear Ester (2012)

Looking closer at the types of weather used in games the use of clouds and rain stand out as the most recurring and notable aspects of weather as well as the most versatile examples of its use in games. For example rain plays different roles in two games, Heavy Rain (2010) and rain (2013). However, without it, the games would not be as compelling as the currently are. Figure 2 shows screenshots from both games. rain (2013) shows how rain is used to affect gameplay by allowing the player only to see the character in the rain, while Heavy Rain (2010) shows how the use of rain adds a depth that increases the film noir feel of the game.

When creating these games a number of different techniques were used to create and render the weather. Ouranos! (1980) uses ASCII because of the limitations

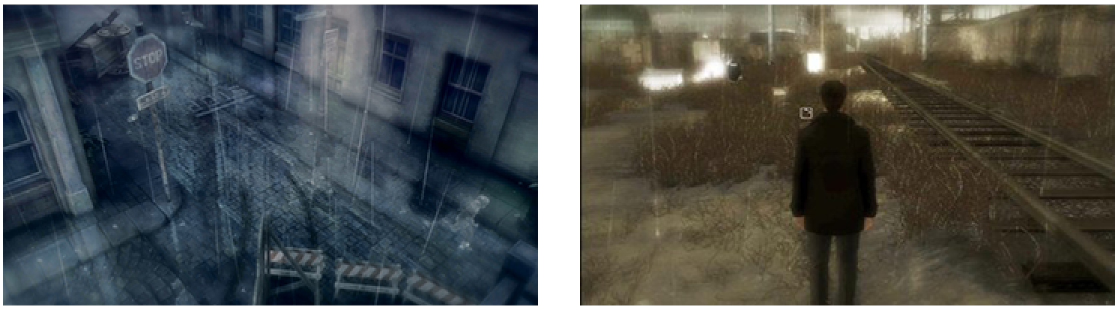

Figure 2: Left: rain (2013), Right: Heavy Rain (2010)

at the time. Games such as Super Mario Bros (1985) use 2D sprites to render clouds on to the background and games like Tomb Raider (2013) uses 3D scripted clouds. 2D games are also more likely to use 2D scrolling rain texture while 3D games are more likely to use a particle system to generate the rain. Some games can use location data to simulate the correct weather at a given location such as NCAA Football 14 (2013). These games usually have a backup dynamic weather system which will be used if the game can't connect to the internet to collect data. Most games created use artistic representation of clouds instead of creating them in real time using equations. Fluid dynamic equations can be used to represent the movement of any object made up of liquid or gas. Basic clouds can be modelled by fluid dynamic equations, more advance clouds need extra equations for water continuity, thermodynamics, and buoyancy. Using equations to generate the clouds will mean that the cloud will behave more like a real cloud than an artist's interpretation of a cloud. When creating the cloud using these equations rain generation can be based proportionally on cloud size or by adding an extra equation to water continuity equations so rainfall is included.

1.2 Research Question:

The project aims to create realistically moving and looking clouds in real time using fluid dynamic equations. Another aim is to create rain in locations and an appropriate amount that relates to the clouds created. The last aim is to compute the equations in the GPU so that the equations can be computed efficiently. These aims lead to the research question:

How can the amount of Precipitation generated in a game be related to realistic simulated clouds generated in real-time using fluid dynamic equations?

Answering the research question results in a number of objectives the need to be completed including using a number of equations, mainly fluid dynamic equations in 3D, space to generate and move clouds. To use these clouds to generate the rain in a 3D scene, the fluid dynamic equations will need to be optimized to run as smoothly as possible in the application whilst producing realistic, or at least plausible, effects.. The final application will be analysed visually and numerically to test how the clouds grow and move over time, as well as generating rain in the most efficient way.

2 Literature

2.1 Background:

This section will look at how clouds and rain are generated currently in games. Weather plays a huge impact to the how a game feels, as Barton (2008) wrote about how "It was a dark and stormy night" not only sets the time and weather of the scene but also sets the tone. Wang (2004) agrees by saying that one of the most fascinating parts of a scene could be the clouds. An example of this is the game Tomb Raider (2013) which at numerous points in the game the user can see a vast sky full of clouds as seen in Figure 3.

Figure 3: A screenshot from Tomb Raider (2013)

2.2 Cloud Generation:

There are a number of different methods for generating clouds from cellular automata, to fluid dynamic equations, to importing 3D objects.

2.2.1 Artist Created:

This technique works not by creating the clouds at run time but instead creates the clouds as models and loads them into the game when needed. Wang (2004) describes a version of this which allows artists to create boxes in 3DS Max in which a plug-in will then generate clouds inside the box. She explained how this method was used when creating Microsoft Flight Simulator 2004: A Century of Flight (2003). A similar method can be used in the CryEngine 3 SDK (2013) which allows the user to alter the properties of the clouds created in real-time within the editor. This method can create very realistic looking clouds but doesn't allow for realistic movement or generation.

2.2.2 Cellular Automata (CA):

A Cellular Automaton can be described as a regular shaped structure which consists of identical cells that are computed synchronously depending on the state of the cell and its neighbours (Dantchev, 2011). Dobashi *et al.* (2000) used a cellular automaton model when generating the clouds which involved giving each cell a number of boolean states that, when coupled with the rules generated clouds.

This method was extended by Miyazaki *et al.* (2001) who used the Coupled Map Lattice (CML) method which is described as "an extension of cellular automaton, and the simulation space is subdivided into lattices". Miyazaki *et al.* (2001) also goes on to explain that the CML model differs from the CA model by using continuous values instead of discreet values. This CML model uses very simple equations for viscosity and pressure effects, advection, diffusion of water vapour, thermal diffusion and buoyancy, and the transition from vapour to water.

Cellular Automaton gives a lot more control to the physics of clouds because of the equations used to define them compared to clouds created by artists. However these equations are not as accurate as using fluid dynamic equations to move and generate clouds.

2.2.3 Fluid Dynamics:

As clouds can be described as an incompressible fluid it can be simulated via the fluid dynamic equations. The Navier-Stokes Equations are used for a "fluid that conserves both mass and momentum." (Stam, 1999). In the Navier-Stokes equations ρ is the density, $\mathbf{f}$ represents all external forces and ν is the kinematic viscosity of the fluid. The velocity and pressure are defined as $\mathbf{u}$ and p respectively. The second equation is the continuity equation which means the fluid is incompressible.

$$\frac{\partial \mathbf{u}}{\partial t} = -\left(\mathbf{u} \cdot \nabla\right)\mathbf{u} - \frac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{u} + \mathbf{f} \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2}$$

The Navier-Stokes equations (1) and (2) can be simplified to Euler's Equation when "the effects of viscosity are negligible in gases" (Fedkiw, Stam and Jensen, 2001). This makes the equations for generating the clouds less computationally heavy and can be shown in equation (3) which has no $\nu\nabla^2\mathbf{u}$. The continuity equation has not changed and can be seen from (4) being the same as (2).

$$\frac{\partial \mathbf{u}}{\partial t} = -\left(\mathbf{u} \cdot \nabla\right)\mathbf{u} - \frac{\nabla p}{\rho} + \mathbf{f} \tag{3}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{4}$$

Fedkiw, Stam and Jensen (2001), Harris *et al.* (2003), and Overby, Melek and Keyser (2002) all used work created by Stam (1999) on stable fluid simulations. Overby, Melek and Keyser (2002) used the actual solver created by Stam (1999) in the application to solve the fluid dynamic part of creating clouds. Whereas Fedkiw, Stam and Jensen (2001) and Harris *et al.* (2003) used the theory in the creation of the smoke and clouds respectively. Even though all three used the same start for simulating cloud generation they have different methods for assigning values to the other equations needed.

The Fedkiw, Stam and Jensen (2001) model uses a Poisson equation to compute the pressure of the system and two scalar functions for advecting the temperature and density. This model also uses a function built up of the temperature, ambient temperature, density, and two other positive constants to create a buoyancy effect. The model also simulates velocity fields, which are dampened out on the coarse grid, by finding where the feature should be and then creating a realistic turbulent effect.

Overby, Melek and Keyser (2002) computes the local temperature based upon the heat energy and the pressure. The pressure is calculated from ground level to the tropopause by an exponentially decreasing value (Overby, Melek and Keyser, 2002). The buoyancy in this model is created using the local temperature, the surrounding temperature and a buoyancy scalar. Relative humidity is calculated based upon current water vapour and saturated water vapour. Water condensation is then calculated based upon relative humidity, hygroscopic nuclei, and a condensation constants. The final equation to be calculated is the latent heat which is calculated by the water condensation and a constant.

The Harris *et al.* (2003) model uses equations for water continuity, thermodynamics, buoyancy, and a Poisson equation for fluid flow. This model also creates velocity fields using the same process as Fedkiw, Stam and Jensen (2001). This model uses more complicated equations than the previous two models to more accurately simulate the creation of clouds. For example this model uses gravity, the mass mixing ratio of hydrometeors and virtual potential temperatures, whereas the previous models use scalars or other constants with the temperature to create the buoyancy force.

The vorticity confinement as defined by Harris *et al.* (2003) model is defined in equation (5). $\boldsymbol{\omega}$ is the vorticity, defined by $\boldsymbol{\omega} = \nabla \times \mathbf{u}$, and $\mathbf{N}$ is the normalized vorticty vector field and points from areas of lower vorticiy to areas of higher vorticity which is defined by equation (6). With h is the grid scale and ϵ is a scale parameter.

$$\mathbf{f}_{vc} = \epsilon h \left(\mathbf{N} \times \boldsymbol{\omega} \right) \tag{5}$$

$$\mathbf{N} = \frac{\nabla \left| \boldsymbol{\omega} \right|}{\left| \nabla \left| \boldsymbol{\omega} \right| \right|} \tag{6}$$

The buoyant force is defined in equation (7) where g is the acceleration due to gravity. q_v is the mixing ratio of hydrometeors and in this case is defined as q_c, the mixing ratio of liquid water. θ_v is the virtual potential temperature and is defined in equation $\theta_v \approx \theta(1 + 0.61q_v)$. θ_{v0} is the reference potential temperature and is between 290 and 300K as defined by Harris *et al.* (2003).

$$B = g \left(\frac{\theta_v}{\theta_{v0}} - q_h \right) \tag{7}$$

The water continuity in Harris *et al.* (2003) model is based upon the Bulk Water Continuity model which described by Houze (1994) as “the simplest type of cloud is a warm non-precipitating cloud”. Houze (1994) describes the model as a set of categories in which clouds can be created. The categories for this simple type of cloud model are vapour q_v and cloud liquid water q_c and are described in equations (8) by Houze (1994). C is the condensation rate.

$$\frac{Dq_v}{Dt} = -C, \frac{Dq_c}{Dt} = C \tag{8}$$

The thermodynamic equation defined in Harris *et al.* (2003) model can be seen in equation (9). c_p is the specific heat capacity of dry air at constant pressure in this case $1005Jkg^{-1}K^{-1}$. L is the latent heat of vaporization of water which is $2.501Jkg^{-1}$. The part of the equation in brackets on the right hand side of the equation can be exchanged for the condensation rate from the water continuity equations.

$$\frac{\partial \theta}{\partial t} + \left(\mathbf{u} \cdot \nabla \right) \theta = \frac{-L}{c_p \Pi} \left(\frac{\partial q_v}{\partial t} + \left(\mathbf{u} \cdot \nabla q_v \right) \right) \tag{9}$$

Π is called the Exner function and is defined in equation 10. Where p_0 is the pressure at the surface, usually taken as $1000hPa$; R_d is the gas constant for dry air a can be taken as $287Jkg^{-1}K^{-1}$; c_p is the heat capacity of dry air at constant pressure, and p is pressure.

$$\Pi = \left(\frac{p}{p_0}\right)^{\frac{R_d}{c_p}} \tag{10}$$

With these extra equations using fluid dynamics for generating and moving clouds will give more accurate simulations compared to the previously mentioned processes. There is a draw back for using fluid dynamic equations as it will use more computing power compared to artistically created, and Cellular Automaton methods.

2.3 Cloud Rendering:

Due to the nature of clouds when light passes through them it becomes scattered. The majority of the models looked at the use two different techniques to accomplish this effect: single scattering and multiple scattering. These models may render clouds using these scattering techniques directly or may use scattering inside other rendering processes such as photon mapping. A number of these models also use billboards or imposters when rendering the clouds as this saves on computation. This section will look at single scattering, multiple scattering, and photon mapping.

2.3.1 Single Scattering:

Harris and Lastra (2001) describe single scattering as a model that simulates scattering in a single direction that is usually the direction leading to the point of view. There are debates concerning whether or not this type of rendering is detailed enough for rendering clouds. Miyazaki *et al.* (2001) states the main topic of his model is the cloud shapes so using single scattering is enough to check the shape of the cloud. Bohren (1987) describes single scattering as insufficient when describing common observations. Due to this being a simpler and less computational heavy method of rendering clouds compared to other methods this process could be best for cloud generated using a number of complex equations.

2.3.2 Multiple scattering:

"Multiple scattering models are more physically accurate, but must account for scattering in all directions ... and therefore are much more complicated and expensive to evaluate" (Harris and Lastra, 2001). Harris *et al.* (2003) uses a version of multiple scattering which is called multiple forward scattering, this differs from the original by instead of calculating scattering in all directions it calculates scattering in the forward direction only. This means the algorithm is less computationally heavy. Fedkiw, Stam and Jensen (2001) describe the multiple scattering of light as necessary for objects made from water vapour, which clouds are.

2.3.3 Photon Mapping:

"Photon mapping is a variation of pure Monte Carlo ray tracing in which photons (particles of radiant energy) are traced through a scene" (Jensen (1996), cited in Harris (2003)). Harris (2003) describes the process of photon mapping as storing position, incoming direction, and radiance of each photon landing on a nonspecular surface that has been traced from the light source. Fedkiw, Stam and Jensen (2001) use photon mapping when rendering smoke and describe the process as a two pass algorithm, one where a volume photon map is built and the second as a rendering pass using a forward ray marching algorithm.

2.4 Rain Rendering:

"Rain is an extremely complex natural atmospheric phenomenon" (Puig-Centelles, Ripolles and Chover, 2009). Puig-Centelles, Ripolles and Chover (2009) describe two main techniques for rendering rain to a scene scrolling textures where a texture the size of the screen scrolls in the direction of the rain, and a particle system where each rain drop is represented as a particle in the system. Tariq (2007) writes “animating rain using a particle system is more useful for realistic looking rain with lots of behaviour (like changing wind).” Puig-Centelles, Ripolles and Chover (2009) sates that texture scrolling “is faster than particle systems, but it does not allow interaction between rain and the environment.”

An extension to the Bulk Water Continuity model which was described in section 2.2.3 allows a warm precipitating cloud with rain as an additional category, Houze (1994). Now instead of two equations there are three which are shown in equations 11 - 13.

$$\frac{dq_v}{dt} = -C + E_c + E_r \tag{11}$$

$$\frac{dq_c}{dt} = C - A - K - E_c \tag{12}$$

$$\frac{dq_r}{dt} = A + K + F - E_r \tag{13}$$

In the above three equations C represents the condensation of vapour into cloud water. A is the autoconversion which is the rate cloud water decreases as particles grow to raining size. E_c and E_r are evaporation variables, the former of cloud water and the latter is evaporation of rainwater. K is the collection of cloud water and F represents the rain fallout of the model. Adding the third equation and the extra variables to the water continuity from section 2.2.3 will allow for more realistic rain generation with a particle system.

3 Methodology:

- choice of practical and evaluation methods
- ethical considerations
- pseudo code

4 Results:

- Visual Test with screen shots at different time to show cloud and rain progression
- Table showing frame rates, cpu/gpu utilisation, memory usage
- talk about any problem with applications and results

5 Discussion:

- talk about the results
- how it affected the topic area
- talk about how aims and RQ was answered

6 Conclusions:

- summaries findings
- clarify questions relating with previous work
- how the project could be improved or extended

References

Barton, M. 2008. How's the weather: Simulating weather in virtual environments 8. Available from: http://gamestudies.org/0801/articles/barton. 2.1

Bohren, C. F. 1987. Multiple scattering of light and some of its observable consequences. *Am.J.Phys* 55(6). pp. 524–533. 2.3.1

Crytek 3 SDK. 2013. *CryEngine* [software] Version 3. Available from: http://mycryengine.com/. 2.2.1

Dantchev, S. 2011. Dynamic neighbourhood cellular automata. *Computer journal* 54(1). pp. 26–30. 2.2.2

Dear Esther. 2012. [Online]. PC. The Chinese Room. (document), 1.1, 1

Dobashi, Y. et al. 2000. A simple, efficient method for realistic animation of clouds. ACM Press/Addison-Wesley Publishing Co. pp. 19–28. 2.2.2

Elek, O. et al. 2012. Interactive cloud rendering using temporally coherent photon mapping. *COMPUTERS and GRAPHICS-UK* 36(8). pp. 1109–1118.

Fedkiw, R., Stam, J., and Jensen, H. 2001. Visual simulation of smoke. ACM. pp. 15–22. (document), 2.2.3, 2.2.3, 2.3.2, 2.3.3, 6

Harris, M. et al. 2003. Simulation of cloud dynamics on graphics hardware. Eurographics Association. pp. 92–101. (document), 2.2.3, 2.2.3, 2.2.3, 2.2.3, 2.3.2, 4, 7

Harris, M. J. 2003. *Real-time cloud simulation and rendering.*. [Online]. 2.3.3

Harris, M. and Lastra, A. 2001. Real-time cloud rendering. *Computer Graphics Forum* 20(3). pp. C76–C76. 2.3.1, 2.3.2

Heavy Rain. 2010. [Disk]. PlayStation 3. Sony Computer Entertainment. (document), 1.1, 2

Houze, R. 1994. *Cloud Dynamics.* International Geophysics. Elsevier Science. 2.2.3, 2.4

Jensen, H. W. 1996. *Global illumination using photon maps.* Springer. pp. 21–30. Rendering Techniques 96. 2.3.3

Microsoft Flight Simulator 2004: A Century of Flight. 2003. [Disk]. PC. Microsoft. 1.1, 2.2.1

Miyazaki, R. et al. 2001. A method for modeling clouds based on atmospheric fluid dynamics. In: *Computer Graphics and Applications, 2001. Proceedings. Ninth Pacific Conference on.* IEEE. pp. 363–372. (document), 2.2.2, 2.3.1, 5

NCAA Football 14. 2013. [Disk]. PlayStation 3. EA Sports. 1.1

Ouranos! 1980. [Cassest]. Commodore PET. The Code Works. (document), 1.1, 1, 1.1

Overby, D., Melek, Z., and Keyser, J. 2002. Interactive physically-based cloud simulation. In: *Computer Graphics and Applications, 2002. Proceedings. 10th Pacific Conference on.* IEEE. pp. 469–470. 2.2.3

Puig-Centelles, A., Ripolles, O., and Chover, M. 2009. Creation and control of rain in virtual environments. *VISUAL COMPUTER* 25(11). pp. 1037–1052. 2.4

rain. 2013. [Online]. PlayStation 3. Sony Computer Entertainment. (document), 1.1, 2

Stam, J. 1999. Stable fluids. ACM Press/Addison-Wesley Publishing Co. pp. 121–128. 2.2.3, 2.2.3

Super Mario Bros. 1985. [Cartridge]. Super Nintendo Entertainment System. Nintendo. 1.1

Tariq, S. 2007. Rain. Tech. rep.. NVIDIA. 2.4

Tomb Raider. 2013. [Online]. PC. Square Enix. (document), 1.1, 2.1, 3

Wang, N. 2004. *Let there be clouds! fast, realistic cloud-rendering in microsoft flight simulator 2004: A century of flight*. [Online]. Available from: http://www.gamasutra.com/view/feature/130434/let_there_be_clouds_fast_.php. 2.1, 2.2.1

Appendix:

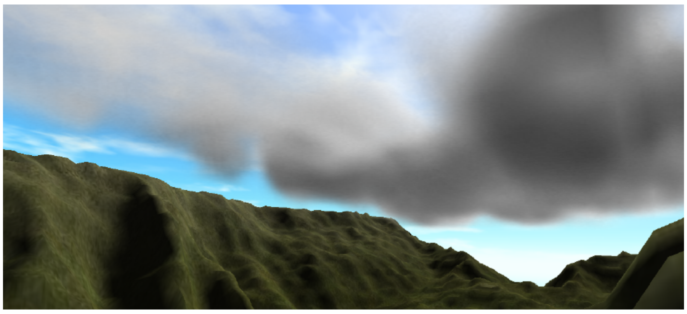

Figure 4: Harris *et al.* (2003)

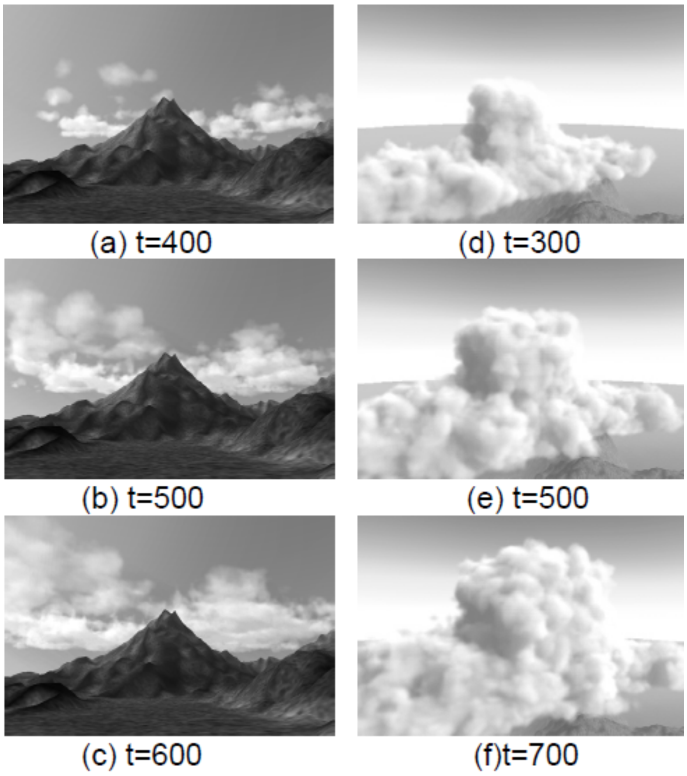

Figure 5: Miyazaki *et al.* (2001)

Figure 6: Fedkiw, Stam and Jensen (2001)

Figure 7: Harris *et al.* (2003)

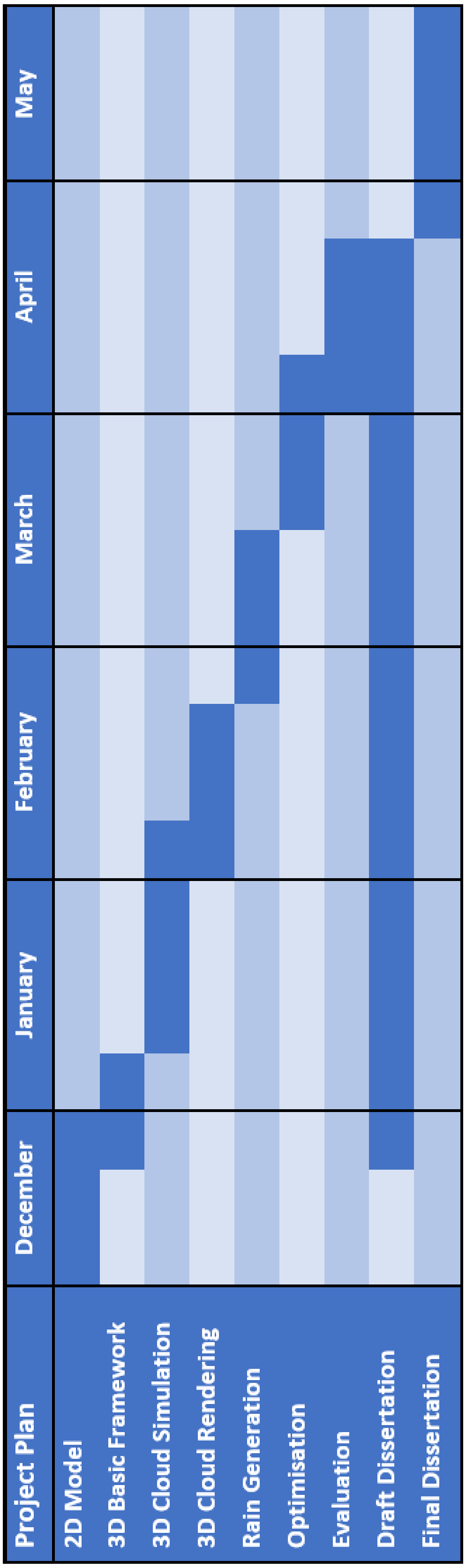

Figure 8: Project Plan