INF585 - Pirate Boat

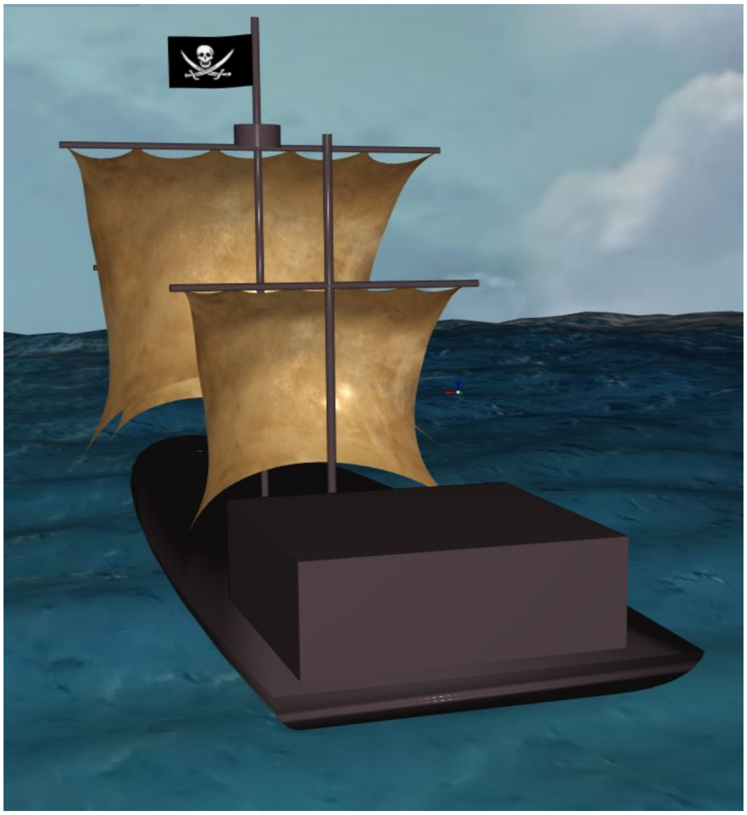
Computer Animation project report

Dmitry Gaynullin - dmitry.gaynullin@polytechnique.edu

Introduction

The project consists of several parts: the boat, the flag, sails and skybox. The project has the same structure as ancient labs: all elements are initialized in the setup_data method and displayed in the frame_draw method. It's not using the last version of cgp library, but rather the old version of vcl library. Everything is packed in with all the necessary libraries and README instructions.

1. Boat



The boat consists of a hull, masts and superstructure.

It is constructed in the method **create_ship (float radius, const vec3 & p0)** as a mesh hierarchy, because it has a complex mesh, constructed from several primitives. I didn't use the ready 3D model from the internet.

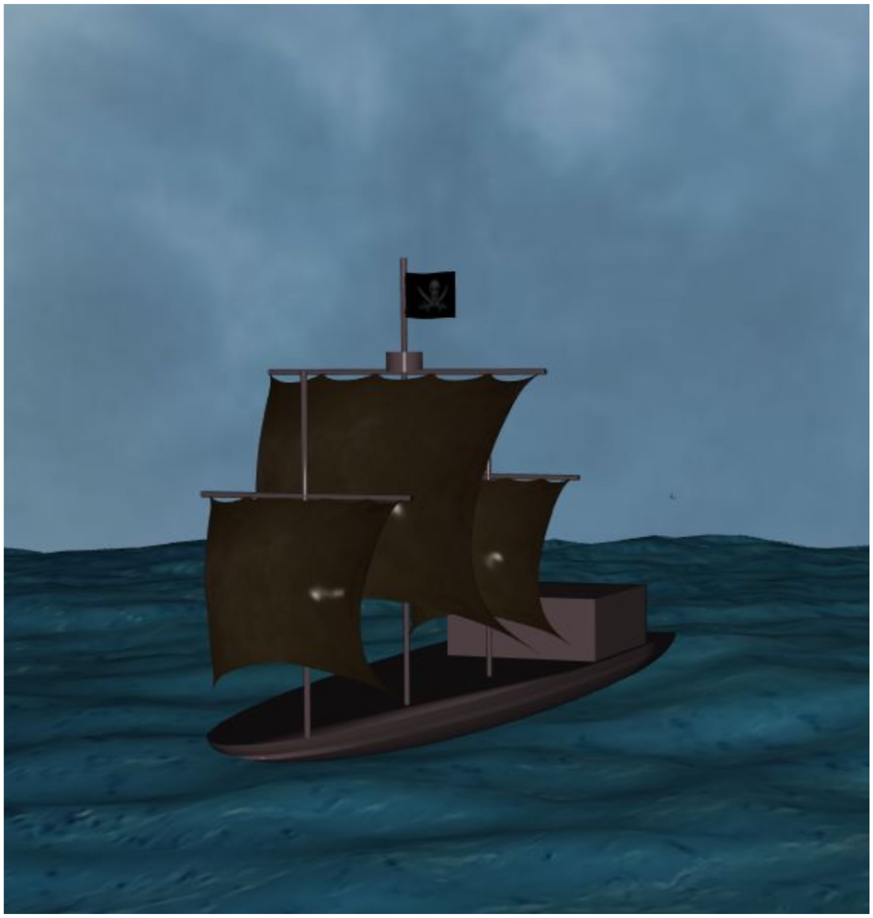
The boat's hull is shaped like a sector of an ellipsoid, with the deck and back built by hand. It is returned by the **create_hull(float radius, float a, float b, float c, const vec3& p0)** method. The masts are cross-shaped, created from two cylinders and an observation platform on the larger one, while the superstructure is a rectangular parallelepiped.

The boat is animated with the help of the hierarchy. So, if we have the current height of the sea at each point and time, we could modify the ship's z-coordinate and rotate it a little around its axis. In addition, it has a certain horizontal speed because of the wind.

2. Sails and the flag

The sails and flag are constructed from a grid of 10-20 points which are linked together in such a way that each point is linked to its neighbors at the top, bottom, left and right, and the connections have different stiffness coefficients. The same forces are applied as during the animation of a cord, with the addition of forces that mimic wind. Wind is an analogue of periodic gravitational forces in all axes, which causes the sails and flag to move, affecting their grid points. The positions of all grid points are saved in **std::vector<std::vector<particle_element> > particles_flag**, **particles_sail_1, particles_sail_2, particles_sail_3.** To create the grid, the **create_grid** method is used, then **update_positions_flag** and **update_positions_sail** functions are applied to update the positions.

3. The sea



The sea is constructed the same way as the terrain was constructed during the labs. Basically, I used Perlin noise with a time-responsive parameter in the **evaluate_terrain(float u, float v, float t)** function, with an addition of a sea texture.

Project structure

As mentioned above, I started from the codebase given during the labs with an old version of the vcl. For the project, I only modified a handful of files, which I'm attaching to this report and you could find them in **INF585/project/src** path:
- **scene.h** and **scene.cpp**: the main work was done here
- **main.cpp**: very minor changes, it's just a standard pipeline.
- also some additional files that was used extensively could be found in the folder **INF585/project/src/boilerplate**

I've included all the files I used for compiling them.

Of course, if you want to recompile and encounter a problem, don't hesitate to contact me (dmitry.gaynullin@polytechnique.edu).

Conclusion and future work

In general, the project consists of lots of small and simple animations, as I struggled a lot with the idea for the project and could not find the project that would be technically feasible for a student project done alone. At first, I tried to implement the animation of the flowing lava from the volcano, but it was too difficult to simulate and I gave up.

The scene I currently have can be improved to simulate a more realistic environment by adding the following possible extensions:

- Better liquid simulation by 3D SPH simulation
- Better wind simulation by having more correlation between animation of different parts of the scene
- Better textures and the models for ship and the flags

Many other improvements can also be added, but due to the limit of time most of these improvements were not applied to our scene.