

CSCI 420 Computer Graphics

**Helper slides,
hw2 (roller coaster)**

Jernej Barbic
University of Southern California

Use `std::vector` to store complex geometry

```
#include <vector>
using namespace std;
vector<float> pos;
vector<float> uvs;

void addTriangle(float posA[3], float posB[3], float posC[3],
                float uvA[2], float uvB[2], float uvC[2])
{
    pos.push_back(posA[0]); pos.push_back(posA[1]); pos.push_back(posA[2]);
    pos.push_back(posB[0]); pos.push_back(posB[1]); pos.push_back(posB[2]);
    pos.push_back(posC[0]); pos.push_back(posC[1]); pos.push_back(posC[2]);
    uvs.push_back(uvA[0]); uvs.push_back(uvA[1]);
    uvs.push_back(uvB[0]); uvs.push_back(uvB[1]);
    uvs.push_back(uvC[0]); uvs.push_back(uvC[1]);
}
```

Init the VBOs using std::vector

```
GLuint buffer;
void initVBO()
{
    glGenBuffers(1, &buffer);
    glBindBuffer(GL_ARRAY_BUFFER, buffer);
    glBufferData(GL_ARRAY_BUFFER, (pos.size() + uvs.size()) * sizeof(float),
        NULL, GL_STATIC_DRAW); // init buffer's size, but don't assign any data

    // upload position data
    glBufferSubData(GL_ARRAY_BUFFER, 0,
        pos.size() * sizeof(float), pos.data());

    // upload uv data
    glBufferSubData(GL_ARRAY_BUFFER, pos.size() * sizeof(float),
        uvs.size() * sizeof(float), uvs.data());
}
```