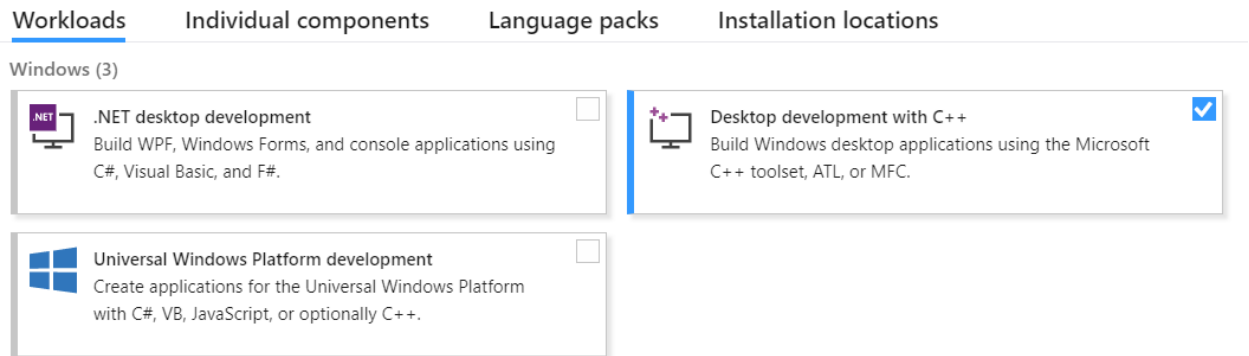


How to build:

The code can be built in both Windows and Linux. I personally recommend **Windows**, since this is the environment I used when developing the program. Also, in my Ubuntu VM, the process was killed often due to high memory usage, and the mouse movement was very fast. Nonetheless, it is possible to build and in Linux. Instructions are at the end.

How to build on Windows:

- Install **Visual Studio**, make sure that the C++ desktop development tools are installed

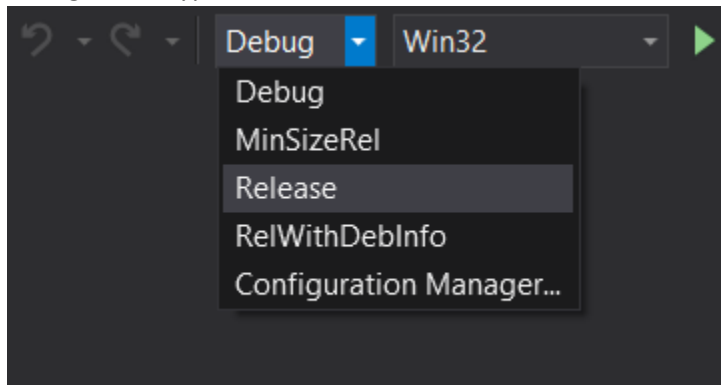


- Install **CMake** for Windows, make sure to add cmake in the PATH environment variable
- Open the source code folder, and run cmake inside the build folder:

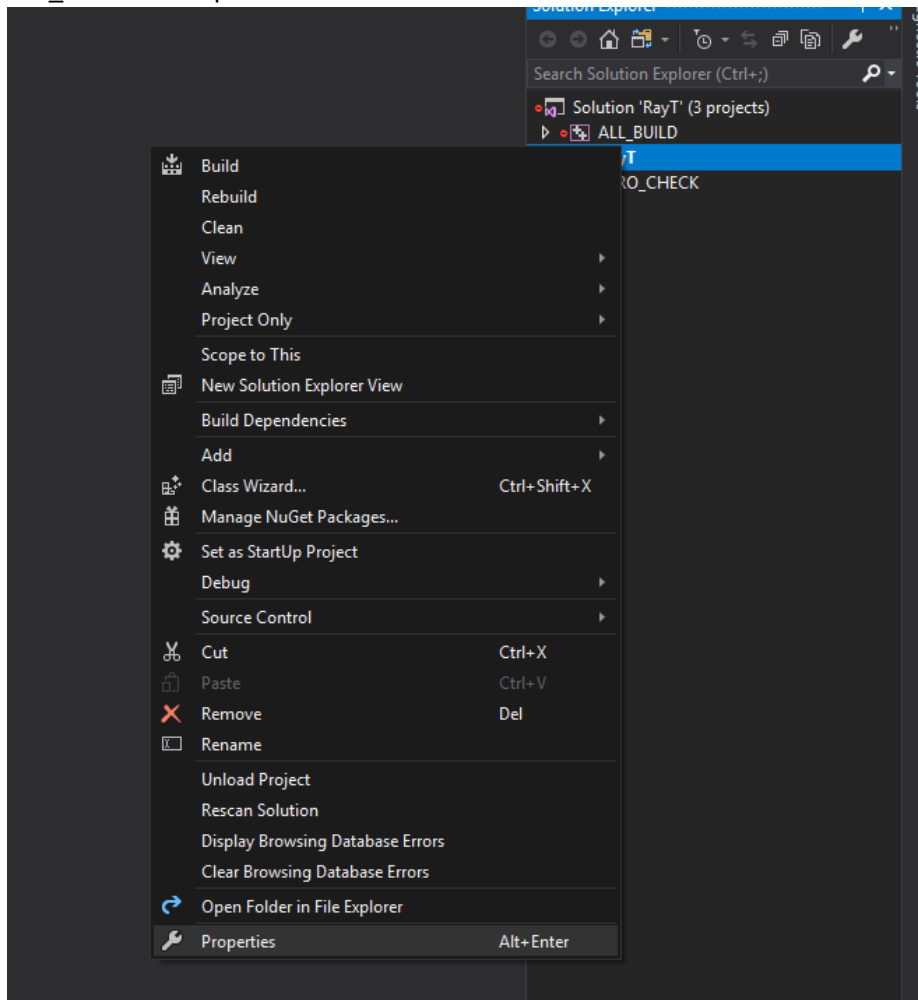
```
Windows PowerShell
PS F:\Documents\dev\Ray-traversal\build> cmake ../_
```

If OpenGL is installed, and Visual Studio is installed correctly, RayT.sln should be created inside the build folder

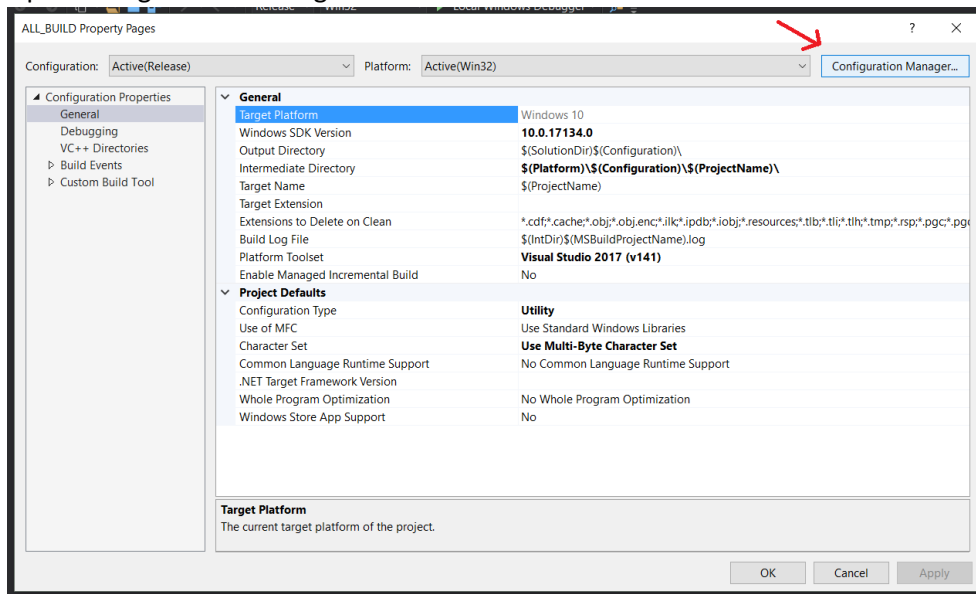
- Open RayT.sln with **Visual Studio**
- Change build type to Release



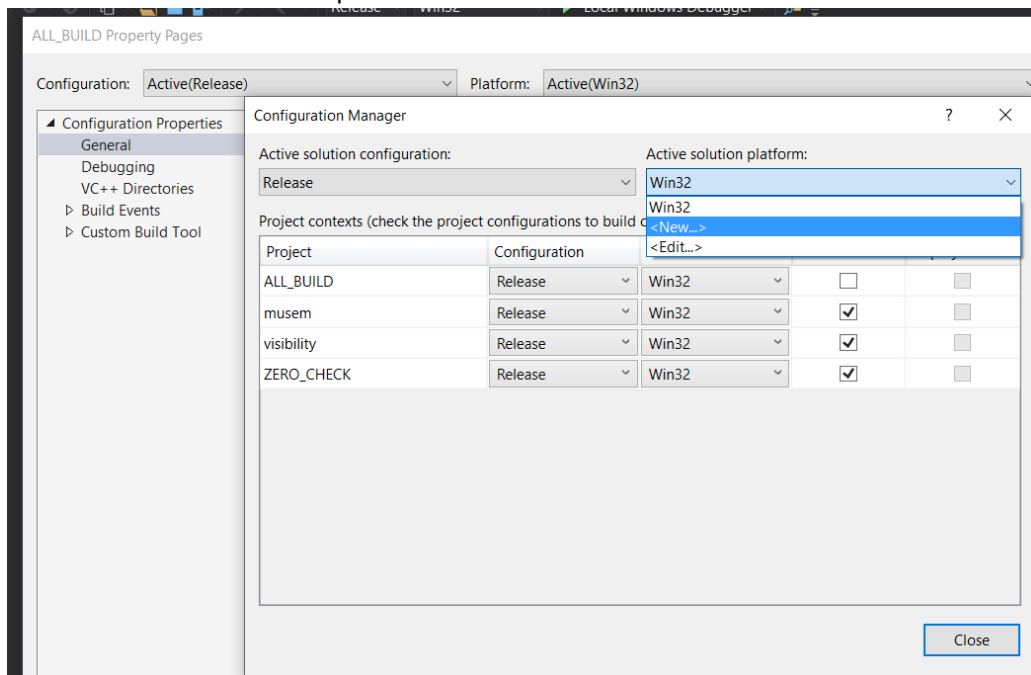
- Set target to x64:
ALL_BUILD -> Properties:

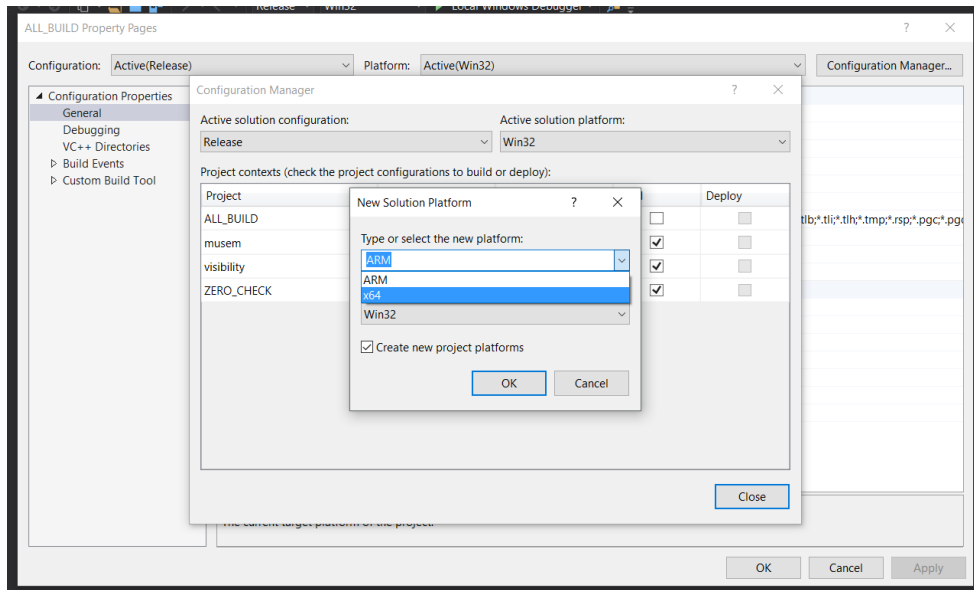


Open configuration Manager:



Create new Active Solution platform for x64:

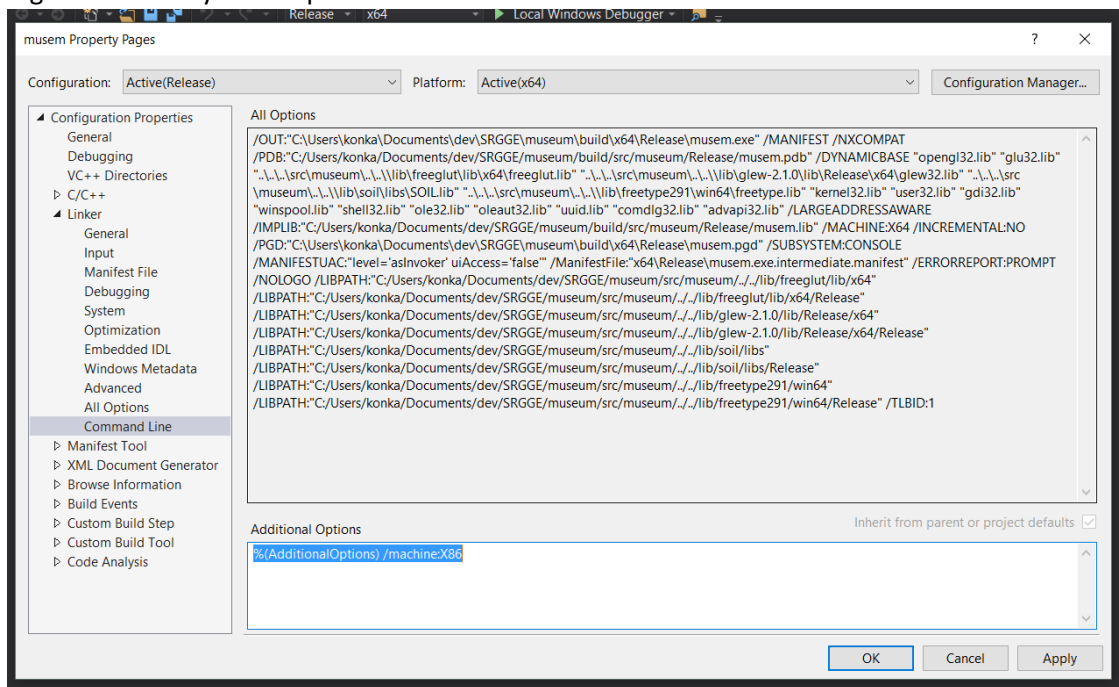




OK -> Close -> OK

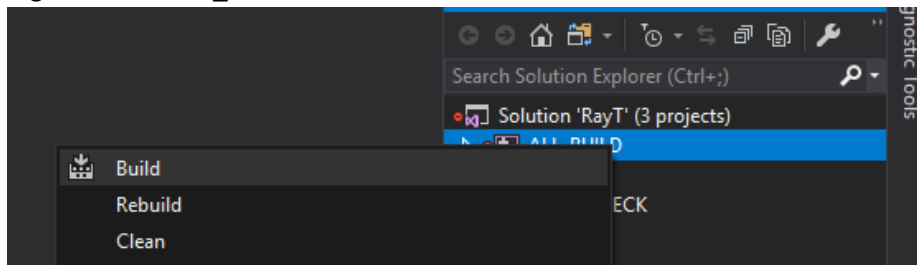
- Remove x86 Linker dependencies:

Right click on RayT -> Properties -> Linker -> Command Line



Delete the Additional Options text, and click Apply.

- Right click on ALL_BUILD -> Build



- Go to directory “build\x64\Release” where the executables are created, and paste the **contents** of the dependencies folder, which is located in the top-level directory

Name	Date modified	Type	Size
fonts	26/8/2019 6:51 μμ	File folder	
RayT.tlog	26/8/2019 6:44 μμ	File folder	
shaders	26/8/2019 6:51 μμ	File folder	
ZERO_CHECK	26/8/2019 6:44 μμ	File folder	
Application.obj	26/8/2019 6:44 μμ	Object File	16 KB
bunny.ply	26/8/2019 6:42 μμ	Αντικείμενο 3D	1.306 KB
Camera.obj	26/8/2019 6:44 μμ	Object File	10 KB
freeglut.dll	26/8/2019 6:42 μμ	Application extens...	230 KB
freetype.dll	26/8/2019 6:42 μμ	Application extens...	799 KB
glew32.dll	26/8/2019 6:42 μμ	Application extens...	413 KB
GraphicsObject.obj	26/8/2019 6:44 μμ	Object File	29 KB
main.obj	26/8/2019 6:44 μμ	Object File	12 KB
moai.ply	26/8/2019 6:42 μμ	Αντικείμενο 3D	372 KB
PLYReader.obj	26/8/2019 6:44 μμ	Object File	85 KB
RayT.exe	26/8/2019 6:44 μμ	Application	247 KB
RayT.log	26/8/2019 6:44 μμ	Text Document	7 KB
Scene.obj	26/8/2019 6:44 μμ	Object File	81 KB

- Run the RayT.exe with your favorite graphics card
- If you wish to run the code through the Visual Studio debugger, make sure to paste all the dependencies inside the working directory of Visual Studio, which is “build/” and change build to Debug.

How to build in Linux:

- Install dependencies: OpenGL, GLUT, FreeGLUT, GLEW, SOIL, Freetype
- Create build folder
- Run *cmake* inside the build folder
- Run *make* inside the build folder
- Copy the required dependencies (shaders, fonts, models) besides the executable

This does not set the target platform to x64 or sets the optimizations for the compiler. This means that it's possible for the execution to be *Killed* from the runtime system or run very slowly.

How to use: Once the program is running, you can manipulate the scene with:

- The **mouse** to look around
- The keys **W, A, S, D** to move the camera
- The letter **Esc** to exit

When doing ray casting, you can see the actual primitives being hit with the ray (red circle) because their colors change to random colors every time.