Documentation for BPUS

(You can consider changing view to web)

[Project repository](https://github.com/CinTheDev/BPUS)

# Contents

Platform specific

[win32\_platform](BPUS/Platform%20specific/win32_platform.docx)

Renderer

[renderer](BPUS/Renderer/renderer.docx)

Utilities

[utils](BPUS/Utils/utils.docx)

[vector](BPUS/Utils/vector.docx)

Image library

[Image](BPUS/Image/Image.docx)

Object

[Object](BPUS/Object/Object.docx)

[Object\_Manager](BPUS/Object/Object_Manager.docx)

Game

[game](BPUS/Game/game.docx)

[Object\_behavior](BPUS/Game/Object_behavior.docx)

[platform\_common](BPUS/Game/platform_common.docx)

Server

[Server](Server/Server.docx)

# 1 Introduction

## 1.1 Setting up the project

If you download the project from the repository, you’ll have to set the Linker to include the following libraries:

* asio (from include path)
* stb\_image (only stb\_image.h and stb\_image\_write.h)

If you do that everything should work.

## 1.2 General Overview

The code starts at [win32\_platform.cpp](BPUS/Platform%20specific/win32_platform.docx) and it includes most stuff. The game runs in the while(running) loop, if the loop ends, the game closes.

Stuff connected to input is in [platform\_common.cpp](BPUS/Game/platform_common.docx).

The game class is in [game.cpp](BPUS/Game/game.docx) and contains everything related to the inner game functionality.

Everything related to rendering is located in [renderer.cpp](BPUS/Renderer/renderer.docx), currently it currently contains only functions that draw specific things on the screen.

[utils.cpp](BPUS/Utils/utils.docx) has some useful functions that greatly improve code readability.

[hvector.h](BPUS/Utils/utils.docx) is a header-based file with four different structs representing vectors. These are used like mathematical vectors and points (Not like std::vector!).

All the classes for single objects are in [Object\_behavior.cpp](BPUS/Game/Object_behavior.docx).

[Object.h](BPUS/Object/Object.docx) contains the definition of the object class and is designed to be inherited from.

[Object\_Manager.h](BPUS/Object/Object_Manager.docx) contains a static class that manages adding, removing and updating objects.

[Image.h](BPUS/Image/Image.docx) reads and converts images to an array of single colors.

font.cpp is an array used for the draw\_text function located in [renderer.cpp](BPUS/Renderer/renderer.docx).

# 2 Workflows

## 2.1 Creating objects

1. Create a pointer (you shouldn’t remove it until proper deletion of object). The type should be the corresponding class from [Object\_behavior](BPUS/Game/Object_behavior.docx).
2. Initialize the pointer with *new*.
3. Call Obj\_M::create(*pointer*);

## 2.2 Deleting objects

1. Call Obj\_M::remove(*pointer*);
2. Delete the pointer. (Not doing it properly would cause a memory leak)