

Getting started instructions

1. Install Visual Studio 2017 (VS2017) (you can download it [here](#))
2. Open Practical.sln in VS2017.
3. Build the project (Build -> Build solution, or Ctrl+Shift+B)
Everything should already be set up in order to build successfully, but you might need to select another Windows SDK version than the one the project is set up for.
If VS2017 complains about this go to Project -> Properties and change Windows SDK Version to one of the available options. It shouldn't matter which one. If nothing is available in the drop-down menu, you will have to download an SDK on Microsoft's [website](#).
4. Run the project (Debug -> Start debugging, or F5)
Building in Release mode results in an optimized executable that runs significantly faster, while Debug mode builds a more or less unoptimized executable that enables you to step through the code and check variable values etc. making it very useful for debugging.

Building the project produces an executable file (Project1.exe) which will be put in a specific folder depending on whether the active configuration combination is Win32 or x64 and Debug or Release, as the following:

```
/Win32 /Debug/Project1.exe  
      /Release/Project1.exe
```

```
/x64   /Debug/Project1.exe  
      /Release/Project1.exe
```

A glew32.dll file needs to be in the same folder as the executable (note: glew32.dll for Win32 and x64 respectively are not the same) as should already be the case in the code you are given.

Overview of the hand-out code

Main.cpp

This is where the program starts.

The first function called is the main() function. The first thing the main function does is initialize GLFW and GLEW which are utilities that allow us to use OpenGL. Then the window is created along with the cloth. The rest of the code consists of the while loop also known as the “render loop” which is where everything will happen from now on until the program is closed.

In the render loop gravitation and wind forces are added to the cloth and the simulation is advanced one time step.

After advancing the simulation, rendering is performed. First, changes in window size are handled and the window is cleared, second, the transformation matrices are set up, followed by executing the cloth’s draw function, and finally user input is handled.

Cloth.h

Contains the Spring and Cloth classes.

The cloth consists of particles in a 2 dimensional grid connected by springs.

Every particle is connected to 3 - 8 other neighbor particles as well as 3 - 8 secondary neighbor particles depending on where it is located in the grid.

ShaderUtility.h

Contains utility functions to load OpenGL shader files.

You won’t have to change anything in this file.

Phong.vert

The code for the vertex shader we use to render the cloth.

You won’t have to change anything in this file.

Phong.frag

The code for the fragment shader we use to render the cloth.

You won’t have to change anything in this file.