



mugen_engine::MEGraphicGpuResource
Manager

- Microsoft::WRL::ComPtr< ID3D12DescriptorHeap > 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_currenntAdditionalVertexBufferViewIndex

- + MEGraphicGpuResourceManager()
- + void Initialize(const MEGraphicDevice &device, UINT numVertexBuffer)
- + void SetGpuResource(MEGraphicCommandList &cmdList)
- + void UploadVertexData(uint32_t index, VERTEX_DATA *vertices, size_t vertexNum)
- + void UploadConstantData(CONSTANT_DATA &constData)
- + void SetRenderCommand(MEGraphicCommandList &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 ...
- void _InitalizeConstantBuffer(const MEGraphicDevice &device)
- size_t _GetAlignmentedSize(size_t size, size_t alignment)
- void _CreateCbv(const MEGraphicDevice &device)
- void _SetBarrierBeforeUploadTexture(const MEGraphicCommandList &cmdList)