Include files and global instances

```
#include <grit.h>
#include <util.h>

grit::engine2d_type  engine;
grit::param_type     parameters;

util::ConfigFile      settings;
util::Log             logging;
```

Main

```
int main()
{
  settings.load(“my_config_file.cfg”)

  std::string  const txt_filename = settings.get_value("txt_filename");
  std::string  const output_path  = settings.get_value("output_path",   ""                  );

  util::LogInfo::on()            = util::to_value<bool>(settings.get_value("logging","true"));
  util::LogInfo::console()       = util::to_value<bool>(settings.get_value("console","true"));
  util::LogInfo::filename()      = output_path
                                 + "/"
                                 + settings.get_value("log_file","log.txt");

  logging << "### " << util::timestamp() << newline;

  parameters = grit::make_parameters_from_config_file(settings);

  assert(grit::is_valid(parameters));

  grit::init_engine_with_mesh_file(
                                      util::get_data_file_path(txt_filename)
                                      , parameters,engine
                                      );

  for (unsigned int i = 0; i < max_steps; ++i)
  {
    write_svg_files(output_path, i);

    do_simulation_step();

    engine.update(parameters);
 }

  logging << "Done" << newline;

  return 0;
}
```