****

**YAŞAR UNIVERSITY**

**FACULTY OF ENGINEERING**

**DEPARTMENT OF COMPUTER ENGINEERING**

**COMP4920 Senior Design Project II, Spring 2020**

**Advisor: Gizem Kayar**

**POF: Performance Optimized Fluid System**

**Product Manual**

**Revision 1.0**

**13.04.2020**

**By:**

**Baran Budak, Student ID: 15070001012**

**Cihanser Çalışkan, Student ID: 16070001020**

**İsmail Mekan, Student ID: 15070001048**

# Revision History

|  |  |  |
| --- | --- | --- |
| **Revision** | **Date** | **Explanation** |
| 1.0 | 13.04.2020 | Initial Product Manual |

# Table of Contents

[Revision History 2](#_Toc36726022)

[Table of Contents 3](#_Toc36726023)

[1. Introduction 4](#_Toc36726024)

[2. POF Software Subsystem Implementation 4](#_Toc36726025)

[2.1. Source Code and Executable Organization 4](#_Toc36726027)

[2.2. Software DevelopmentTools 5](#_Toc36726028)

[2.3. Hardware and System Software Platform 5](#_Toc36726029)

[3. XYZAPP Hardware and Software Subsystem Testing 5](#_Toc36726030)

[4. XYZAPP Installation, Configuration and Operation 5](#_Toc36726031)

[References 6](#_Toc36726032)

**List of Figures**

[Fig 1: Nvidia Flex in Unity asset store](#_Toc501993332) 7

[Fig 2: Import Unity package window](#_Toc501993332) 8

[Fig 3: Creating Flex container in Unity](#_Toc501993332) 8

[Fig 4: Assigning Flex container to the Flex Array Actor table](#_Toc501993332) 10

[Fig 6: Make right click and select create](#_Toc501993332) 10

[Fig 7: Select Flex Array Asset](#_Toc501993332) 10

[Fig 8: Selecting Boundary Mesh](#_Toc501993332) 11

[Fig 9: Creating Flex Source Asset](#_Toc501993332) 11

[Fig 10: Selecting surface mesh of flex source asset](#_Toc501993332) 12

[Fig 11: Surface mesh type menu](#_Toc501993332) 12

[Fig 12: Creating Flex Source Actor and adding component](#_Toc501993332) 13

[Fig 13: Select custom package in Unity](#_Toc501993332) 13

[Fig 14: Find POF unity package in file explorer](#_Toc501993332) 14

[Fig 15: Import package file window](#_Toc501993332) 14

[Fig 16: POF system set up on your unity project 15](#_Toc501993332)

**List of Tables**

[Table 1: Minimum Requirements of software Platform](#_Toc501993332) 6

**1. Introduction**

The purpose of this product manual is to document the implementation, testing, installation and operation of the POF system as a software product.

The POF system is implemented and tested as it is described in Design Specification Document, Revision 2.0 [3], satisfying the requirements in POF system Requirements Specification Document, Revision 1.0.

Implementation, testing and operation details are given in the following sections of this document.

# 2. POF System Software Subsystem Implementation

This section describes the implementation of the POF system and its subsystems.

# 2.1. Source Code and Executable Organization

This section is separately evaluated in POF Code Organization document.

# 2.2. Software Development Tools

In this following section, we describe software tools that we have used in the POF system project.

***2.2.1 Unity***

Unity game engine is used in our project as a visual tool for testing and implementation. Various programs can be used such as Unreal Engine 4. We used Unity because learning speed is faster compared to the Unreal Engine (Unity GUI is relatively easier for us). Also, because some of the members in our team have experience with Unity engine, our advisor suggested us to use the Unity engine. Aforementioned conceivable reasons, we determined to use Unity in our project.

Details: Unity 3D version 2018.3.11 (29 March 2019), Unity Technologies.

***2.2.2 Visual Studio 2017***

Visual Studio is an integrated development environment (ide). We write our code in C# by using Visual Studio. The reason we use Visual Studio is we are developing project in Windows operating system and Unity has Visual Studio support. You can import Unity library to Visual Studio.

Details: Visual Studio 2017 v15.9.15 (13 August 2019), Microsoft.

***2.2.3 Github***

GitHub, Inc. is a company that provides hosting for software development version control using Git. We used Github in our project for storing safely. Keeping our data in local is not an efficient way and it confuses version order. Besides, the importance of tools as Github vastly shows its importance in telecommuting.

Details: GitHub Inc., Subsidiary to Microsoft.

***2.2.4 Gitkraken***

GitKraken is another Git GUI client is used from developers to increase productivity. It has the same operation as Github. However, Gitkraken has a reasonable advantage when it comes to code handling. Gitkraken shows the changing parts of the code and it makes easier to reduce confusions and accelerates the project speed.

Details: Gitkraken, Axosoft.

***2.2.5 NVIDIA FleX***

We used NfleX as a third-party software which serves us to the purpose of having and initialization of particle-based fluid simulation. As we emphasized in the final report [1], since creating a particle-based fluid simulation is another complex thesis topic,

we aim to improve both visualization and performance as much as we can in a research paper implementation manner. We do it by using already existed particle-based fluid simulation.

Flex used in as an asset for Unity. The software operates on Windows or Linux, but it operates on windows in our project. It can be executed on Unity or Unreal Engine 4 platforms, but we use unity for the reasons that we mentioned before.

NVIDIA FleX Requirements:

* Windows 7 (64-bit) or newer.
* DX11 or CUDA capable graphics cart
* Unity 2017.3 or later version

Details: NVIDIA FleX v1.0 (19 July 2018), NVIDIA company.

# 2.3. Hardware and System Software Platform

The minimum specification requirements are listed below:

|  |
| --- |
| D3D11 capable graphics card. |
| NVIDIA: GeForce Game Ready Driver 372.90 or above. |
| AMD: Radeon Software Version 16.9.1 or above. |
| Microsoft Visual Studio 2013 or above. |
| G++ 4.6.3 or higher |
| CUDA 8.0.44 or higher |
| DirectX 11/12 SDK |
| Windows 7 (64-bit) or higher |
| Unity 3D 2017.3 version or higher |
|  |

**Table 1:** Minimum Requirements of software Platform

You can see the system that we used while developing the POF system in Final Report [1].

# 3. POF System Software Testing

// İSMAİLDE\*\*

Since our project is predominantly based on research and development, it takes a lot of time to synthesize scientific articles and implement mathematical algorithms, and try out algorithms, so we used the buttom-up integration test technique, which has a gradual structure in the testing process of our project.As a result, our software works correctly in general.

Bottom-up testing is an approach to integrated testing where the lowly level components are tested firstly, then used to facilitate the testing of high level components. The process is repeated until the component at the top of the hierarchy is tested. This approach is helpful only when all or most of the component of the same development level are ready.

**3.1 Testing of Hash System**

In our POF project, we first tested whether the Hash System was working properly.The purpose of the Hashing system is to increase the performance by reaching the particles faster.

We tried to color the scene mode of the Unity platform. We did the test by painting the selected particle to blue colur on gizmos. Neighbour particles coloured to red and cell that particle in is drawn as a red wire cube. The meaning of white particles is neutral and offset the colour of the particles. The outmost yellow wire cube is AABB. Additionally, Cell System is a mathematical imaginary structure that can be calculated as a function with the position data of a particle and it does not exist in the POF system.

The images in figures 1,2 and 3 below were taken in Unity game mode.

tablo, pasta, oturma, beyaz içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Figure 1:**Particle and its neighbours in the cell.

Particles start to move in game mode. We must track the selected particle and its neighbours including the cell that particle belongs to. as shown in the figure(2) below, tracking particle has been achieved.

**bilgisayar içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Figure 2:**Tracking particle in a cell.

Initially, we managed to track the particle that we want by changing the particle id which proves that our hash system works correctly. Then,in AABB, we proved by calculating the manual particle order. Tracking neighbour particles and the cell boundaries is as significant as tracking the particle itself.

**tablo, farklı, grup, geniş içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Figure 3:**Tracking particle and its neighbours in a cell.

Surface kısmındada sormam gerekenler var.

***3.2 Testing of Surface Recognizer:***

**Marching cubesde fotoğraf gelmesini bekliyorum ve bir şeyler soracağım**

***3.3 Testing of Marching Cubes:***

# 4. POF system Installation, Configuration and Operation

This section describes how user installs, configures and operates the POF system. Firstly, the user must have the required system mentioned before.

***4.1 NVIDIA Flex installation and set up***

Assuming that a normal user has the POF system digital copy. User should open an empty 3D project from Unity. You can download Unity from the official website [6]. Firstly, you should import the NVIDIA Flex asset, you can download from the Unity asset store as the figure below. Access the NVIDIA Flex asset page in the Unity asset store. Click download.

When download finished, import button will appear. Click import. Also, you can import the POF system so you do not necessarily need to import NVIDIA Flex from the Unity asset store. However, importing from the Unity asset store is shown here.

ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu  **Figure 1:** Nvidia Flex in Unity asset store

Another window will appear, and you select all files and click import again.

ekran görüntüsü, elektronik eşyalar, bilgisayar içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Figure 2:** Import Unity package window

NVIDIA Flex is imported, the next step is making ready our scene by setting up the Flex. Make right-click on the project files which positioned below of the screen. Select create, from the new window select Flex and then select flex container.

A screenshot of a computer

Description automatically generated

**Figure 3:** Creating Flex container in Unity

A screenshot of a cell phone

Description automatically generatedWe have created flex container and a new table of flex container should appear as the figure below.

**Figure 4:** Flex container table

Next step is making a right-click in the hierarchy window and select create empty and Game object is created. Make right-click and change its name to Flex array actor. Click the Flex array actor and at the inspector menu click add component. Write Flex Array Actor to the search bar and press enter. On the new open menu, we need to specify flex container which is not assigned right now. Drag and drop Flex Container to space across Container of the Container title in the Flex Array Actor inspector menu. It should look like this in the figure below.

A screenshot of a cell phone screen with text

Description automatically generated**Figure 5:** Assigning Flex container to the Flex Array Actor table

A screenshot of a cell phone

Description automatically generatedMake right-click and select flex and from there select flex array asset. We create Flex array asset from the same place that we created a flex container. This step is similar to the previous one.

**Figure 6:** Make right click and select create

A screenshot of a cell phone

Description automatically generated

**Figure 7:** Select Flex Array Asset

Click the small button to select the boundary mesh as shown in the figure. You can select different boundary meshes. However, it is preferred to select the cube mesh.

A screenshot of a computer

Description automatically generated

**Figure 8:** Selecting Boundary Mesh

Next, we click our flex array actor and assign flex array asset by drag and drop to the array asset row. Enable the Fluid checkbox so our particles can act as a fluid. This is a method of creating particles to our scene. There are particles which are stored like an array in somewhere. When you play the scene, particle-based fluid simulation behaves as the NVIDIA Flex operates with the CUDA supported GPU in the back end. Another way is creating a Flex source asset. Flex source asset operates a kind of tap or fountain. Here is how you can set up Flex source asset into your scene.

Right-click in the project file window and create a flex source asset as shown in the figure.A screenshot of a cell phone

Description automatically generated

**Figure 9:** Creating Flex Source Asset

Change the value of mesh tessellation to 1 and then click flex source actor and select surface mesh as shown in the figure.

A screenshot of a cell phone

Description automatically generated

**Figure 11:** Selecting surface mesh of flex source asset

Select mesh from the newly opened menu. We prefer quad mesh however you can prefer different meshes such as a cylinder. It will change the way of the release of the particles.

A screenshot of a cell phone

Description automatically generated

**Figure 12:** Surface mesh type menu

Create an empty game object in scene hierarchy and rename it as flex source actor. Then, click add component and add Flex source actor script to our game object. Flex source actor script has not assigned parts so we should assign Flex container to the container row by drag and drop. Assign the flex source asset to the relevant row in flex source actor script as shown in the figure with a red square and an arrow.

A screenshot of a cell phone

Description automatically generated

**Figure 13:** Creating Flex Source Actor and adding component

Click flex array actor and click add component. Write flex fluid renderer and press enter. Do this part again for the flex source actor. Particles look like water or fluid-like because of the fluid renderer.

***4.2 POF installation and set up***

**ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu**Now, you can import the POF system. Click assets, from the opening menu please click import package. Click custom package as shown in the figure.

**Figure 14:** Select custom package in Unity.

Find POF unity package file location in your computer and select.

A screenshot of a social media post

Description automatically generated

**Figure 15:** Find POF unity package in file explorer.

New window will open as a next step. You can select which files to be imported on your unity project because of this window. Please select all and click import.

A screenshot of a cell phone

Description automatically generated

**Figure 16:** Import package file window.

You may wait a little while; it depends on your computer speed. After the process finished, you can see the POF package is set up to your unity project. As you can see, all components are included, NVIDIA Flex is one of them. However, you can import Flex asset from both Unity asset store and just import from our project. Then, you prepare the NVIDIA Flex because POF is dependent on the fact that Flex is executable. Marching cubes file is about visualization part which we described with all details in the Design Specifications Document [3]. SimuSystem is most of the part that our project covers. Visualization makes visual output completely different so we think that user can change the main setting from Marching cubes folder.

A screenshot of a computer

Description automatically generated

**Figure 17:** POF system set up on your unity project.

***4.3 Learning the basics of POF!***

Welcome to the POF system! In this section we describe about the parameters and configurations that user can change from the inspector menu. The other unity settings such as lighting etc. is irrelevant and depends on the user.

***4.3.1 NVIDIA Flex inspector settings***

***// Intro gir***

***A screenshot of a cell phone

Description automatically generated***

**Figure 18:** Description ofFlex array asset parameters.

***A screenshot of a cell phone

Description automatically generated***

**Figure 19:** Description ofFlex Container parameters.

***4.3.2 POF inspector settings***

# References

1. Final Report revision 1.0
2. Requirement Specification Document revision 2.0 (RSD 2.0)
3. Design Specification Document revision 2.0 (DSD 2.0)
4. NVIDIA FleX manual

<<https://docs.nvidia.com/gameworks/content/gameworkslibrary/physx/flex/manual.html#manual>> (Last review: 10.4.2020)

1. Unity 2018.3 manual

<<https://docs.unity3d.com/2018.3/Documentation/Manual/index.html>>

(Last review: 10.4.2020)

1. NVIDIA Flex Set up tutorial Youtube video playlist

<<https://www.youtube.com/watch?v=Fp1SMb3SWoo&list=PL4FII4B-zM0dMI-GgR3KsfJwm100MH3TT>> (Last review: 10.4.2020)