CIS 29, Winter 2021
Prof. Bentley
Group #1

Tank Game Compile Instructions

Note 1: For the following instructions, we assume that you have the same file structure provided. If you are using a different file structure, you may need to change Maps::filePrefix(which is stored in Maps.h) appropriately.

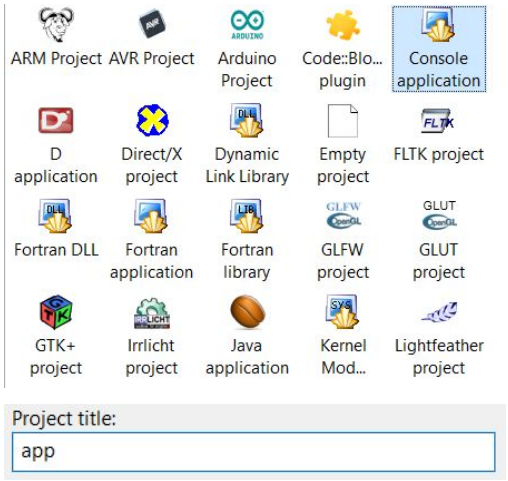
Note 2: The git repository does not contain the theme song wav file(due to it being too large). If cloning from version control, make sure to download the wav song from here: https://drive.google.com/file/d/115hatLqbAjB8-aqrWrl8g-f5q0ITc3k0/view

Note 3: The code was written for SFML 2.5.1

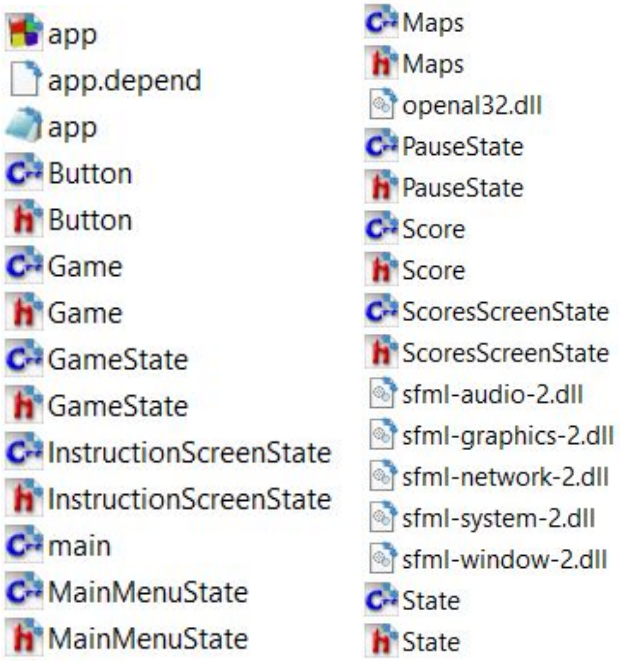
Note 4: You do not need the main.c file from the TankGraphics library, and you may delete it (you may have to make some additional changes/builds if you do so however).

Windows 10, Code::Blocks 20.03

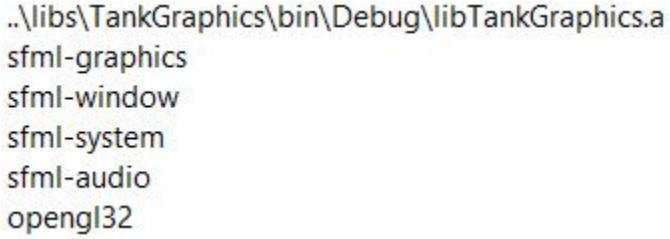
1. Firstly, either download the files or clone from version control using:
   $git clone https://github.com/yousefh409/tankGame.git
2. Next, create a new file directory that will hold a game. Now you want we want to create a C++ console application project in the game directory, and name it ``app``:



3. We then want to add all the files from the ``app`` folder you downloaded previously. **Make sure to add them through Code::Blocks**. This will assure that the appropriate compiler command is used. You should also copy the `sfml-audio-2.dll`, `sfml-graphics-2.dll`, `sfml-network-2.dll`, `sfml-system-2.dll`, `sfml-window-2.dll`, and `openal32.dll` files into the directory as well. After adding all the files, your ``app`` directory should contain the following files:



   Note: if you ran the project you will also find the ``bin`` and ``obj`` folders.
4. Next, make sure to copy the ``libs`` and ``data`` folder from the folder you downloaded previously into the game directory you created. Important: your ``libs``, ``data``, and ``app`` folder should all be at the same “level”
5. Next, you should download SFML into your ``libs`` directory from: https://www.sfml-dev.org/download/sfml/2.5.1/ (the Code::Blocks version is best suited). You should make sure it is named “SFML-2.5.1”
6. Next, we should link in the libraries to our Code::Blocks app that we created previously. Open the project in Code::Blocks, and go to Project => Build Options. In Linker Settings, you want to add the following Link Libraries(order is important):



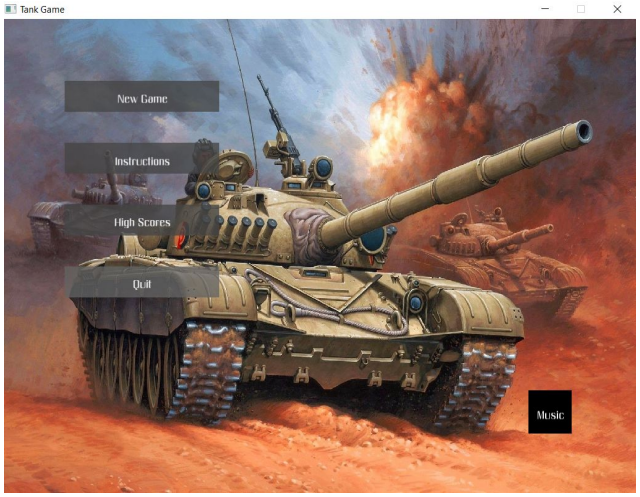
Next, in Search Directories => Compiler, add the following:

```
..\libs\SFML-2.5.1\include
..\libs\TankGraphics
```

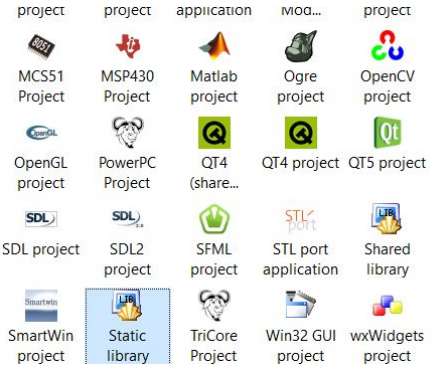
And in Search Directories => Linker, add the following:

```
..\libs\SFML-2.5.1\lib
..\libs\TankGraphics\bin\Debug
```

7. Now, you may run your program in Code::Blocks, and the appropriate Game Window should appear:



8. (**IMPORTANT**). If you encounter an error with the TankGraphics library, or decide to make a change, you should also re-built the library. To do this, you want to create a new static library through Code::Blocks in the ``libs`` directory:



With the name ``TankGraphics``. You should then add in all the files through Code::Blocks as we did with ``app``. After this, you should set the following Project => Build Options:

Linker Settings

sfml-graphics
sfml-window
sfml-system
sfml-audio

Search Directories => Linker

..\SFML-2.5.1\lib

Search Directories => Compiler

..\SFML-2.5.1\include

Now you should build the project through Code::Blocks, and when you're-ren the ``app`` project, you should be good to go.

Linux (Ubuntu 20.04.2 LTS)

Cloning from version control(if not already downloaded):

$git clone https://github.com/yousefh409/tankGame.git

Compiling the TankGraphics Library (assuming gcc and other dependencies already installed):

$cd tankGame/libs/TankGraphics
$gcc -Wall -c *.cpp -lsfml-graphics -lsfml-window -lsfml-system -lsfml-audio -lsfml-network
$ar ru TankGraphics.a *.o
$ranlib TankGraphics.a

Compiling the game:

$cd ../../app
$gcc -Wall -c *.cpp -I ../libs/TankGraphics
$gcc -Wall -o tankGame *.o ../libs/TankGraphics/TankGraphics.a -lsfml-audio -lsfml-graphics -lsfml-network -lsfml-system -lsfml-window -lstdc++ -lm

Running the game:

./tankGame (may need to set permissions using chmod)

To make this easier, we can put all of these terminal commands in a shell script:

compile.sh (in git directory):

#!bin/sh
cd libs/TankGraphics
gcc -Wall -c *.cpp -lsfml-graphics -lsfml-window -lsfml-system -lsfml-audio -lsfml-network
ar ru TankGraphics.a *.o
ranlib TankGraphics.a
cd ../../app
gcc -Wall -c *.cpp -I ../libs/TankGraphics

```
gcc -Wall -o tankGame *.o ../libs/TankGraphics/TankGraphics.a -lsfml-audio -lsfml-graphics
-lsfml-network -lsfml-system -lsfml-window -lstdc++ -lm
```