



# B4 - Year-End Project

---

B-YEP-400

## Indie Studio

---

Time to go cross-platform





# Indie Studio

binary name: bomberman  
language: C++  
compilation: CMake 3.17



- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

The aim of the **Indie Studio** is to implement a **cross-platform** 3D video game with **real-world** tools.

Making an enjoyable game is just as important as writing clean code.  
A working game which is no fun to play is pointless!



Let's be honest. Implementing a video game is generally not fun (at least not all the time).  
But it can be, and having someone else enjoy a game you made is an amazing feeling.  
The result will be worth the effort, so do your best! And most importantly, **have fun**.



## COMPILATION

---

You **MUST** use **CMake** as a cross-platform build system:

- **Linux:** it must generate a `Makefile` to be able to build the project
- **Windows:** it must generate a Visual Studio solution `.sln` (will be tested with Visual Studio 2019).



We will provide a `CMAKE_MODULE_PATH` cmake variable when building the project. Irrlicht 1.8.4 binaries and includes are provided for VS2019 v142 / SDK 10.0.18362.0

## LIBRARIES

---

You can choose between two library for this project:

- `raylib`, a very simple C library for 2D/3D graphics. As of today, 3D animation for `.glTF` models files is badly handled. We advised you to use multiple `.obj` files to animate your assets. `raylib` is a C library, ever heard of encapsulation ?
- `Irrlicht Engine`, a C++ 3D engine. Harder to use than `raylib`, but it fully supports some old 3D animations file formats. As Irrlicht doesn't provide audio module, you can use `SFML-audio 2.5` (and ONLY `-audio`) to play sounds and musics.

CMakes modules are provided for `raylib` and `Irrlicht`, you must use it to generate your project. You may use `Boost` to help you with some aspects of the project (or for bonuses).



There are plenty of free assets on the Internet. Creating your own should be far from being a priority... Do not waste time.

## THE GAME

---



The game you must implement is a Bomberman-like game.

**Bomberman** is one of the most famous video games of all time.

With over 70 franchises, ranging from the first version on MSX, ZX Spectrum and Sharp MZ-700 in 1983 to the latest versions on modern consoles, over 10 million units have been sold.

Our gameplay reference is **Neo Bomberman**, released on the Neo Geo and MVS systems in 1997.

Here's the required features :

- Local multi-player with the possibility to play with a friend + AI's controlled bots.
- A main game menu (to start a new game or load an existing one).
- Your game must feature 3D graphics but a 2D gameplay.
- Your game must feature **procedurally generated maps** (random obstacles, random bonus...).
- Save/Load a game
- Animation and sounds to give life to the game (especially during various interactions)
  - Think about bomb explosion, player walking, background music, ...
- Some Power-ups
  - Bomb Up
  - Speed Up
  - Fire Up
  - Wall Pass

The goal of this project is not only to code a video game, but it's to code a fully finished video game. This means that the efforts you make to polish your game will count as much as the technical aspect.

Do your best to avoid clunky animations or graphical glitches, manage your camera in a way that the "outside" of the game is never visible, pay attention to game design, level design and sound design, add variations in enemies and environments, etc.