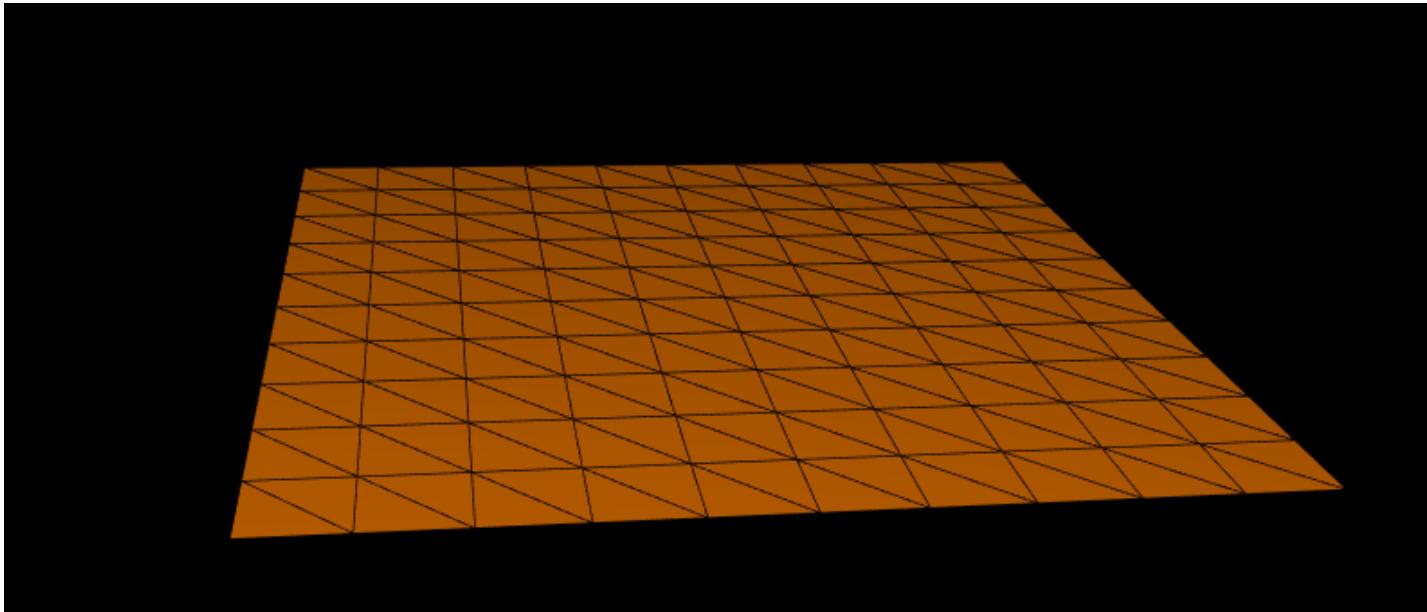# MP2 Starter Code Walkthrough

CS 418: Interactive Computer Graphics

Professor Eric Shaffer

ILLINOIS

Generating a Terrain

Today, we will generate a flat triangulated mesh

This can be the start of your code for MP 2

Creating a Terrain Class

We will create a JavaScript class for our terrain in the **Terrain.js** file

It will be rendered using the `drawElements` method

We will need to generate three buffers

- A buffer of the vertex positions

- A buffer of the triangles, specified by vertex indices

- A buffer of normal vectors
 - Each vertex will have a normal vector associated with it

The Terrain Geometry

We will create a set of triangles in the z=0 plane

- Each vertex will have coordinates $(x_i, y_i, 0)$

Example `div` = 10

The parameters for the triangle set will be:

- `div` the number of triangles along each axis
- `minX, maxX` the extent of the terrain along the x axis
- `minY, maxY` the extent of the terrain along the y axis

ILLINOIS
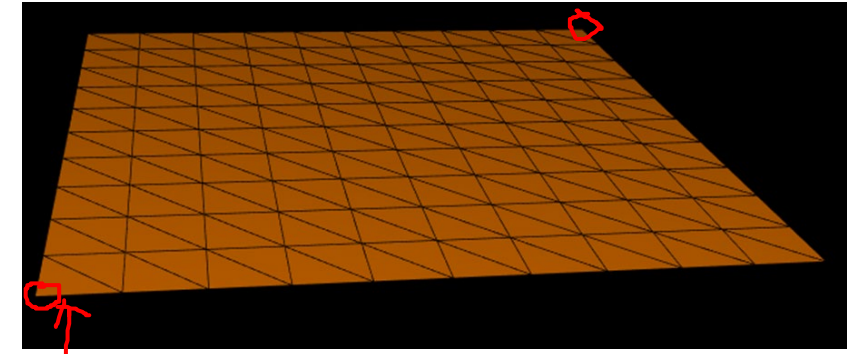
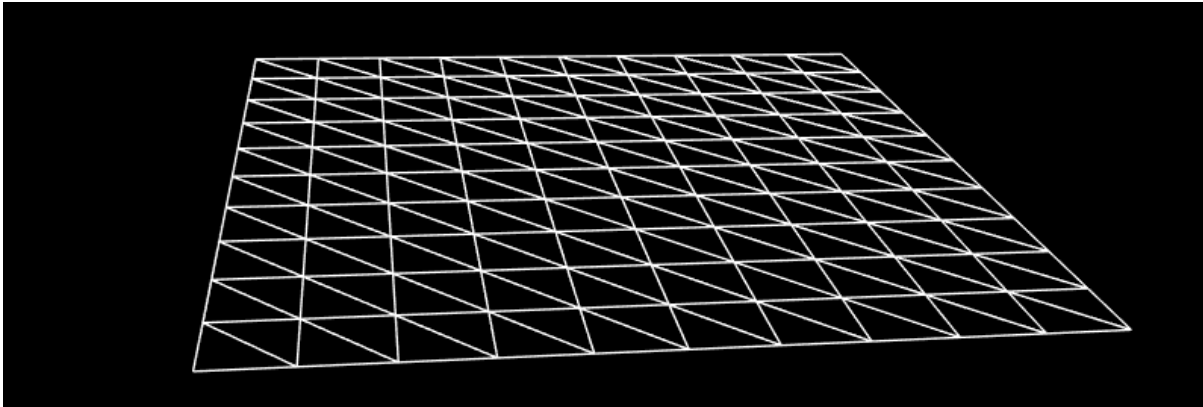# Terrain Geometry

- `div` the number of triangles along each axis
- `minX, maxX` the extent of the terrain along the x axis
- `minY, maxY` the extent of the terrain along the y axis

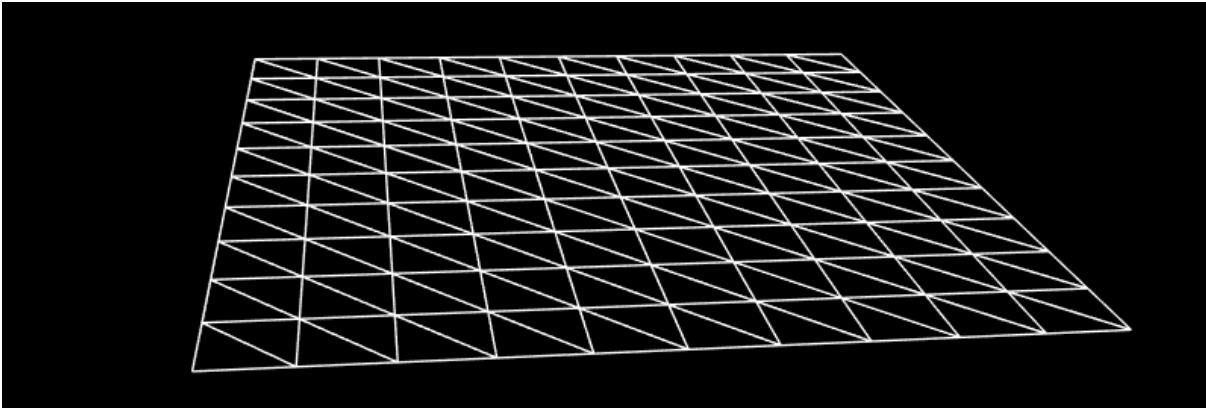

Vertex (i,j) is in the ith row and jth column with i,j in the range [0,div]

Generate it so that (0,0) is the first vertex with coordinates (minX,minY,0)

What are the indices of the faces?

- How many total vertices?
- How many total triangles?

Terrain Geometry

- `div` the number of triangles along each axis
- `minX, maxX` the extent of the terrain along the x axis
- `minY, maxY` the extent of the terrain along the y axis

If there are **div** edges along an axis then there are **div**+1 vertices along the axis

There are (**div**+1)^2 vertices in the terrain

There are 2**div**^{2} triangles

- How many total vertices?
- How many total triangles?

Your Tasks

Write the following functions in the terrain.js file

- getVertex
  - ...this is a function you can use (or not) to get the coordinates of a vertex

- setVertex
  - ...this is a function you can use (or not) to set the coordinates of a vertex

- generateTriangles
  - ...this function fills the javascript arrays containing the
    - Vertex positions
    - Triangle connectivity
    - Vertex normals

GenerateTriangles: Vertex Positions

...you can write it any way you want that works

...our solution computes the following values:

deltaX
the distance along the x axis between two neighboring vertices in a row

deltaY
the distance along the y axis between two neighboring vertices in a column

Then, a nested loop, iterating over the rows and columns, sets the vertex positions

The delta values are used to increment the X and Y coordinates in the loop

ILLINOIS

GenerateTriangles: Triangle connectivity

...our solution loops over the rows and columns of triangles
- in the same order as the vertices were created
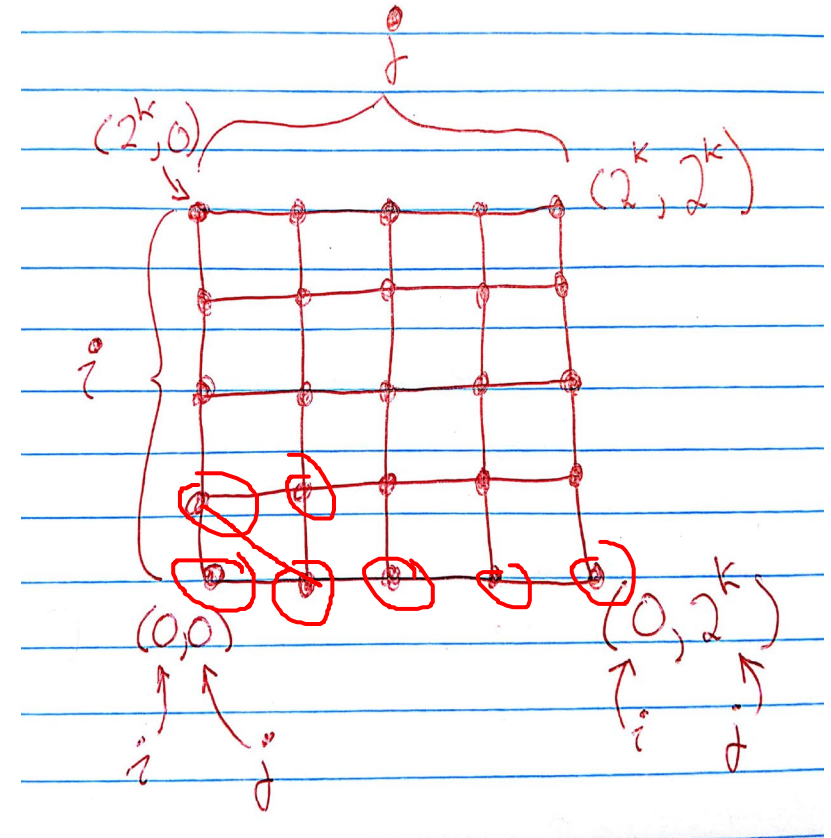- the body of the loop inserts two triangles into the array

The illustration here shows a grid with $div=2^k=4$

You can split each quad into two triangles and

You need to figure what the indexing for the vertices is

For example the vertex in the bottom left corner has index 0

The vertex above it would have index 5

One Solution

One possible solution

```
getVertex(v,i,j)
{
    //Your code here
    var vid = 3*(i*(this.div+1) + j);
    v[0]= this.vBuffer[vid];
    v[1]= this.vBuffer[vid+1];
    v[2]= this.vBuffer[vid+2];
}
```

```
setVertex(v,i,j)
{
    //Your code here

    var vid = 3*(i*(this.div+1) + j);
    this.vBuffer[vid] =v[0];
    this.vBuffer[vid+1]=v[1];
    this.vBuffer[vid+2]=v[2];
}
```

Indexing with `drawElements`

```
void gl.drawElements(mode, count, type, offset);
```

type

A `GLenum` specifying the type of the values in the element array buffer. Possible values are:

- `gl.UNSIGNED_BYTE`
- `gl.UNSIGNED_SHORT`
- When using the `OES_element_index_uint` extension:
  - `gl.UNSIGNED_INT`

- In WebGL 1.0 you are limited to using 2-byte integer indices
  - How may vertices could you index?
- There's an extension to let you use 4-byte integer indices
- Need to initialize/check with

```
var ext = gl.getExtension('OES_element_index_uint');
```

I ILLINOIS

Computing Normal Vectors

For shading you need per vertex normal

Vertex buffer contains N vertices with x,y,z coordinates

So 3N floating point numbers
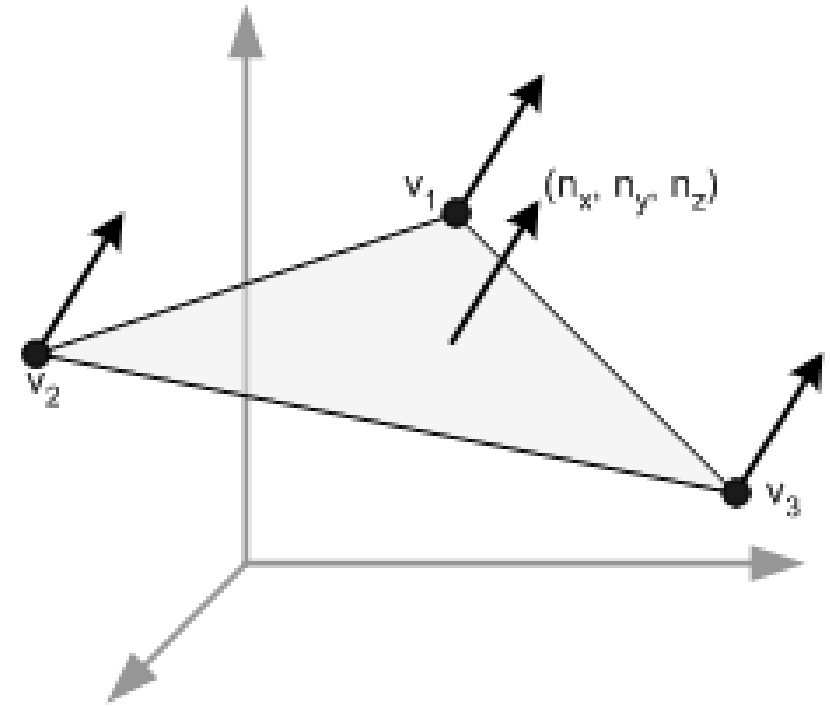
Normal buffer will contain N normals with x,y,z

So 3N floating point numbers

The Kth vertex will start at location 3(K-1) in the buffer
The normal for vertex K will start at location 3(K-1)

What are the normal vectors for the terrain in our code?

How would you compute them in general?

generateTriangles()

```javascript
var deltaX=(this.maxX-this.minX)/this.div;
var deltaY=(this.maxY-this.minY)/this.div;

for(var i=0;i<=this.div;i++)
    for(var j=0;j<=this.div;j++)
    {

        this.vBuffer.push(this.minX+deltaX*j);
        this.vBuffer.push(this.minY+deltaY*i);
        this.vBuffer.push(0);

        this.nBuffer.push(0);
        this.nBuffer.push(0);
        this.nBuffer.push(1);
    }

for(var i=0;i<this.div;i++)
    for(var j=0;j<this.div;j++)
    {
        var vid = i*(this.div+1) + j;
        this.fBuffer.push(vid);
        this.fBuffer.push(vid+1);
        this.fBuffer.push(vid+this.div+1);

        this.fBuffer.push(vid+1);
        this.fBuffer.push(vid+1+this.div+1);
        this.fBuffer.push(vid+this.div+1);
    }
```

The rest is up to you....

https://illinois-cs418.github.io/assignments/mp2-1.html

Name	Points	Description
Runs and renders	1	Program runs without crashing and renders visible polygons.
Commented	1	Each function in your code is commented in the required style.
Normals	1	Normal vectors for the terrain are computed and handled correctly.
Terrain Generation	2	The terrain is modeled using the random partition method.
Elevation Color Map	2	The shading of the terrain changes by mapping different ranges of elevation to different colors.
Phong Shading	1	Shading is accomplished using the Phong reflection model in the fragment shader.
Creativity	1	The color map is visually interesting and the terrain exhibits good aesthethic quality (e.g. not totally random).
Total	9	

ILLINOIS