

- README -

2D City Defense

Context : a destructive meteor shower is descending on your town ! Destroy meteorites and retrieve the energy they contain to defend the city with your silos !

1 Time management

You will find the real *Gantt diagram* below, that will gives you an idea of how much time I passed for each task (1 dashed box width corresponding to 1 day, around 2h) :

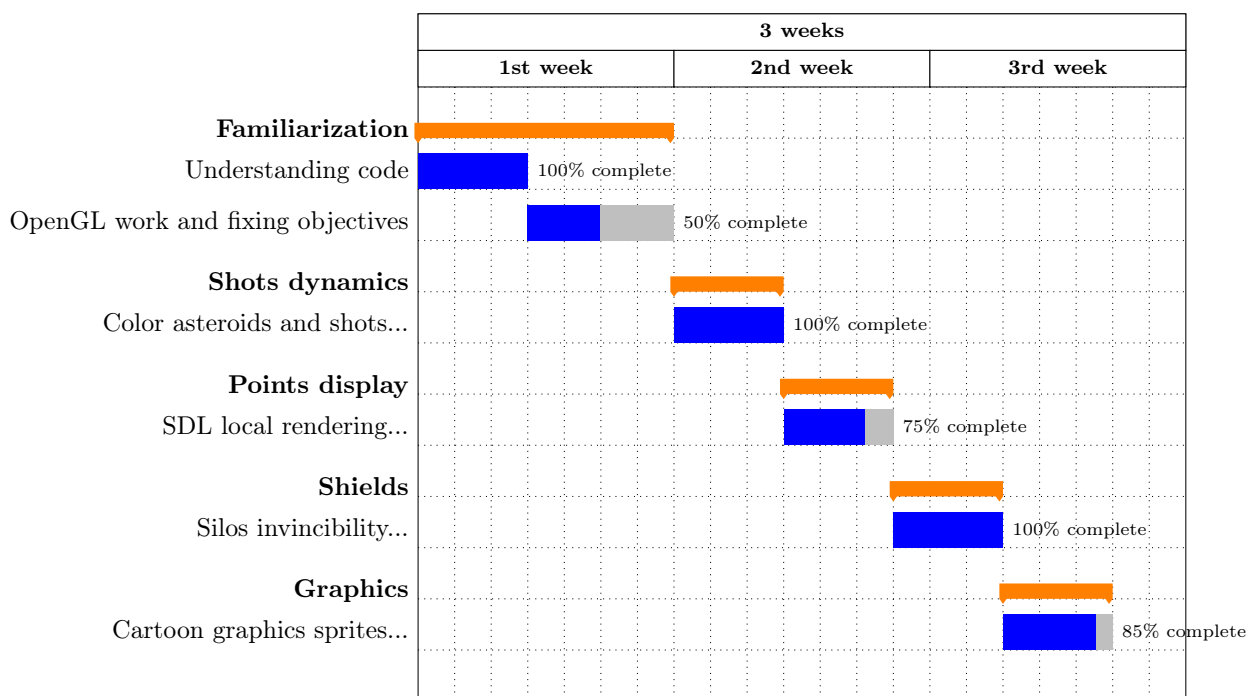


FIGURE 1 – Gantt Diagram

Please bear in mind that my schedule has been busy, and my work scattered among my available moments.

2 Decisions made

One of the main difficulty of this test consisted in finding a real stake, that could stimulate desire for players to keep playing. A stake is often created in video games by inventing some *resource* and finding some *objective*.

The city is composed of 6 warehouses where each can store 100 Energy units (thus 600 Energy units could be stored at maximum). They are 50 reserve energy units initially. We earn energy units only when we destroy asteroids with our energy (silo shots or shields), not when asteroids destroyed themselves by colliding the scenery. When a warehouse is destroyed by an asteroid, we lose all the energy it contained (assuming that the same amount of energy units is stored in each warehouse).

Little asteroid are red-coloured and can only be destroyed by laser gun, whereas big asteroid are blue-coloured and can only be destroyed by electromagnetic gun. A laser shot costs 5 units and an electromagnetic shot costs 10 units. However, 10 energy units can be recycled (and then automatically and equally stored in warehouses) if we destroy a little asteroid, against 20 energy units for big ones.

So the main objective of this game is to quickly reach the maximum capacity of warehouses (600 energy units) so as to be able to use a supreme final shot, that will destroyed the meteor downpour forever (not coded unfortunately...).

3 Changes and corresponding implementations

3.1 Gameplay

- Shots dynamic (`LEFT MOUSE` and `RIGHT MOUSE` button for laser and electromagnetic shots respectively) : in the initial game, shots were unique and could potentially annoy the player. By associating a mouse button click to a shot type, and a shot type to an asteroid type, the player must be more focused in thinking to where he shots. It is assumed that electromagnetic shots are inert on small hard core meteorites and laser shots are inefficient on big meteorites.
- Energy shields (`SPACE` button) : the only gadget of the game is to pay 100 energy units to generate shields that protect all silos during approximately 10 seconds. Every meteorites that collides on these shields are instantaneously destroyed and the corresponding energy retrieved from their destruction.
- Duality between energy and time : either we are focusing on filling the warehouses to give the final shot, or we are focusing on improving our performance on time.

3.2 Graphics

The original graphics design proposed was relatively basic, so I opted for a *cartoon* graphics style, because spatial realistic rendering would have been difficult to implement in SDL (except by using some libraries combining SDL with GPU for instance ...). All sprites have been downloaded from Internet, and slightly modified with Gimp to bring manual corrections on images, due to resizing and noise.

When meteorites are destroyed, some text is now displayed to inform the player the amount of energy retrieved and stored.

4 Problems encountered and outlook

4.1 Main hardships

The major problem of this task has been to restrict reasonably the objective to something feasible in the given time frame because "Impress us" is a pretty large guideline and theoretically an infinite objective! I wanted to demonstrate conveniently my gameplay (because this application is for a Gameplay developer) and graphics (because it's one of my major skills) abilities. At the beginning, I planned to convert this SDL video game into an OpenGL one in order to have more control on the graphics pipeline and primitives to draw. But due to my schedule and more stimulating objectives for this game, I decided to change my plans.

Sometimes, code bugs when arriving at game over window, for an unknown reason, even after several debug attempts with Visual 2015.

4.2 Outlook

- Implement a real end for the game, with a special effect for the supreme final shot, when all 6 warehouses are filled.
- A new gadget like a powerful weapon could be interesting to be implemented, in order to introduce more dynamics in the game (or evolution of silos when certain time reached, like ages of evolution for cities).
- Collision between asteroids could be coded as well, to make the game more realistic. Nevertheless, the number of asteroids is increasing with time and more oriented when one silo and one warehouse in the same area remain. So algorithm that weights pros and cons should be thought.
- Clarify the stake.

5 Acknowledgements

I would like to warmly thank you in particular for giving me such an exercise, original for a video game company. Indeed, this exercise combines both to impose limits in one's objectives set beforehand, but also to manage one's time properly, to understand the algorithmic architecture of a video game already functional and to study the SDL in depth.