**Table of Contents**

**1 Model 3**

1.1 Model tried 3

1.1.1 src 3

1.1.1.1 API 3

1.1.1.1.1 OpenGL 3

1.1.1.2 Platform 5

1.1.1.2.1 Windows 5

1.1.1.3 XYZ 5

1.1.1.3.1 Class Stopwatch 7

1.1.1.3.2 AssetManager 8

1.1.1.3.2.1 Class AssetManager 8

1.1.1.3.3 Core 9

1.1.1.3.3.1 Enumeration AnsiCode 10

1.1.1.3.3.2 Event 11

1.1.1.3.3.2.1 Class event\_ptr 12

1.1.1.3.4 ECS 13

1.1.1.3.5 Particle 13

1.1.1.3.5.1 Class ParticleProps2D 14

1.1.1.3.6 Physics 15

1.1.1.3.6.1 Class PhysicsSystem 15

1.1.1.3.7 Renderer 16

1.1.1.3.7.1 Class Animation 18

1.1.1.3.7.2 Class BufferElement 18

1.1.1.3.7.3 Class BufferLayout 18

1.1.1.3.7.4 Class Command 19

1.1.1.3.7.5 Class CommandI 19

1.1.1.3.7.6 Class IndexBuffer 19

1.1.1.3.7.7 Class MaterialManager 19

1.1.1.3.7.8 Class OrthoCamera 20

1.1.1.3.7.9 Class OrthoCameraBounds 21

1.1.1.3.7.10 Class OrthoCameraController 21

1.1.1.3.7.11 Class Renderable2D 22

1.1.1.3.7.12 Class RenderCommand 23

1.1.1.3.7.13 Class RenderCommandQueue 23

1.1.1.3.7.14 Class Renderer 24

1.1.1.3.7.15 Class RendererAPI 24

1.1.1.3.7.16 Class RendererBatchSystem2D 25

1.1.1.3.7.17 Class Routine 26

1.1.1.3.7.18 Class Shader 26

1.1.1.3.7.19 Class ShaderStorageBuffer 27

1.1.1.3.7.20 Class SortingGroup 28

1.1.1.3.7.21 Class SpriteAnimation 28

1.1.1.3.7.22 Class SpriteAnimationController 29

1.1.1.3.7.23 Class SpriteSystem 29

1.1.1.3.7.24 Class SubRoutine 30

1.1.1.3.7.25 Class Texture 30

1.1.1.3.7.26 Class Texture2D 30

1.1.1.3.7.27 Class TextureUniform 31

1.1.1.3.7.28 Class Uniform 31

1.1.1.3.7.29 Class VertexArray 31

1.1.1.3.7.30 Class VertexBuffer 32

1.1.1.3.7.31 Enumeration BufferUsage 32

1.1.1.3.7.32 Enumeration RenderFlags 32

1.1.1.3.7.33 Enumeration ShaderDataType 33

1.1.1.3.7.34 Enumeration UniformDataType 33

1.1.1.3.8 Utils 33

1.1.1.3.8.1 DataStructures 33

1.2 Package Diagram 34

1.2.1 API 34

1.2.2 Asset Manager 34

1.2.3 Assets 34

1.2.4 Business Logic 34

1.2.5 Core 35

1.2.6 Data 35

1.2.7 ECS 35

1.2.8 Particle 35

1.2.9 Presentation 35

1.2.10 Renderer 35

1.2.11 Physics 35

1.2.12 Renderer 35

1.3 Save/Load Diagram 35

1.3.1 Class Asset Package 36

1.3.2 Class Editor 36

# **Model**

## **Model tried**

### **src**

#### **API**

##### **OpenGL**



Obrázek 1 - OpenGL

#### **Platform**

##### **Windows**



Obrázek 2 - Windows

#### **XYZ**



Obrázek 3 - Debug



Obrázek 4 - XYZ

##### **Class Stopwatch**

Basic scope timer. At begining of the sope construct object of Stopwatch (Stopwatch sw()). Name of scope can be passed in constructor. Function/lambda can be also passed in constructor that will be passed args of const std::string, const float args. const String [0] -> Scope name, const float [1] -> time in ms. Function will be called on scope end. After object goes out of scope result's will be written to std::cout. results.json will be generated. This can be viewed in "chrome://tracing". Process id, thread id's are passed.

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| m\_End | TimePoint |  |

|  |  |  |
| --- | --- | --- |
| m\_Function | OptionalFunction |  |

|  |  |  |
| --- | --- | --- |
| m\_Name | std::string | referene\_wrapper can reference function so no need for function.. + its broken using OptionalFunction = OptionalReference<std::function<void(const std::string, const float)>>; using OptionalString = OptionalReference<const std::string&>; |

|  |  |  |
| --- | --- | --- |
| m\_Start | TimePoint |  |

|  |  |  |
| --- | --- | --- |
| **Název metody** | **Návratový typ** | **Popis** |
| GetProcessID | size\_t |  |

|  |  |  |
| --- | --- | --- |
| Stopwatch |  | Parametry:  **name: std::string&** -  Parametry:  **func: OptionalFunction** - |

|  |  |  |
| --- | --- | --- |
| ~Stopwatch |  |  |

##### **AssetManager**



Obrázek 5 - AssetManager

###### **Class AssetManager**

! @class AssetManager

@brief The class to manage game assets

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| ASSETS\_FOLDER | std::string | Path to the folder with assets |

|  |  |  |
| --- | --- | --- |
| m\_AssetIdCounter | uint16\_t | Counter to track next free id to assign to an asset |

|  |  |  |
| --- | --- | --- |
| m\_FreedIds | std::queue<uint16\_t> | Queue with ids of uloaded assets |

|  |  |  |
| --- | --- | --- |
| m\_Instance | std::unique\_ptr<AssetManager> | Instance of the class |

|  |  |  |
| --- | --- | --- |
| m\_Shaders | std::unordered\_map<std::string, std::pair<uint16\_t, std::shared\_ptr<XYZ::Shader>>> | Shaders collection |

|  |  |  |
| --- | --- | --- |
| m\_Textures | std::unordered\_map<std::string, std::pair<uint16\_t, std::shared\_ptr<XYZ::Texture2D>>> | Textures collection |

|  |  |  |
| --- | --- | --- |
| **Název metody** | **Návratový typ** | **Popis** |
| AssetManager |  |  |

|  |  |  |
| --- | --- | --- |
| Get | std::unique\_ptr<AssetManager>& |  |

|  |  |  |
| --- | --- | --- |
| GetShader | std::shared\_ptr<XYZ::Shader> | Parametry:  **name: std::string&** - |

|  |  |  |
| --- | --- | --- |
| GetTexture | std::shared\_ptr<XYZ::Texture2D> | Parametry:  **name: std::string&** - |

|  |  |  |
| --- | --- | --- |
| LoadShader | void | Parametry:  **name: std::string&** -  Parametry:  **path: std::string&** - |

|  |  |  |
| --- | --- | --- |
| LoadTexture | void | Parametry:  **wrap: TextureWrap** -  Parametry:  **name: std::string&** -  Parametry:  **path: std::string&** - |

|  |  |  |
| --- | --- | --- |
| UnloadShader | void | Parametry:  **name: std::string&** - |

|  |  |  |
| --- | --- | --- |
| UnloadTexture | void | Parametry:  **name: std::string&** - |

##### **Core**



Obrázek 6 - Core

###### **Enumeration AnsiCode**

Enum Numeric ansi code representation

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| COLOR\_BLACK\_F |  |  |

|  |  |  |
| --- | --- | --- |
| COLOR\_RED\_F |  |  |

|  |  |  |
| --- | --- | --- |
| COLOR\_GREEN\_F |  |  |

|  |  |  |
| --- | --- | --- |
| COLOR\_YELLOW\_F |  |  |

|  |  |  |
| --- | --- | --- |
| COLOR\_BLUE\_F |  |  |

|  |  |  |
| --- | --- | --- |
| COLOR\_MAGENTA\_F |  |  |

|  |  |  |
| --- | --- | --- |
| COLOR\_CYAN\_F |  |  |

|  |  |  |
| --- | --- | --- |
| COLOR\_WHITE\_F |  |  |

|  |  |  |
| --- | --- | --- |
| COLOR\_BLACK\_B |  |  |

|  |  |  |
| --- | --- | --- |
| COLOR\_RED\_B |  |  |

|  |  |  |
| --- | --- | --- |
| COLOR\_GREEN\_B |  |  |

|  |  |  |
| --- | --- | --- |
| COLOR\_YELLOW\_B |  |  |

|  |  |  |
| --- | --- | --- |
| COLOR\_BLUE\_B |  |  |

|  |  |  |
| --- | --- | --- |
| COLOR\_MAGENTA\_B |  |  |

|  |  |  |
| --- | --- | --- |
| COLOR\_CYAN\_B |  |  |

|  |  |  |
| --- | --- | --- |
| COLOR\_WHITE\_B |  |  |

###### **Event**



Obrázek 7 - Event

**Class event\_ptr**

typedef std::shared\_ptr<Event> event\_ptr;

##### **ECS**



Obrázek 8 - ECS

##### **Particle**



Obrázek 9 - Particle

###### **Class ParticleProps2D**

@struct ParticleProps2D

@brief properties of single particle

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| colorBegin | glm::vec4 |  |

|  |  |  |
| --- | --- | --- |
| colorEnd | glm::vec4 |  |

|  |  |  |
| --- | --- | --- |
| lifeTime | float |  |

|  |  |  |
| --- | --- | --- |
| position | glm::vec2 |  |

|  |  |  |
| --- | --- | --- |
| rotation | float |  |

|  |  |  |
| --- | --- | --- |
| sizeBegin | float |  |

|  |  |  |
| --- | --- | --- |
| sizeEnd | float |  |

|  |  |  |
| --- | --- | --- |
| velocity | glm::vec2 |  |

##### **Physics**



Obrázek 10 - Physics

###### **Class PhysicsSystem**

@class PhysicsSytem

@brief Represents system for handling physics

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| m\_Components | std::vector<Component> |  |

|  |  |  |
| --- | --- | --- |
| m\_Gravity | float |  |

|  |  |  |
| --- | --- | --- |
| **Název metody** | **Návratový typ** | **Popis** |
| Add | void | Parametry:  **entity: Entity** - |

|  |  |  |
| --- | --- | --- |
| Contains | bool | Parametry:  **entity: Entity** - |

|  |  |  |
| --- | --- | --- |
| PhysicsSystem |  |  |

|  |  |  |
| --- | --- | --- |
| Remove | void | Parametry:  **entity: Entity** - |

|  |  |  |
| --- | --- | --- |
| Update | void | Parametry:  **dt: float** - |

##### **Renderer**



Obrázek 11 - Renderer

###### **Class Animation**

@struct Animation

@brief encapsulation for animation data

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| currentFrame | int |  |

|  |  |  |
| --- | --- | --- |
| frameInterval | std::pair<int, int> |  |

|  |  |  |
| --- | --- | --- |
| frameLen | float |  |

###### **Class BufferElement**

@struct BufferElement Store information about buffer element Each element contains information about it's size in vertex buffer. The vertex buffers can store only raw data, the buffer element let us use custom ShaderDataType values in the vertex buffers.

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| Divisor | unsigned int |  |

|  |  |  |
| --- | --- | --- |
| Index | unsigned int |  |

|  |  |  |
| --- | --- | --- |
| Offset | unsigned int |  |

|  |  |  |
| --- | --- | --- |
| Size | unsigned int |  |

|  |  |  |
| --- | --- | --- |
| Type | ShaderDataType |  |

|  |  |  |
| --- | --- | --- |
| **Název metody** | **Návratový typ** | **Popis** |
| BufferElement |  | Parametry:  **index: unsigned int** -  Parametry:  **type: ShaderDataType** -  Parametry:  **name: std::string&** -  Parametry:  **divisor: unsigned int** - |

|  |  |  |
| --- | --- | --- |
| GetComponentCount | uint32\_t |  |

###### **Class BufferLayout**

@class BufferLayout Represents layout of data in buffer. Consists of multiple BufferElements, stored in vector, let us structure data in the vertex buffer.

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| m\_Elements | std::vector<BufferElement> |  |

|  |  |  |
| --- | --- | --- |
| m\_Stride | unsigned int |  |

|  |  |  |
| --- | --- | --- |
| **Název metody** | **Návratový typ** | **Popis** |
| begin | auto |  |

|  |  |  |
| --- | --- | --- |
| begin | auto |  |

|  |  |  |
| --- | --- | --- |
| BufferLayout |  |  |

|  |  |  |
| --- | --- | --- |
| BufferLayout |  | Parametry:  **elements: std::initializer\_list<BufferElement>&** - |

|  |  |  |
| --- | --- | --- |
| CalculateOffsetsAndStride | void |  |

|  |  |  |
| --- | --- | --- |
| CreateMat4 | void |  |

|  |  |  |
| --- | --- | --- |
| end | auto |  |

|  |  |  |
| --- | --- | --- |
| end | auto |  |

|  |  |  |
| --- | --- | --- |
| GetElements | std::vector<BufferElement>& |  |

|  |  |  |
| --- | --- | --- |
| GetStride | uint32\_t& |  |

###### **Class Command**

@class Command

@brief represents command, stores void function and it's arguments.

@tparam[in] ...Args Pack of the arguments of the function

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| m\_Args | std::tuple<Args...> |  |

|  |  |  |
| --- | --- | --- |
| **Název metody** | **Návratový typ** | **Popis** |
| Command |  | Parametry:  **prm1: void** -  Parametry:  **args: Args...** - |

|  |  |  |
| --- | --- | --- |
| Execute | void |  |

###### **Class CommandI**

@interface CommandI pure virtual (interface) class.

| **Název metody** | **Návratový typ** | **Popis** |
| --- | --- | --- |
| Execute | void |  |

###### **Class IndexBuffer**

@interface IndexBuffer pure virtual (interface) class. Storage of the indices, send them to the GPU for further processing. The GPU use stored indices for indexing the vertices in the vertex buffer

| **Název metody** | **Návratový typ** | **Popis** |
| --- | --- | --- |
| Bind | void |  |

|  |  |  |
| --- | --- | --- |
| Create | std::shared\_ptr<IndexBuffer> | Parametry:  **indices: uint32\_t\*** -  Parametry:  **count: unsigned int** - |

|  |  |  |
| --- | --- | --- |
| GetCount | unsigned int |  |

|  |  |  |
| --- | --- | --- |
| ~IndexBuffer |  |  |

|  |  |  |
| --- | --- | --- |
| UnBind | void |  |

###### **Class MaterialManager**

@class MaterialManager

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| m\_AvailableIDs | std::queue<int16\_t> |  |

|  |  |  |
| --- | --- | --- |
| m\_Materials | std::unordered\_map<int16\_t, std::shared\_ptr<Material>> | Temporary untill we have asset manager |

|  |  |  |
| --- | --- | --- |
| m\_MaterialsInExistence | int16\_t |  |

|  |  |  |
| --- | --- | --- |
| s\_Instance | MaterialManager |  |

|  |  |  |
| --- | --- | --- |
| sc\_MaxNumberOfMaterials | int16\_t |  |

|  |  |  |
| --- | --- | --- |
| **Název metody** | **Návratový typ** | **Popis** |
| Get | MaterialManager& |  |

|  |  |  |
| --- | --- | --- |
| GetMaterial | std::shared\_ptr<Material> | Parametry:  **id: int16\_t** - |

|  |  |  |
| --- | --- | --- |
| MaterialManager |  |  |

|  |  |  |
| --- | --- | --- |
| MaterialManager |  | Parametry:  **prm1: MaterialManager&** - |

|  |  |  |
| --- | --- | --- |
| RegisterMaterial | int16\_t | Parametry:  **material: std::shared\_ptr<Material>** - |

|  |  |  |
| --- | --- | --- |
| RemoveMaterial | void | Parametry:  **id: int16\_t** - |

###### **Class OrthoCamera**

@class OrthoCamera

@brief represents camera for 2D scene, calculates projection and view matrix

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| m\_Position | glm::vec3 |  |

|  |  |  |
| --- | --- | --- |
| m\_ProjectionMatrix | glm::mat4 |  |

|  |  |  |
| --- | --- | --- |
| m\_Rotation | float |  |

|  |  |  |
| --- | --- | --- |
| m\_ViewMatrix | glm::mat4 |  |

|  |  |  |
| --- | --- | --- |
| m\_ViewProjectionMatrix | glm::mat4 |  |

|  |  |  |
| --- | --- | --- |
| **Název metody** | **Návratový typ** | **Popis** |
| GetPosition | glm::vec3& |  |

|  |  |  |
| --- | --- | --- |
| GetProjectionMatrix | glm::mat4& |  |

|  |  |  |
| --- | --- | --- |
| GetRotation | float |  |

|  |  |  |
| --- | --- | --- |
| GetViewMatrix | glm::mat4& |  |

|  |  |  |
| --- | --- | --- |
| GetViewProjectionMatrix | glm::mat4& |  |

|  |  |  |
| --- | --- | --- |
| OrthoCamera |  | Parametry:  **left: float** -  Parametry:  **right: float** -  Parametry:  **bottom: float** -  Parametry:  **top: float** - |

|  |  |  |
| --- | --- | --- |
| RecalculateViewMatrix | void |  |

|  |  |  |
| --- | --- | --- |
| SetPosition | void | Parametry:  **pos: glm::vec3&** - |

|  |  |  |
| --- | --- | --- |
| SetProjection | void | Parametry:  **left: float** -  Parametry:  **right: float** -  Parametry:  **bottom: float** -  Parametry:  **top: float** - |

|  |  |  |
| --- | --- | --- |
| SetRotation | void | Parametry:  **rot: float** - |

###### **Class OrthoCameraBounds**

@class OrthoCameraBound

@brief represents boundaries of view frustum of the camera

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| Bottom | float |  |

|  |  |  |
| --- | --- | --- |
| Left | float |  |

|  |  |  |
| --- | --- | --- |
| Right | float |  |

|  |  |  |
| --- | --- | --- |
| Top | float |  |

|  |  |  |
| --- | --- | --- |
| **Název metody** | **Návratový typ** | **Popis** |
| GetHeight | float |  |

|  |  |  |
| --- | --- | --- |
| GetWidth | float |  |

###### **Class OrthoCameraController**

@class OrthoCameraController

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| m\_AspectRatio | float |  |

|  |  |  |
| --- | --- | --- |
| m\_Bounds | OrthoCameraBounds |  |

|  |  |  |
| --- | --- | --- |
| m\_Camera | OrthoCamera |  |

|  |  |  |
| --- | --- | --- |
| m\_CameraPosition | glm::vec3 |  |

|  |  |  |
| --- | --- | --- |
| m\_CameraRotation | float | In degrees, in the anti-clockwise direction |

|  |  |  |
| --- | --- | --- |
| m\_CameraRotationSpeed | float |  |

|  |  |  |
| --- | --- | --- |
| m\_CameraTranslationSpeed | float |  |

|  |  |  |
| --- | --- | --- |
| m\_MouseScroll | HandlerID |  |

|  |  |  |
| --- | --- | --- |
| m\_Rotation | bool |  |

|  |  |  |
| --- | --- | --- |
| m\_WindowResize | HandlerID |  |

|  |  |  |
| --- | --- | --- |
| m\_ZoomLevel | float |  |

|  |  |  |
| --- | --- | --- |
| **Název metody** | **Návratový typ** | **Popis** |
| GetBounds | OrthoCameraBounds& |  |

|  |  |  |
| --- | --- | --- |
| GetCamera | OrthoCamera& |  |

|  |  |  |
| --- | --- | --- |
| GetCamera | OrthoCamera& |  |

|  |  |  |
| --- | --- | --- |
| GetZoomLevel | float |  |

|  |  |  |
| --- | --- | --- |
| OnMouseScrolled | void | Parametry:  **event: event\_ptr** - |

|  |  |  |
| --- | --- | --- |
| OnUpdate | void | Parametry:  **dt: float** - |

|  |  |  |
| --- | --- | --- |
| OnWindowResized | void | Parametry:  **event: event\_ptr** - |

|  |  |  |
| --- | --- | --- |
| OrthoCameraController |  | Parametry:  **aspectRatio: float** -  Parametry:  **rotation: bool** - |

|  |  |  |
| --- | --- | --- |
| ~OrthoCameraController |  |  |

|  |  |  |
| --- | --- | --- |
| SetZoomLevel | void | Parametry:  **level: float** - |

###### **Class Renderable2D**

@struct Renderable2D

@brief represents 2D renderable object.

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| color | glm::vec4 |  |

|  |  |  |
| --- | --- | --- |
| material | std::shared\_ptr<Material> |  |

|  |  |  |
| --- | --- | --- |
| position | glm::vec3 |  |

|  |  |  |
| --- | --- | --- |
| rotation | float |  |

|  |  |  |
| --- | --- | --- |
| size | glm::vec2 |  |

|  |  |  |
| --- | --- | --- |
| texCoord | glm::vec4 | (x = left, y = bottom, z = right, w = top) |

|  |  |  |
| --- | --- | --- |
| textureID | int |  |

|  |  |  |
| --- | --- | --- |
| visible | bool |  |

|  |  |  |
| --- | --- | --- |
| **Název metody** | **Návratový typ** | **Popis** |
| Renderable2D |  | Parametry:  **Material: std::shared\_ptr<Material>** -  Parametry:  **Color: glm::vec4&** -  Parametry:  **TexCoord: glm::vec4&** -  Parametry:  **Position: glm::vec3&** -  Parametry:  **Size: glm::vec2&** -  Parametry:  **Rotation: float** -  Parametry:  **Visible: bool** -  Parametry:  **TextureID: int** - |

###### **Class RenderCommand**

@class RenderCommand

@brief holds RendererAPI, provide static functions to call RendererAPI function

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| s\_RendererAPI | std::unique\_ptr<RendererAPI> |  |

|  |  |  |
| --- | --- | --- |
| **Název metody** | **Návratový typ** | **Popis** |
| Clear | void |  |

|  |  |  |
| --- | --- | --- |
| DrawIndexed | void | Parametry:  **vertexArray: std::shared\_ptr<VertexArray>&** -  Parametry:  **indexCount: uint32\_t** - |

|  |  |  |
| --- | --- | --- |
| DrawInstanced | void | Parametry:  **vertexArray: std::shared\_ptr<VertexArray>&** -  Parametry:  **count: uint32\_t** - |

|  |  |  |
| --- | --- | --- |
| Init | void |  |

|  |  |  |
| --- | --- | --- |
| SetClearColor | void | Parametry:  **color: glm::vec4&** - |

|  |  |  |
| --- | --- | --- |
| SetViewPort | void | Parametry:  **x: uint32\_t** -  Parametry:  **y: uint32\_t** -  Parametry:  **width: uint32\_t** -  Parametry:  **height: uint32\_t** - |

###### **Class RenderCommandQueue**

class represents queue of the CommandI commands, stores them in continuous block of memory. Can loop through them and execute them

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| m\_CommandBuffer | unsigned char\* |  |

|  |  |  |
| --- | --- | --- |
| m\_CommandBufferPtr | unsigned char\* |  |

|  |  |  |
| --- | --- | --- |
| m\_CommandCount | unsigned int |  |

|  |  |  |
| --- | --- | --- |
| **Název metody** | **Návratový typ** | **Popis** |
| Allocate | void | Parametry:  **cmd: CommandI\*** -  Parametry:  **size: unsigned int** - |

|  |  |  |
| --- | --- | --- |
| Clear | void |  |

|  |  |  |
| --- | --- | --- |
| Execute | void |  |

|  |  |  |
| --- | --- | --- |
| RenderCommandQueue |  |  |

|  |  |  |
| --- | --- | --- |
| ~RenderCommandQueue |  |  |

###### **Class Renderer**

@class Renderer

@brief represents encapsulation for systems, that takes care of sorting and rendering objects

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| m\_CommandQueue | RenderCommandQueue |  |

|  |  |  |
| --- | --- | --- |
| s\_Instance | Renderer\* |  |

|  |  |  |
| --- | --- | --- |
| s\_SceneData | SceneData\* |  |

|  |  |  |
| --- | --- | --- |
| **Název metody** | **Návratový typ** | **Popis** |
| BeginScene | void | Parametry:  **camera: OrthoCamera&** - |

|  |  |  |
| --- | --- | --- |
| EndScene | void |  |

|  |  |  |
| --- | --- | --- |
| Flush | void |  |

|  |  |  |
| --- | --- | --- |
| GetAPI | RendererAPI::API |  |

|  |  |  |
| --- | --- | --- |
| Init | void |  |

|  |  |  |
| --- | --- | --- |
| OnWindowResize | void | Parametry:  **width: uint32\_t** -  Parametry:  **height: uint32\_t** - |

|  |  |  |
| --- | --- | --- |
| Submit | void | Parametry:  **command: CommandI&** -  Parametry:  **size: unsigned int** - |

###### **Class RendererAPI**

@interface RendererAPI pure virtual (interface) class.

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| s\_API | API |  |

|  |  |  |
| --- | --- | --- |
| **Název metody** | **Návratový typ** | **Popis** |
| Clear | void |  |

|  |  |  |
| --- | --- | --- |
| Create | std::unique\_ptr<RendererAPI> |  |

|  |  |  |
| --- | --- | --- |
| DrawIndexed | void | Parametry:  **vertexArray: std::shared\_ptr<VertexArray>&** -  Parametry:  **indexCount: uint32\_t** - |

|  |  |  |
| --- | --- | --- |
| DrawInstanced | void | Parametry:  **vertexArray: std::shared\_ptr<VertexArray>&** -  Parametry:  **count: uint32\_t** - |

|  |  |  |
| --- | --- | --- |
| GetAPI | API |  |

|  |  |  |
| --- | --- | --- |
| Init | void |  |

|  |  |  |
| --- | --- | --- |
| SetClearColor | void | Parametry:  **color: glm::vec4&** - |

|  |  |  |
| --- | --- | --- |
| SetViewport | void | Parametry:  **x: uint32\_t** -  Parametry:  **y: uint32\_t** -  Parametry:  **width: uint32\_t** -  Parametry:  **height: uint32\_t** - |

###### **Class RendererBatchSystem2D**

@class RendererBatchSystem2D

@brief represents system , that groups ( batches ) renderables dependent on the material and z distance from the camera

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| m\_Components | std::vector<Component> |  |

|  |  |  |
| --- | --- | --- |
| m\_OpaqueGroup | SortingGroup<ZDescend> | Opaque group z descend ordering |

|  |  |  |
| --- | --- | --- |
| m\_TransparentGroup | SortingGroup<ZAscend> | Transparent group z ascend ordering |

|  |  |  |
| --- | --- | --- |
| **Název metody** | **Návratový typ** | **Popis** |
| Add | void | Parametry:  **entity: Entity** - |

|  |  |  |
| --- | --- | --- |
| Contains | bool | Parametry:  **entity: Entity** - |

|  |  |  |
| --- | --- | --- |
| EntityUpdated | void | If renderable is updated and keys do not match, reinsert it and update key  Parametry:  **entity: Entity** - |

|  |  |  |
| --- | --- | --- |
| Remove | void | Parametry:  **entity: Entity** - |

|  |  |  |
| --- | --- | --- |
| RendererBatchSystem2D |  |  |

|  |  |  |
| --- | --- | --- |
| SubmitToRenderer | void |  |

###### **Class Routine**

@struct Routine

@brief represents routine of the shader

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| activeSubRoutine | SubRoutine |  |

|  |  |  |
| --- | --- | --- |
| subRoutines | std::vector<SubRoutine> |  |

###### **Class Shader**

@class Shader

@brief Shader encapsulates graphics API shader program. Creates abstraction above shader programs

| **Název metody** | **Návratový typ** | **Popis** |
| --- | --- | --- |
| AddReloadCallback | void | Parametry:  **callback: std::function<void()>** - |

|  |  |  |
| --- | --- | --- |
| Bind | void |  |

|  |  |  |
| --- | --- | --- |
| Compute | void | Parametry:  **groupX: unsigned int** -  Parametry:  **groupY: unsigned int** -  Parametry:  **groupZ: unsigned int** - |

|  |  |  |
| --- | --- | --- |
| Create | std::shared\_ptr<Shader> | Parametry:  **path: std::string&** - |

|  |  |  |
| --- | --- | --- |
| Create | std::shared\_ptr<Shader> | Parametry:  **name: std::string&** -  Parametry:  **path: std::string&** - |

|  |  |  |
| --- | --- | --- |
| FindTexture | TextureUniform\* | Parametry:  **name: std::string&** - |

|  |  |  |
| --- | --- | --- |
| FindUniform | Uniform\* | Parametry:  **name: std::string&** - |

|  |  |  |
| --- | --- | --- |
| GetName | std::string |  |

|  |  |  |
| --- | --- | --- |
| GetPath | std::string |  |

|  |  |  |
| --- | --- | --- |
| GetUniformSize | unsigned int |  |

|  |  |  |
| --- | --- | --- |
| Reload | void |  |

|  |  |  |
| --- | --- | --- |
| SetSubRoutine | void | Parametry:  **name: std::string&** - |

|  |  |  |
| --- | --- | --- |
| SetUniforms | void | Parametry:  **buffer: unsigned char\*** - |

|  |  |  |
| --- | --- | --- |
| ~Shader |  |  |

|  |  |  |
| --- | --- | --- |
| Unbind | void |  |

|  |  |  |
| --- | --- | --- |
| UploadRoutines | void |  |

###### **Class ShaderStorageBuffer**

@interface ShaderStorageBuffer pure virtual (interface) class. Storage of the data, can be processed by compute shaders.

| **Název metody** | **Návratový typ** | **Popis** |
| --- | --- | --- |
| Bind | void |  |

|  |  |  |
| --- | --- | --- |
| BindRange | void | Parametry:  **offset: uint32\_t** -  Parametry:  **size: uint32\_t** -  Parametry:  **index: uint32\_t** - |

|  |  |  |
| --- | --- | --- |
| Create | std::shared\_ptr<ShaderStorageBuffer> | Parametry:  **size: uint32\_t** - |

|  |  |  |
| --- | --- | --- |
| Create | std::shared\_ptr<ShaderStorageBuffer> | Parametry:  **vertices: float\*** -  Parametry:  **size: uint32\_t** -  Parametry:  **usage: BufferUsage** - |

|  |  |  |
| --- | --- | --- |
| GetLayout | BufferLayout& |  |

|  |  |  |
| --- | --- | --- |
| Resize | void | Parametry:  **vertices: void\*** -  Parametry:  **size: uint32\_t** - |

|  |  |  |
| --- | --- | --- |
| SetLayout | void | Parametry:  **layout: BufferLayout&** - |

|  |  |  |
| --- | --- | --- |
| ~ShaderStorageBuffer |  |  |

|  |  |  |
| --- | --- | --- |
| Update | void | Parametry:  **vertices: void\*** -  Parametry:  **size: uint32\_t** -  Parametry:  **offset: uint32\_t** - |

###### **Class SortingGroup**

@class SortingGroup

@brief Sort renderables dependent on specfied Comparator

@tparam T Comparator

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| m\_Renderables | RenderablesGroup |  |

|  |  |  |
| --- | --- | --- |
| **Název metody** | **Návratový typ** | **Popis** |
| AddRenderable | void | Parametry:  **renderable: Renderable2D\*** - |

|  |  |  |
| --- | --- | --- |
| GetRenderables | RenderablesGroup& |  |

|  |  |  |
| --- | --- | --- |
| RemoveRenderable | void | Parametry:  **renderable: Renderable2D\*** - |

###### **Class SpriteAnimation**

@class SpriteAnimation

@brief handles sprite animation, updates Renderable2D texture coordinates, texture is split to single sized quads, each quad representing single frame.

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| m\_Animation | Animation |  |

|  |  |  |
| --- | --- | --- |
| m\_AnimationLen | float |  |

|  |  |  |
| --- | --- | --- |
| m\_Columns | int |  |

|  |  |  |
| --- | --- | --- |
| m\_CurrentTime | float |  |

|  |  |  |
| --- | --- | --- |
| m\_Height | uint32\_t |  |

|  |  |  |
| --- | --- | --- |
| m\_NumFrames | int |  |

|  |  |  |
| --- | --- | --- |
| m\_Rows | int |  |

|  |  |  |
| --- | --- | --- |
| m\_Width | uint32\_t |  |

|  |  |  |
| --- | --- | --- |
| **Název metody** | **Návratový typ** | **Popis** |
| calcTexCoords | glm::vec4 |  |

|  |  |  |
| --- | --- | --- |
| SetFrameInterval | void | Parametry:  **first: int** -  Parametry:  **last: int** -  Parametry:  **timeFrame: float** - |

|  |  |  |
| --- | --- | --- |
| SpriteAnimation |  | Parametry:  **numRows: int** -  Parametry:  **numCols: int** -  Parametry:  **width: uint32\_t** -  Parametry:  **height: uint32\_t** - |

|  |  |  |
| --- | --- | --- |
| Update | void | Parametry:  **dt: float** -  Parametry:  **sprite: Renderable2D\*** - |

###### **Class SpriteAnimationController**

@class SpriteAnimationController

@brief Stores and updates animations

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| m\_Animations | std::unordered\_map<std::string, Animation> |  |

|  |  |  |
| --- | --- | --- |
| m\_Current | Animation\* |  |

|  |  |  |
| --- | --- | --- |
| **Název metody** | **Návratový typ** | **Popis** |
| AddAnimation | void | Parametry:  **name: std::string&** -  Parametry:  **anim: Animation** - |

|  |  |  |
| --- | --- | --- |
| StartAnimation | void | Parametry:  **name: std::string&** - |

|  |  |  |
| --- | --- | --- |
| UpdateSpriteAnimation | void | Parametry:  **anim: SpriteAnimation\*** - |

###### **Class SpriteSystem**

@class SpriteSystem

@brief Updates entities with components Renderable2D and SpriteAnimation

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| m\_Components | std::vector<Component> |  |

|  |  |  |
| --- | --- | --- |
| **Název metody** | **Návratový typ** | **Popis** |
| Add | void | Parametry:  **entity: Entity** - |

|  |  |  |
| --- | --- | --- |
| Contains | bool | Parametry:  **entity: Entity** - |

|  |  |  |
| --- | --- | --- |
| Remove | void | Parametry:  **entity: Entity** - |

|  |  |  |
| --- | --- | --- |
| SpriteSystem |  |  |

|  |  |  |
| --- | --- | --- |
| Update | void | Parametry:  **dt: float** - |

###### **Class SubRoutine**

@struct SubRoutine

@brief represents sub routine of the shader

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| index | unsigned int |  |

|  |  |  |
| --- | --- | --- |
| name | std::string |  |

|  |  |  |
| --- | --- | --- |
| shaderType | unsigned int |  |

###### **Class Texture**

@interface Texture pure virtual (interface) class.

| **Název metody** | **Návratový typ** | **Popis** |
| --- | --- | --- |
| Bind | void | Parametry:  **slot: uint32\_t** - |

|  |  |  |
| --- | --- | --- |
| GetHeight | uint32\_t |  |

|  |  |  |
| --- | --- | --- |
| GetWidth | uint32\_t |  |

|  |  |  |
| --- | --- | --- |
| SetData | void | Parametry:  **data: void\*** -  Parametry:  **size: uint32\_t** - |

|  |  |  |
| --- | --- | --- |
| ~Texture |  |  |

###### **Class Texture2D**

@class Texture2D

@brief class Derived from Texture

| **Název metody** | **Návratový typ** | **Popis** |
| --- | --- | --- |
| Create | std::shared\_ptr<Texture2D> | Parametry:  **format: TextureFormat** -  Parametry:  **wrap: TextureWrap** -  Parametry:  **width: uint32\_t** -  Parametry:  **height: uint32\_t** - |

|  |  |  |
| --- | --- | --- |
| Create | std::shared\_ptr<Texture2D> | Parametry:  **wrap: TextureWrap** -  Parametry:  **path: std::string&** - |

###### **Class TextureUniform**

@struct TextureUniform

@brief represents texture uniform of the shader

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| count | unsigned int |  |

|  |  |  |
| --- | --- | --- |
| location | uint32\_t |  |

|  |  |  |
| --- | --- | --- |
| name | std::string |  |

|  |  |  |
| --- | --- | --- |
| slot | unsigned int |  |

###### **Class Uniform**

@struct Uniform

@brief represents uniform of the shader

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| count | unsigned int |  |

|  |  |  |
| --- | --- | --- |
| isArray | bool |  |

|  |  |  |
| --- | --- | --- |
| location | uint32\_t |  |

|  |  |  |
| --- | --- | --- |
| name | std::string |  |

|  |  |  |
| --- | --- | --- |
| offset | unsigned int |  |

|  |  |  |
| --- | --- | --- |
| size | unsigned int |  |

|  |  |  |
| --- | --- | --- |
| type | UniformDataType |  |

###### **Class VertexArray**

@interface VertexArray pure virtual (interface) class.

| **Název metody** | **Návratový typ** | **Popis** |
| --- | --- | --- |
| AddShaderStorageBuffer | void | Parametry:  **shaderBuffer: std::shared\_ptr<ShaderStorageBuffer>&** - |

|  |  |  |
| --- | --- | --- |
| AddVertexBuffer | void | Parametry:  **vertexBuffer: std::shared\_ptr<VertexBuffer>&** - |

|  |  |  |
| --- | --- | --- |
| Bind | void |  |

|  |  |  |
| --- | --- | --- |
| Create | std::shared\_ptr<VertexArray> |  |

|  |  |  |
| --- | --- | --- |
| GetIndexBuffer | std::shared\_ptr<IndexBuffer>& |  |

|  |  |  |
| --- | --- | --- |
| GetVertexBuffer | std::vector<std::shared\_ptr<VertexBuffer>>& |  |

|  |  |  |
| --- | --- | --- |
| SetIndexBuffer | void | Parametry:  **indexBuffer: std::shared\_ptr<IndexBuffer>&** - |

|  |  |  |
| --- | --- | --- |
| ~VertexArray |  |  |

###### **Class VertexBuffer**

@interface VertexBuffer pure virtual (interface) class. Storage of data for rendering. Send the data to the GPU for further processing. Vertices are rendered by shader program. VertexArray stores VertexBuffers, must have BufferLayout set before being stored.

| **Název metody** | **Návratový typ** | **Popis** |
| --- | --- | --- |
| Bind | void |  |

|  |  |  |
| --- | --- | --- |
| Create | std::shared\_ptr<VertexBuffer> | Parametry:  **size: uint32\_t** - |

|  |  |  |
| --- | --- | --- |
| Create | std::shared\_ptr<VertexBuffer> | Parametry:  **vertices: float\*** -  Parametry:  **size: uint32\_t** -  Parametry:  **usage: BufferUsage** - |

|  |  |  |
| --- | --- | --- |
| GetLayout | BufferLayout& |  |

|  |  |  |
| --- | --- | --- |
| Resize | void | Parametry:  **vertices: float\*** -  Parametry:  **size: uint32\_t** - |

|  |  |  |
| --- | --- | --- |
| SetLayout | void | Parametry:  **layout: BufferLayout&** - |

|  |  |  |
| --- | --- | --- |
| UnBind | void |  |

|  |  |  |
| --- | --- | --- |
| Update | void | Parametry:  **vertices: void\*** -  Parametry:  **size: uint32\_t** -  Parametry:  **offset: uint32\_t** - |

|  |  |  |
| --- | --- | --- |
| ~VertexBuffer |  |  |

###### **Enumeration BufferUsage**

Represents usage of the vertex buffer If Static, the data is not expected to be updated, if Dynamic, the data is expected to be updated

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| None |  |  |

|  |  |  |
| --- | --- | --- |
| Static |  |  |

|  |  |  |
| --- | --- | --- |
| Dynamic |  |  |

###### **Enumeration RenderFlags**

enum class represents flags for the material sort key

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| MaterialFlag |  |  |

|  |  |  |
| --- | --- | --- |
| TransparentFlag |  |  |

|  |  |  |
| --- | --- | --- |
| InstancedFlag |  |  |

|  |  |  |
| --- | --- | --- |
| LayerFlag |  |  |

###### **Enumeration ShaderDataType**

Represents data types in shader program

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| None |  |  |

|  |  |  |
| --- | --- | --- |
| Float |  |  |

|  |  |  |
| --- | --- | --- |
| Float2 |  |  |

|  |  |  |
| --- | --- | --- |
| Float3 |  |  |

|  |  |  |
| --- | --- | --- |
| Float4 |  |  |

|  |  |  |
| --- | --- | --- |
| Mat3 |  |  |

|  |  |  |
| --- | --- | --- |
| Mat4 |  |  |

|  |  |  |
| --- | --- | --- |
| Int |  |  |

|  |  |  |
| --- | --- | --- |
| Int2 |  |  |

|  |  |  |
| --- | --- | --- |
| Int3 |  |  |

|  |  |  |
| --- | --- | --- |
| Int4 |  |  |

|  |  |  |
| --- | --- | --- |
| Bool |  |  |

###### **Enumeration UniformDataType**

enum class represents uniform data types

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| NONE |  |  |

|  |  |  |
| --- | --- | --- |
| SAMPLER2D |  |  |

|  |  |  |
| --- | --- | --- |
| SAMPLERCUBE |  |  |

|  |  |  |
| --- | --- | --- |
| BOOL |  |  |

|  |  |  |
| --- | --- | --- |
| FLOAT |  |  |

|  |  |  |
| --- | --- | --- |
| FLOAT\_VEC2 |  |  |

|  |  |  |
| --- | --- | --- |
| FLOAT\_VEC3 |  |  |

|  |  |  |
| --- | --- | --- |
| FLOAT\_VEC4 |  |  |

|  |  |  |
| --- | --- | --- |
| INT |  |  |

|  |  |  |
| --- | --- | --- |
| INT\_VEC2 |  |  |

|  |  |  |
| --- | --- | --- |
| INT\_VEC3 |  |  |

|  |  |  |
| --- | --- | --- |
| INT\_VEC4 |  |  |

|  |  |  |
| --- | --- | --- |
| FLOAT\_MAT4 |  |  |

|  |  |  |
| --- | --- | --- |
| STRUCT |  |  |

##### **Utils**

###### **DataStructures**



Obrázek 12 - DataStructures

## **Package Diagram**



Obrázek 13 - Package Diagram

### **API**

### **Asset Manager**

### **Assets**

### **Business Logic**

### **Core**

### **Data**

### **ECS**

### **Particle**

### **Presentation**

### **Renderer**

### **Physics**

### **Renderer**

## **Save/Load Diagram**



Obrázek 14 - Save/Load diagram

### **Class Asset Package**

Present in filesystem in packed or unpacked format

### **Class Editor**

! @class AssetManager

@brief The class to manage game assets

| **Název atributu** | **Datový typ** | **Popis** |
| --- | --- | --- |
| ASSETS\_FOLDER | std::string | Path to the folder with assets |

|  |  |  |
| --- | --- | --- |
| m\_AssetIdCounter | uint16\_t | Counter to track next free id to assign to an asset |

|  |  |  |
| --- | --- | --- |
| m\_FreedIds | std::queue<uint16\_t> | Queue with ids of uloaded assets |

|  |  |  |
| --- | --- | --- |
| m\_Instance | std::unique\_ptr<Editor> | Instance of the class |

|  |  |  |
| --- | --- | --- |
| m\_Shaders | std::unordered\_map<std::string, std::pair<uint16\_t, std::shared\_ptr<XYZ::Shader>>> | Shaders collection |

|  |  |  |
| --- | --- | --- |
| m\_Textures | std::unordered\_map<std::string, std::pair<uint16\_t, std::shared\_ptr<XYZ::Texture2D>>> | Textures collection |

|  |  |  |
| --- | --- | --- |
| **Název metody** | **Návratový typ** | **Popis** |
| Editor |  |  |

|  |  |  |
| --- | --- | --- |
| Get | std::unique\_ptr<Editor>& |  |

|  |  |  |
| --- | --- | --- |
| SaveShader | std::bool | Parametry:  **name: std::string&** - |

|  |  |  |
| --- | --- | --- |
| SaveTexture | std::bool | Parametry:  **name: std::string&** - |