```cpp
#include "shapes.h"
#include <iostream>
#include <sstream>

#include <GL/glew.h>
#include <GLFW/glfw3.h>
#include <glm/glm.hpp>

Shapes::Shapes() {

};

Shapes::~Shapes() {

}

void Shapes::LoadObj() {

    std::vector< glm::vec3 > obj_vertices;
    std::vector< unsigned int > vertexIndices;
    istringstream rawDataStream(rawData);
    string dataLine;  int linesDone = 0;

    while (std::getline(rawDataStream, dataLine)) {
        if (dataLine.find("v ") != string::npos) {  // does this line have a vector?
            glm::vec3 vertex;

            int foundStart = dataLine.find(" ");  int foundEnd = dataLine.find(" ", foundStart + 1);
            vertex.x = stof(dataLine.substr(foundStart, foundEnd - foundStart));

            foundStart = foundEnd; foundEnd = dataLine.find(" ", foundStart + 1);
            vertex.y = stof(dataLine.substr(foundStart, foundEnd - foundStart));

            foundStart = foundEnd; foundEnd = dataLine.find(" ", foundStart + 1);
            vertex.z = stof(dataLine.substr(foundStart, foundEnd - foundStart));

            obj_vertices.push_back(vertex);
        }
        else if (dataLine.find("f ") != string::npos) { // does this line defines a triangle face?
            string parts[3];

            int foundStart = dataLine.find(" ");  int foundEnd = dataLine.find(" ", foundStart + 1);
            parts[0] = dataLine.substr(foundStart + 1, foundEnd - foundStart - 1);

            foundStart = foundEnd; foundEnd = dataLine.find(" ", foundStart + 1);
            parts[1] = dataLine.substr(foundStart + 1, foundEnd - foundStart - 1);

            foundStart = foundEnd; foundEnd = dataLine.find(" ", foundStart + 1);
            parts[2] = dataLine.substr(foundStart + 1, foundEnd - foundStart - 1);

            for (int i = 0; i < 3; i++) {       // for each part

                vertexIndices.push_back(stoul(parts[i].substr(0, parts[i].find("/"))));

                int firstSlash = parts[i].find("/"); int secondSlash = parts[i].find("/", firstSlash + 1);

                if (firstSlash != (secondSlash + 1)) {  // there is texture coordinates.
                                                        // add code for my texture coordintes here.
                }
            }
        }

        linesDone++;
    }

    for (unsigned int i = 0; i < vertexIndices.size(); i += 3) {
        vertexPositions.push_back(obj_vertices[vertexIndices[i + 0] - 1].x);
        vertexPositions.push_back(obj_vertices[vertexIndices[i + 0] - 1].y);
        vertexPositions.push_back(obj_vertices[vertexIndices[i + 0] - 1].z);

        vertexPositions.push_back(obj_vertices[vertexIndices[i + 1] - 1].x);
        vertexPositions.push_back(obj_vertices[vertexIndices[i + 1] - 1].y);
        vertexPositions.push_back(obj_vertices[vertexIndices[i + 1] - 1].z);

        vertexPositions.push_back(obj_vertices[vertexIndices[i + 2] - 1].x);
        vertexPositions.push_back(obj_vertices[vertexIndices[i + 2] - 1].y);
        vertexPositions.push_back(obj_vertices[vertexIndices[i + 2] - 1].z);
    }
}


void Shapes::Load() {
    static const char * vs_source[] = { R"(
#version 330 core

in vec4 position;
uniform mat4 mv_matrix;
uniform mat4 proj_matrix;
```

```

void main(void){
    gl_Position = proj_matrix * mv_matrix * position;
}
)" };

    static const char * fs_source[] = { R"(
#version 330 core

uniform vec4 inColor;
out vec4 color;

void main(void){
    color = inColor;
}
)" };

    program = glCreateProgram();
    GLuint fs = glCreateShader(GL_FRAGMENT_SHADER);
    glShaderSource(fs, 1, fs_source, NULL);
    glCompileShader(fs);
    checkErrorShader(fs);

    GLuint vs = glCreateShader(GL_VERTEX_SHADER);
    glShaderSource(vs, 1, vs_source, NULL);
    glCompileShader(vs);
    checkErrorShader(vs);

    glAttachShader(program, vs);
    glAttachShader(program, fs);

    glLinkProgram(program);

    mv_location = glGetUniformLocation(program, "mv_matrix");
    proj_location = glGetUniformLocation(program, "proj_matrix");
    color_location = glGetUniformLocation(program, "inColor");

    glGenVertexArrays(1, &vao);
    glBindVertexArray(vao);

    glGenBuffers(1, &buffer);
    glBindBuffer(GL_ARRAY_BUFFER, buffer);
    glBufferData(GL_ARRAY_BUFFER,
        vertexPositions.size() * sizeof(GLfloat),
        &vertexPositions[0],
        GL_STATIC_DRAW);
    glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, 0, NULL);
    glEnableVertexAttribArray(0);

    glLinkProgram(0);   // unlink
    glDisableVertexAttribArray(0); // Disable
    glBindVertexArray(0);   // Unbind
}
```

```cpp

void Shapes::Draw() {
    glUseProgram(program);
    glBindVertexArray(vao);
    glEnableVertexAttribArray(0);

    glUniformMatrix4fv(proj_location, 1, GL_FALSE, &proj_matrix[0][0]);
    glUniformMatrix4fv(mv_location, 1, GL_FALSE, &mv_matrix[0][0]);

    glUniform4f(color_location, fillColor.r, fillColor.g, fillColor.b, 
      fillColor.a);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glDrawArrays(GL_TRIANGLES, 0, vertexPositions.size() / 3);

    glUniform4f(color_location, lineColor.r, lineColor.g, lineColor.b, 
      lineColor.a);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);  glLineWidth(lineWidth);
    glDrawArrays(GL_TRIANGLES, 0, vertexPositions.size() / 3);
}


void Shapes::checkErrorShader(GLuint shader) {
    // Get log length
    GLint maxLength;
    glGetShaderiv(shader, GL_INFO_LOG_LENGTH, &maxLength);

    // Init a string for it
    std::vector<GLchar> errorLog(maxLength);

    if (maxLength > 1) {
        // Get the log file
        glGetShaderInfoLog(shader, maxLength, &maxLength, &errorLog[0]);

        cout << "--------------Shader compilation error-------------\n";
        cout << errorLog.data();
    }
}

Cube::Cube() {
    // Exported from Blender a cube by default (OBJ File)
    rawData = R"(
o Plane
v -1.000000 0.000000 1.000000
v 1.000000 0.000000 1.000000
v -1.000000 0.000000 -1.000000
v 1.000000 0.000000 -1.000000
vn 0.0000 1.0000 0.0000
usemtl None
s off
f 2 3 1
f 2 4 3
)";

```

```
    LoadObj();
}

Cube::~Cube() {

}

Sphere::Sphere() {

    rawData = R"(
o Sphere
v -0.097545 0.490393 0.000000
v -0.277785 0.415735 0.000000
v -0.415735 0.277785 0.000000
v -0.490393 0.097545 0.000000
v -0.490393 -0.097545 0.000000
v -0.415735 -0.277785 0.000000
v -0.277785 -0.415735 0.000000
v -0.097545 -0.490393 0.000000
v -0.090120 0.490393 -0.037329
v -0.256640 0.415735 -0.106304
v -0.384089 0.277785 -0.159095
v -0.453064 0.097545 -0.187665
v -0.453064 -0.097545 -0.187665
v -0.384089 -0.277785 -0.159095
v -0.256640 -0.415735 -0.106304
v -0.090120 -0.490393 -0.037329
v -0.068975 0.490393 -0.068975
v -0.196424 0.415735 -0.196424
v -0.293969 0.277785 -0.293969
v -0.346760 0.097545 -0.346760
v -0.346760 -0.097545 -0.346760
v -0.293969 -0.277785 -0.293969
v -0.196424 -0.415735 -0.196424
v -0.068975 -0.490393 -0.068975
v -0.037329 0.490393 -0.090120
v -0.106304 0.415735 -0.256640
v -0.159095 0.277785 -0.384089
v -0.187665 0.097545 -0.453064
v -0.187665 -0.097545 -0.453064
v -0.159095 -0.277785 -0.384089
v -0.106304 -0.415735 -0.256640
v -0.037329 -0.490393 -0.090120
v 0.000000 0.490393 -0.097545
v 0.000000 0.415735 -0.277785
v 0.000000 0.277785 -0.415735
v 0.000000 0.097545 -0.490393
v 0.000000 -0.097545 -0.490393
v 0.000000 -0.277785 -0.415735
v 0.000000 -0.415735 -0.277785
v 0.000000 -0.490393 -0.097545
v 0.037329 0.490393 -0.090120
v 0.106304 0.415735 -0.256640
```

```
v 0.159095 0.277785 -0.384089
v 0.187665 0.097545 -0.453064
v 0.187665 -0.097545 -0.453064
v 0.159095 -0.277785 -0.384089
v 0.106304 -0.415735 -0.256640
v 0.037329 -0.490393 -0.090120
v 0.068975 0.490393 -0.068975
v 0.196424 0.415735 -0.196424
v 0.293969 0.277785 -0.293969
v 0.346760 0.097545 -0.346760
v 0.346760 -0.097545 -0.346760
v 0.293969 -0.277785 -0.293969
v 0.196424 -0.415735 -0.196424
v 0.068975 -0.490393 -0.068975
v 0.090120 0.490393 -0.037329
v 0.256640 0.415735 -0.106304
v 0.384089 0.277785 -0.159095
v 0.453064 0.097545 -0.187665
v 0.453064 -0.097545 -0.187665
v 0.384089 -0.277785 -0.159095
v 0.256640 -0.415735 -0.106304
v 0.090120 -0.490393 -0.037329
v 0.097545 0.490393 0.000000
v 0.277785 0.415735 -0.000000
v 0.415735 0.277785 0.000000
v 0.490393 0.097545 0.000000
v 0.490393 -0.097545 0.000000
v 0.415735 -0.277785 0.000000
v 0.277785 -0.415735 0.000000
v 0.097545 -0.490393 -0.000000
v 0.090120 0.490393 0.037329
v 0.256640 0.415735 0.106304
v 0.384089 0.277785 0.159095
v 0.453064 0.097545 0.187665
v 0.453064 -0.097545 0.187665
v 0.384089 -0.277785 0.159095
v 0.256640 -0.415735 0.106304
v 0.090120 -0.490393 0.037329
v 0.068975 0.490393 0.068975
v 0.196424 0.415735 0.196424
v 0.293969 0.277785 0.293969
v 0.346760 0.097545 0.346760
v 0.346760 -0.097545 0.346760
v 0.293969 -0.277785 0.293969
v 0.196424 -0.415735 0.196424
v 0.068975 -0.490393 0.068975
v 0.000000 -0.500000 0.000000
v 0.037329 0.490393 0.090120
v 0.106304 0.415735 0.256640
v 0.159095 0.277785 0.384089
v 0.187665 0.097545 0.453064
v 0.187665 -0.097545 0.453064
v 0.159095 -0.277785 0.384089
```

```
v 0.106304 -0.415735 0.256640
v 0.037329 -0.490393 0.090120
v 0.000000 0.490393 0.097545
v 0.000000 0.415735 0.277785
v 0.000000 0.277785 0.415735
v 0.000000 0.097545 0.490392
v 0.000000 -0.097545 0.490392
v 0.000000 -0.277785 0.415735
v 0.000000 -0.415735 0.277785
v 0.000000 -0.490393 0.097545
v -0.037329 0.490393 0.090120
v -0.106304 0.415735 0.256640
v -0.159095 0.277785 0.384089
v -0.187665 0.097545 0.453063
v -0.187665 -0.097545 0.453063
v -0.159095 -0.277785 0.384089
v -0.106304 -0.415735 0.256640
v -0.037329 -0.490393 0.090120
v -0.068975 0.490393 0.068975
v -0.196424 0.415735 0.196424
v -0.293969 0.277785 0.293969
v -0.346760 0.097545 0.346760
v -0.346760 -0.097545 0.346760
v -0.293969 -0.277785 0.293969
v -0.196423 -0.415735 0.196424
v -0.068975 -0.490393 0.068975
v 0.000000 0.500000 0.000000
v -0.090120 0.490393 0.037329
v -0.256640 0.415735 0.106304
v -0.384088 0.277785 0.159095
v -0.453063 0.097545 0.187665
v -0.453063 -0.097545 0.187665
v -0.384088 -0.277785 0.159095
v -0.256640 -0.415735 0.106304
v -0.090120 -0.490393 0.037329
s off
f 7 14 15
f 3 10 11
f 12 3 11
f 8 15 16
f 5 12 13
f 2 125 124
f 2 9 10
f 6 13 14
f 89 8 16
f 122 17 9
f 7 128 6
f 20 27 28
f 8 129 7
f 22 29 30
f 19 26 27
f 29 36 37
f 31 22 30
```

```
f 89 16 24
f 26 33 34
f 24 31 32
f 28 35 36
f 122 25 17
f 27 34 35
f 37 44 45
f 38 29 37
f 89 24 32
f 42 33 41
f 32 39 40
f 36 43 44
f 31 38 39
f 122 33 25
f 43 34 42
f 45 52 53
f 46 37 45
f 89 32 40
f 43 50 51
f 48 39 47
f 52 43 51
f 39 46 47
f 50 41 49
f 122 41 33
f 53 60 61
f 47 54 55
f 46 53 54
f 48 55 56
f 60 51 59
f 58 49 57
f 122 49 41
f 89 40 48
f 61 68 69
f 55 62 63
f 54 61 62
f 51 58 59
f 58 65 66
f 68 59 67
f 122 57 49
f 56 63 64
f 89 48 56
f 63 70 71
f 62 69 70
f 59 66 67
f 69 76 77
f 66 73 74
f 122 65 57
f 64 71 72
f 76 67 75
f 89 56 64
f 79 70 78
f 70 77 78
f 67 74 75
```

```
f 77 84 85
f 72 79 80
f 122 73 65
f 76 83 84
f 89 64 72
f 74 81 82
f 87 78 86
f 86 77 85
f 75 82 83
f 85 93 94
f 80 87 88
f 84 92 93
f 122 81 73
f 89 72 80
f 91 81 90
f 87 95 96
f 86 94 95
f 83 91 92
f 94 101 102
f 93 100 101
f 89 80 88
f 122 90 81
f 91 98 99
f 88 96 97
f 95 102 103
f 92 99 100
f 102 109 110
f 96 103 104
f 122 98 90
f 89 88 97
f 99 106 107
f 105 96 104
f 109 100 108
f 108 99 107
f 110 117 118
f 104 111 112
f 122 106 98
f 89 97 105
f 107 114 115
f 103 110 111
f 117 108 116
f 113 104 112
f 108 115 116
f 118 126 127
f 120 111 119
f 122 114 106
f 115 123 124
f 111 118 119
f 89 105 113
f 113 120 121
f 126 116 125
f 119 127 128
f 116 124 125
```

```
f 120 128 129
f 89 113 121
f 121 129 130
f 122 123 114
f 89 121 130
f 122 1 123
f 89 130 8
f 3 126 125
f 5 126 4
f 15 22 23
f 10 17 18
f 24 15 23
f 13 20 21
f 18 25 26
f 14 21 22
f 21 28 29
f 12 19 20
f 11 18 19
f 1 124 123
f 122 9 1
f 6 127 5
f 7 6 14
f 3 2 10
f 12 4 3
f 8 7 15
f 5 4 12
f 2 3 125
f 2 1 9
f 6 5 13
f 7 129 128
f 20 19 27
f 8 130 129
f 22 21 29
f 19 18 26
f 29 28 36
f 31 23 22
f 26 25 33
f 24 23 31
f 28 27 35
f 27 26 34
f 37 36 44
f 38 30 29
f 42 34 33
f 32 31 39
f 36 35 43
f 31 30 38
f 43 35 34
f 45 44 52
f 46 38 37
f 43 42 50
f 48 40 39
f 52 44 43
f 39 38 46
```

```
f 50 42 41
f 53 52 60
f 47 46 54
f 46 45 53
f 48 47 55
f 60 52 51
f 58 50 49
f 61 60 68
f 55 54 62
f 54 53 61
f 51 50 58
f 58 57 65
f 68 60 59
f 56 55 63
f 63 62 70
f 62 61 69
f 59 58 66
f 69 68 76
f 66 65 73
f 64 63 71
f 76 68 67
f 79 71 70
f 70 69 77
f 67 66 74
f 77 76 84
f 72 71 79
f 76 75 83
f 74 73 81
f 87 79 78
f 86 78 77
f 75 74 82
f 85 84 93
f 80 79 87
f 84 83 92
f 91 82 81
f 87 86 95
f 86 85 94
f 83 82 91
f 94 93 101
f 93 92 100
f 91 90 98
f 88 87 96
f 95 94 102
f 92 91 99
f 102 101 109
f 96 95 103
f 99 98 106
f 105 97 96
f 109 101 100
f 108 100 99
f 110 109 117
f 104 103 111
f 107 106 114
```

```
f 103 102 110
f 117 109 108
f 113 105 104
f 108 107 115
f 118 117 126
f 120 112 111
f 115 114 123
f 111 110 118
f 113 112 120
f 126 117 116
f 119 118 127
f 116 115 124
f 120 119 128
f 121 120 129
f 3 4 126
f 5 127 126
f 15 14 22
f 10 9 17
f 24 16 15
f 13 12 20
f 18 17 25
f 14 13 21
f 21 20 28
f 12 11 19
f 11 10 18
f 1 2 124
f 6 128 127

)";

    LoadObj();
}

Sphere::~Sphere() {

}

Arrow::Arrow() {

    rawData = R"(
o Cone
v 0.000000 0.800000 -0.100000
v 0.070711 0.800000 -0.070711
v 0.100000 0.800000 -0.000000
v 0.000000 1.000000 0.000000
v 0.070711 0.800000 0.070711
v -0.000000 0.800000 0.100000
v -0.070711 0.800000 0.070711
v -0.100000 0.800000 -0.000000
v -0.070711 0.800000 -0.070711
s off
f 4 7 6
f 5 7 2
```

```
f 4 8 7
f 3 4 5
f 5 4 6
f 4 9 8
f 4 1 9
f 2 1 4
f 2 4 3
f 9 1 2
f 2 3 5
f 5 6 7
f 7 8 9
f 9 2 7
o Cylinder
v 0.000000 0.000000 -0.050000
v 0.009755 0.900000 -0.049039
v 0.019134 0.000000 -0.046194
v 0.027779 0.900000 -0.041573
v 0.035355 0.000000 -0.035355
v 0.041573 0.900000 -0.027779
v 0.046194 0.000000 -0.019134
v 0.049039 0.900000 -0.009755
v 0.050000 0.000000 -0.000000
v 0.049039 0.900000 0.009755
v 0.046194 0.000000 0.019134
v 0.041573 0.900000 0.027779
v 0.035355 0.000000 0.035355
v 0.027779 0.900000 0.041573
v 0.019134 0.000000 0.046194
v 0.009755 0.900000 0.049039
v -0.000000 0.000000 0.050000
v -0.009755 0.900000 0.049039
v -0.019134 0.000000 0.046194
v -0.027779 0.900000 0.041573
v -0.035355 0.000000 0.035355
v -0.041574 0.900000 0.027778
v -0.046194 0.000000 0.019134
v -0.049039 0.900000 0.009754
v -0.050000 0.000000 -0.000000
v -0.049039 0.900000 -0.009755
v -0.046194 0.000000 -0.019134
v -0.041573 0.900000 -0.027779
v -0.035355 0.000000 -0.035355
v -0.027778 0.900000 -0.041574
v -0.019134 0.000000 -0.046194
v -0.009754 0.900000 -0.049039
s off
f 13 15 14
f 16 14 15
f 17 19 18
f 18 16 17
f 19 21 20
f 20 18 19
f 21 23 22
```

```
f 22 20 21
f 23 25 24
f 24 22 23
f 25 27 26
f 26 24 25
f 27 29 28
f 28 26 27
f 29 31 30
f 30 28 29
f 31 33 32
f 32 30 31
f 33 35 34
f 34 32 33
f 35 37 36
f 36 34 35
f 37 39 38
f 38 36 37
f 41 40 39
f 40 38 39
f 41 10 40
f 29 21 37
f 11 12 10
f 24 32 16
f 15 17 16
f 11 13 12
f 14 12 13
f 10 41 11
f 13 11 41
f 41 39 37
f 37 35 33
f 33 31 29
f 29 27 25
f 25 23 29
f 21 19 17
f 17 15 13
f 13 41 37
f 37 33 29
f 29 23 21
f 21 17 13
f 13 37 21
f 40 10 12
f 12 14 16
f 16 18 20
f 20 22 24
f 24 26 28
f 28 30 32
f 32 34 36
f 36 38 40
f 40 12 16
f 16 20 24
f 24 28 32
f 32 36 40
f 40 16 32
```

```
)";

    LoadObj();
}

Arrow::~Arrow() {

}
```