

EBU7405

3D Graphics Programming Tools

OpenGL Programming Techniques

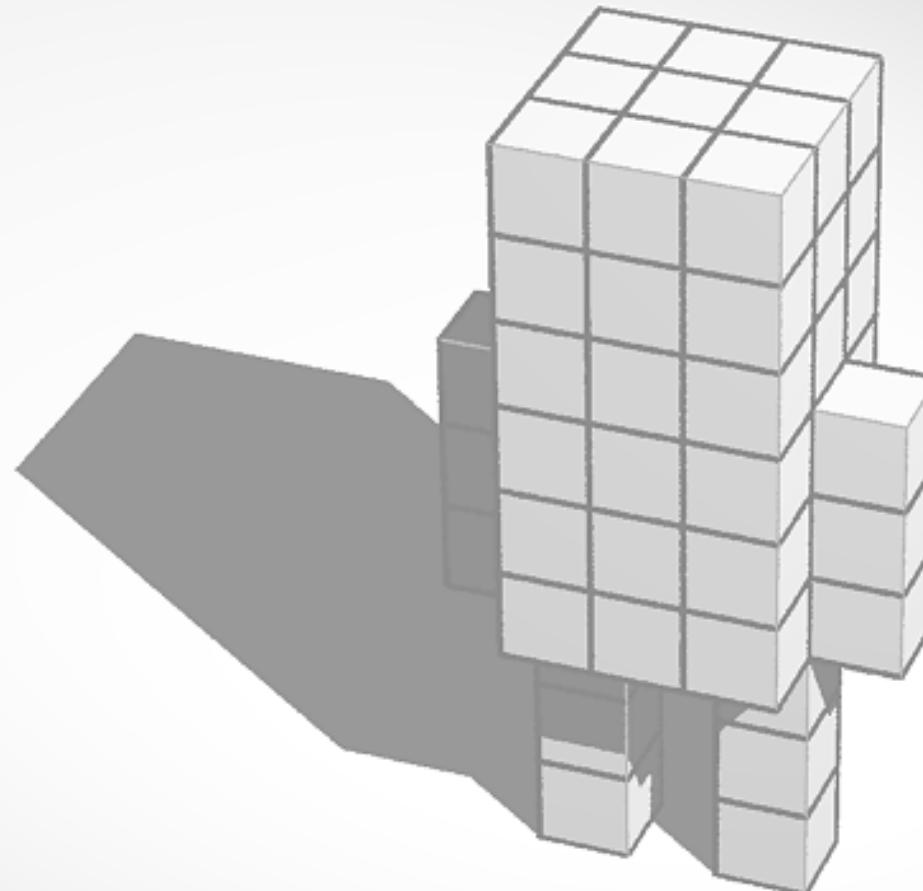
Dr. Xianhui Cherry Che

x.che@qmul.ac.uk

Topics

- Pixel Art
- Modular Programming
- Troubleshooting

Pixel Art



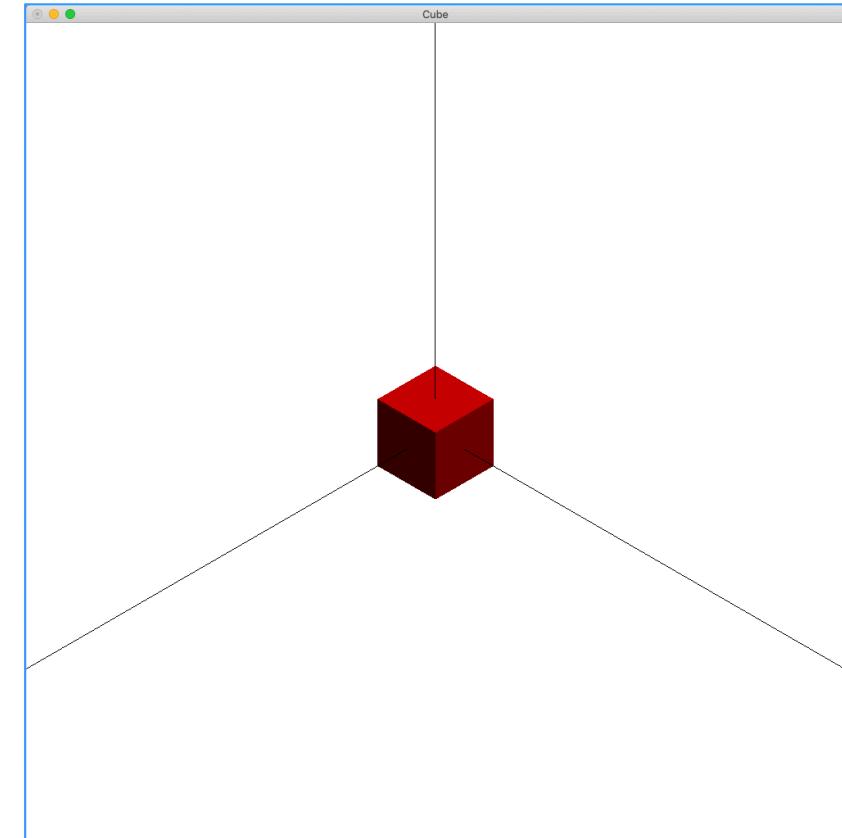
Cube Pixels

- Define a function to draw a cube with customised position and size:

```
void oneCube(GLfloat x, GLfloat y, GLfloat z, GLfloat L){  
    glPushMatrix();  
    glTranslated(x, y, z);  
    glutSolidCube(L);  
    glPopMatrix();  
}
```

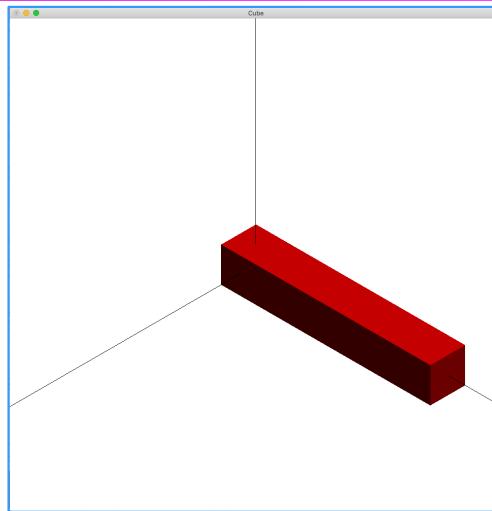
- And use it however you like:

```
oneCube(0, 0, 0, 1);
```

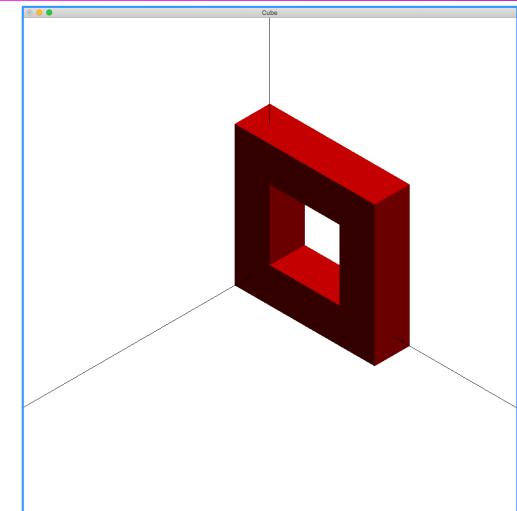


Cube Pixels

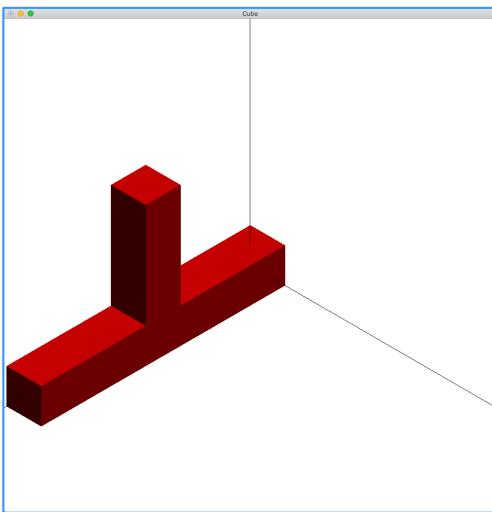
```
oneCube(0, 0, 0, 1);
oneCube(1, 0, 0, 1);
oneCube(2, 0, 0, 1);
oneCube(3, 0, 0, 1);
oneCube(4, 0, 0, 1);
oneCube(5, 0, 0, 1);
```



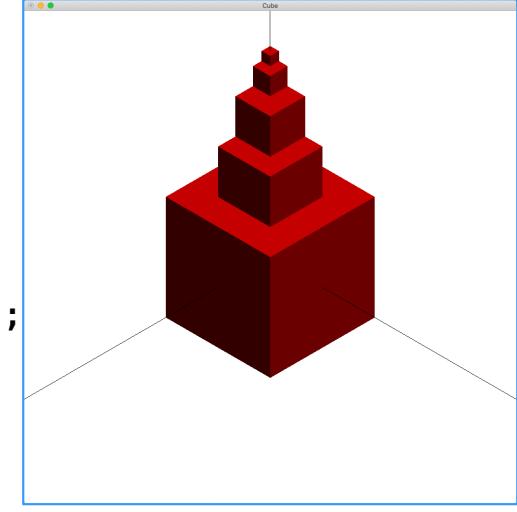
```
oneCube(0, 0, 0, 1);
oneCube(1, 0, 0, 1);
oneCube(2, 0, 0, 1);
oneCube(3, 0, 0, 1);
oneCube(3, 1, 0, 1);
oneCube(3, 2, 0, 1);
oneCube(3, 3, 0, 1);
oneCube(2, 3, 0, 1);
oneCube(1, 3, 0, 1);
oneCube(0, 3, 0, 1);
oneCube(0, 2, 0, 1);
oneCube(0, 1, 0, 1);
```



```
oneCube(0, 0, 0, 1);
oneCube(0, 0, 1, 1);
oneCube(0, 0, 2, 1);
oneCube(0, 0, 3, 1);
oneCube(0, 0, 4, 1);
oneCube(0, 0, 5, 1);
oneCube(0, 0, 6, 1);
oneCube(0, 1, 3, 1);
oneCube(0, 2, 3, 1);
oneCube(0, 3, 3, 1);
```



```
oneCube(0, 0, 0, 3);
oneCube(0, 2, 0, 1.5);
oneCube(0, 3.5, 0, 1);
oneCube(0, 4.5, 0, 0.5);
oneCube(0, 5, 0, 0.25);
```



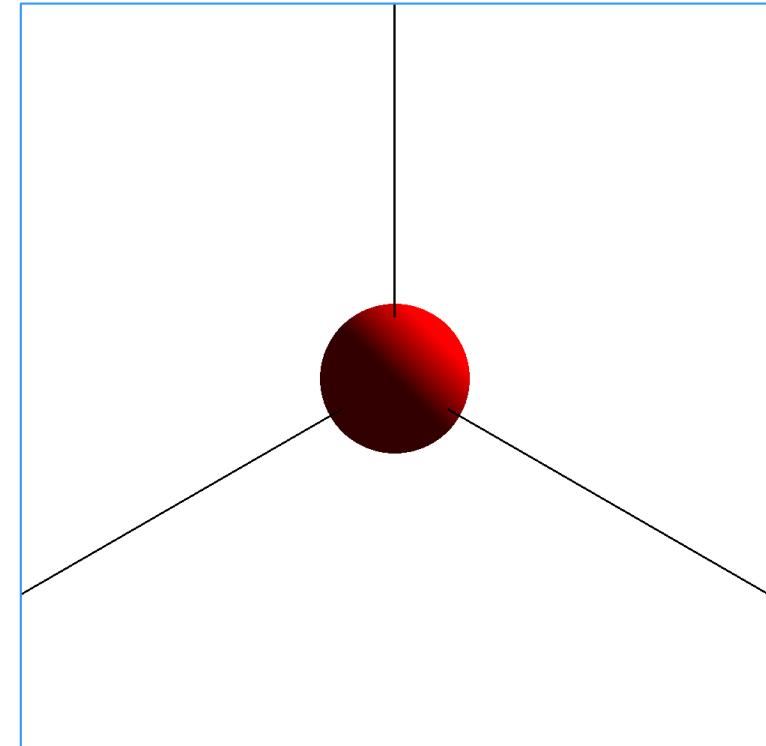
Sphere Pixels

- Define a function to draw a cube with customised position and size:

```
void oneSphere(GLfloat x, GLfloat y, GLfloat z, GLfloat R){  
    glPushMatrix();  
    glTranslated(x, y, z);  
    glutSolidSphere(R, 50, 50);  
    glPopMatrix();  
}
```

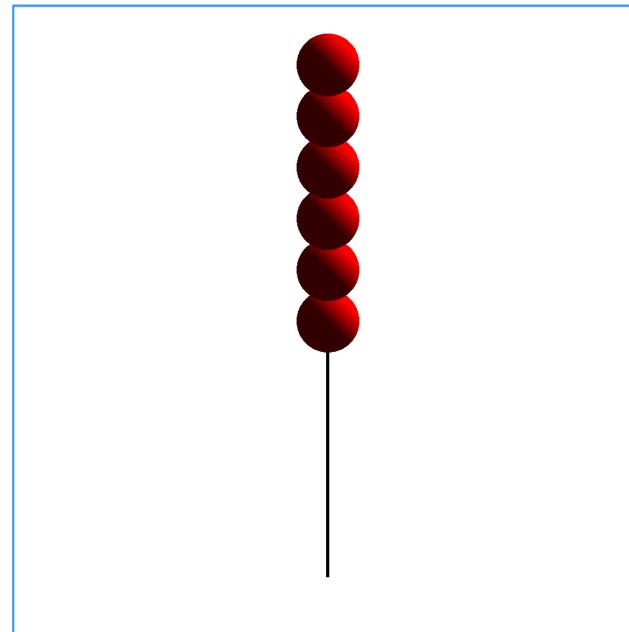
- And use it however you like:

```
oneSphere(0, 0, 0, 1);
```

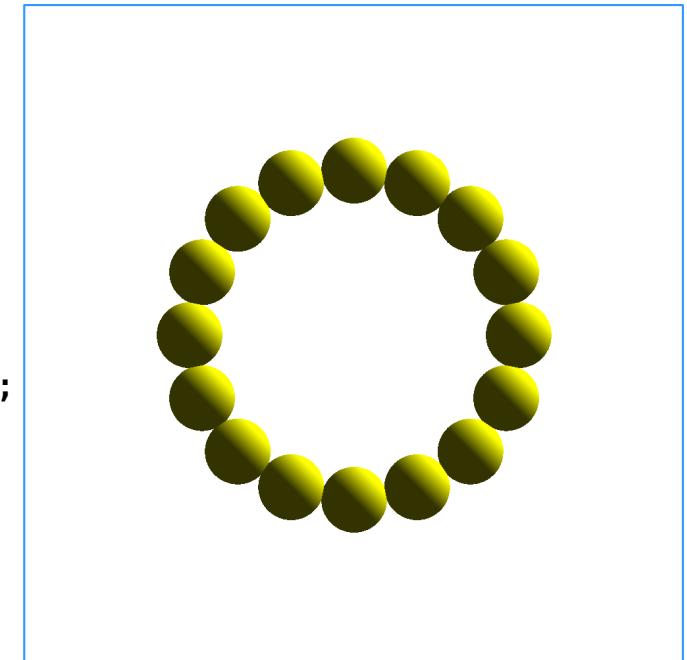


Sphere Pixels

```
oneSphere(0, 0, 0, 0.5);
oneSphere(0, 1, 0, 0.5);
oneSphere(0, 2, 0, 0.5);
oneSphere(0, 3, 0, 0.5);
oneSphere(0, 4, 0, 0.5);
oneSphere(0, 5, 0, 0.5);
glColor3f (0.0, 0.0, 0.0);
glLineWidth(5);
 glBegin(GL_LINES);
 glVertex3i(0, 0, 0);
 glVertex3i(0, -5, 0);
 glEnd();
```

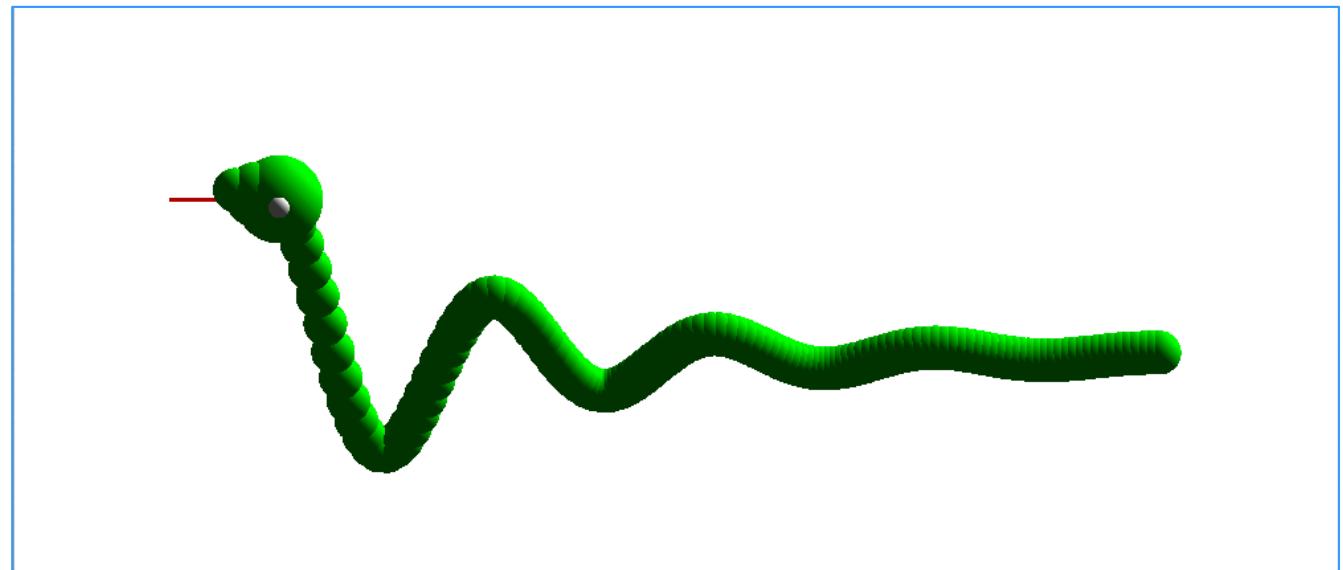


```
float r = 0.5;
int num_segments = 16;
for(int i = 0; i < num_segments; i++) {
    float theta = 2.0f * 3.1415926f * float(i) / float(num_segments);
    float x = r * cosf(theta);
    float y = r * sinf(theta);
    oneSphere(x, y, 0, 0.1); }
```

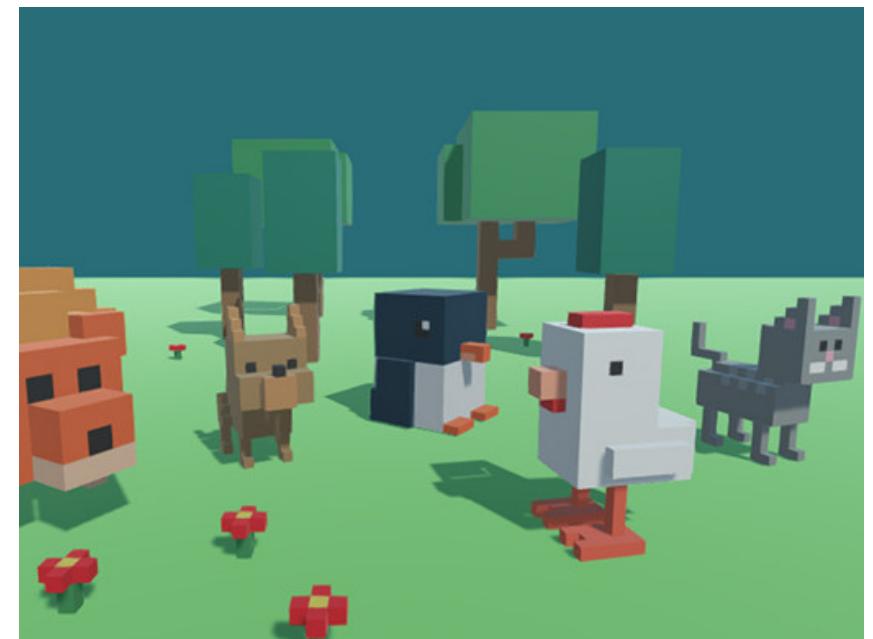
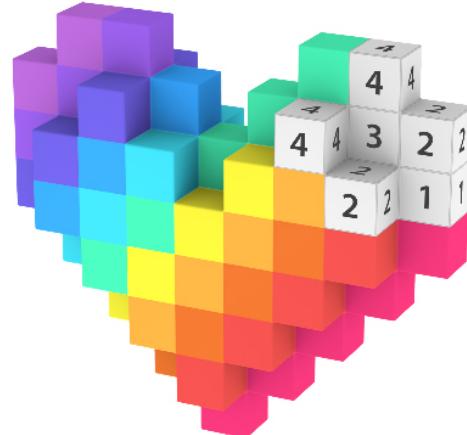
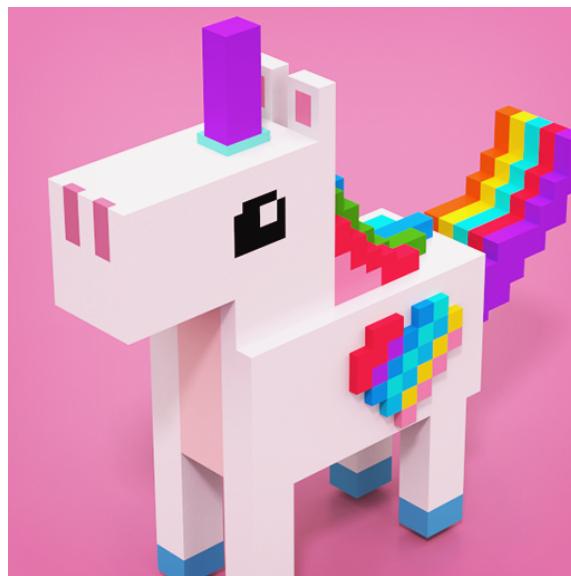
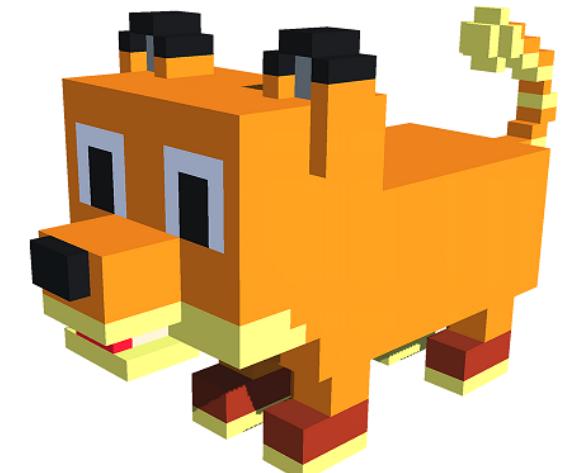
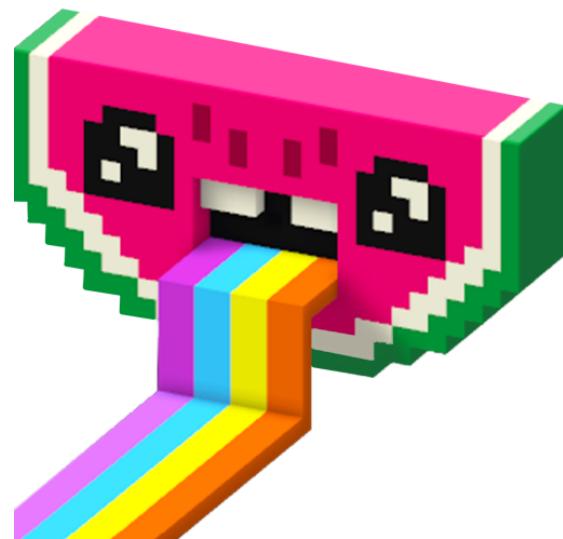
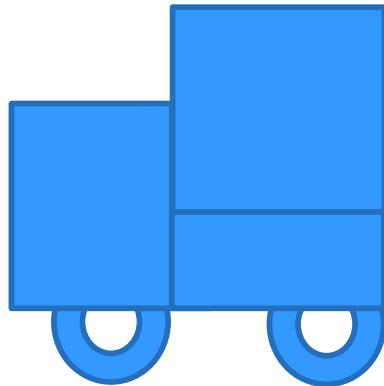


Sphere Pixels

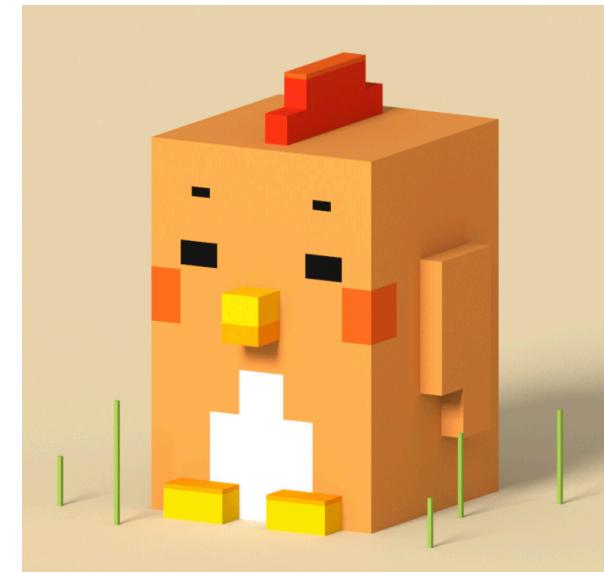
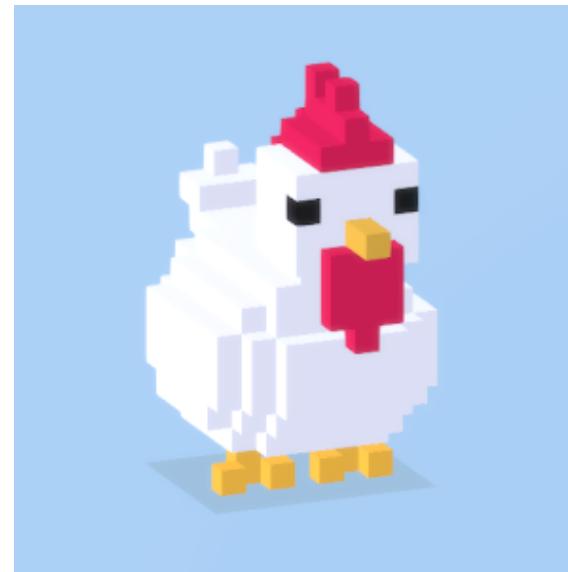
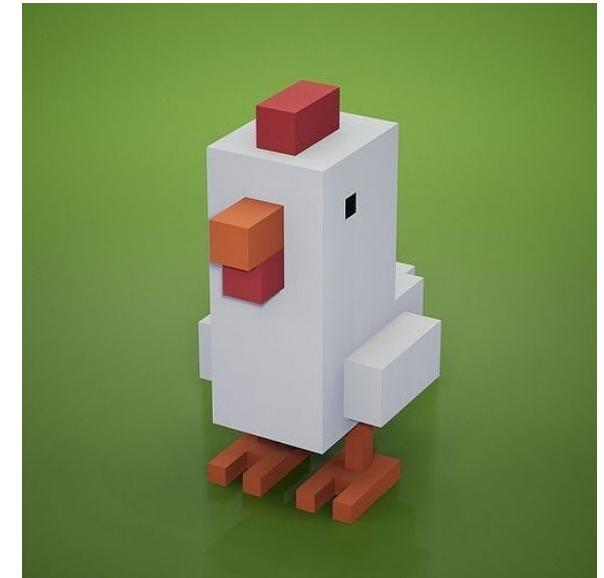
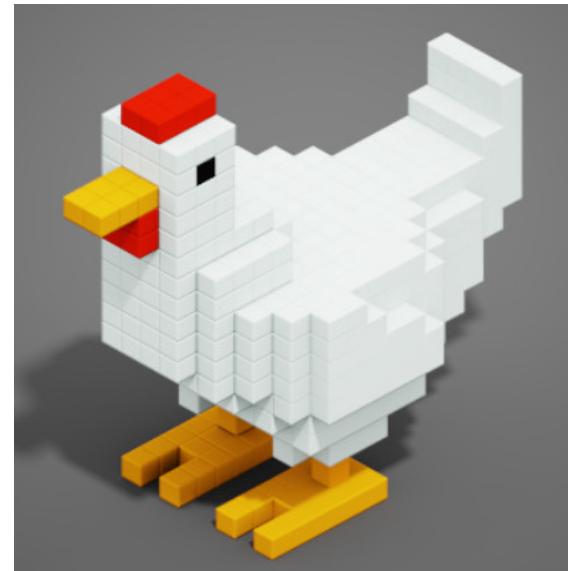
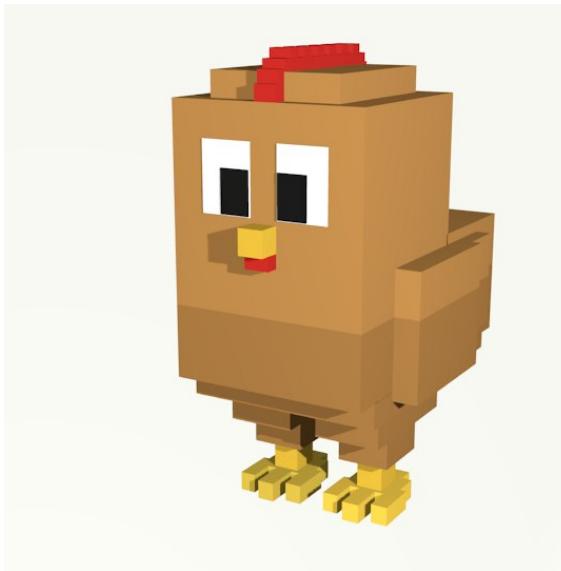
```
glColor3f (0, 1, 0);
GLdouble x;
for(x = 0; x < 4.0 ; x += 0.035) {
    glColor3f (0.0, 1.0, 0.0);
    GLdouble func = exp(-x) * cos(2 * 3.14159265 * x);
    oneSphere(x, func, 0, 0.1);
}
oneSphere(0,1, 0, 0.2);
oneSphere(-0.1, 1.03, 0, 0.15);
oneSphere(-0.2, 1.06, 0, 0.1);
glColor3f (1, 1, 1);
oneSphere(0, 1.1, 0.15, 0.05);
glLineWidth(3);
 glBegin(GL_LINES);
 glColor3f (1, 0, 0);
 glVertex3f(0, 1, 0);
 glVertex3f(-0.5, 1, 0);
 glEnd();
```



What do these images consist of?



Different Ways of Drawing Chickens



Topics

- Pixel Art
- Modular Programming
- Troubleshooting

Modular Programming

- Modular programming: breaking a program up into smaller, manageable functions or modules
- Function: a collection of statements to perform a task
- Motivation for modular programming:
 - Improves maintainability of programs
 - Simplifies the process of writing programs
 - Enable the reusability of function modules

One Bad Example

- All the code are clogged inside the `display()`;

```
void display(){
    glClearColor(0.0, 1.0, 1.0, 0.0);
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f (0.0, 0.0, 0.0);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ();
    gluOrtho2D(0, 0.5, 0, 0.5);
    GLfloat roof[3][2] = {{0.0,0.8}, {0.5,0.2}, {-0.5,0.2}};
    GLfloat wall[4][2] = {{0.5,0.2}, {0.5,-0.7}, {-0.5,-0.7}, {-0.5,0.2}};
    GLfloat chimney[4][2] = {{0.2, 0.7}, {0.4, 0.7}, {0.4, 0.2}, {0.2, 0.2}};
    GLfloat door[4][2] = {{0.1, 0.0}, {0.4, 0.0}, {0.4, -0.7}, {0.1, -0.7}};
    //Draw chimney
    glBegin(GL_POLYGON);
        glVertex2fv(chimney[0]);
        glVertex2fv(chimney[1]);
        glVertex2fv(chimney[2]);
        glVertex2fv(chimney[3]);
    glEnd();
    //Draw roof
    glColor3f (1.0, 0.0, 0.0);
    glBegin(GL_TRIANGLES);
        glVertex2fv(roof[0]);
        glVertex2fv(roof[1]);
        glVertex2fv(roof[2]);
    glEnd();
    //Draw door
    glLineWidth (2);
    glColor3f (0.0, 0.0, 0.0);
    glBegin(GL_LINE_LOOP);
        glVertex2fv(door[0]);
        glVertex2fv(door[1]);
        glVertex2fv(door[2]);
        glVertex2fv(door[3]);
    glEnd();
    //Draw Fence
    glPointSize(10);
    glBegin(GL_POINTS);
        glVertex2f(0.15, -0.35);
    glEnd();
    glBegin(GL_LINES);
    for(int i=0; i<3; i++){
        glVertex2f(-0.4+i*0.15, 0.0);
        glVertex2f(-0.4+i*0.15, -0.3);
        glVertex2f(-0.4, 0-i*0.15);
        glVertex2f(-0.1, 0-i*0.15);
    }
    glEnd();
}
```

Full Template – Part 1

```
#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif
#include <iostream>
using namespace std;

GLdouble W = 600.0;
GLdouble H = 500.0;

void idle() {
    glutPostRedisplay();
}

void drawX() {
    glColor3f (1.0, 0.0, 0.0);
    glutSolidTeapot(0.5);
}
```

```
void display(){
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    //glMatrixMode(GL_MODELVIEW);
    //glLoadIdentity();
    //gluLookAt(1.0, 1.0, 1.0 , 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);
    drawX();
    //glutSwapBuffers(); //Animation
    glFlush();          //Still image
}

void init(void) {
    glClearColor(1, 1, 1, 1.0);
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    GLfloat light_position[] = { 1.0, 1.0, 0.0, 0.0 };
    glLightfv(GL_LIGHT0, GL_POSITION, light_position);
    glEnable(GL_COLOR_MATERIAL);
    glShadeModel(GL_SMOOTH);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-1, 1, -1, 1, -1, 100.0);
}
```

Full Template – Part 2

```
//void mykey(unsigned char key, int x, int y) {}
//void mySpecialKey(int key, int x, int y) {}
//void mymouse(int button,int state,int x,int y) {}
//void myMouseMotion(int x, int y){}

void reshape(GLsizei width, GLsizei height) {
    if (height == 0) height = 1;
    GLfloat aspect = (GLfloat)width / (GLfloat)height;
    glLoadIdentity();
    if (width >= height) {
        gluOrtho2D(-1.0 * aspect, 1.0 * aspect, -1.0, 1.0);
    } else {
        gluOrtho2D(-1.0, 1.0, -1.0 / aspect, 1.0 / aspect);}
}

int main(int argc, char** argv){
    glutInit(&argc, argv);
    //glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);      //Animation
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);      //Still image
    glutInitWindowSize(W, H);
    glutCreateWindow("X");
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    //glutIdleFunc(idle);
    //glutKeyboardFunc(mykey);
    //glutSpecialFunc(mySpecialKey);
    //glutMouseFunc(mymouse);
    //glutMotionFunc(myMouseMotion);
    init();
    glEnable(GL_DEPTH_TEST);
    glutMainLoop();
}
```

Use Global Data

Declare global variables to define the width and height of the display window → Futureproof.

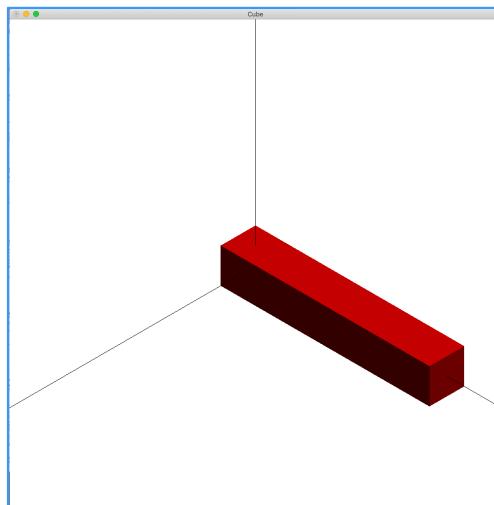
```
#include <GLUT/glut.h>
GLdouble W = 600.0;
GLdouble H = 500.0;

int main(int argc, char** argv){
    glutInit(&argc, argv);
    glutInitWindowSize(W, H);
    glutCreateWindow("...");
    glutDisplayFunc(mydisplay);
    myInit();
    glutMainLoop();}
```

Use Variables

If a particular value appears more than once in the code, declare variable instead of giving actual values.

Benefit: The values can be easily passed on between functions.



Bad practice:

```
oneCube(0, 0, 0, 1);
oneCube(1, 0, 0, 1);
oneCube(2, 0, 0, 1);
oneCube(3, 0, 0, 1);
oneCube(4, 0, 0, 1);
oneCube(5, 0, 0, 1);
```

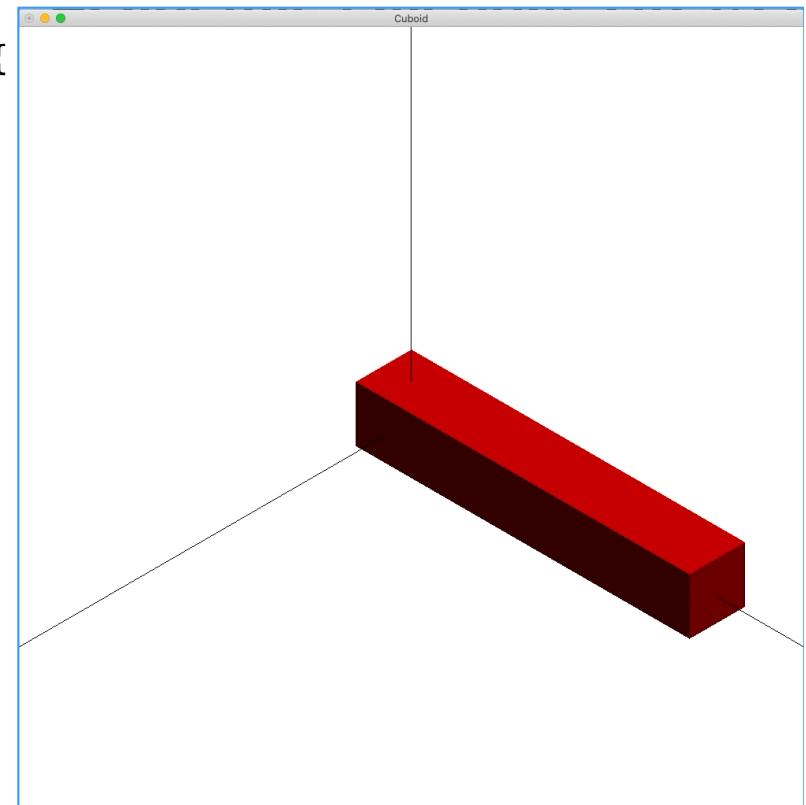
Good practice:

```
GLfloat x=0, y=0, z=0, l=1;
oneCube( x, y, z, l );
oneCube( x+1, y, z, l );
oneCube( x+2, y, z, l );
oneCube( x+3, y, z, l );
oneCube( x+4, y, z, l );
oneCube( x+5, y, z, l );
```

Case Study

```
GLfloat a=0, b=0, c=0, size=1;  
GLfloat dx=0, dy=0, dz=0;  
  
void idle() {  
    dy += 0.05;  
    glutPostRedisplay();  
}  
void oneCube(GLfloat x, GLfloat y, GLfloat z, GLfloat L){  
    glPushMatrix();  
    glTranslated(x, y, z);  
    glutSolidCube(L);  
    glPopMatrix();}  
void drawCuboid(GLfloat x, GLfloat y, GLfloat z, GLfloat L){  
    oneCube( x, y, z, L);  
    oneCube(x+1, y, z, L);  
    oneCube(x+2, y, z, L);  
    oneCube(x+3, y, z, L);  
    oneCube(x+4, y, z, L);  
    oneCube(x+5, y, z, L);  
}  
void display(){  
    ...//Other stuffs  
    glPushMatrix();  
    glTranslated(dx, dy, dz);  
    drawCuboid(a, b, c, size);  
    glPopMatrix();  
    glutSwapBuffers();  
}
```

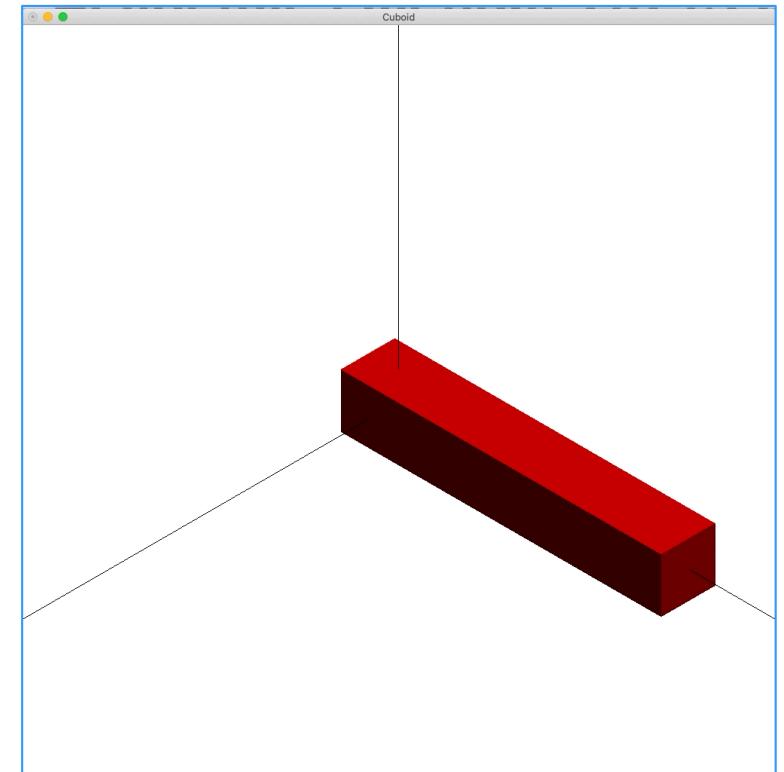
Effect 1:



Case Study

```
GLfloat a=0, b=0, c=0, size=1;  
GLfloat dx=0, dy=0, dz=0;  
  
void idle() {  
    dy += 0.05;  
    size += 0.007;  
    a -= 0.02;  
    c += 0.02;  
    glutPostRedisplay();  
}  
void oneCube(GLfloat x, GLfloat y, GLfloat z, GLfloat L){  
    glPushMatrix();  
    glTranslated(x, y, z);  
    glutSolidCube(L);  
    glPopMatrix();}  
void drawCuboid(GLfloat x, GLfloat y, GLfloat z, GLfloat L){  
    oneCube( x, y, z, L);  
    oneCube(x+1, y, z, L);  
    oneCube(x+2, y, z, L);  
    oneCube(x+3, y, z, L);  
    oneCube(x+4, y, z, L);  
    oneCube(x+5, y, z, L);  
}  
void display(){  
    ...//Other stuffs  
    glPushMatrix();  
    glTranslated(dx, dy, dz);  
    drawCuboid(a, b, c, size);  
    glPopMatrix();  
    glutSwapBuffers();  
}
```

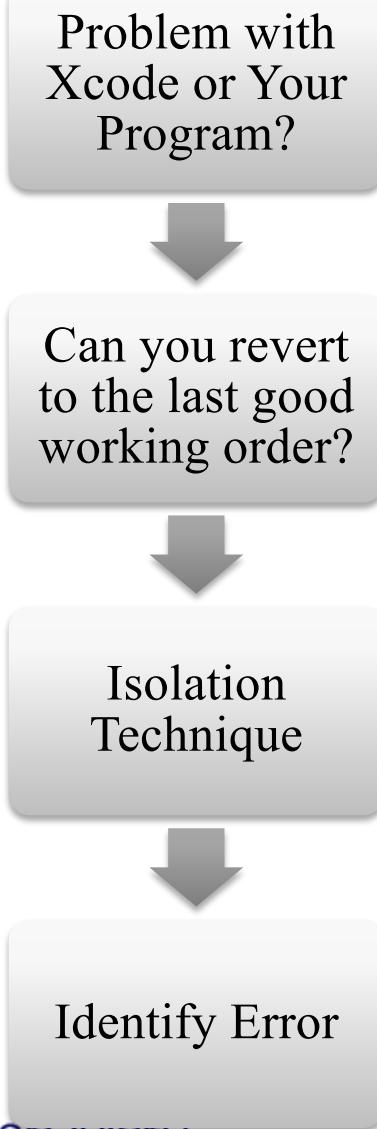
Effect 2:



Topics

- Pixel Art
- Modular Programming
- Troubleshooting

Troubleshooting Strategy



Run a "Hello World" or a demo project to find out. Eliminate environment/compatibility problems.

Version control
(Next section)

(Next slide)

Read error message and try to understand it.
(Next few slides)

Isolation Technique

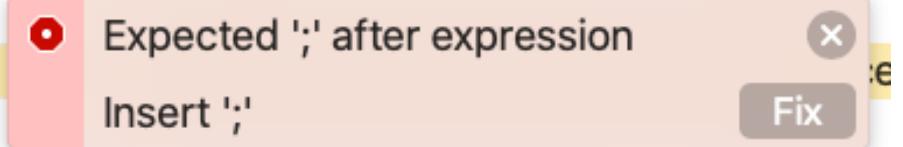
```
//  
//    for (anotherView) in superview!.subviews {  
//        if (self.frame.intersects(anotherView.frame) && (self != anotherView)) {  
//            self.image = UIImage(named:"explosion.png")  
//        }  
//    }  
//  
//    for (anotherView) in superview!.subviews {  
//        if (self.frame.intersects(anotherView.frame) && (self != anotherView)) {  
//            let stringCollision = Mirror(reflecting: anotherView)  
//            self.image = UIImage(named:"explosion.png")  
//            //print("Collision: \(stringCollision.subjectType)")  
//        }  
//    }  
  
}  
  
override func touchesEnded(_ touches: Set<UITouch>, with event: UIEvent?) {  
    //self.image = UIImage(named: "flower.png")  
}
```

- Comment out blocks of code, and only run the code in question.
- If you don't know where the problem is, then divide the program into multiple blocks and run one block at a time, until you identify the block that gives an error.

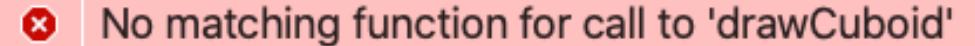
Understand the Error Messages

- Error indication in Xcode:

```
glEnable(GL_LIGHTING);
glEnable(GL_LIGHT0);
GLfloat light_position[] = { 1.0, 1.0, 0.0, 0.0 };
glLightfv(GL_LIGHT0, GL_POSITION, light_position)
glEnable(GL_COLOR_MATERIAL);
glShadeModel(GL_SMOOTH);
glMatrixMode(GL_PROJECTION);
```



```
glPushMatrix();
glTranslated(dx, dy, dz);
drawCuboid(a, b, c);
glPopMatrix();
```



Seeking Answers – Google

- Type the error message and wrap it with a quote
- E.g. “value of optional string! not unwrapped”



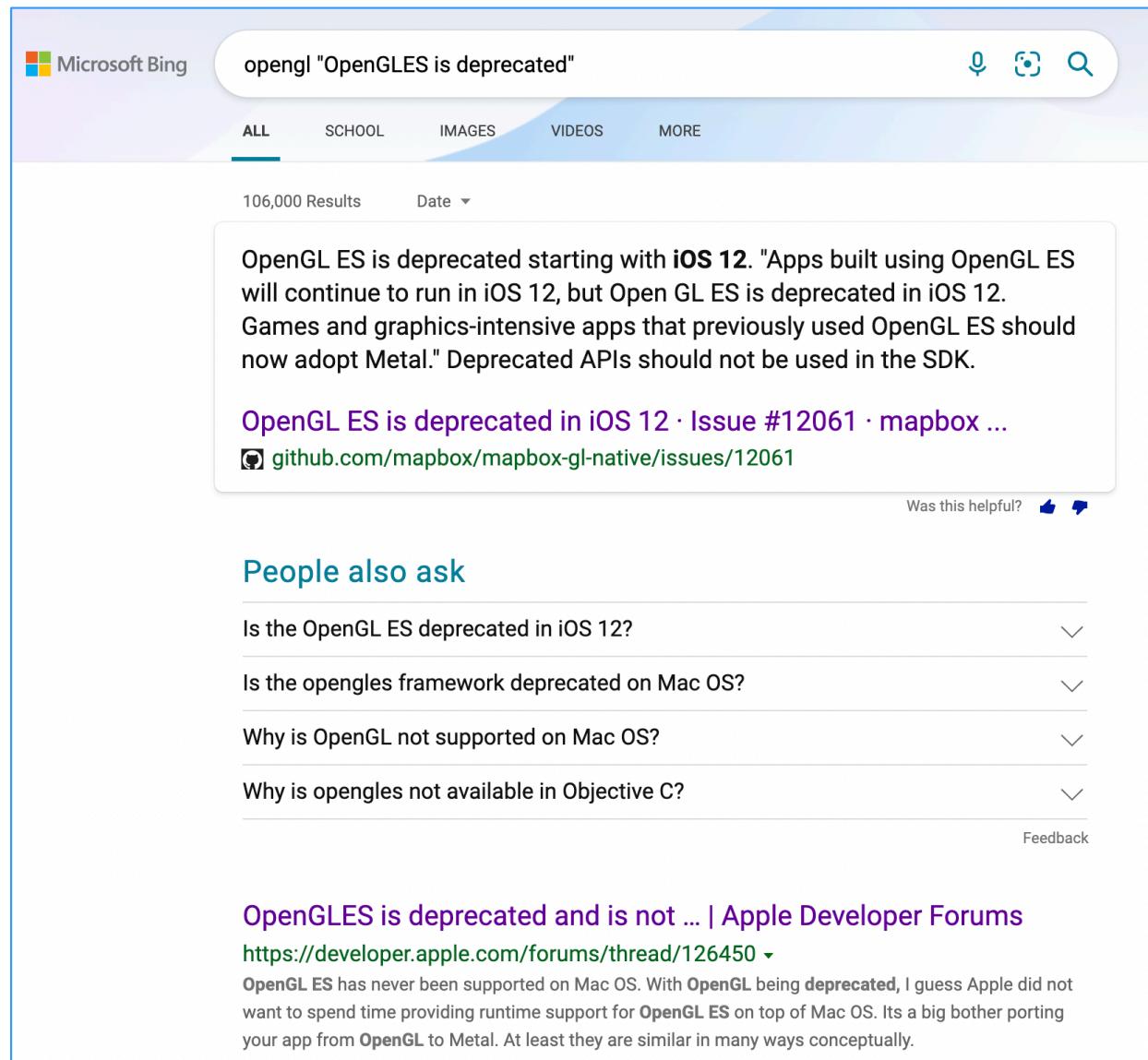
Using quotes will match the exact search term, therefore increasing your chance of finding the right answer.

The screenshot shows a Google search results page with the following details:

- Search Query:** "value of optional string! not unwrapped"
- Results:** 5 results (0.50 seconds)
- First Result:** [Learning Swift, Value of optional type not unwrapped when ...](https://stackoverflow.com/.../learning-swift-value-of-optional-type-not-unwrap...)
stackoverflow.com/.../learning-swift-value-of-optional-type-not-unwrap... ▾
10 Jan 2016 - and also value of optional 'String?' not unwrapped; did you mean to use '!' or '?' I'm little confused, because in the lesson example they have ...
- Second Result:** [Value of optional type String? not unwrapped](https://techspider.techjaffa.info/value-of-optional-type.../MjU3OTk1Mjk=...)
techspider.techjaffa.info/value-of-optional-type.../MjU3OTk1Mjk=... ▾
... -not-unwrapped-value,optional-type-string-not-unwrapped-value-of,type-string-not-unwrapped-value-of-optional,string-not-unwrapped-value-of-optional-type ...
- Third Result:** [Value of optional type String? not unwrapped with text.toInt](https://techspider.techjaffa.info/value-of-optional-type.../MzI2NzM2MzQ=...)
techspider.techjaffa.info/value-of-optional-type.../MzI2NzM2MzQ=... ▾
... -not-unwrapped-with-texttoint-value-of,type-string-not-unwrapped-with-texttoint-value-of-optional,string-not-unwrapped-with-texttoint-value-of-optional-type ...
- Fourth Result:** [Project 8 Discussion : hackingwithswift - Reddit](https://www.reddit.com/r/hackingwithswift/.../project_8_discussion/)
https://www.reddit.com/r/hackingwithswift/.../project_8_discussion/ ▾
11 Apr 2015 - But then I get an error: "Value of optional 'String?' not unwrapped, did you mean to use '!' or '??'" So I tried changing it to solutions.
- Fifth Result:** [Tutorial CLGeocoder in Swift. Come cercare nella mappa iOS](https://www.xcoding.it/tutorial-clgeocoder-swift-cerca...)
https://www.xcoding.it/tutorial-clgeocoder-swift-cerca... ▾ Translate this page
23 Nov 2014 - ... {{placemarks, error}->Void in if error == nil { mi ritorna value of optional 'String?' not unwrapped; anche scaricando il progetto completo ...

Alternative English Search Engine

- www.bing.com



The screenshot shows a Microsoft Bing search results page. The search query is "opengl \"OpenGLES is deprecated\"". The results are filtered under the "ALL" tab, showing 106,000 results. A snippet of the top result discusses OpenGL ES deprecation in iOS 12, mentioning Metal and Deprecated APIs. Below the snippet is a link to an issue on GitHub: "OpenGL ES is deprecated in iOS 12 · Issue #12061 · mapbox ...". The GitHub URL is github.com/mapbox/mapbox-gl-native/issues/12061. At the bottom right of the search results, there is a "Was this helpful?" section with thumbs up and thumbs down icons.

People also ask

- Is the OpenGL ES deprecated in iOS 12?
- Is the opengles framework deprecated on Mac OS?
- Why is OpenGL not supported on Mac OS?
- Why is opengles not available in Objective C?

Feedback

OpenGL ES is deprecated and is not ... | Apple Developer Forums
<https://developer.apple.com/forums/thread/126450>

OpenGL ES has never been supported on Mac OS. With OpenGL being **deprecated**, I guess Apple did not want to spend time providing runtime support for OpenGL ES on top of Mac OS. Its a big bother porting your app from OpenGL to Metal. At least they are similar in many ways conceptually.

Debugging

- Same as other programming language, printing a message in the console is always a useful way to find out how things are going at a certain stage.
- Required header and namespace declaration:

```
#include <iostream>
using namespace std;
```

- Command to print a message in C++:

```
cout << "x=" << x << "y=" << y << "z=" << z << endl;
```

```
3  AppKit
4  AppKit
5  AppKit
6  AppKit
7  AppKit
8  GLUT
9  GLUT
10 MyFirst
11 libdyld.dylib
12 ???

}
x=-2.07152 y=-0.131284 z=-3.45745
0x00007fff351c86e2 -[NSCarbonMenuI
0x00007fff351c80b2 -[NSCarbonMenuI
0x00007fff351c10d5 -[NSApplication
0x00007fff354aad91 _NSApplicationB
0x00007fff35662595 -[NSApplication
0x00007fff3a99b523 -[GLUTApplicati
0x00007fff3a9a77e7 glutMainLoop + 1
0x00000001000030af main + 111
0x00007fff7201fcc9 start + 1
0x0000000000000001 0x0 + 1
```

Summary of Reference Documents

- **OpenGL data types:**
https://www.khronos.org/opengl/wiki/OpenGL_Type
- **GL and GLU Functions:**
<https://www.khronos.org/registry/OpenGL-Refpages/gl2.1/>
- **GLUT Functions:**
QMPlus: Background Reading – GLUT Documentation or
<https://www.opengl.org/resources/libraries/glut/spec3/spec3.html>
- **WebGL:**
 - <https://www.khronos.org/registry/webgl/specs/latest/2.0/>
 - https://www.khronos.org/files/webgl/webgl-reference-card-1_0.pdf

Importance of Comments in Programming

- Writing comments on programs is an essential **ethic** for software engineers.
- Two main reasons:
 - **For yourself:** Appropriate use of commenting can enhance the readability of programs (especially long ones), and hence makes code maintenance and troubleshooting much easier.
 - **For others:** In an organization, there are typically many programmers who work on the same project as a team. Well-commented functions/logics are helpful to other programmers to better understand code (e.g. imagine what will happen if a programmer leaves the team).
- Three key objectives for code comments:
 - Give out key information about the code
 - Help with debugging
 - Enhance readability

Commenting Techniques 1 – Key Info

- There is no need to give meaningless comments.
 - Bad example: *//This is a function*
- In the example below, every line of code is **useful**.

```
for( student = 1; student <= numStudents; student++) //Outer loop for students
{
    for(day = 1; day <= numDays; day++) //Inner loop for days
    {
        cin >> numHours; //Input the number of hours
        total = total + numHours; //Calculate the total
        if (total > 10) //Check if boundary is exceeded
        {
            cout << "Number exceeded"; //Give warning
            total = 0; //Reset hours
        } //End if
    } //End inner for
} // End outer for
return 0;
} // End main
```

Helpful for debugging

Key
information
for the code

Commenting Techniques 2 – Readability

- Enhance readability of the code by taking advantage of various symbols to divide a long code into more visible blocks.

```
1 // Implementation file for the Pharmacy class
2 #include "Pharmacy.h"
3 #include <iostream>
4 using namespace std;
5
6 //*****
7 // Default constructor
8 //*****
9
10 Pharmacy::Pharmacy()
11 {
12 }
13
14 //*****
15 // Default destructor
16 //*****
17
18 Pharmacy::~Pharmacy()
19 {
20 }
21
22
```

```
22 ****
23 // Accessor for the price
24 ****
25 ****
26
27 int Pharmacy::getPrice()
28 {
29     return price[menuPrint()];
30 }
31
32
33 ****
34 // Function for the user interface
35 ****
36
37 int Pharmacy::menuPrint()
38 {
39     int choice;
40
41     // Display the medication menu
42 }
```

Commenting Technique 3 – Header

- Add info about portfolio of the program at the beginning.

```
///////////////////////////////
// Solution Example for 4COM1037 Coursework //
//This is NOT the only solution for the system. //
//Please refer to the marking scheme for the assessment criteria. //
//-----
//This system stores supermarket stock information, //
//allows user to check out items by entering barcodes, //
//and deals with cash transactions. //
//Developed by Barry Ip //
//Date: 15/11/2013 //
//Last Update: 16/09/2019 //
//1 mark will be deducted if the program is written without a heading. //
/////////////////////////////
#include <iostream>
#include <string>
#include <fstream>
#include <iomanip>
using namespace std;

//=====
//      Global Variables
//=====

const int NUM = 10;
```

Essential information include:

- Purpose of the program
- Author name
- Date of creation
- Date of last modification

Questions?

x.che@qmul.ac.uk