****

**YAŞAR UNIVERSITY**

**FACULTY OF ENGINEERING**

**DEPARTMENT OF COMPUTER ENGINEERING**

**COMP4910 Senior Design Project 1, Fall 2019**

**Supervisor: Dr. Gizem Kayar**

**POF: Performance Optimized Fluid System**

**Final Report**

**By:**

**Baran Budak -15070001012**

**Cihanser Çalışkan -16070001020**

**İsmail Mekan -15070001048**

# PLAGIARISM STATEMENT

This report was written by the group members and in our own words, except for quotations from published and unpublished sources which are clearly indicated and acknowledged as such. We are conscious that the incorporation of material from other works or a paraphrase of such material without acknowledgement will be treated as plagiarism according to the University Regulations. The source of any picture, graph, map or other illustration is also indicated, as is the source, published or unpublished, of any material not resulting from our own experimentation, observation or specimen collecting.

**Project Group Members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name, Last name** | **Student Number** | **Signature** | **Date** |
| Baran Budak | 15070001012 |  |  |
| Cihanser Çalışkan | 1607000020 |  |  |
| İsmail Mekan | 15070001048 |  |  |

**Project Supervisors:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name, Lastname** | **Department** | **Signature** | **Date** |
| Gizem Kayar | Computer Engineering |  |  |

# ACKNOWLEDGEMENTS

We thank Gizem Kayar for discussions and suggestions in the development of the project. This project is supported by Yasar University Computer Engineering Department.

# 

# KEYWORDS

|  |  |
| --- | --- |
| **Term** | **Description** |
| Cell | Axis aligned bounding box is divided into small identical cubes. |
| Color field quantity | It is a functionthat calculates how each particle is affected by all the other particles. |
| Gradient | The directional derivative of a scalar field gives a vector field directed towards where the increment is most, and its magnitude is equal to the greatest value of the change. |
| Grid | Series of vertical and horizontal lines that are used to subdivide AABB vertically and horizontally into cells in three-dimensional space. |
| Iso-surface | An isosurface is a 3D surface representation of points with equal values in a 3D data distribution which is the 3D equivalent of a contour line. |
| Marching Cubes | Marching cubes is a computer graphics algorithm, published in 1987 for extracting a polygonal mesh of an isosurface from a three-dimensional discrete scalar field. |
| NVIDIA Flex | NVIDIA Flex is a particle-based simulation technique for real-time visual effects. |
| Polygonal Mesh | A polygon mesh is the collection of vertices, edges, and faces that make up a 3D object. |
| Unity 3D | Unity is a cross-platform game engine developed by Unity Technologies. Unity is used for developing video games and simulations for consoles and mobile devices. |
| Visual Studio | Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. |

**Table 1:** Keywords

# ABSTRACT

The POF system offers more efficient and better performance particle-based fluid simulation. To increase efficiency, we use hashing to reach particles faster. We used the Marching cubes algorithm to visualize particles. We only render the surface particles by using Marching cubes to increase system performance.

\*\* burayı düzenle.

# ÖZET

POF sistemi daha efektif ve daha iyi performanslı bir parçacık temelli sıvı simülasyonu sunar. Hash algoritmasını taneciklere daha hızlı ulaşmak ve verimliliği arttırmak için kullanıyoruz. Tanecikleri görselleştirmek için ise Marching cubes algoritmasını kullanıyoruz. Marching cubes ile yüzey taneciklerini görselleştirip diğer tanecikler sıvının içinde kaldığı için çizmiyoruz ve bu da performansı arttırıyor.

# TABLE OF CONTENTS

[PLAGIARISM STATEMENT ii](#_Toc533517651)

[ACKNOWLEDGEMENTS iii](#_Toc533517652)

[KEYWORDS iv](#_Toc533517653)

[ABSTRACT v](#_Toc533517654)

[ÖZET vi](#_Toc533517655)

[TABLE OF CONTENTS vii](#_Toc533517656)

[LIST OF FIGURES viii](#_Toc533517657)

[LIST OF TABLES ix](#_Toc533517658)

[LIST OF ACRONYMS/ABBREVIATIONS x](#_Toc533517659)

[1. INTRODUCTION 1](#_Toc533517660)

[1.1. Description of the Problem 1](#_Toc533517661)

[1.2. Project Goal 1](#_Toc533517662)

[1.3. Project Output 1](#_Toc533517663)

[1.4. Project Activities and Schedule 1](#_Toc533517664)

[2. DESIGN 1](#_Toc533517665)

[2.1. High Level Design 1](#_Toc533517666)

[2.2. Detailed Design 1](#_Toc533517667)

[2.3. Realistic Restrictions and Conditions in the Design 1](#_Toc533517668)

[3. IMPLEMENTATION, TESTS and TEST DISCUSSIONS 2](#_Toc533517669)

[3.1. Implementation of the Product 2](#_Toc533517670)

[3.2. Tests and Results of Tests 2](#_Toc533517671)

[4. CONCLUSIONS 2](#_Toc533517672)

[4.1. Summary 2](#_Toc533517673)

[4.2. Cost Analysis 2](#_Toc533517674)

[4.3. Benefits of the Project 2](#_Toc533517675)

[4.4. Future Work 2](#_Toc533517676)

[References 3](#_Toc533517677)

[APPENDICES 4](#_Toc533517678)

[APPENDIX A: REQUIREMENTS SPECIFICATION DOCUMENT 5](#_Toc533517679)

[APPENDIX B: DESIGN SPECIFICATION DOCUMENT 6](#_Toc533517680)

**Table 2:** Table of Contents

# LIST OF FIGURES

Figure 1: The pof system……………..2

# LIST OF TABLES

Table 1: Keywords…………………3

Table 2: Table of Contents …………………3

Table 3: List of acronyms/abbreviations……..3

Table 4: Cost Analysis of Workers…………3

# LIST OF ACRONYMS/ABBREVIATIONS

|  |  |
| --- | --- |
| AABB | Axis Aligned Bounding Box. Bounding volume for a set of objects is a closed volume that completely contains the union of the objects in the set. |
| API | Application Programming Interface. |
| CPU | Central Processing Unit. |
| GPU | Graphic Processing Unit. |
| OPENGL | Open Graphics Library is a cross-language, cross-platform application programming interface for rendering 2D and 3D vector graphics. |
| POF | The Performance Optimized Fluid system. |

**Table 3:** List of acronyms/abbreviations

# 1. INTRODUCTION

# 1.1. Description of the Problem

The main problem of the particle-based fluid simulation system is excessive numbers of the particles. There are millions of particles in a small number of liquids such as water. A particle is a rigid body sphere. Simulation applies physics to particles and these particles act as a liquid. Simulation having difficulties in calculations hinge upon a surplus of particles. Indirectly, time and space complexity increasing.

# 1.2. Project Goal

The main goal of the project researches whether there is a way to enhance fluid simulation. Increasing the efficiency and performance of an existing particle-based fluid simulation is a major goal. We aim to achieve these goals by implementing a variety of methods to the POF system such as using special structures to find store particles and visualize it by using various methods like the Marching cubes. In our project, there is no certain way because it is a research and development project and new more effective ways can be found during the project. Various methods and techniques will be researched and implemented while the project is in the development process.

# 1.3. Project Output

* Better performance.
* Better memory efficiency.
* Fluid-like appearance and behaviour.
* Testing of different algorithms for performance and efficiency.
* Higher frame rates per second.

# 1.4. Project Activities and Schedule

\*\* BU KISIM DEĞİŞTİRİLECEK!

1. **1. Semester schedule**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| WBS | Task Name | Duration | Start | Finish | Predecessors |
| **0** | **POF project** | **70 days** | **Mon 16/09/19** | **Fri 20/12/19** |  |
| 1 | Select project topic. | 5 days | Mon 16/09/19 | Fri 20/09/19 |  |
| 2 | Researching necessary platform. | 10 days | Mon 23/09/19 | Fri 04/10/19 | 1 |
| **3** | **Algorithm\_1** | **11 days** | **Mon 07/10/19** | **Mon 21/10/19** |  |
| 3.1 | Researching divide into cells algorithm. | 5 days | Mon 07/10/19 | Fri 11/10/19 | 2 |
| 3.2 | Implementation of dividing into cells algorithm. | 6 days | Mon 14/10/19 | Mon 21/10/19 | 4 |
| 3.3 | Testing dividing into cells algorithm. | 2 days | Fri 18/10/19 | Mon 21/10/19 | 5SS+4 days |
| 4 | Preparing the RSD document. | 5 days | Mon 07/10/19 | Fri 11/10/19 |  |
| **5** | **Algorithm\_2** | **11 days** | **Tue 22/10/19** | **Tue 05/11/19** |  |
| 5.1 | Researching surface recognizer algorithm. | 5 days | Tue 22/10/19 | Mon 28/10/19 | 6 |
| 5.2 | Find surface particle constant. | 2 days | Fri 25/10/19 | Mon 28/10/19 | 9SS+3 days |
| 5.3 | Implementation of surface recognizer algorithm. | 6 days | Tue 29/10/19 | Tue 05/11/19 | 10 |
| 5.4 | Testing surface recognizer algorithm. | 2 days | Mon 04/11/19 | Tue 05/11/19 | 11SS+4 days |
| **6** | **Algorithm\_3** | **10 days** | **Wed 06/11/19** | **Tue 19/11/19** |  |
| 6.1 | Researching Zhu Bridson algorithm. | 5 days | Wed 06/11/19 | Tue 12/11/19 | 12 |
| 6.2 | Implementation of Zhu Bridson algorithm. | 5 days | Wed 13/11/19 | Tue 19/11/19 | 14 |
| 6.3 | Testing Zhu Bridson algorithm. | 2 days | Mon 18/11/19 | Tue 19/11/19 | 15SS+3 days |
| 7 | Preparing the RSD 2.0 document. | 10 days | Mon 18/11/19 | Fri 29/11/19 |  |
| 8 | Preparing the DSD document. | 10 days | Mon 25/11/19 | Fri 06/12/19 |  |
| **9** | **Algorithm\_4** | **20 days** | **Mon 25/11/19** | **Fri 20/12/19** |  |
| 9.1 | Researching the Marching Cubes algorithm. | 10 days | Mon 25/11/19 | Fri 06/12/19 |  |
| 9.2 | Implementing the Marching Cubes algorithm. | 10 days | Mon 09/12/19 | Fri 20/12/19 | 20 |
| 9.3 | Testing the Marching Cubes algorithm. | 4 days | Tue 17/12/19 | Fri 20/12/19 | 21SS+6 days |
| 10 | Preparing the Final report document. | 15 days | Mon 02/12/19 | Fri 20/12/19 |  |

# 

# 2. Semester schedule

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| WBS | Task Name | Duration | Start | Finish | Predecessors |
| **0** | **POF Project second semester** | **71 days** | **Mon 03/02/20** | **Mon 11/05/20** |  |
| **1** | **Algorithms** | **71 days** | **Mon 03/02/20** | **Mon 11/05/20** |  |
| **1.1** | **Researching Algorithms** | **66 days** | **Mon 03/02/20** | **Tue 05/05/20** |  |
| 1.1.1 | Time and Space Complexity comparison of the algorithm to be selected. | 30 days | Mon 03/02/20 | Fri 13/03/20 |  |
| 1.1.2 | Performance testing of system. | 10 days | Mon 02/03/20 | Fri 13/03/20 | 3SS+20 days |
| 1.1.3 | Development of algorithms. | 15 days | Mon 16/03/20 | Fri 03/04/20 | 4 |
| 1.1.4 | Deciding algorithms. | 0 days | Tue 05/05/20 | Tue 05/05/20 | 11 |
| **1.2** | **Implementing Algorithms** | **16 days** | **Mon 06/04/20** | **Mon 27/04/20** |  |
| 1.2.1 | Implementation of surface particle recognizer. | 8 days | Mon 06/04/20 | Wed 15/04/20 | 5 |
| 1.2.2 | Implementing visualization. | 8 days | Thu 16/04/20 | Mon 27/04/20 | 8 |
| **1.3** | **Testing** | **10 days** | **Tue 28/04/20** | **Mon 11/05/20** |  |
| 1.3.1 | Testing surface particles. | 5 days | Tue 28/04/20 | Mon 04/05/20 | 9 |
| 1.3.2 | Testing visualization | 5 days | Tue 05/05/20 | Mon 11/05/20 | 11 |

# 2. DESIGN \*\* DSD den sonra bakılacak

# 2.1. High Level Design

# 2.2. Detailed Design

This section will be completed in future works.

# 2.3. Realistic Restrictions and Conditions in the Design

We had to neglect some aspects of the project to implement the project in a year. The security issue is ignored because the project aims to help everybody who has interested fluid simulations and contribute to science. We assumed that users of the POF system have the necessary equipment and software and know to how to use them.

# 3. IMPLEMENTATION, TESTS and TEST DISCUSSIONS

# 3.1. Implementation of the System

// kod atma bölümü burası

This section will be completed in future works.

# 3.2. Tests and Results of Tests

We have tested different particle-based fluid simulation systems to measure efficiency and performance. Uflex had small bugs and errors in the code, even though we have fixed minor bugs, the particles were not recognizing the collider of the objects. Collider of the simple primitive objects was not recognized by the Uflex and particles were penetrating the objects. The only plane object was being recognized by the Uflex. The problem could not be solved, and we have changed the fluid simulation. Obifluid is eliminated because of performance problems. The expected result was not satisfied by the Obifluid compared to other fluid simulations. We have tested the performance and functionality of different particle-based fluid simulations in the scene that we have created. This section will be elaborated vastly in the future.

# 4. CONCLUSIONS

In this section, the cost table of workers is given and explained. The cost of software and hardware is given with details and benefits of the projects are explained. This part of the final report summarizes our project and gives a cost analysis for the project. Future works mentioned.

# 4.1. Summary

The massive amounts of particles can be a computational hardship for the computer.

We implement various methods to get better results by making a research. Our project focuses on catalyzing computational difficulties by increasing the performance and efficiency. Project should make easier to simulate with higher quantities of particles or getting better results with the same number of particles.

# 4.2. Cost Analysis Table

**4.2.1 Cost of workers**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Members** | **Day/Hour** | **Week/Hour** | **Semester/Hour** | **Salary/Hour** | **Salary/Monthly** | **TOTAL** |
| Member | 8 | 40 | 560 | 30 TL | 4800 TL | 16800 TL |

**Table 4:** Cost Analysis of Workers

As shown in the cost analysis table, three people works in the project. Every people work equally as workload. Therefore, only one member is represented on the cost table.

Every member works 8 hours a day and 5 days a week. A semester consists of 14 weeks and salary is 30 Turkish lira per hour. Each member costs 4800 TL per month and costs 16800 in a semester. The salary costs of all three members are 50400 TL per semester. The equivalent of 16800 TL is $2894,67. Currency translation has made from Dollar / Turkish Lira = 1 / 5.80 in 10 December 2019.

**4.2.2 Cost of Software**

**4.2.2.1 Uflex**

Unity Uflex is an asset, software purchased for our project. We used the Uflex fluid simulation asset to study on particle-based fluid simulation. This asset is eliminated because of the adaptive feature cannot be closed. The adaptive attribute is a change in the particles to look particles more realistic and physics based. However, the POF system is not able to work with the adaptive feature. Uflex asset costs 30$ in the unity asset store.

**4.2.2.2 Obi Fluid**

Obi fluid is another assed particle-based fluid simulation. This asset not selected because performance was very low compared to NVIDIA flex. Obi Fluid asset costs 30$ in the unity asset store.

**4.2.2.3 Screen space fluids (SSF)**

Screen space fluids asset is used for fluid system visualization. It recommended that SSF works with Uflex best. Since Uflex is elected in our project, SSF is not necessarily needed. Besides, visualization is a part of our project and we will use the Marching cubes algorithm to visualize. SSF asset costs 7$ in the unity asset store.

**4.2.2.4 Tecnie collider creator (TCC)**

Tecnie collider creator asset is a collider asset for the objects. TCC aimed to use create better interactions with the fluids and rigid body objects. It is not decided to use this asset, but it can be used in future works. TCC asset costs 30$ in the unity asset store.

Total Software Costs: 97$

**4.2.3 Cost of Hardware**

**4.2.3.1 PC component costs that used in Project (PC 1)**

Operating System: Windows 10 (64-bit)

Processor: Intel Core i7-4700 HQ CPU

Memory: 16 GB RAM – DDR3L-1600 Mhz

GPU: NVIDIA GeForce GTX850M 4GB DDR3

Cost of PC 1 in project per user: $1693, 21

Total cost = Total employee cost + Total software cost + Total Hardware cost (Hardware that used in project.)

Total cost per user = $2894,67 + $97 + $1693, 21 (PC1) = $4684,88

Total cost = Total cost per user \* 3 = ($2894,67 + $1790, 21) \* 3 = $14054,64

**4.2.3.2 Optimal Simulation Computer (PC 2)**

Operating System: Windows 10(64-bit Pro)

Processor: 8-core Intel i7 5.1 GHz

Memory: 32 GB RAM- DDR4- 2666MHz

GPU: NVIDIA Quadro P2200 5GB

Cost of PC 2 per user: $5017

Total cost = Total employee cost + Total software cost + Total Hardware cost (Hardware that used in project.)

Total cost per user = $2894,67 + $97 + $5017 (PC2) = $8008,67

Total cost = Total cost per user \* 3 = ($2894,67 +$5114) \* 3 = $24026,01

# 4.3. Benefits of the Project

Our project can benefit in all areas where liquid simulation is available.

**4.3.1 Animations and Movies:** The POF system can be used in any movies, animations that used fluids.

**4.3.2 Scientific work:** Our project benefit scientific areas the most because the project is heavily research and development based of the research papers about the particle-based fluid simulations. Scientist and researchers can use the POF system for their scientific researches

**4.3.3 Games:** Some games need a fluid simulation system to make more realistic games. The POF system can be a good factor for the makes realistic games. For instance, in sailing simulator game is a perfect match for our system.

**4.3.4 Construction:** The construction and Architecture sector can benefit from our system because the simulation is physics-based which means the POF system is almost realistic. The POF system neglects some imperceptible elastic deformations. For instance, a civil engineer can build a barrage and want to test endurance, on the computer simulation. Therefore, our system can be used for construction and architecture testing.

# 4.4. Future Work

We will develop our project in order to achieve performance and efficiency goals. The functionality of the project will remain the same. However, small changes in the calculations will be changed to get better results. For instance, the research paper written by Solenthaler [SOL??] will be implemented to our project.

# References

Requirement Specification Document revision 1.0 (RSD 1.0)

Requirement Specification Document revision 2.0 (RSD 2.0)

Design Specification Document revision 1.0 (DSD 1.0)