mugen_engine::MEGraphicPipeline
- std::vector< char > m_vsBlob
- std::vector< char > m_psBlob
- Microsoft::WRL::ComPtr< ID3DBlob > m_errorBlob
- Microsoft::WRL::ComPtr< ID3D12PipelineState > m_pipelineState
- Microsoft::WRL::ComPtr< ID3D12RootSignature > 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< ID3D12DescriptorHeap > 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, MEGraphicCommandList &cmdList)
+ void SetBarrierBeforePresent(MEGraphicCommandList &cmdList)
+ void Clear(float clearColor[4], MEGraphicCommandList &cmdList)
+ void SetRenderArea(MEGraphicCommandList &cmdList, const int topX, const int topY, const int bottomX, const int bottomY)
+ void SetRenderBaseCommand(MEGraphicCommandList &cmdList)
mugen_engine::MEGraphicDevice
- Microsoft::WRL::ComPtr< ID3D12Device > m_device
- Microsoft::WRL::ComPtr< IDXGIFactory6 > m_dxgiFactory
+ MEGraphicDevice()
+ MEGraphicDevice(const MEGraphicDevice &)=delete
+ void Initialize()
+ ID3D12Device *const GetDevice() const
+ IDXGIFactory4 *const GetFactory() const
- void _EnableDebugLayer()
mugen_engine::MEGraphicCommandList
- Microsoft::WRL::ComPtr< ID3D12CommandAllocator > m_cmdAllocator
- Microsoft::WRL::ComPtr< ID3D12GraphicsCommandList > m_cmdList
- Microsoft::WRL::ComPtr< ID3D12CommandQueue > m_cmdQueue
- Microsoft::WRL::ComPtr< ID3D12Fence > m_fence
- UINT64 m_fenceVal
+ MEGraphicCommandList()
+ void Initialize(const MEGraphicDevice &device)
+ void Execute()
+ ID3D12CommandQueue *const GetCommandQueue() const
+ ID3D12GraphicsCommandList *const GetCommandList() const
mugen_engine::VERTEX_DATA
+ DirectX::XMFLOAT3 pos
+ DirectX::XMFLOAT2 uv
mugen_engine::MEGraphicGpuResourceManager
- 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_currentAdditionalVertexBufferViewIndex
+ 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)
-m_pPipeline
-m_pPipeline
-m_pRenderTarget
-m_pRenderTarget
-m_pDevice
-m_pDevice
-m_pCmdList
-m_pCmdList
-m_vertices
-m_resourceManager
mugen_engine::MEImage
- size_t m_width
- size_t m_height
- size_t m_xDivideNum
- size_t m_yDivideNum
- DirectX::XMFLOAT4 m_brightness
- BLEND_TYPE m_blendType
+ MEImage()
+ MEImage(const std::wstring &filepath, MEGraphicDevice &device, size_t xDivideNum, size_t yDivideNum, MEGraphicCommandList &cmdList, MEGraphicPipeline &pipeline, MEGraphicRenderTarget &renderTarget)
+ void DrawGraph(int x, int y, float priority, int index=0)
+ void DrawRotaGraph(int x, int y, float scale, float angle, float priority, int index=0)
+ void DrawGraph2X(int x, int y, float priority, int index=0)
+ void DrawRotaGraph2X(int x, int y, float scale, float angle, float priority, int index=0)
+ void DrawModiGraph(int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3, float priority, int index=0)
+ void DrawModiGraph2X(int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3, float priority, int index=0)
+ void SetBrightness(const float R, const float G, const float B, const float A)
+ void SetBlendType(BLEND_TYPE blendType)
+ void ResetAdditionalVertexBuffer()
mugen_engine::MEFontData
- HDC m_hdc
- HFONT m_oldFont
- std::unordered_map< wchar_t, MEGraphicCharacterUnit > m_loadedCharacters
+ MEFontData()
+ MEFontData(std::wstring fontName, int fontSize, MEGraphicDevice &device, MEGraphicCommandList &cmdList, MEGraphicPipeline &pipeline, MEGraphicRenderTarget &renderTarget)
+ MEFontData & operator=(const MEFontData &rhs)
+ void DrawString(const int x, const int y, const float color[4], float priority, const std::wstring text)
+ void DrawFormatString(const int x, const int y, const float color[4], float priority, const std::wstring text,...)
magica_rogue::MRRandom
- std::mt19937 m_engine
+ MRRandom(uint32_t seed)
+ uint32_t GetRanged(uint32_t minimum, uint32_t maximum)
+ std::mt19937 & GetDevice()
-m_font
-m_mapchipImg
-m_minimapImg
-m_random
magica_rogue::MRMapData
- const int m_width
- const int m_height
- int m_startX
- int m_startY
- std::vector< std::vector< int > > m_mapData
- std::vector< std::vector< int > > m_graphicData
- std::vector< ROOM_NODE > m_roomList
- std::vector< ROOM_NODE > m_pathList
- std::vector< ROOM_NODE > m_regionList
- float m_chipSize
- std::vector< ROOM_INDEX > m_roomIndex
+ MRMapData(const int width, const int height, uint32_t seed, MRStaticObjectManager &staticList)
+ void Update(const MRTransform &playerTransform)
+ void Render(const MRCamera &camera) const
+ void RenderMiniMap(const MRTransform &playerTransform, MRStaticObjectManager &staticList) const
+ float GetStartX() const
+ float GetStartY() const
+ void HitWithWall(MRTransform &transform, const float size)
- void _ConvertGraphFromMap()
- void _DivideRooms()
- void _SetStartPosition(std::vector< ROOM_NODE > &rooms)
- void _SpawnTreasureBox(MRStaticObjectManager &staticList)