Sapphire Engine

Design Document

# Engine

## UC001: Engine Startup

### Description

The main goal of this Use Case is to get the engine up and running. The procedure should create and display the window of the given height. It should also create and initialize the entire render system.

### Procedure

1. Execute UC002: Create The Display Window
2. Execute UC003: Create The Render System

## UC002: Create The Display Window

### Description

Do all the necessary steps to create a window with given dimensions to show a windows on a desktop, on which our engine will be presenting the frames.

### Inputs

The width and height of the window’s client area

### Outputs

An empty window displayed on the screen.

### Procedure

1. Register the Window Class
2. Adjust the Window’s Client Area so it fits exactly the given window dimensions
3. Create Window Instance

# Render System

## UC003: Create Render System

### Description

This Use Case will create all the necessary DirectX 12 resources to start producing the fames.

### Inputs

The window handle, the width and height of the render area.

### Outputs

### Procedure

1. Create the DXGI Factory
2. Check system capabilities
3. Enumerate all the available adapters
4. Create DirectX 12 Device
5. Create Command Queue
6. Create Command List
7. Create Swap Chain
8. Create Descriptor Heap
9. Create two Render Targets, one for Back Buffer and one for Front Buffer. Those two resources were created along with the Swap Chain.

# Command Queue

In Sapphire Engine there is only one command queue. It’s main responsibility is to execute the Command Lists, created by the renderer. The Command Queue should be responsible for the synchronization with the GPU. The synchronization is really simple – you can’t do anything on the CPU side until the Command Queue is finish executing.

## UC0004: Create Command Queue

### Description

The engine should have just one Command Queue. It is a simple, direct queue with a normal priority.

### Inputs

### Outputs

### Procedure

1. Create and store the DirectX 12 Command Queue. It should be of the type direct, with normal priority and with no additional flags.
2. Create and store the DirectX 12 Fence. It should have the default value of zero.
3. Create an empty Windows Event. Store the handle to it.
4. Flush the GPU by executing the UC0005: Flush

## UC0005: Flush

### Description

### Inputs

### Outputs

### Procedure

## UC0006: Execute Command List

### Description

### Inputs

### Outputs

### Procedure

# Command List

## UC0007: Create Command List

### Description

### Inputs

### Outputs

### Procedure

# Render Target

## UC0008: Create Render Target

### Description

### Inputs

### Outputs

### Procedure