Suicide project 3D

Master I Informatique *FORNALI Damien*

Université Nice-Sophia-Antipolis

2017-2018 *24/12/17*

Introduction

With a colleague we came to see you at the beginning of the year, we asked you if we could directly use an OpenGL library instead of the LibGraph library.

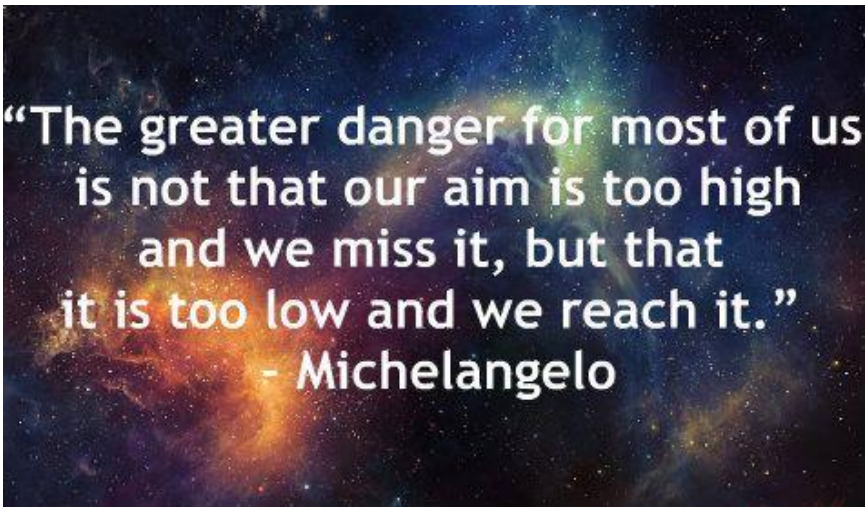
As a result, I started to learn OpenGL. Yes I had never done it seriously, and so I started to create a 2D mini-engine... until I became crazy… *What ? Damoy, please, keep reasonable…*

Why would I do that?

As a 2D game developer on my spare time, I thought I would not really learn anything new by doing a tower-defense-like in a new language with a limited library. *Oh God, it does not smell good O.o*

Sooner in the year, I discovered OpenGL and I never dove myself seriously into it, I thought a strong challenge was the occasion for me to start to learn it. *Well, seems legit*

I started to think, as I always do at the beginning of a project, big, too big.

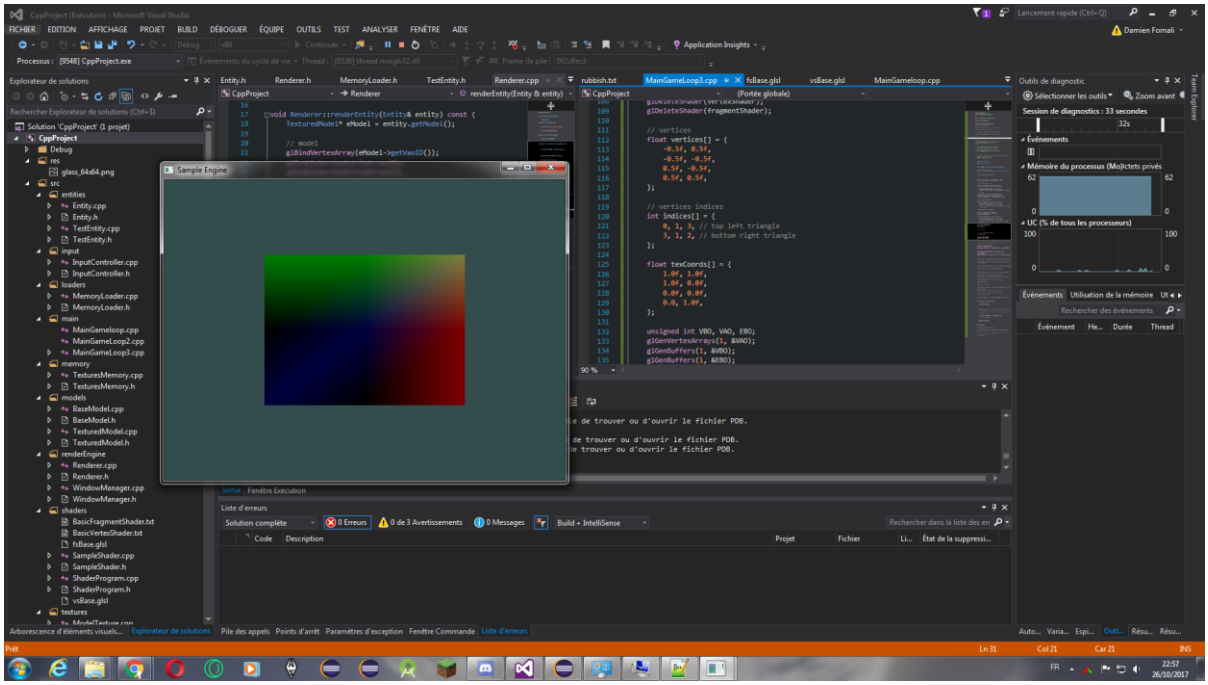


Explain me a bit please…

I started by learning models and shaders, how to place a vertex on the screen, how to color it, how to store the data required into OpenGL, etc.

Hum, okay seems legit for now

Oh a colored rectangle



I started to learn about vertices, indices (which purpose is to increase performance), glsl extension, GPU processing, double buffering…

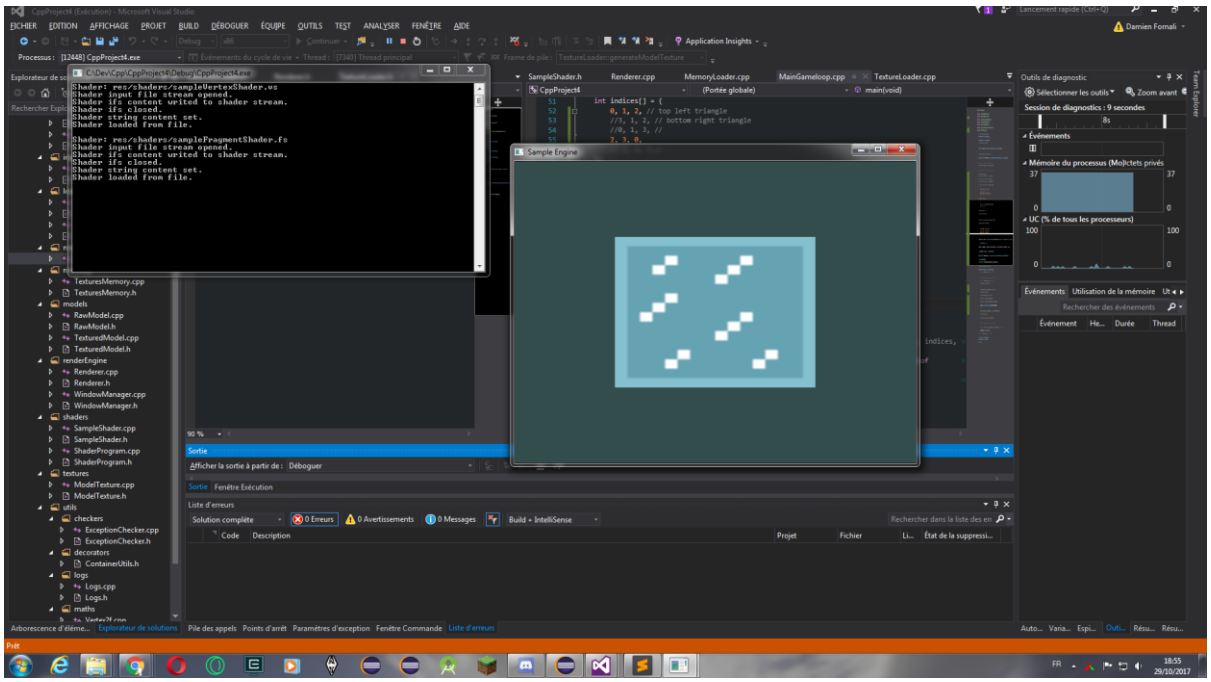
I already discovered OpenGL difficulties at this point. *WHY DOES NOT MY RECTANGLE RENDER AND WHY OPENGL DOES NOT TELL ME ANYTHING ?*

Indeed, OpenGL is very silent when it is about errors, this can be enjoyable and hateful.

Then I started to think about textures, raw colors were too simple.

Hum ? What are you talking about ? You are supposed to do a tower defense !

So many hours of work for this…

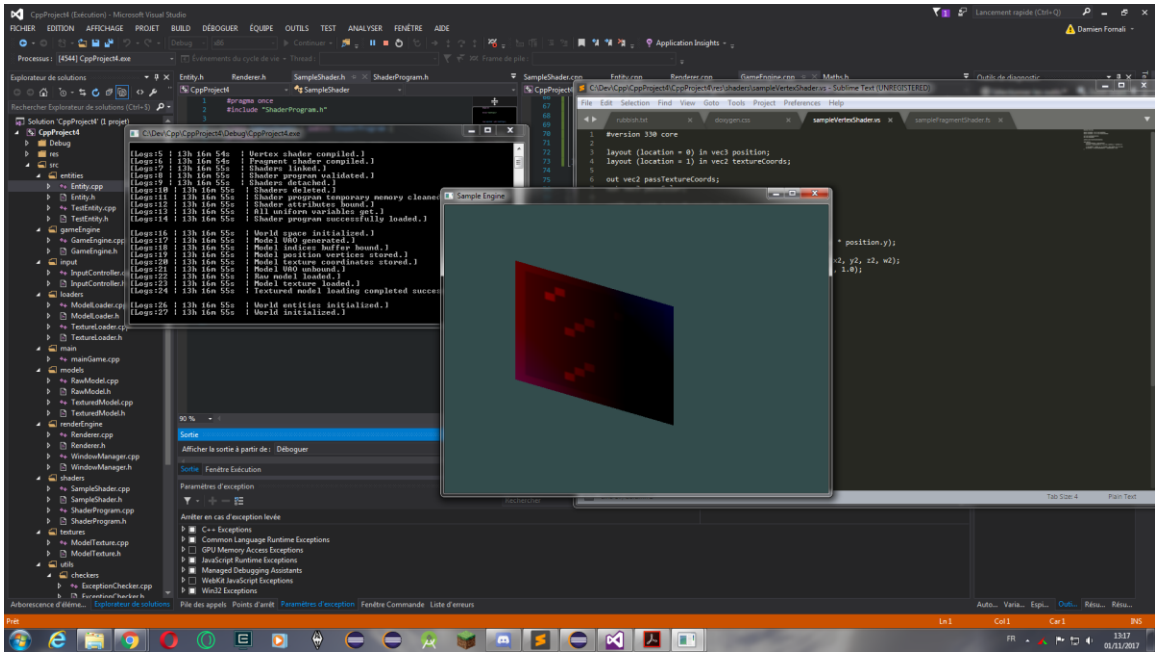


And again I started to understand the scale of making an OpenGL game alone. ***God bless the independent game developers.***

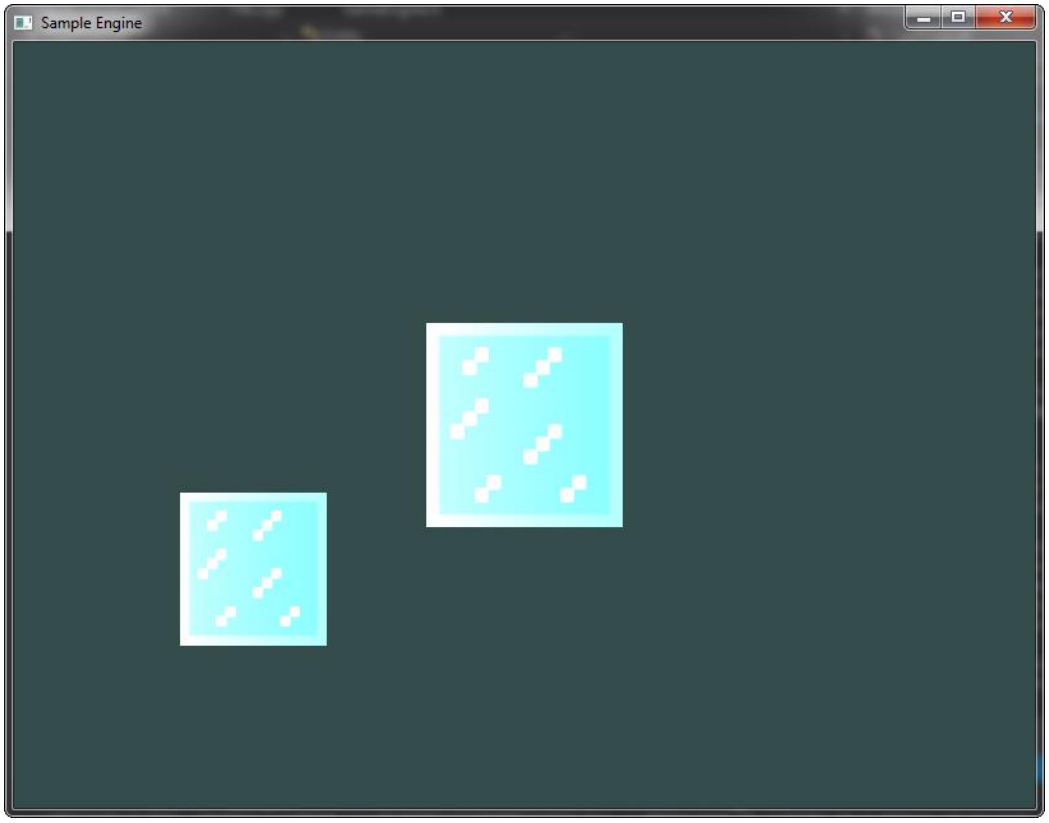
During month I spent week-ends and evenings on the base engine that would stay alive until the final result, I made the WindowManager to easily control the window, the InputController for the user to interact with the game and I started to fill the ModelLoader that would be essential to use models and make the whole work.

So old code is still used, ok

I then learnt about matrices, the transformation matrix (position, rotation and scale).



And more about shaders.



In fact it was okay, I could start to make the game based on this simple engine.

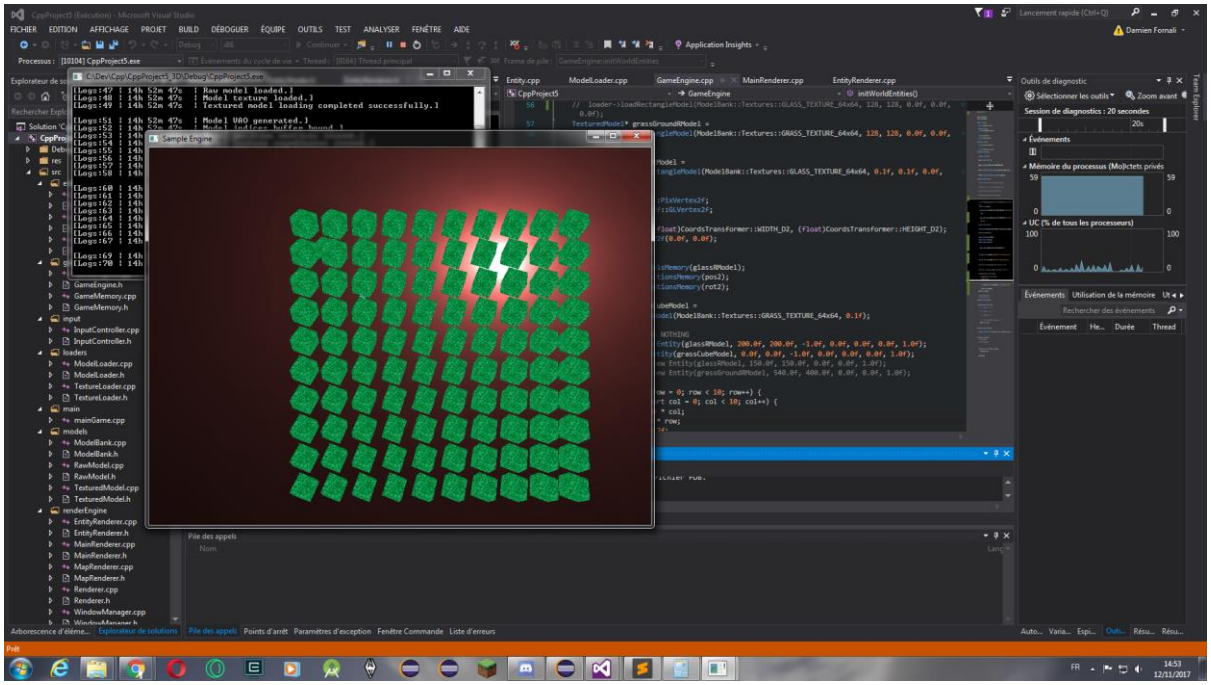
Until I realised… I was, finally, very close to 3D. I understood that when I learnt about the projection and matrices which allow to add depth to the rendering and movement for the entities.

And here we go…

So I challenged myself, if I was able to have a 3D rendering in 24 hours I would try to. do. the. cpp. project. in. 3D. even if I knew nothing about 3D.

And yes, I did:

No, no, that's not lightning you see behind the cubes ☺



And so the game was on, each day I looked for 3D information that could be useful for me, a beginner, for a short-term project. I evolved the ModelLoader, the rendering, the shaders.

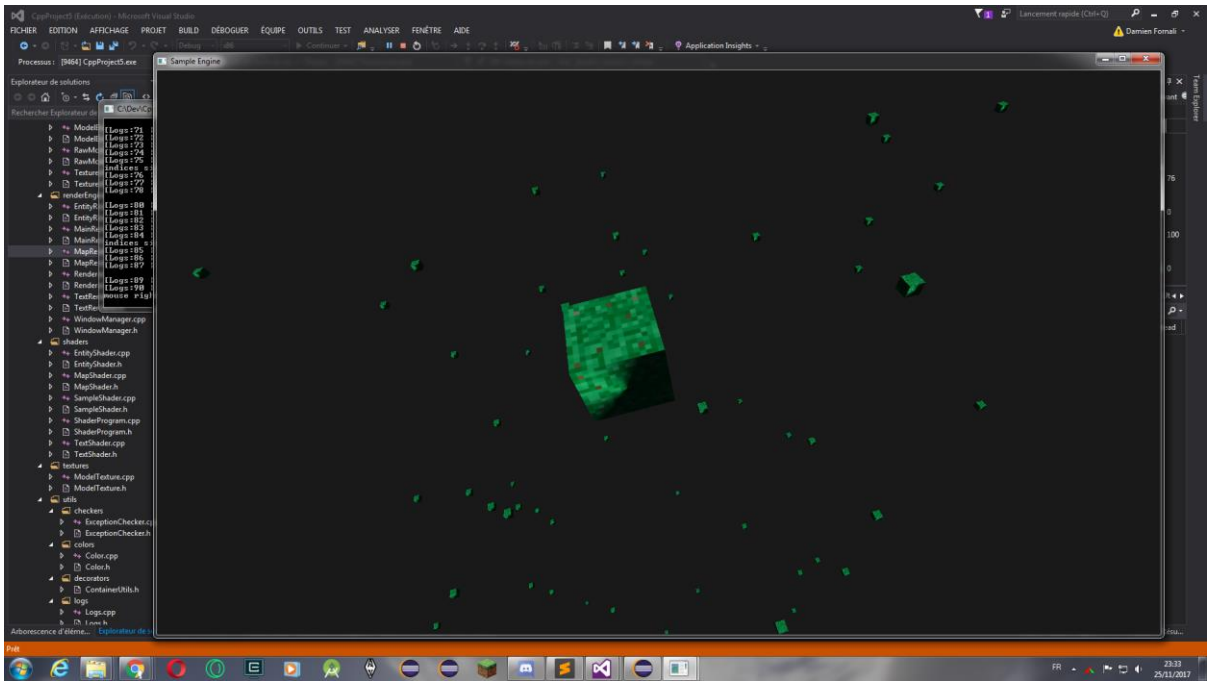
But the whole architecture was bad (yes it is still now) and I encountered manyy problems. Invisible / wrong placed models faces while cull facing and map not showing were particularly annoying.

I thought to abandon this crazy idea many many times, at this point I learnt many things and I still had the time to do a 2D game, with my code, or the LibGraph library whatever.

Yes, go for the 2D one !!! You did learn a lot !!

But I could not abandon one hundred hours of work like that. So I kept going and going and, miraculously, I managed each time to solve the problems I encountered.

Yep, that is truly lightning :O



I developed a Logs system and an Exception Checker that allowed me to accelerate debugging and be more productive.

Oh, cool, that seems indeed a useful idea.

I am running out of time, what to do?

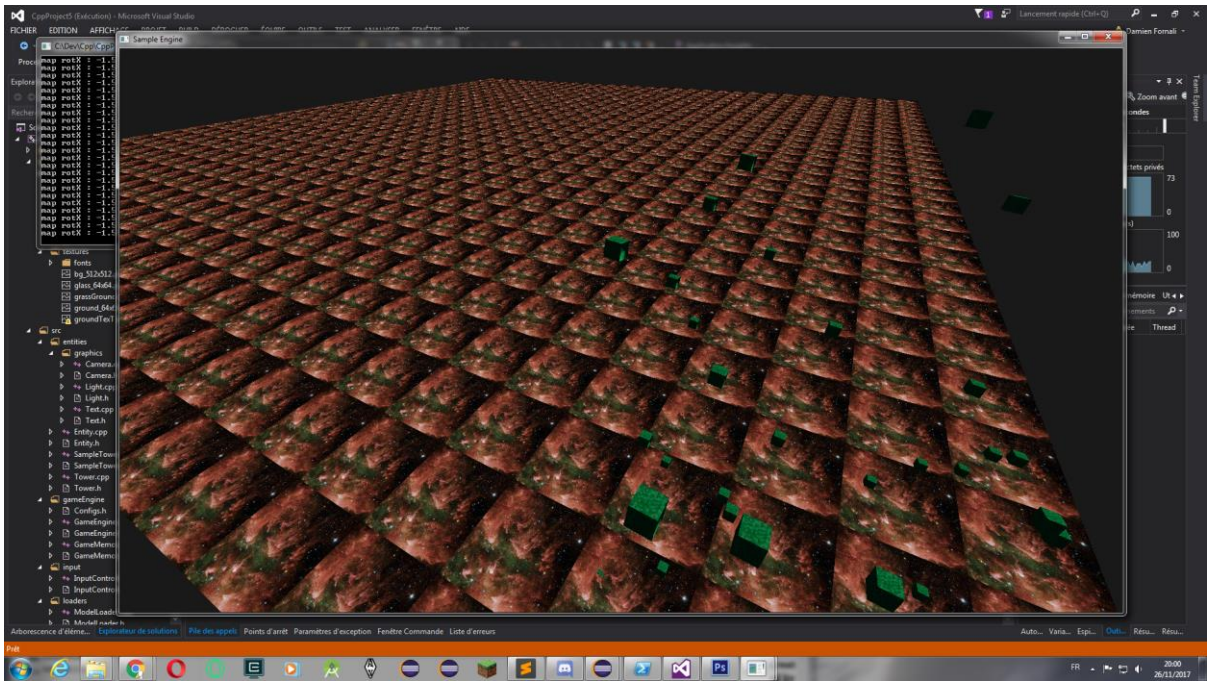
University started to give us many projects, and as an apprentice in an IT company it was hard to keep going on this project, on side.

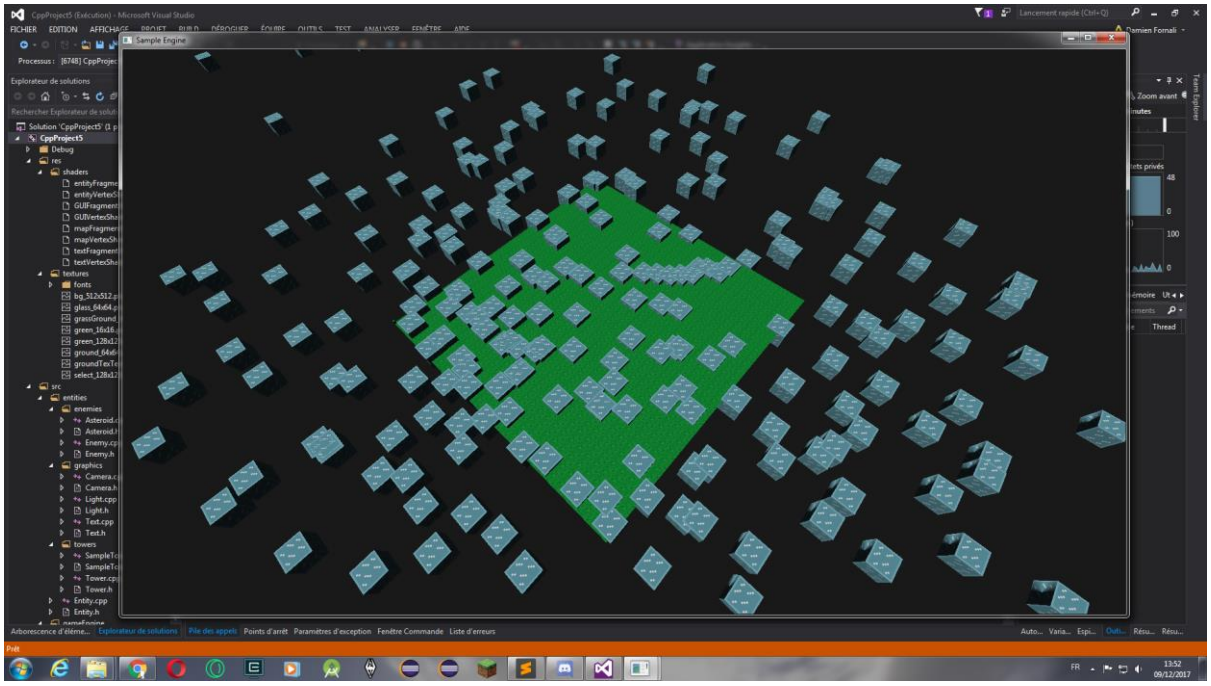
New doubts came: no map, end soon, many projects, should I keep going?

Oh, no, go 2D it will be saferrrr !

Yep, I did, map finally came !

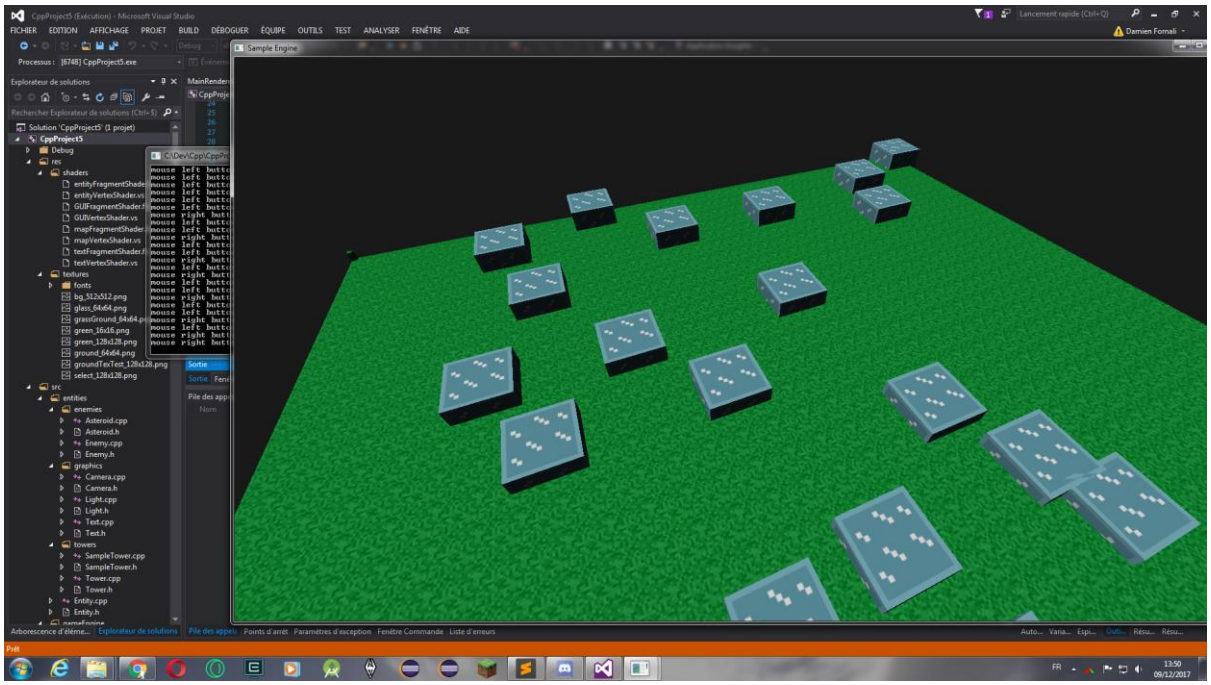
Why don’t you listen to me ?





As you imagine, too hard for me to add complexes models as this point, I decided to go on cubic ones.

Anddd that is a mouse-ray / map collision !



Viewport space (mouse screen coordinates) to world space made me crazy too but I am happy I managed to implement it.

The game prototype

Okay, now it is time to make at least one level playable.

I started to implement the tiled map, enemies, asteroids.

Quickly I have been able to feel a sense of gameplay and I spent December trying to “finish” it, the more properly and stable I could given the context in which I was.

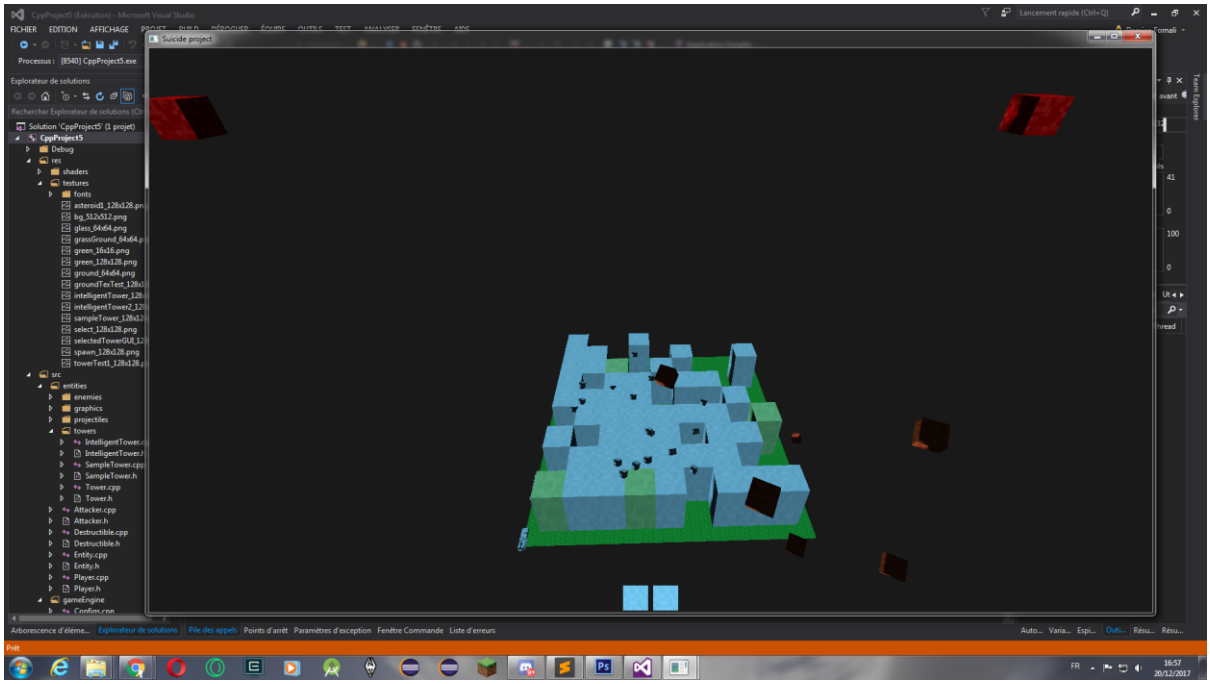
I refactored a lot, added a Player class, a Shop, interfaces-like and more important I spent a long time trying to make work at least two towers. They still feel odd but they are not so bad, now, after all.

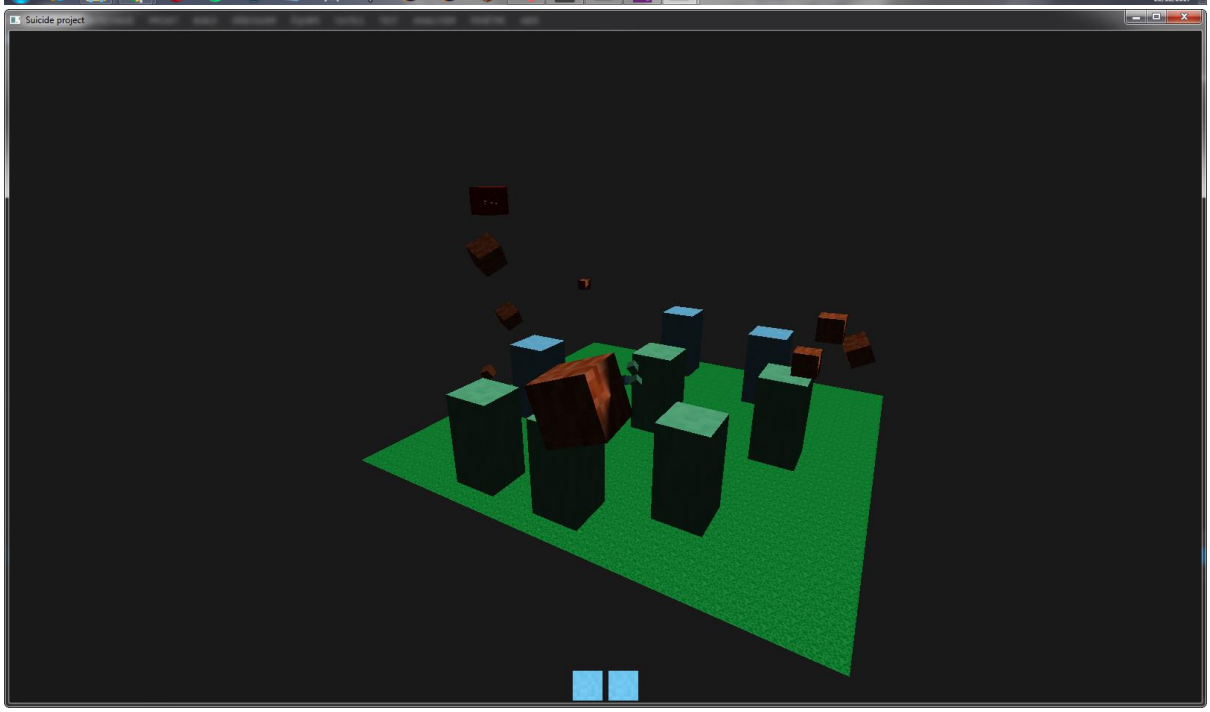
I also wanted to add GUIs and texts but I failed in both.

My GUIs have rendering problem and text are too complex to add as I can’ t add *glutbitmapcharacter* in a recent OpenGL context.

Ok, Damoy, but I still think the 2D would have been better :P

Final product





Problems and improvements

- I abused of the Entity class in most of the project methods. I did not want to do that but that was for me the quickest way to make something for the release date, breaking types is definitively not something I would redo.
- Collisions are perfectible as they don't take rotation in count
- Memory management is perfectible too, there are objects that are not deleted and I make many copies throughout the code
- I did try to implement a second Level, which was just a bigger map with more asteroids, I successfully chained levels, then I got a problem that made me think I should let only the stable one. Level chaining is perfectible / TODO
- Game would have been more enjoyable with more different enemies, towers, terrain textures…
- Removing a tower while its projectiles are alive does not kill them

Dependencies

GLEW (OpenGL), GLFW (Window), GLM (Maths), GLUT (Text, unused), STBI (Texture loading)

How to play

Controls are classic, WASD / ZQSD / SPACE / CONTROL for movement, right mouse click for camera rotation, left mouse click for placing a tower and middle mouse click to remove a tower.

C/V to change the current tower type (V is better ☺)

The documentation has been made, it should make the user understand easily most of the code ☺

Conclusion

I learnt a lot, that's why I think it was truly worth to work like that (200-300 hours). Game programming, OpenGL, C++, memory management, 3D, GPU, CPU, compilation, optimisation.

I am happy of the result, the obstacles I overcame even though I left some feathers ☺