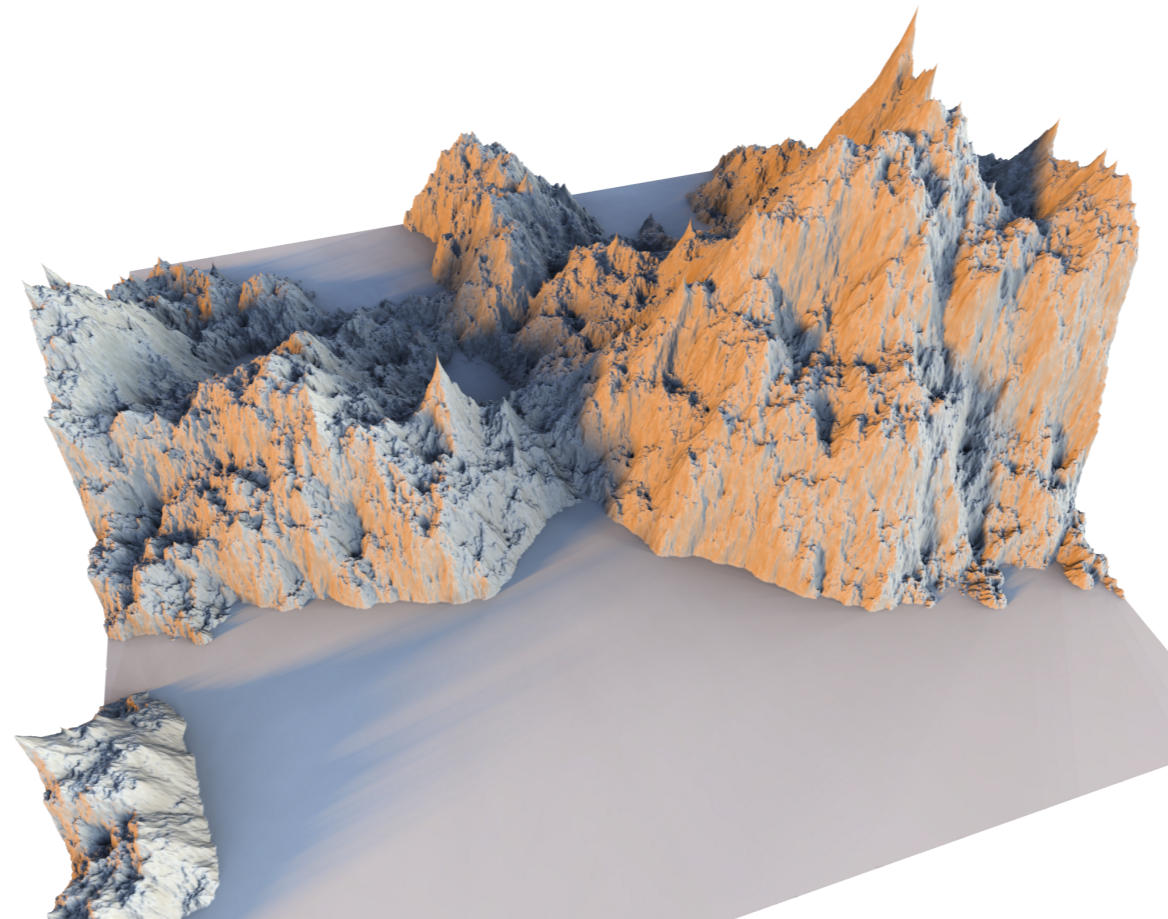


Fall 2018

CSCI 420: **Computer Graphics**



Exercise 1 - Height Field



Kyle Morgenroth

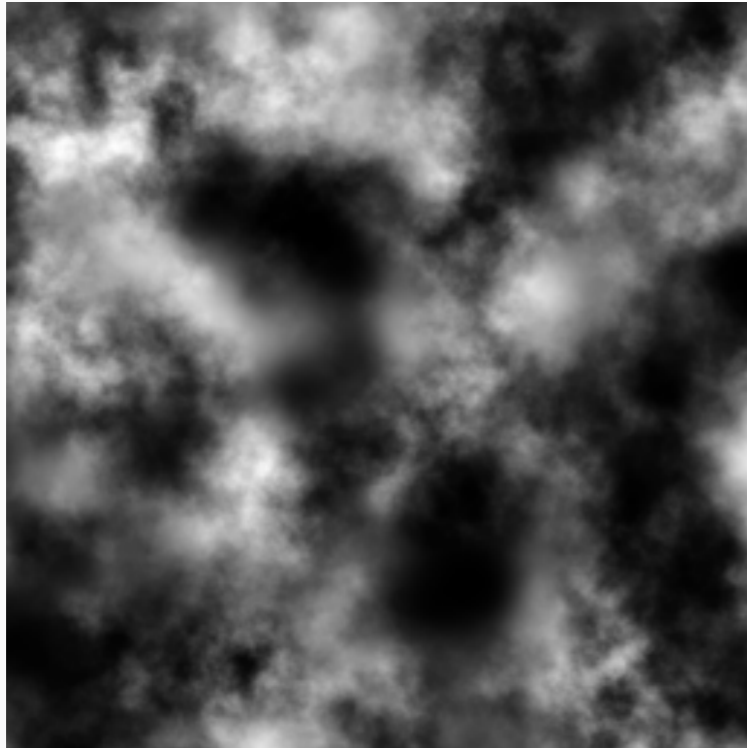
<http://cs420.hao-li.com>

Height Field



Height Field

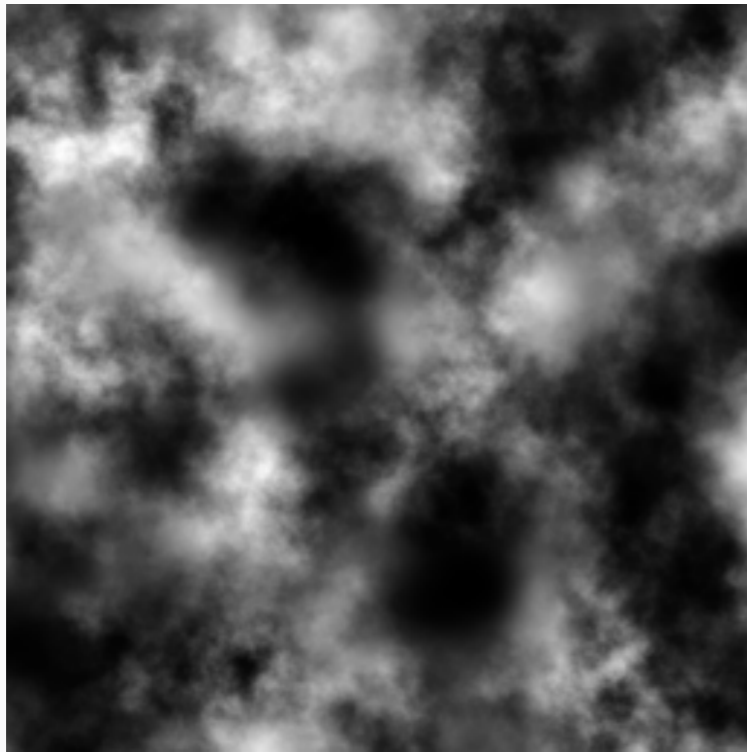
Interactive 3D Height Field Viewer and Fly-through!



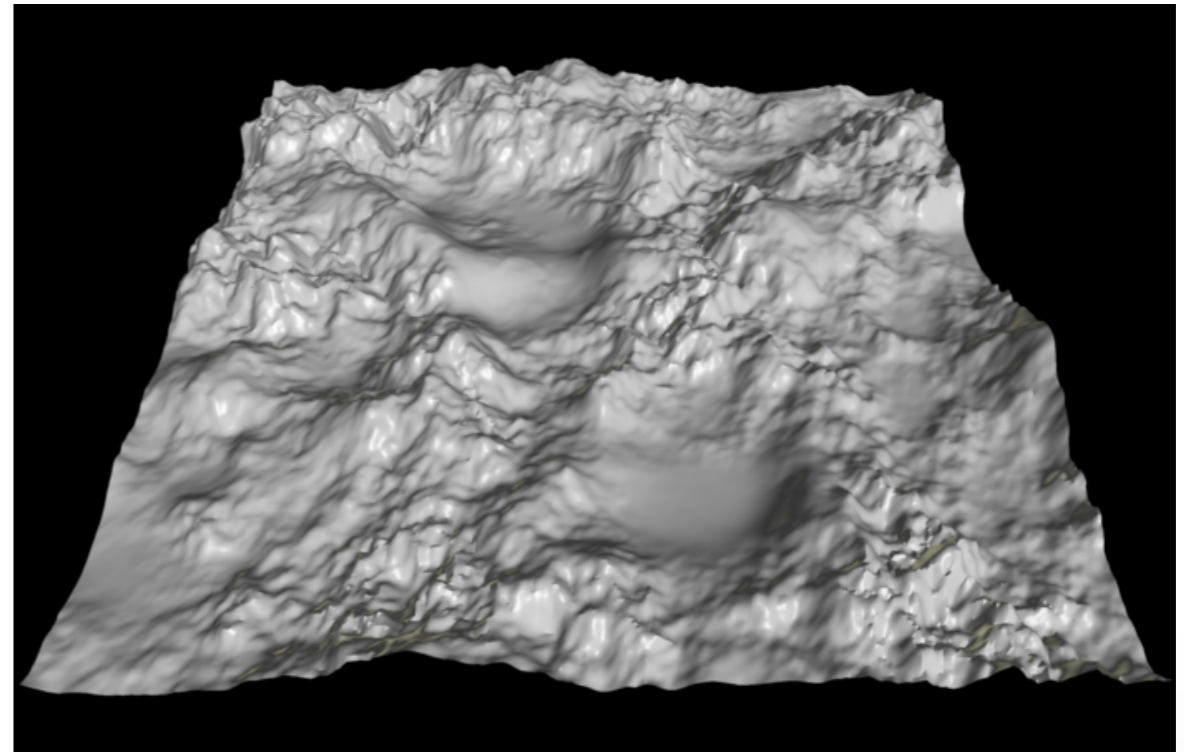
Input

Height Field

Interactive 3D Height Field Viewer and Fly-through!

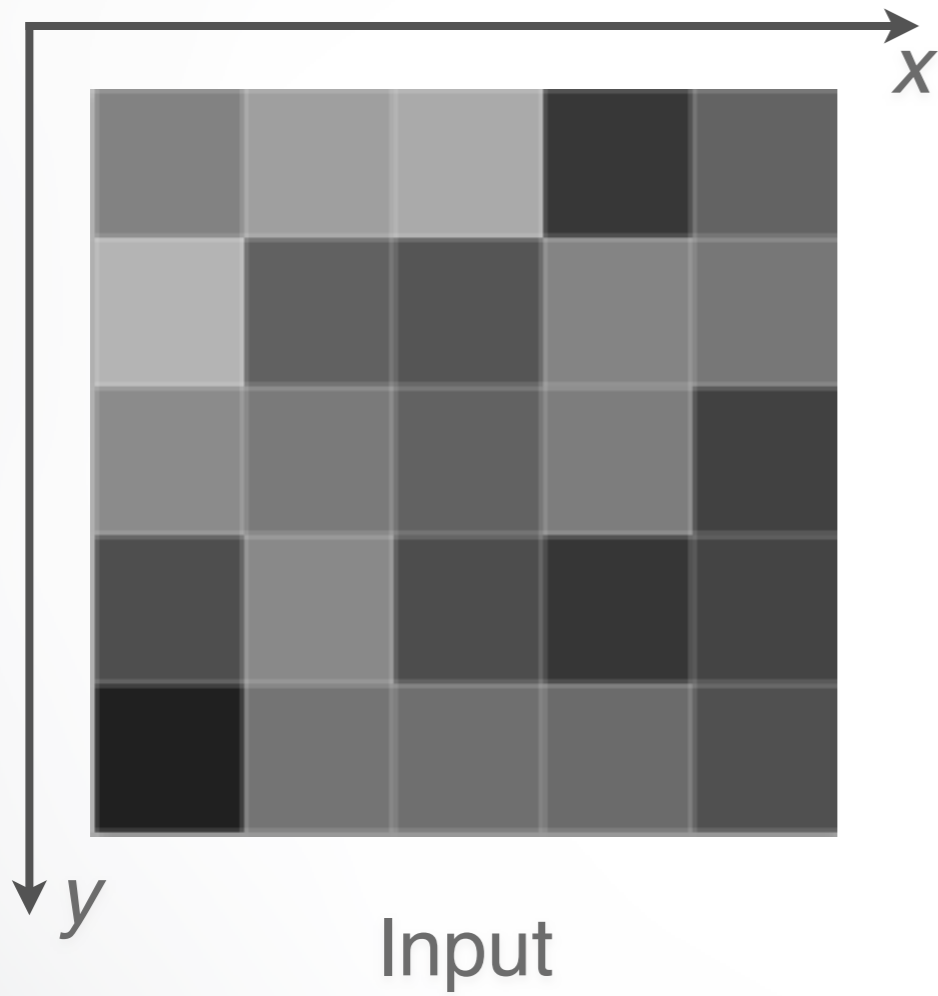


Input

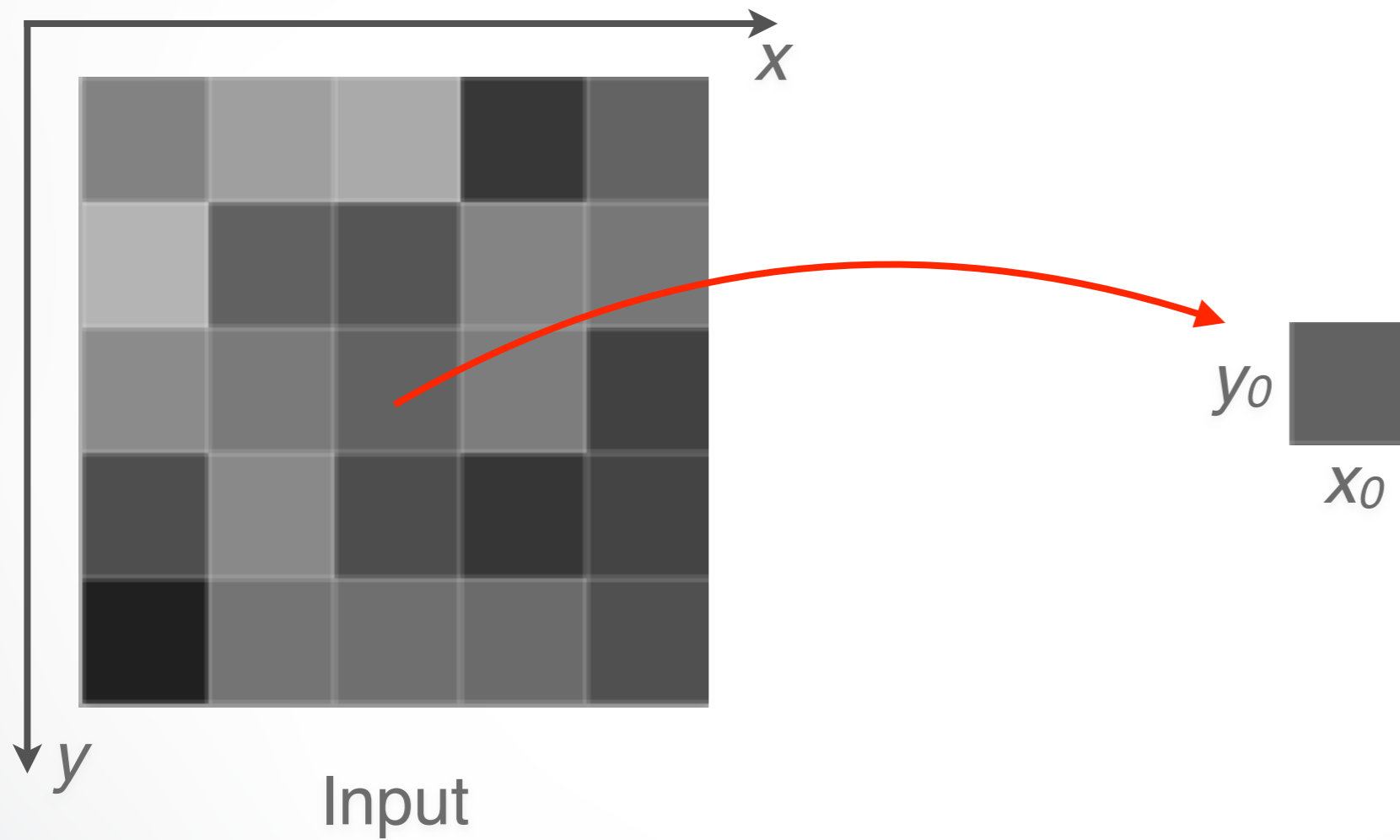


Output

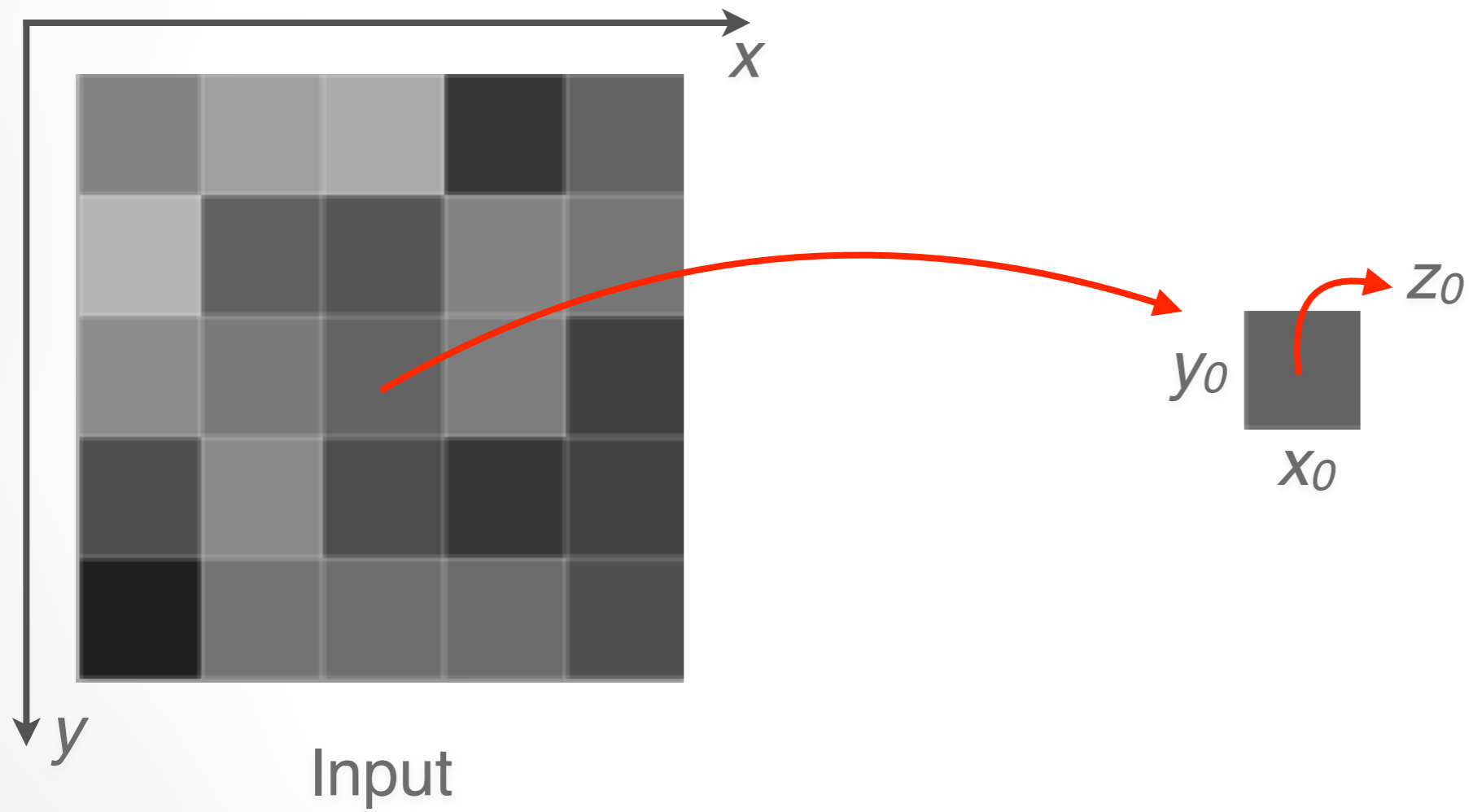
Height Field



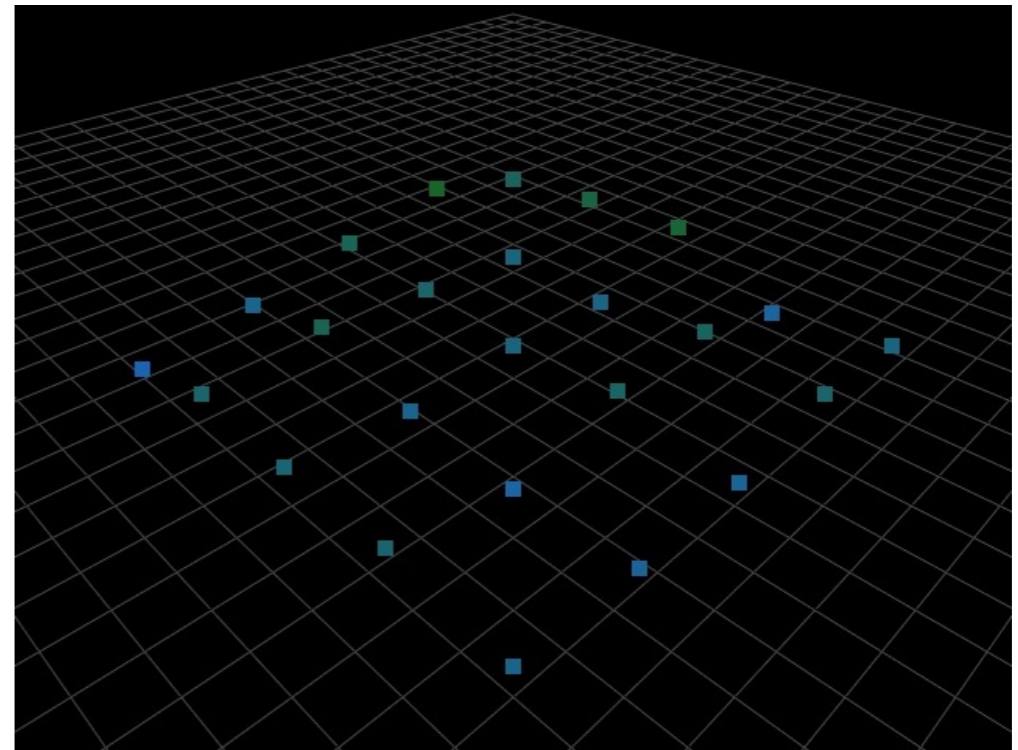
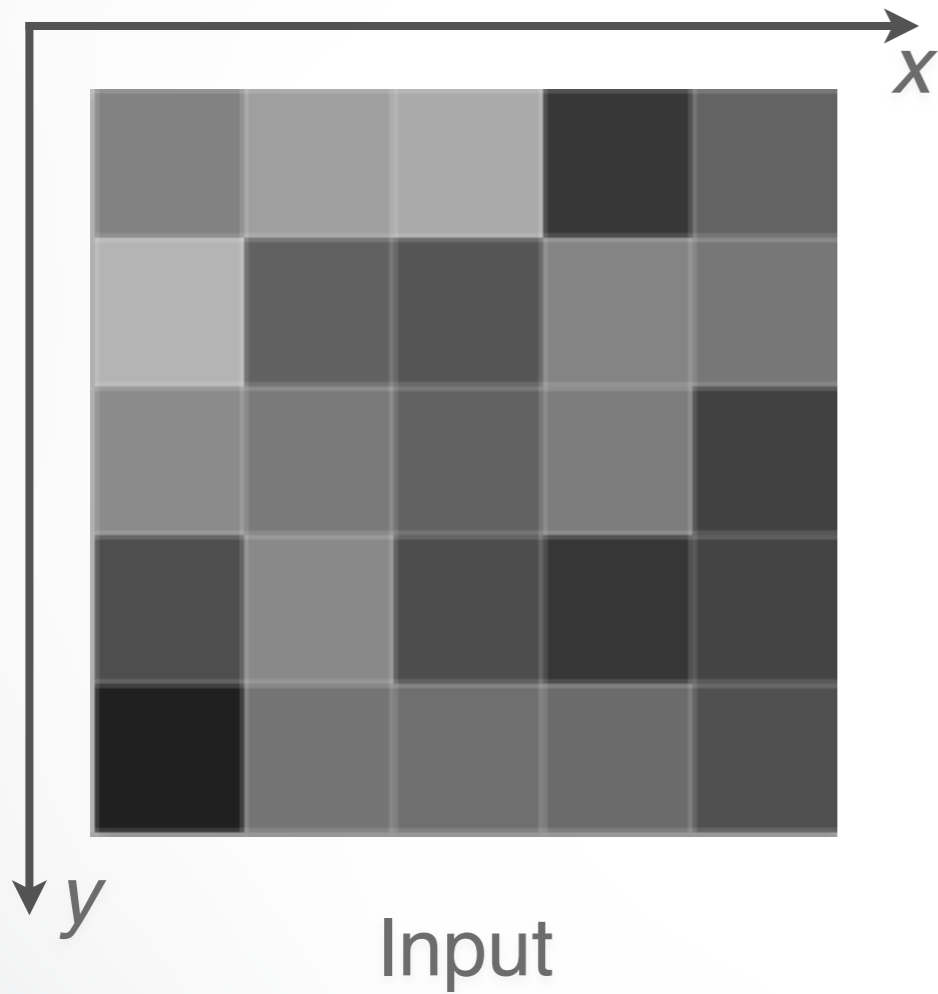
Height Field



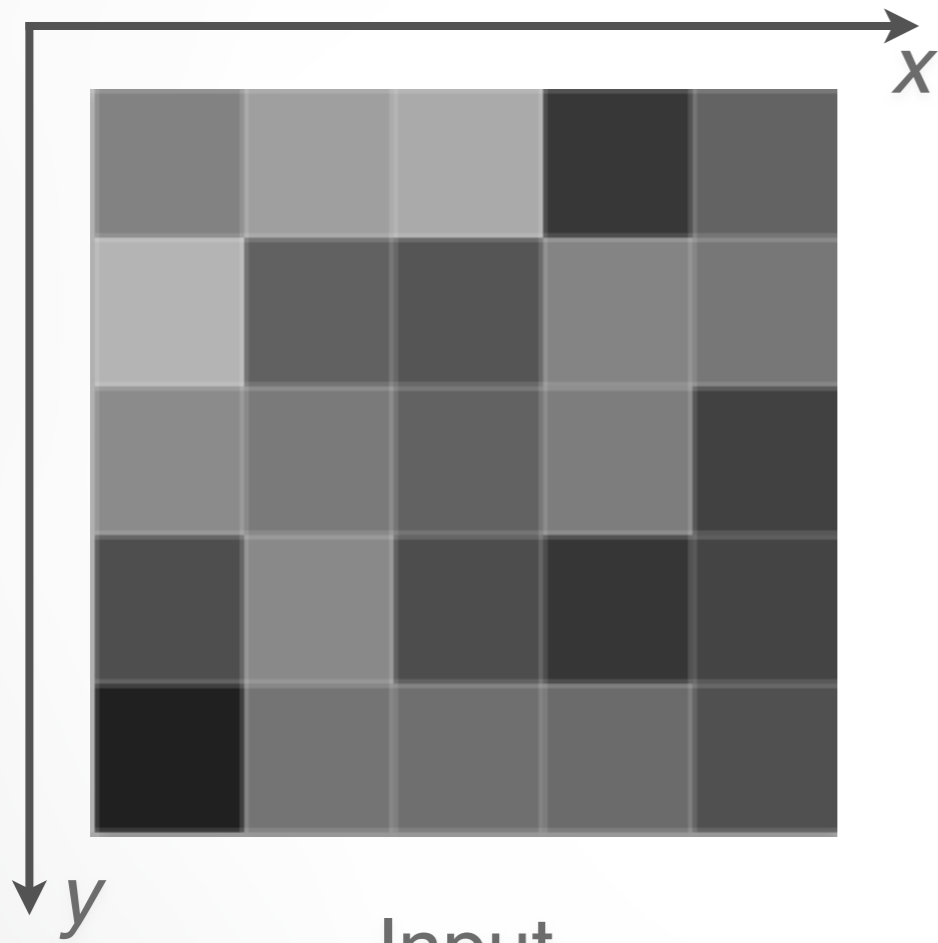
Height Field



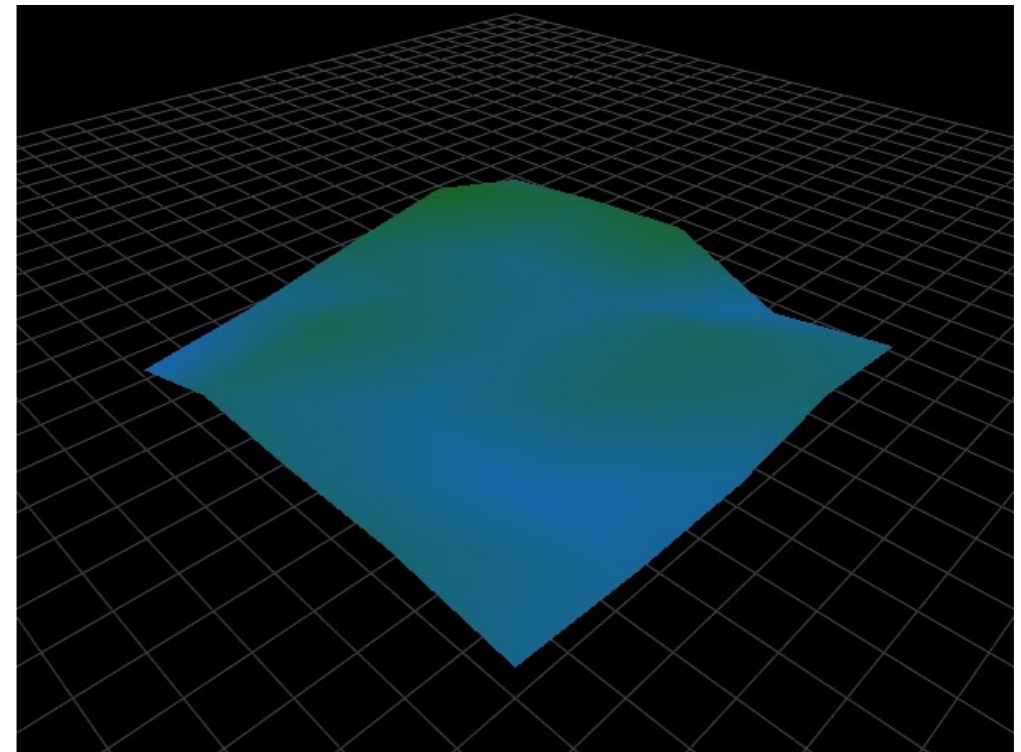
Height Field



Height Field



Input

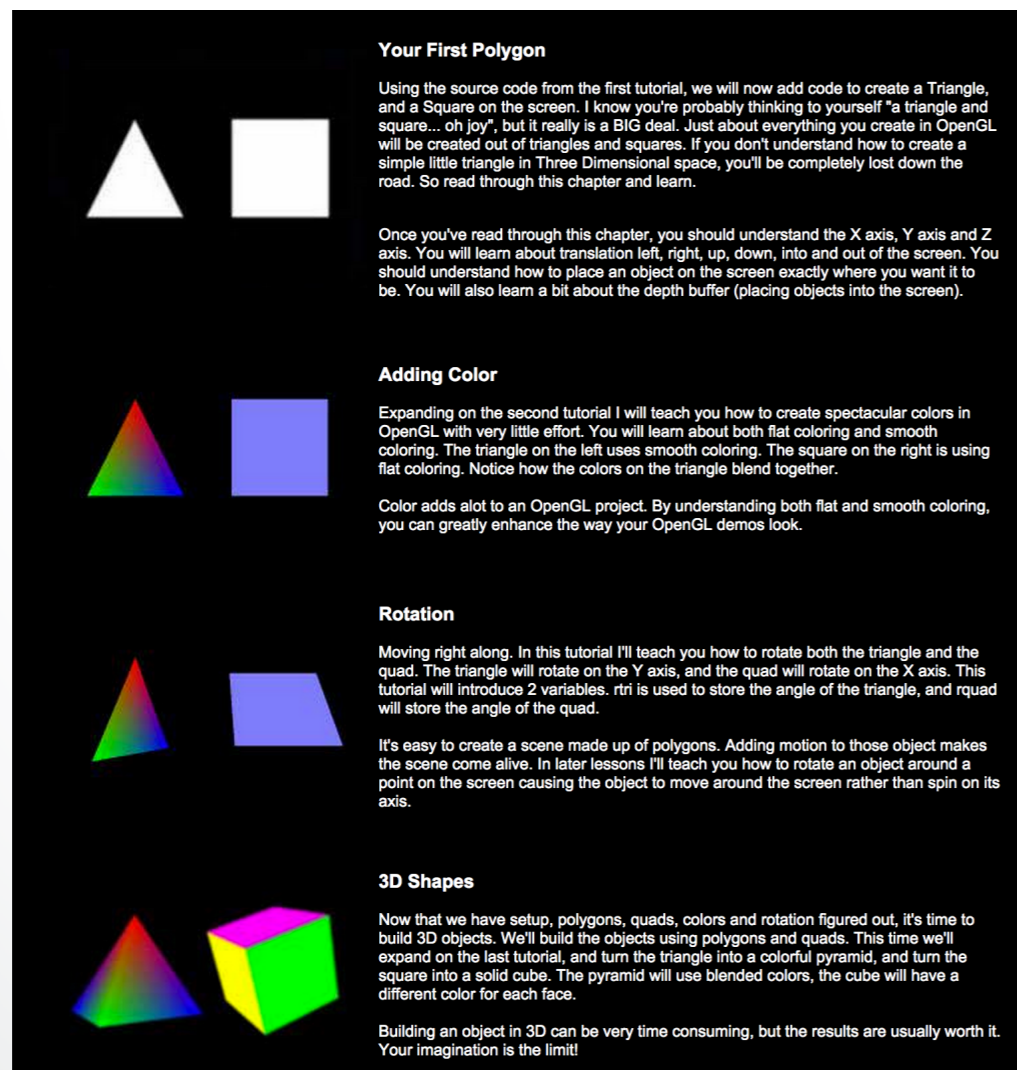


Output

Start coding ...

- NeHe Productions

http://nehe.gamedev.net/tutorial/lessons_01_05/22004/



Your First Polygon

Using the source code from the first tutorial, we will now add code to create a Triangle, and a Square on the screen. I know you're probably thinking to yourself "a triangle and square... oh joy", but it really is a BIG deal. Just about everything you create in OpenGL will be created out of triangles and squares. If you don't understand how to create a simple little triangle in Three Dimensional space, you'll be completely lost down the road. So read through this chapter and learn.

Once you've read through this chapter, you should understand the X axis, Y axis and Z axis. You will learn about translation left, right, up, down, into and out of the screen. You should understand how to place an object on the screen exactly where you want it to be. You will also learn a bit about the depth buffer (placing objects into the screen).

Adding Color

Expanding on the second tutorial I will teach you how to create spectacular colors in OpenGL with very little effort. You will learn about both flat coloring and smooth coloring. The triangle on the left uses smooth coloring. The square on the right is using flat coloring. Notice how the colors on the triangle blend together.

Color adds alot to an OpenGL project. By understanding both flat and smooth coloring, you can greatly enhance the way your OpenGL demos look.

Rotation

Moving right along. In this tutorial I'll teach you how to rotate both the triangle and the quad. The triangle will rotate on the Y axis, and the quad will rotate on the X axis. This tutorial will introduce 2 variables. rtri is used to store the angle of the triangle, and rquad will store the angle of the quad.

It's easy to create a scene made up of polygons. Adding motion to those object makes the scene come alive. In later lessons I'll teach you how to rotate an object around a point on the screen causing the object to move around the screen rather than spin on its axis.

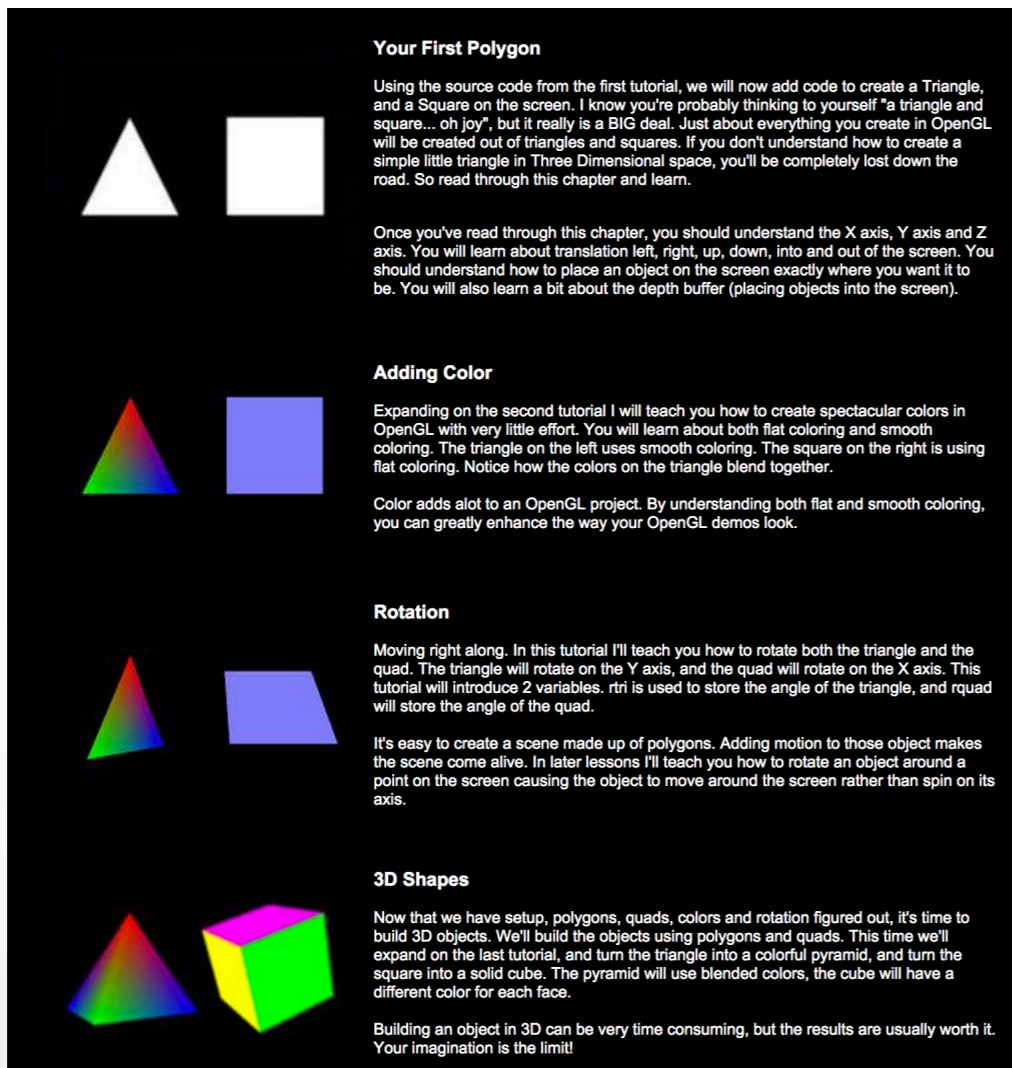
3D Shapes

Now that we have setup, polygons, quads, colors and rotation figured out, it's time to build 3D objects. We'll build the objects using polygons and quads. This time we'll expand on the last tutorial, and turn the triangle into a colorful pyramid, and turn the square into a solid cube. The pyramid will use blended colors, the cube will have a different color for each face.

Building an object in 3D can be very time consuming, but the results are usually worth it. Your imagination is the limit!

Start coding ...

- NeHe Productions
http://nehe.gamedev.net/tutorial/lessons_01_05/22004/
- OpenGL Programming Guide
<http://glprogramming.com/red/>



Your First Polygon

Using the source code from the first tutorial, we will now add code to create a Triangle, and a Square on the screen. I know you're probably thinking to yourself "a triangle and square... oh joy", but it really is a BIG deal. Just about everything you create in OpenGL will be created out of triangles and squares. If you don't understand how to create a simple little triangle in Three Dimensional space, you'll be completely lost down the road. So read through this chapter and learn.

Once you've read through this chapter, you should understand the X axis, Y axis and Z axis. You will learn about translation left, right, up, down, into and out of the screen. You should understand how to place an object on the screen exactly where you want it to be. You will also learn a bit about the depth buffer (placing objects into the screen).

Adding Color

Expanding on the second tutorial I will teach you how to create spectacular colors in OpenGL with very little effort. You will learn about both flat coloring and smooth coloring. The triangle on the left uses smooth coloring. The square on the right is using flat coloring. Notice how the colors on the triangle blend together.

Color adds alot to an OpenGL project. By understanding both flat and smooth coloring, you can greatly enhance the way your OpenGL demos look.

Rotation

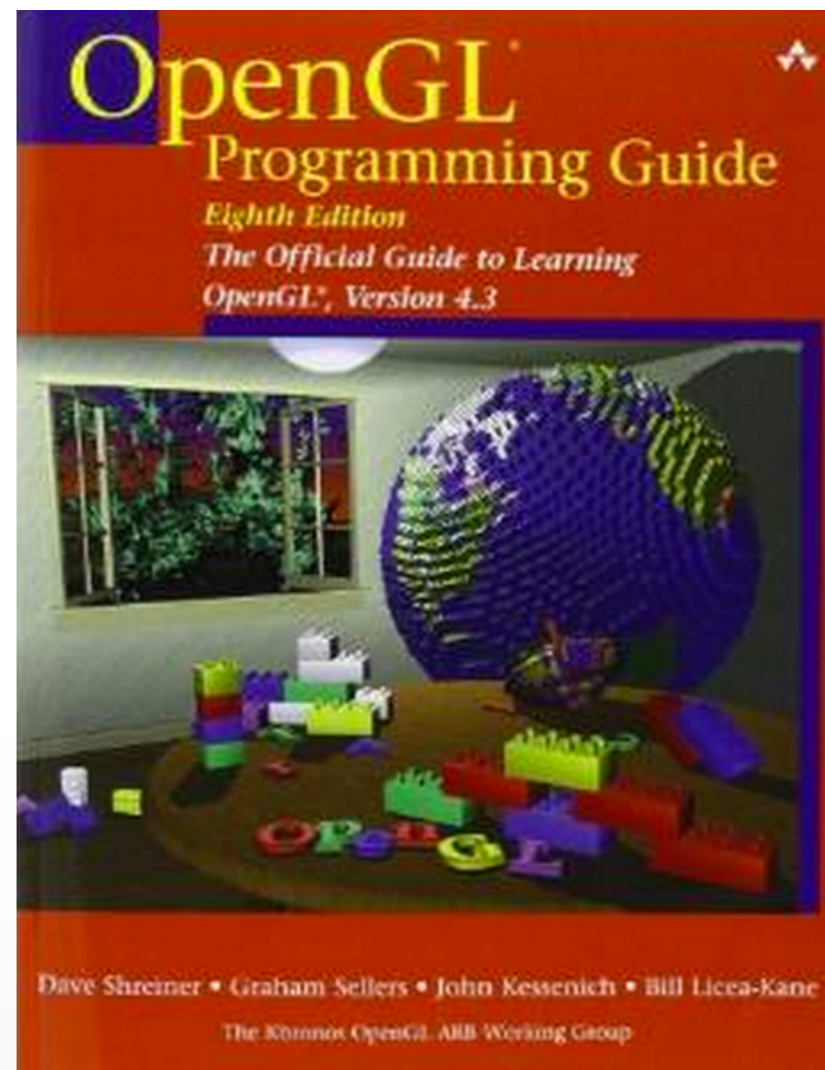
Moving right along. In this tutorial I'll teach you how to rotate both the triangle and the quad. The triangle will rotate on the Y axis, and the quad will rotate on the X axis. This tutorial will introduce 2 variables. rtri is used to store the angle of the triangle, and rquad will store the angle of the quad.

It's easy to create a scene made up of polygons. Adding motion to those object makes the scene come alive. In later lessons I'll teach you how to rotate an object around a point on the screen causing the object to move around the screen rather than spin on its axis.

3D Shapes

Now that we have setup, polygons, quads, colors and rotation figured out, it's time to build 3D objects. We'll build the objects using polygons and quads. This time we'll expand on the last tutorial, and turn the triangle into a colorful pyramid, and turn the square into a solid cube. The pyramid will use blended colors, the cube will have a different color for each face.

Building an object in 3D can be very time consuming, but the results are usually worth it. Your imagination is the limit!



Submission

The screenshot shows a Blackboard course interface. On the left, a navigation sidebar lists course items: '20153_csci_420_30053: Computer Graphics', 'Announcements', 'Syllabus', 'Content', 'Assignments' (circled in red), 'Discussion Board', 'Contacts', 'Tools', 'USC Code of Ethics', and 'Course Home'. The main content area is titled 'Assignments' and features a dark navigation bar with 'Build Content', 'Assessments', 'Tools', and 'Partner Content'. Below this, an 'Exercise 1' assignment is displayed with a document icon and the text: 'Please submit all the required materials for Exercise 1 here.'

- Deadline: **Monday, Sep 24, 2018 11:59 pm**
- Follow submission instructions on Ex. 1 webpage:
 - Upload a .zip compressed file named “Exercise1-YourName.zip” to blackboard
 - Include your code with comments
 - Include a readme file
 - Include JPEG frames or a video

Contact

The screenshot shows a Piazza forum interface for the course CSCI 420. The top navigation bar includes the Piazza logo, the course name 'CSCI 420', and links for 'Q & A', 'Resources', 'Statistics', and 'Manage Class'. Below this, there are tabs for 'polls', 'hw1', 'hw2', 'hw3', 'hw4', and 'other'. The main content area is split into a left sidebar and a right main area. The sidebar contains a 'New Post' button and a search bar. The main area displays a 'note' titled 'Welcome to CSCI 420 - Computer Graphics'. The note includes a link to 'Course Information and Syllabus', details about the 'Lecture' (Time: Tuesday @ 3:30pm - 6:50pm, Location: ZHS 352 (Zumberge Hall)), 'Discussion' (As needed, Time: Thursday @ TBA, Location: TBA), and 'Instructor' information (Prof. Dr. Hao Li, Office Hours: SAL 244, Tuesday @ 2:00pm - 3:00pm, Email: hao.li@usc.edu).

- Office Hours:
Prof. Dr. Hao Li - SAL 244, Tues @ 2:00pm - 3:00pm
Haiwei Chen - PHE 108, Weds @ 12:30pm - 1:30pm
Kyle Morgenroth - PHE 108, Mon @ 3:00pm - 4:00pm
- General questions about the assignment can be posted on Piazza
- Emails (include “CSCI_420” in the subject):
haiweich@usc.edu, kmorgenr@usc.edu

<http://cs420.hao-li.com>

Thanks!

