Graphics and Game Technology

Lab Assignments

Getting Started

1 General remarks

Welcome to the exciting world of Computer Graphics! We hope that the lectures and lab assignments will stimulate your interest and get you acquainted with the theory, concepts and practice of this subject.

Before you begin with the lab assignments, there are a number of rules that we expect you to abide by. *Read these carefully!*

1. Check the schedules for the time and location of the lab sessions. Make sure that you are aware of any last-minute changes by checking the roster, visiting the lectures and checking any announcements that may have been left on Canvas.

2. All assignments will be posted on the Canvas section with this course. If you cannot access that section, you must first enroll through SIS.

3. An assistant will be present at the lab sessions to answer questions or help you in case of problems. **This is the *only* method of support that will be provided. The assistants will *not* answer questions by email.**

4. You must work on the assignments **in pairs**. Do not switch partner during the course. You are expected to contribute *equally* to the *whole* assignment. The assistants will check this by asking you questions on your submission. If you contributed equally, you will receive the same grade. If not, corrections will be made.

5. The submission deadline for each assignment is set to the end of the week; Friday at 23:59 hours. Submit a working version of your code in the form of a tarball through Canvas. Each student pair only needs to submit the assignment once. Make sure *both* your names and student numbers are added to the marked comment sections of the code. If the assignment came with datasets, do *not* submit the datasets. **NOTE: you may only submit your work *once***, so make sure you double check your submission.

6. Grading of your assignment will be done **during lab sessions** by an assistant, but only *after* you have submitted your work. You must *both* be present to get graded.

Ideally you would ask an assistant to grade your submission at the earliest lab session after you submitted your work.

7. All assignments for this course must be completed in the C programming language and must compile and work under Linux.

8. All assignments come with a framework for which the software dependencies are described in the assignment text. It is *your* responsibility to resolve these software dependencies. The first thing you should do with each assignment is to compile and run the framework. If you cannot get the framework running, contact an assistant.

2 Using the assignment framework

Each assignment comes with a PDF description of the assignment and a framework package (.tar.gz or .tgz) that contains files written in C and a `Makefile` that generates a running executable for Linux. The framework should compile and run without problems. It will just not do a lot because usually one of the source files contains one or more empty functions that you must complete. The assignment description provides details on what to do.

2.1 Unpacking the assignment framework

Download the assignment package from Canvas and unpack it (enter the commands that are preceded by a `$`, without entering the `$`):

```
$ mkdir cg-pract
$ cd cg-pract
$ tar zxvf ../framework2.tgz
framework2/
framework2/normals.c
framework2/Makefile
framework2/main.c
framework2/normals.h
framework2/polys.c
framework2/polys.h
$ cd framework2
$ ls
main.c  Makefile  normals.c  normals.h  polys.c  polys.h
```

2.2 Building an executable

Now you can directly compile and run it, as follows:

```
$ make
gcc -c -g -O2 -std=c99 -Wall -Wextra \
```

```
  -Werror-implicit-function-declaration \
  -Wshadow -Wstrict-prototypes -pedantic-errors main.c
main.c: In function 'keyPressed':
main.c:261: warning: unused parameter 'x'
main.c:261: warning: unused parameter 'y'
main.c: In function 'specialKeyPressed':
main.c:272: warning: unused parameter 'key'
main.c:272: warning: unused parameter 'x'
main.c:272: warning: unused parameter 'y'
main.c: In function 'createSphere':
main.c:88: warning: 'p.pts[3].x' may be used uninitialized in this function
main.c:88: warning: 'p.pts[3].y' may be used uninitialized in this function
main.c:88: warning: 'p.pts[3].z' may be used uninitialized in this function
main.c:88: warning: 'p.pts[2].x' may be used uninitialized in this function
main.c:88: warning: 'p.pts[2].y' may be used uninitialized in this function
main.c:88: warning: 'p.pts[2].z' may be used uninitialized in this function
gcc -c -g -O2 -std=c99 -Wall -Wextra \
  -Werror-implicit-function-declaration \
  -Wshadow -Wstrict-prototypes -pedantic-errors normals.c
normals.c: In function 'calcNormalsFlat':
normals.c:21: warning: unused parameter 'list'
normals.c: In function 'calcNormalsGouraud':
normals.c:26: warning: unused parameter 'list'
gcc -c -g -O2 -std=c99 -Wall -Wextra \
  -Werror-implicit-function-declaration \
  -Wshadow -Wstrict-prototypes -pedantic-errors polys.c
gcc -g -lGL -lglut -o main main.o normals.o polys.o
$ ./main
6 polygons to flat shade
648 polygons to gouraud shade
```

Don't worry about the warnings for now. Most of these should disappear as you start to fill in the required functions. When a window appears with a picture in it, everything should be fine. In most cases the assignment description also shows one or more pictures of the expected output.

3 Submitting files

When your program is ready for submission, check that you have written your name and student number in the comments at the top of all file(s) that you have changed. When you submit your assignment, submit a working version (excluding datasets, if any) of your project in `.tgz` format (see below). Do not submit executables, object files or editor backup files. You can use "`make clean`" to remove most of these files.

```
$ cd framework2
$ make clean
```

```
rm -f *.o
rm -f main
$ cd ..
$ tar zcvf framework2.tgz framework2/
framework2/
framework2/polys.c
framework2/normals.h
framework2/main.c
framework2/normals.c
framework2/polys.h
framework2/Makefile
$ ls
framework2  framework2.tgz
```

Submit the `.tgz` file through the Assignments section on Canvas.

4 Grading

Each assignment contains a small section on how grades will be determined. In most cases you will therefore know in advance (before you ask to be graded) how many points you can get for each part of the assignment. In general an assignment will be split into several subtasks that will build upon one another. You will receive points for each of the subtasks individually. There are some general rules that apply to the grading of all assignments:

- You can get up to 10 points if your program has all the requested properties.
- If you exceed the deadline, you lose two points.
- If your program is unclear or has a confusing layout (i.e. missing or wrong indents), you can lose up to one point.
- If your program doesn't compile, but we recognize what it would have done if it compiled, you get the points according to the above list minus three points.
- If we have reason to believe that you cheated, i.e. copied and pasted from other students or from the internet, you receive - in the best case - no points at all.
- In individual cases there can be deviations from these rules.

Good luck!