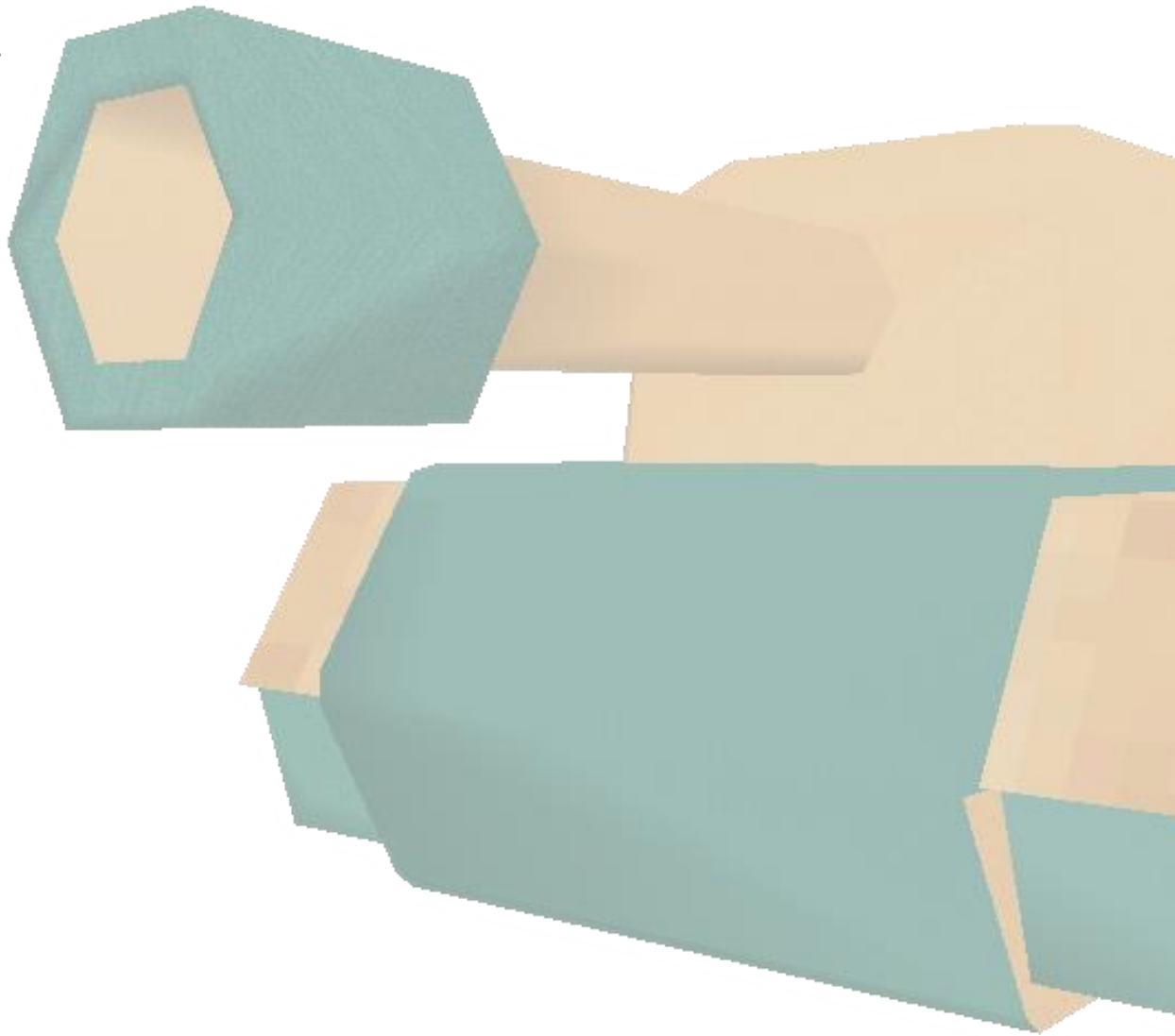


Design.

MyBuasIntake

Tygo Boons.



Concept

I will be creating a game as my intake assignment for my next course as a software developer. This game will be created using the C++ programming languages together with the minimal amount of necessary libraries and the required intake vs2022 template that is provided.

Engine:

The game-engine will support simple 3D rendering. lack of a user-interface most of the components will be read by using color reference maps. Things like tile rendering and collision maps will use these color reference maps so that the user can easily change and create levels and collisions. Furthermore, the engine's 2D rendering, Text rendering and other features like sounds and texture loading will be handled by a third-party library. The 3D part of the engine will be written from scratch.

Gameplay:

It will be a 2-dimensional game where you play as a tank. You can move the tank around with your keyboard and move the gun on top of the tank with your mouse. The field you play on is a small level that consists of walls and other AI-controlled tanks. The objective of the game is to shoot and defeat all the AI tanks. You do this by shooting at them and hitting them with your bullets. However, the AI tanks will shoot back and try to hit you too. The level is completed if all the tanks are defeated, after which you will go on to the next level. The art-style will be decided by the assets that are freely available, as I am not qualified enough to make high-quality assets, and it is not the main goal of this project. Therefore this will be updated later in the project.

Note: The game is heavily based on the tanks minigame in the game Wii Play.

Gameplay example: <https://www.youtube.com/watch?v=orLxrg51xL8>

Technical design

This page will contain all the tools and programs/libraries that will be used to create this project.

Visual studio 2022

Visual studio 2022 is the IDE (Integrated development environment) that I will write all the C++ code in. It includes a compiler and all the tools required to develop software. It is also widely used in the professional industry.

<https://visualstudio.microsoft.com/>

C++

C++ will be the programming language used. It is widely used in the game industry and most applications are created using it.

GitHub - GitKraken

GitHub will be used for source control together with GitKraken as Git GUI application.

<https://github.com/> <https://www.gitkraken.com/>

BUAS vs2022 Template

This template is required for the intake assignment. I will probably be stripping most of the parts out. The template can be downloaded with this link.

<https://www.buas.nl/documents/portfolio-programming>

SFML

Sfml is a graphics library for c++, this will be used as for input, window creation and rendering.

<https://www.sfml-dev.org/>

GLM

Glm is a math library commonly used in c++. It will be used for matrix multiplication and vector maths.

<https://github.com/g-truc/glm>

GLAD

Glad will be used as a header to load OpenGL methods.

<https://glad.dav1d.de/>



Gameplay

This page will go over the gameplay elements that will be (not)included in the project, a detailed description of the mechanics that have to be created and general details about gameplay-specific elements of the game. The bounce mechanic of the game will be represented by bullet Type 1 being able to bounce ricochet off of the walls once.

Camera view

The camera position will be in the center of the level looking down at the playfield as a top view. The whole level can be seen from this point of view.

What will be possible.

- Player tank
 - The player tank can be controlled using the keyboard. The player can move up down left and right
 - The gun on the player's tank can be rotated by moving the mouse, the gun points at the mouse direction relative to the player's position.
 - The gun on the player's tank can shoot a bullet in the direction the gun is currently facing by using the mouse buttons.
 - The gun on the player's tank can shoot 3 bullets fast, after which there is a small delay.
- Bullet
 - Bullet gets destroyed if it contacts any type of tank or hits a wall if it does not have any ricochets left.
 - Type 1
 - Bullet will travel at a fixed slow speed.
 - Bullet can ricochet off of any surface once.
 - Type 2
 - Bullet will travel fast.
 - Bullet can not ricochet off any surface.
- Wall
 - Wall will be collidable.
- Enemy Tank
 - Tank is AI controlled
 - Tank can do everything the player tank can also do.
 - Type 1
 - Uses bullet type 1.
 - Type 2
 - Uses bullet type 2.
 - Type 3
 - Uses bullet type 2.
 - Moves faster than other tank types.

MyBuasIntake

What might be possible.

- A UI start screen.
- 3D Instead of 2D rendering.
- UI elements.
 - Amount of tanks left.
 - Current score.
- Basic audio.
 - Soundtrack
 - Tank movement sounds.
 - Bullet sounds.

What will not be possible.

- More than 10 levels.
- More than 3 types of enemy AI tanks.
- Advanced AI controlled tanks.



Engine components

This page will contain the required engine components to make this engine flexible enough to properly build the game in.

Must have.

- 3D Mesh rendering
- Textures
- Color reference mapping (to read things like levels and collider positions etc.)
- Collisions
 - Only AABB Rectangle collisions.
 - AABB's can be rotated.
- Input
 - Keyboard
 - Mouse
- Entity component system
 - Transform component.
 - MeshRenderer / SpriteRenderer component.
 - CameraComponent

Might have.

- Entity component system
 - UiTextComponent.
 - UiImageComponent.
- UI rendering layer.
- 3D rendering.
- Special effects layer.
 - Vignette.
 - Bloom.
 - Color filters.

Will not have.

- Lighting

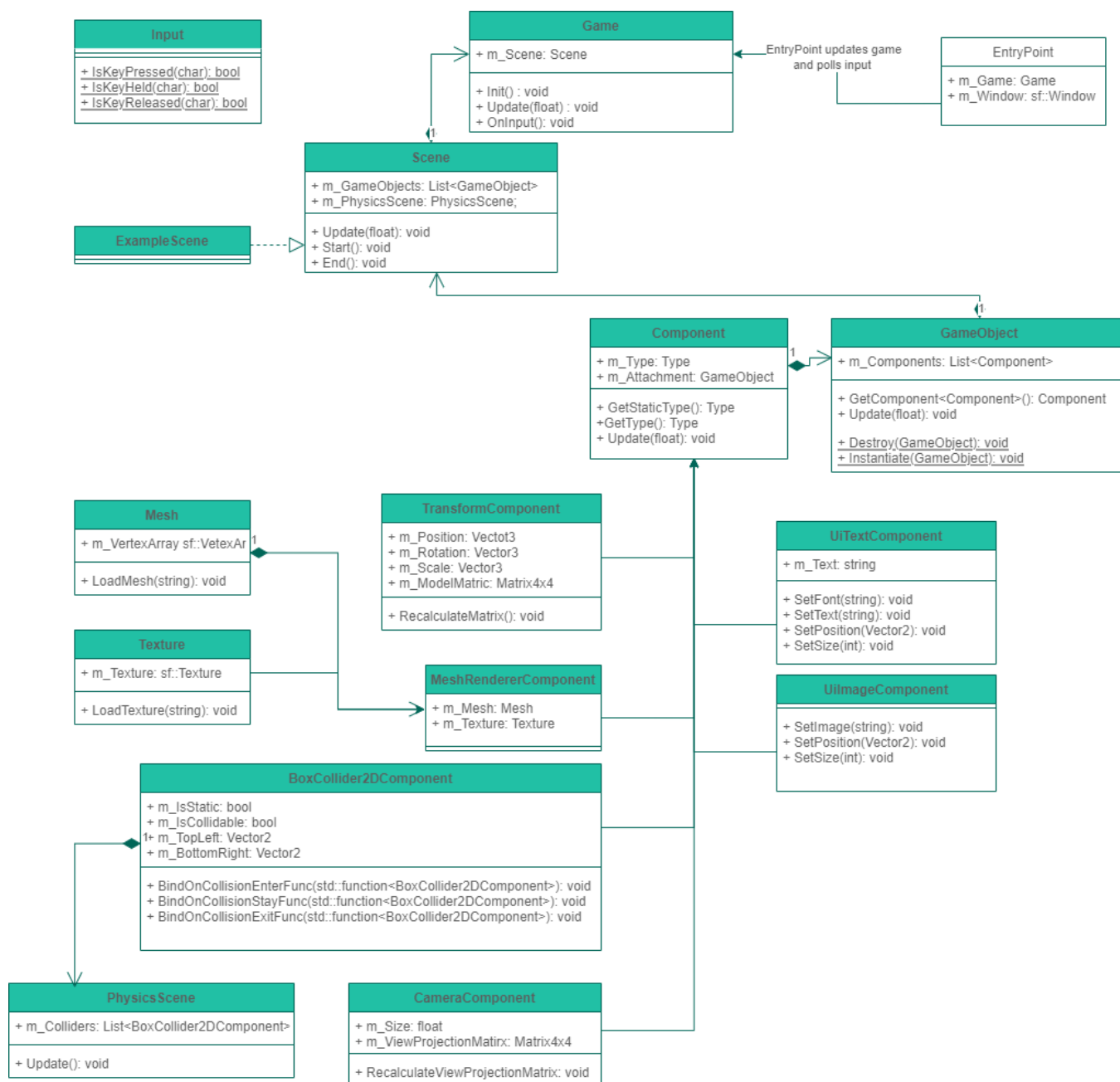
Engine Functional Design

This page will contain the more code-specific architecture of the engine. Going in to detail about which tools have to be created and how the classes fit together.

UML

Uml diagram gives you a rough understanding of the class hierarchy and architecture of in this case the engine part of this project.

Source: https://github.com/TygoB-B5/MyBuasIntake/blob/main/MyBuasIntake_UML



Gameplay Functional Design

This page will contain the code-specific architecture of the gameplay elements required in the game.

UML

Uml diagram gives you a rough understanding of the class hierarchy and architecture of in this case the gameplay specific parts of this project.

Source: https://github.com/TygoB-B5/MyBuasIntake/blob/main/MyBuasIntake_UML

