mugen_engine::VERTEX_DATA

- + DirectX::XMFLOAT3 pos
- + DirectX::XMFLOAT2 uv

mugen_engine::MEGraphicGpuResource Manager

- Microsoft::WRL::ComPtr< ID3D12Descriptor Heap > m basicDescHeap
- · uint32_t m_descriptorHeapIncrementSize
- Microsoft::WRL::ComPtr< ID3D12Resource > m textureBuffer
- Microsoft::WRL::ComPtr< ID3D12Resource > m constantBuffer
- Microsoft::WRL::ComPtr< ID3D12Resource > m_uploadBuffer
- std::vector< Microsoft::WRL ::ComPtr< ID3D12Resource > > m vertexBuffer
- std::vector< D3D12 VERTEX BUFFER VIEW > m vertexBufferView
- UINT m numVertexBuffer
- std::vector< Microsoft::WRL ::ComPtr< ID3D12Resource > > m additionalVertexBuffer
- UINT m_numAdditionalVertexBuffer
- UINT m_currerntAdditionalVertexBuffer ViewIndex
- + MEGraphicGpuResourceManager()
- + void Initialize(const MEGraphicDevice &device. UINT numVertexBuffer)
- + void SetGpuResource(MEGraphicCommand List &cmdList)
- + void UploadVertexData(uint32 _t index, VERTEX_DATA *vertices, size t vertexNum)
- + void UploadConstantData(CONSTANT DATA &constData)
- + void SetRenderCommand(MEGraphicCommand List &cmdList)
- + void CreateSrv(const DXGI FORMAT format, const MEGraphicDevice &device)
- + void CreateTextureBuffer(const DirectX:TexMetadata &metadata, const MEGraphicDevice &device)
- + void ResetUploadBuffer(const size t rowPitch, const size _t height, const MEGraphicDevice &device)
- + void UploadDataToUploadBuffer (uint8 t *srcData, const size _t rowPitch, const size_t height)

8 8 8 8 8

- void InitalizeConstantBuffer (const MEGraphicDevice &device)
- size t GetAlignmentedSize (size_t size, size_t alignment)
- void _CreateCbv(const MEGraphicDevice &device)
- void _SetBarrierBeforeUploadTexture (const MEGraphicCommandList &cmdList)

mugen engine::MEGraphicPipeline

- std::vector< char > m_vsBlob
- std::vector< char > m psBlob
- Microsoft::WRL::ComPtr< ID3DBlob > m_errorBlob
- Microsoft::WRL::ComPtr< ID3D12Pipeline State > m_pipelineState
- Microsoft::WRL::ComPtr< ID3D12Root Signature > m_rootSignature
- + MEGraphicPipeline()
- + void Initialize(const MEGraphicDevice &device, const D3D12_INPUT_ELEMENT DESC inputLayout[], const int layoutSize)
- + void SetPipelineState(const int type, MEGraphicCommandList &cmdList)
- void ProcessBlobError(HRESULT result)
- · void CreateRootSignarure (const MEGraphicDevice &device)
- void LoadShader()
- void CreatePipelineState (const MEGraphicDevice &device. const D3D12 INPUT ELEMENT DESC inputLayout[], const int layoutSize)

mugen engine::MEGraphicRenderTarget

- const int m_numBackBuffer
- Microsoft::WRL::ComPtr< IDXGISwapChain4 > m swapchain
- Microsoft::WRL::ComPtr< ID3D12Descriptor Heap > m_rtvHeaps
- std::vector< Microsoft::WRL ::ComPtr< ID3D12Resource > > m backBuffers
- D3D12 CPU DESCRIPTOR HANDLE m_renderTargetHandle
- D3D12_VIEWPORT m_viewport
- D3D12_RECT m_scissorRect
- + MEGraphicRenderTarget()
- void Initialize(const MEGraphicDevice &device, const MEGraphicCommandList &cmdList, HWND hwnd, const int window _width, const int window_height)
- + void Present()
- + void SetBarrierBeforeRender (MEGraphicDevice &device, MEGraphicCommand List &cmdList)
- + void SetBarrierBeforePresent (MEGraphicCommandList &cmdList)
- + void Clear(float clearColor [4], MEGraphicCommandList &cmdList)
- + void SetRenderArea(MEGraphicCommand List &cmdList, const int topX, const int topY, const int bottomX, const int bottomY)
- void SetRenderBaseCommand (MEGraphicCommandList &cmdList)

m pCmdList

-m_pPipeline -m resourceManager

m_pRenderTarget

-m vertices

mugen_engine::MEGraphicCommandList

Microsoft::WRL::ComPtr< ID3D12Graphics</p>

Microsoft::WRL::ComPtr< ID3D12Command

Microsoft::WRL::ComPtr< ID3D12Fence

· void Initialize(const MEGraphicDevice

Allocator > m cmdAllocator

CommandList > m cmdList

Queue > m_cmdQueue

+ MEGraphicCommandList()

+ ID3D12CommandQueue *const

GetCommandQueue() const

ID3D12GraphicsCommandList

*const GetCommandList() const

> m fence

&device)

+ void Execute()

- UINT64 m fenceVal

Microsoft::WRL::ComPtr< ID3D12Command

mugen_engine::MEGraphicCharacterUnit

- size_t m_width
- size_t m_height
- friend MEFontData
- + MEGraphicCharacterUnit()
- + MEGraphicCharacterUnit(const wchar_t character, HFONT fontData, HDC hdc, MEGraphicDevice &device, MEGraphicCommandList &cmdList, MEGraphicPipeline &pipeline, MEGraphicRenderTarget &renderTarget)
- void DrawCharacter(const int x, const int y, const float color[4], const float priority)