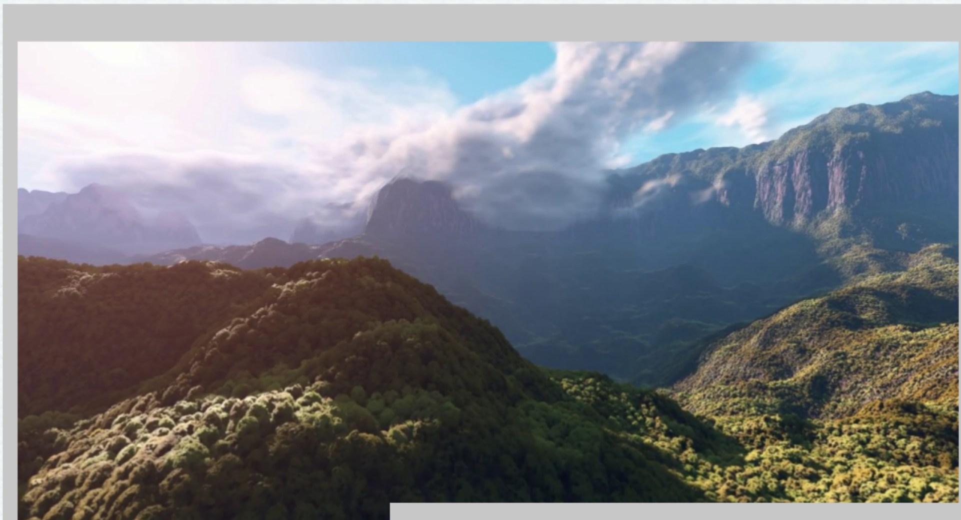
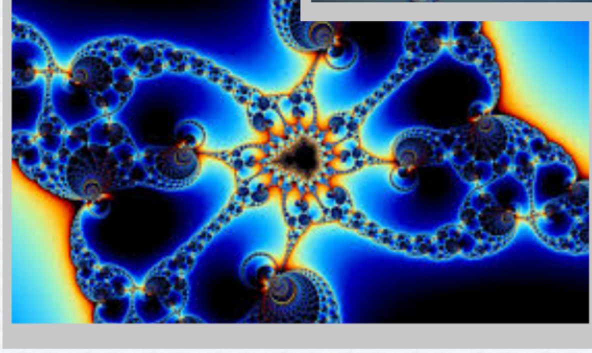
The Design and Implementation of an Advanced Graphics Screensaver Suite

By James Balajan





Contents

List of Figures

Chapter 1

Project Proposal

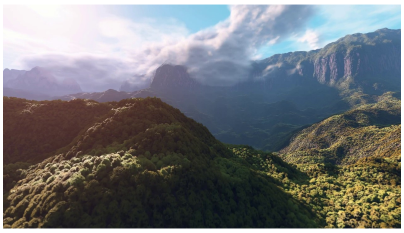


Figure 1.1: *Inspiration for Screensaver. Computer generated Rainforest by Inigo Quilez*

1.1 Executive Summary

Much promise has been shown in the applicability of basic, computer generated graphics for creating screen savers for computers in the past, an example of this being that all of the screensavers for Windows XP were rendered. The ability to create aesthetically pleasing, dynamic graphics has made computer generated graphics find use in this field. For this reason a study will be undertaken to investigate if utilising advanced computer graphics techniques to generate a visually appealing scene, which will act as a screensaver, is possible without being overly burdensome on the computer's hardware. Some proposed techniques to be utilised to make this scene are physically based rendering, atmospheric scattering and volumetric raymarching, though this list of techniques is subject to change throughout the project as the needs and challenges of the project cannot fully be anticipated.

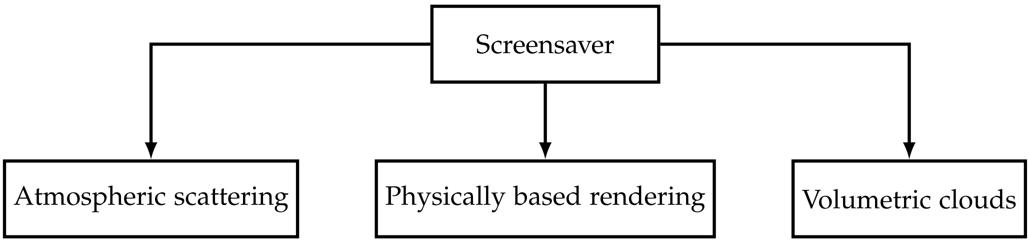


Figure 1.2: *Graphics techniques that may be used*

To manage the project, the structured approach will be utilised and documented in a portfolio. Our clientele would like for us to use VB.NET to create the project, so the project will be programmed utilising it. As a result, challenges may be faced regarding the performance of the screensaver. Since the development of this screensaver will be only undertaken by one developer, the anticipated completion date of the project is June 2019, although this deadline is subject to change based on circumstances regarding the project.

Chapter 2

Project Plan

2.1 Defining and Understanding the Problem

Our initial thoughts on what software construction methodology to use leaned towards one which would benefit a small development team, such as the prototyping methodology. However, it is mandated by our project sponsor that a portfolio be developed to document our development process in a structured manner, so a structured approach is the best choice due to its reliance on documentation.

Thus, the project will be split into 5 stages according to the structured approach:

- Planning
- Design
- Building / Implementing
- Testing
- Deployment

These stages are reflected in our Gantt Chart. Please see figure 3.6 on page 8 for the Gantt Chart.

During the design phase, a survey will be conducted, asking customers what features they would like in a screensaver suite. The survey responses will be considered in our designs and implementation. After initial implementation, a survey will be conducted requesting the opinions of our customers on the product. As of such, there will be some elements of the prototyping methodology in our approach as well, iterating over our initial product to improve it and satisfy the expectations of our customers. However, as stated previously, the methodology of development will be primarily structured.

2.2 Design

When developing the designs for the product, the following will be created.

- A Context Diagram
- A Data Flow Diagram
- A Structure Diagram
- High level flowcharts for key functions in the program.
- A Data Dictionary.

All besides the data dictionary will be created before the implementation stage. The data dictionary will be created along with the product itself.

2.3 Implementation

The implementation stage will firstly begin with a technical research stage. Advanced graphics techniques involve use of heavy amounts of mathematics, and as of such, research papers must be studied and cited.

After the initial research, implementation will commence. Testing will be done regularly as the project progresses, to ensure each module is functioning before moving onto the next one and to avoid the accumulation of errors.

2.4 Testing

On completing the implementation, the screensaver suite will be tested on a variety of hardwares, to ensure it functions correctly on other GPUs. This is critical, as graphics shaders may perform differently and possibly unexpectedly based on each graphics card provider's OpenGL implementation.

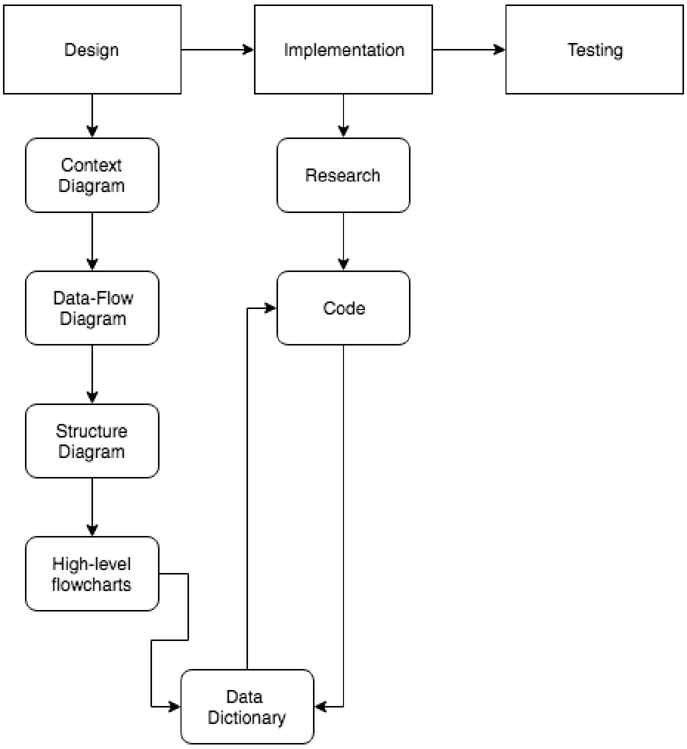


Figure 2.1: *High-level diagram summarising project plan*

Chapter 3

Feasibility Study

3.1 Executive Summary

Screensavers have been a medium of graphics experimentation during the times of early computing. However, advanced graphics techniques have never been applied to the context of screensavers, leaving an excellent vacant marketplace to test cutting edge techniques in the context of screensavers. This document analyzes the feasibility of undertaking a project where advanced graphics techniques are applied to create a screensaver from the lenses of technological feasibility, staffing feasibility and marketplace feasibility.

3.2 Description of Products and Services

Screensavers, during the times of early computing, have been the medium for the showcasing of various graphics techniques. From the Windows 98 era with the notable screensaver maze, bringing rasteurized graphics to the customer in a display of engineering creativity to the classic Windows XP pipes, using random generation to provide the customer with a new and consistently unique scene, graphics techniques have found their place in the world of screensavers. However, these screensavers have often been quite basic, using simple Phong shading or even no shading at all with only rasteur based techniques lacking lighting shading. This makes one wonder why advanced graphics techniques, such as the ones explored by video game companies and animation studies such as Pixar, have not been explored in the avenue of screensaver creation.

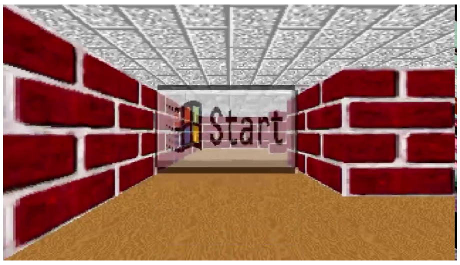


Figure 3.1: *Windows 98 Maze*

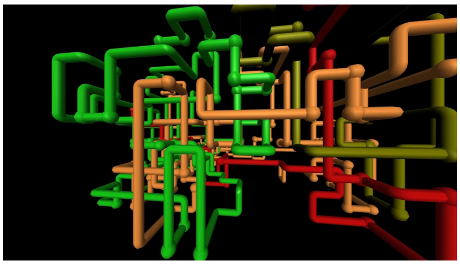


Figure 3.2: *Windows XP Pipes*

This project aims to investigate this avenue, with the aim of creating a suite of 3 screensavers. The first one will investigate the application of volumetric clouds and randomly generated terrain to screensavers. The second will investigate how mathematical fractal geometry may be applied to generating interesting screensavers. The final will investigate the application of the interesting effect of metaballs to screensavers.

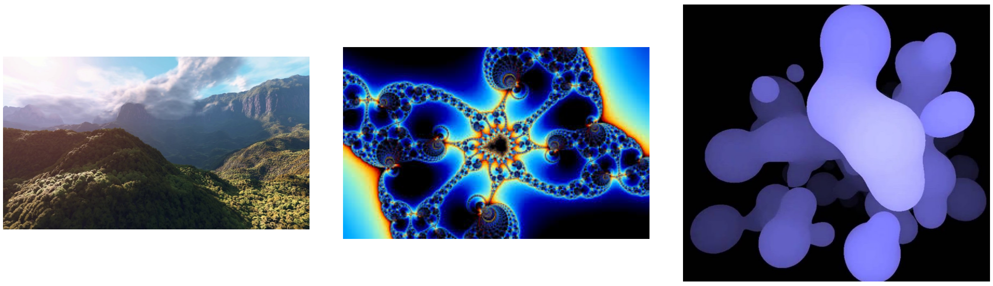


Figure 3.3: *From left to right example images of: Terrain with Volumetric Clouds, Mandelbrot Fractal and Metaballs.*

3.3 Technology Considerations

3.3.1 Hardware Considerations

The personnel possess a single HP Pavilion laptop running Windows 10 OS that uses a Nvidia Geforce GTX 1050 graphics card. Although the graphics card is powerful, it is not out of reach of most customers, making it an excellent benchmark for what our users may use.

3.3.2 Software Considerations

The development will be carried out using the VB.NET programming language through the Visual Studio IDE. VB.NET was chosen as a matter of preference from our project sponsor. Due to this unorthodox choice for a programming language for graphics programming, in which C and C++ are most commonly used due to their performance, extra considerations may need to be taken regarding the overhead of the VB.NET language. Furthermore, as VB.NET is being used to create the screensaver suite, the project is only going to work on the Windows operating system as it required the .NET framework. However, projects such as .NET core and Mono aim to port the .NET framework to linux and mac, so cross compatibility is not completely out of the question, and can be implemented later if the demand is ever shown.

3.4 Product/Service Marketplace

The marketplace for the utilisation of advanced graphics techniques in the context of screensavers has been fairly limited. No repositories on github were found for screensavers which utilised volumetric clouds, which is not surprising as it is a relatively new graphics technique which is still being researched. No procedural terrain screensavers were found either. This makes the first screensaver idea, of procedurally generated terrain with volumetric clouds, a venture into entirely uncharted territory, as the marketplace is entirely empty.

As for the mandelbrot fractal screensaver, mandelbrot fractals are well documented mathematical structures which produce beautiful results when rendered, however have also, surprisingly not been applied in the context of screensavers. There have, however, been large numbers of mandelbrot fractal explorers generated, as it is seen as an excellent exercise to develop one's programming skills. An example written in Javascript is shown below:

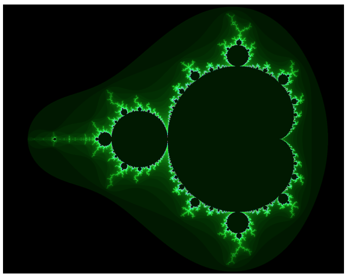


Figure 3.4: *Mandelbrot Fractal Renderered by Rafrex's "Fractal" from https://github.com/rafrex/fractal*

Metaballs are also a well documented technique, created by Jim Blinn, the father of Blinn-Phong shading, which produce organic looking balls that change shape depending on their positions relative to other metaballs. Metaballs have been applied to a wide range of areas, however, one particular use of them is in simulating the behaviour of water droplets efficiently, without the use of complex and expensive fluid dynamics simulations. Metaballs have not been applied to the context of screensavers, however, various eye candy demos have been created such as the one below written in Javascript:

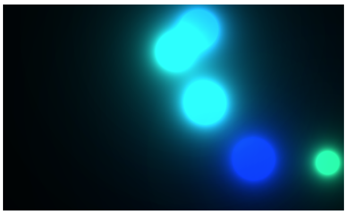


Figure 3.5: *Metaballs Rendered by FlannelHead's "Dynaballs" from http://flannelhead.github.io/dynaballs/*

3.5 Social and Ethical Considerations

3.5.1 Legal Considerations

As this project is primarily a non-profit, research experiment, the project will be released using the Open Source MIT License. This license limits our liability and makes warranty not legally enforcable, legally protecting us in the case of a potential misuse of our software. The license permits commerical use, modification, distribution and private use of our not for profit, research project.

3.5.2 Ergonomics

- The screensaver should be designed to be ergonomic and user friendly. This is easily achievable, as screensavers lack the traditional GUIs of many other products, their main purpose being to be eye candy. Options for these screensavers may be programmed to be directly accessible through the Windows control panel, or could be programmed to be configurable through a separate GUI all together.
- The design of the screensaver options dialog will aim to keep the interface simple and intuitive to use, not overflowing the user with unnecessary options. Advanced options will be hidden behind a drop down menu, for experienced users to utilise.
- The screensavers will aim to maximise FPS without jeopardising the quality of the screensavers.

3.5.3 Inclusivity

- An English locale will be implemented into the screensaver suite options dialog, as English is one of the most widely spoken languages in the world. Furthermore, a Russian locale will be implemented as it is the 7th most widely spoken language in the world, and the developer has knowledge of the language. As this project is open source, if more locales are desired to be programmed in, open source contributors may add them.

- Data formats that differ internationally, such as dates and currency, are not being handled and hence these factors do not need to be accounted for.

- The screensaver would be free and released under the open source licence, allowing anyone of any economic background to utilise it.

- The scenes rendered in the screensavers will be designed to be universally appealing to all demographics.

- The option for the screensaver to play music may be programmed in at a later date. One option used to cater for people with hearing disabilities in software is to include subtitles. However, this solution is not suitable, as the person would still not be able to experience the music through a simple text prompt which says "Music Plays". As of such, this issue will be left open to the open source community to solve, if a solution is found in the future.

- People with visual impairments must also be catered for. Legal blindness cannot be catered for, as the product is a purely visual one. However, for people with mild visual impairments, text can be written using a clear serif or sans-serif font with a large font size. Poor color combinations will be avoided like yellow on white, only using colors which contrast well, such as black on white.

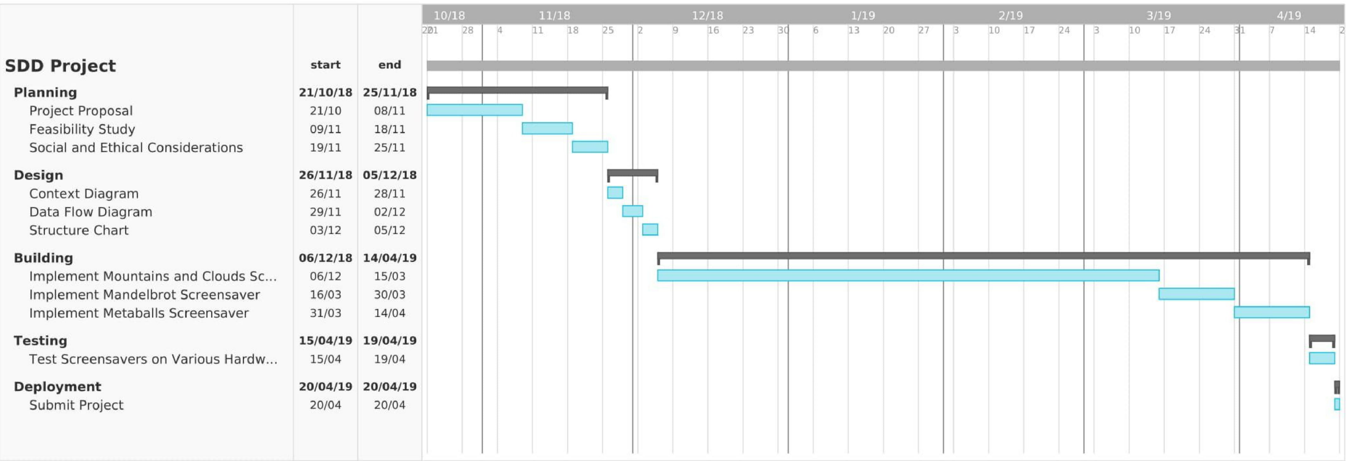
- Issues pertaining to cultural inclusivity need not be worried about, as the screensavers aim to have universal beauty, showcasing the remarkable natures of both the natural and mathematical worlds.

3.6 Organization and Staffing

In considering to create this product, considerations must be made regarding whether the developers of this project possess the expertise and the technology required to complete the project. The development team consists of one student programmer, indicating that the personnel may lack the expertise required to develop an advanced graphics screensaver. However, the deadline for the end of the project provides enough time for the programmer to research and become knowledgable on these techniques before implementation.

3.7 Schedule

The schedule of this project will closely follow the Gantt Chart. The structure of this gantt chart aims to reflect the standard software development life cycle.



SDD Project	start	end
Planning	**21/10/18**	**25/11/18**
Project Proposal	21/10	08/11
Feasibility Study	09/11	18/11
Social and Ethical Considerations	19/11	25/11
Design	**26/11/18**	**05/12/18**
Context Diagram	26/11	28/11
Data Flow Diagram	29/11	02/12
Structure Chart	03/12	05/12
Building	**06/12/18**	**14/04/19**
Implement Mountains and Clouds Sc...	06/12	15/03
Implement Mandelbrot Screensaver	16/03	30/03
Implement Metaballs Screensaver	31/03	14/04
Testing	**15/04/19**	**19/04/19**
Test Screensavers on Various Hardw...	15/04	19/04
Deployment	**20/04/19**	**20/04/19**
Submit Project	20/04	20/04

Figure 3.6: *Gantt chart*

3.8 Financial Projections

As the project manager is also the developer and has direct interests in completing this project, the developer demands no pay. No extra software must be purchased, as all tools that will be used are free. The only costs associated with this project are those associated with electricity for recharging the laptop.

3.9 Findings and Recommendations

In summary:

- There is ample time for research, development and refinement.
- Through investigating advanced graphics techniques in screensavers, one can set the groundwork for their future integration into other mediums such as animated movies and video games.
- The cost of development is negligible.

Hence, it is in our best interests that the project proceed.

Glossary

Atmospheric Scattering Atmospheric scattering, in the context of computer graphics, refers to the modeling and emulation of the process by which light particles are scattered by particles in the atmosphere. Scattering from these particles leads to the blue color of our sky at noon, and its orange color at dawn and dusk. 1

OpenGL a cross-language, cross-platform API standard for rendering 2D and 3D vector graphics. The API is used to interact with a graphics processing unit, to achieve hardware-accelerated rendering. 3

Participating Media Participating media are materials which may absorb, emit and/or scatter light. Participating media usually consist of many particles suspended in the air. 9

Physically Based Rendering In its broadest sense, physically based rendering refers to the usage of physics equations to accurately emulate the trajectories of photons when in contact with materials. An example of a physically based lighting model is the Cook-Torrance model, as opposed to the old, non-physically based Blinn-Phong model. 1

Shader a type of computer program originally run on graphics card to do shading, aka lighting simulation. However, shaders now perform a variety of tasks, and the term now refers to a category of computer programs that are designed to be interpreted and run by the graphics card. 3

Volumetric Clouds Volumetric clouds are rendered using the technique of volumetric raymarching over real-time generated volumes that are designed to emulate the shapes and behaviours of clouds. 1

Volumetric Raymarching Volumetric raymarching is a graphics technique used to accurately render participating media, such as smoke, fog and clouds. 1, 9