OpenGL Tutorial

Computer Graphics

CMU 15-462/15-662, Fall 2016

What is OpenGL?

- Cross-language, cross-platform application programming interface (API) for rendering 2D/3D graphics
- Originally released by Silicon Graphics Inc. (SGI) in 1992
- Now managed by non-profit technology consortium Khronos Group
- Closely outlines what GPUs are meant to do

Source: Wikipedia

Things You Can Do with OpenGL

Source: UNiGiNE

Things You Can Do with OpenGL

Source: http://www.heyuguys.com/astonishing-game-of-thrones-recreated-in-minecraft/

Disclaimer

- This tutorial does NOT cover the “modern” OpenGL (version 3.x and higher, latest is 4.5) which uses lower level API’s to give you more flexibility.

- Instead, we focus on the “older” OpenGL (version 2.1) to get your feet wet with high-level API’s.

Drawing Primitive Shapes

Drawing a Triangle

Starts the draw triangles state

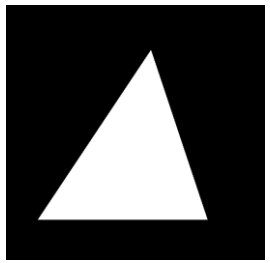
- glBegin(GL_TRIANGLES);
- glVertex2f(-2, -1);
- glVertex2f(1, -1);
- glVertex2f(0, 2);

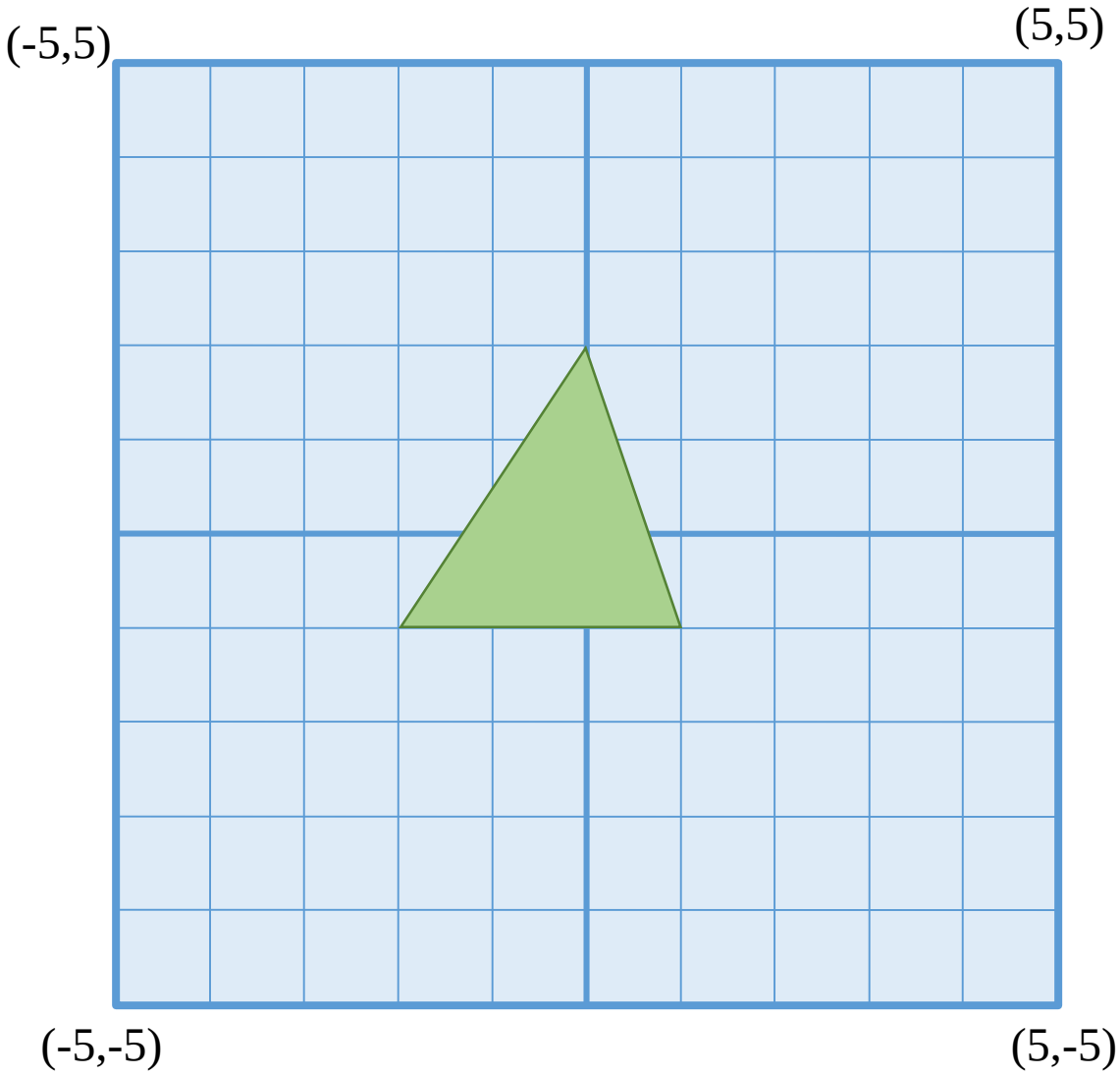
Vertices

- glEnd();

Ends the draw triangles state

- *Default color is actually white





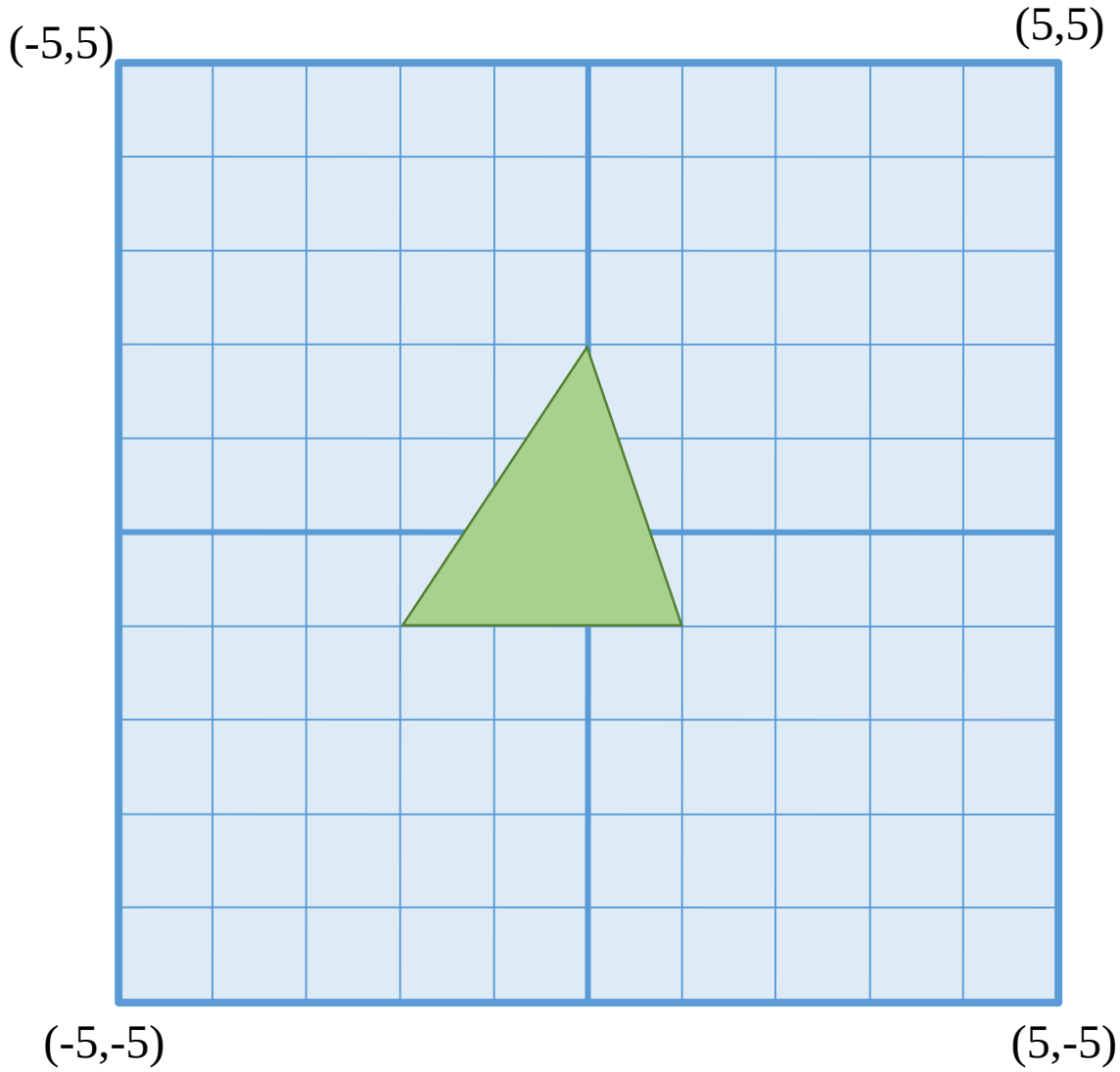
OpenGL API Convention

OpenGL

- glBegin(GL_TRIANGLES);
- glVertex2f(-2, -1);
- glVertex2f(1, -1);
- glVertex2f(0, 2);
- glEnd();

2D

float

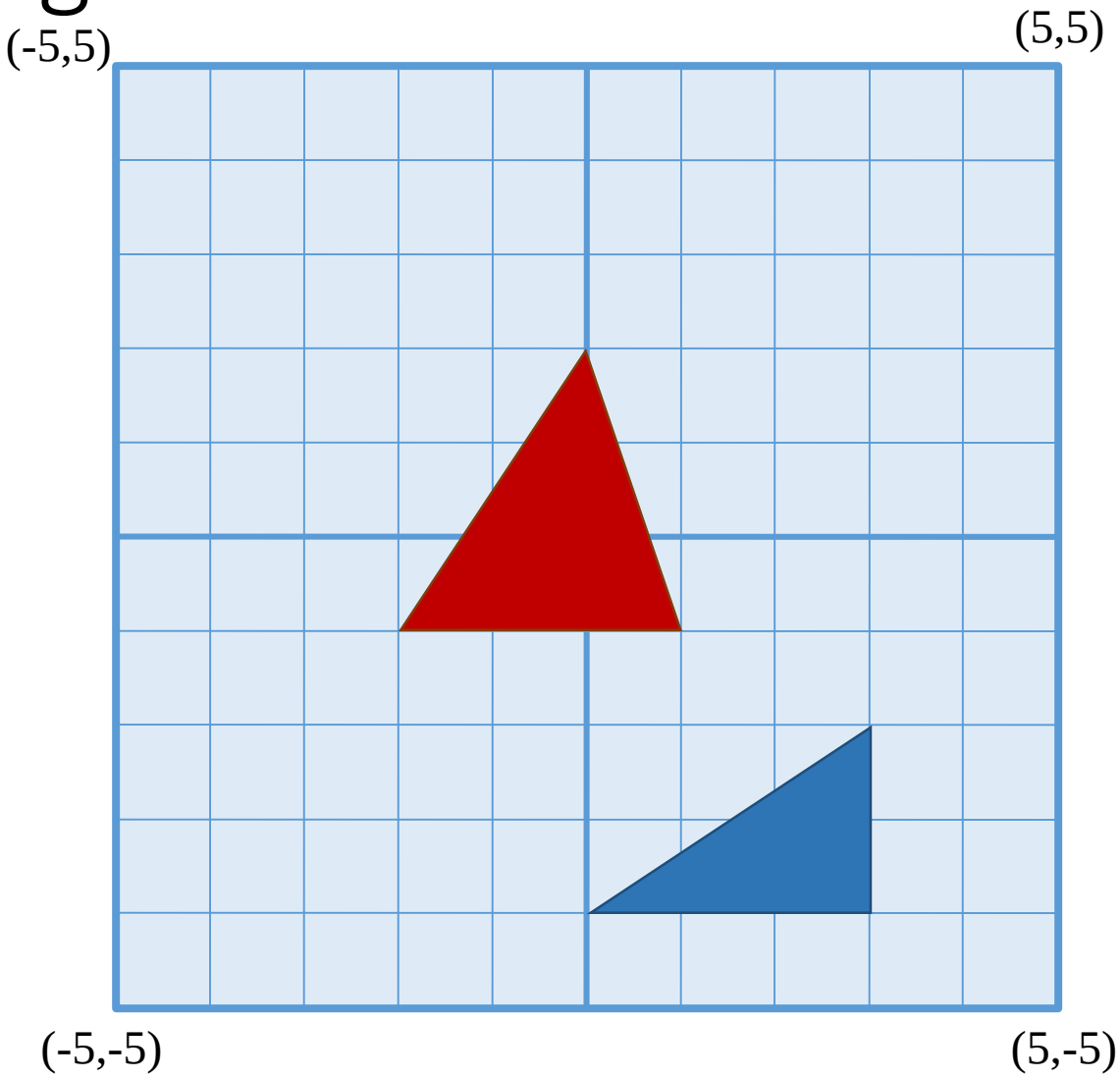


Drawing Multiple Triangles

- glBegin(GL_TRIANGLES);
- glVertex2f(-2, -1);
- glVertex2f(1, -1);
- glVertex2f(0, 2);

- glVertex2f(0, -4);
- glVertex2f(3, -4);
- glVertex2f(3, -2);
- glEnd();

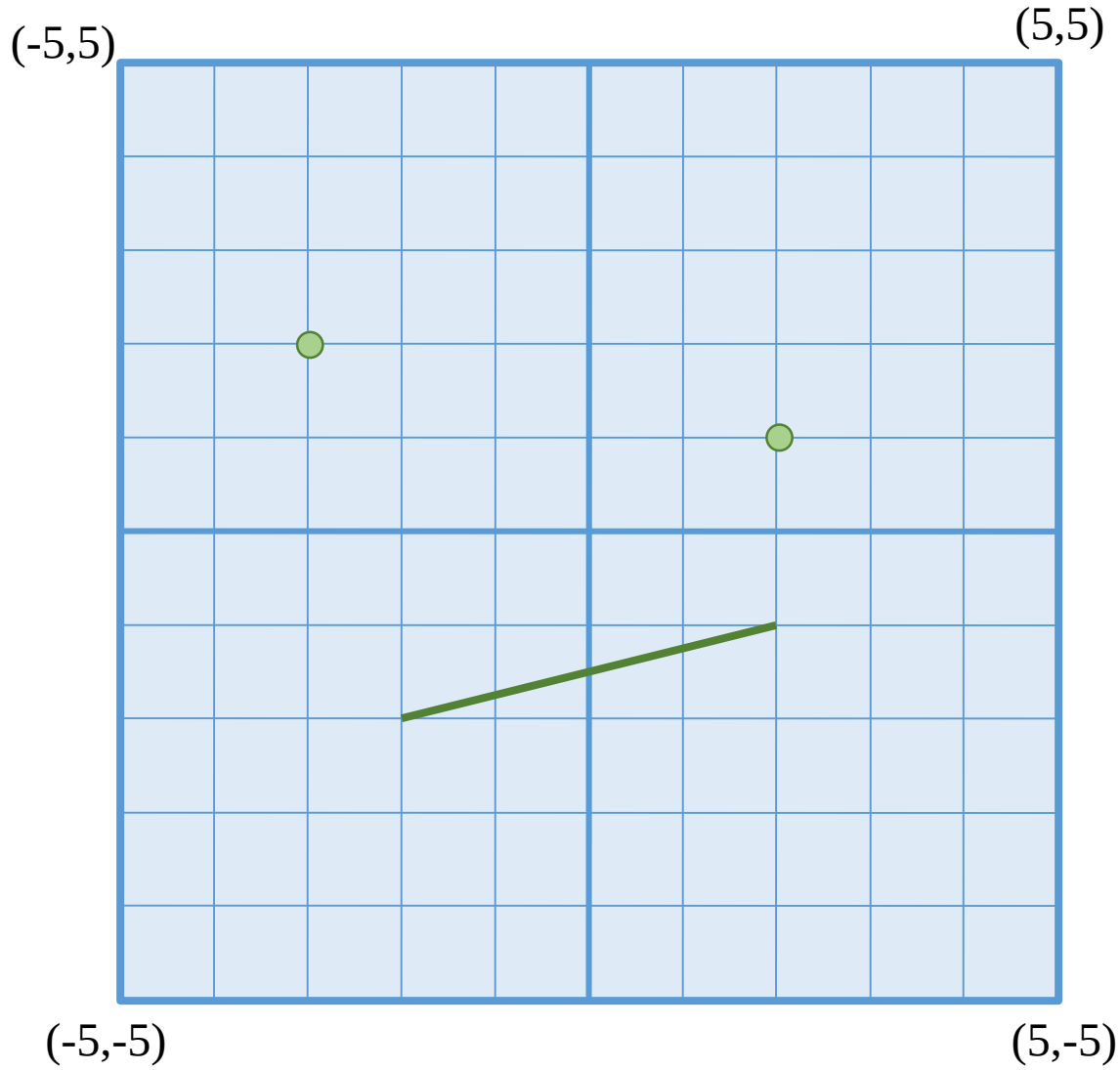
- What happens if number of vertices are not 3n?



Drawing Other Shapes

- glBegin(GL_POINTS);
- glVertex2f(-3, 2);
- glVertex2f(2, 1);
- glEnd();

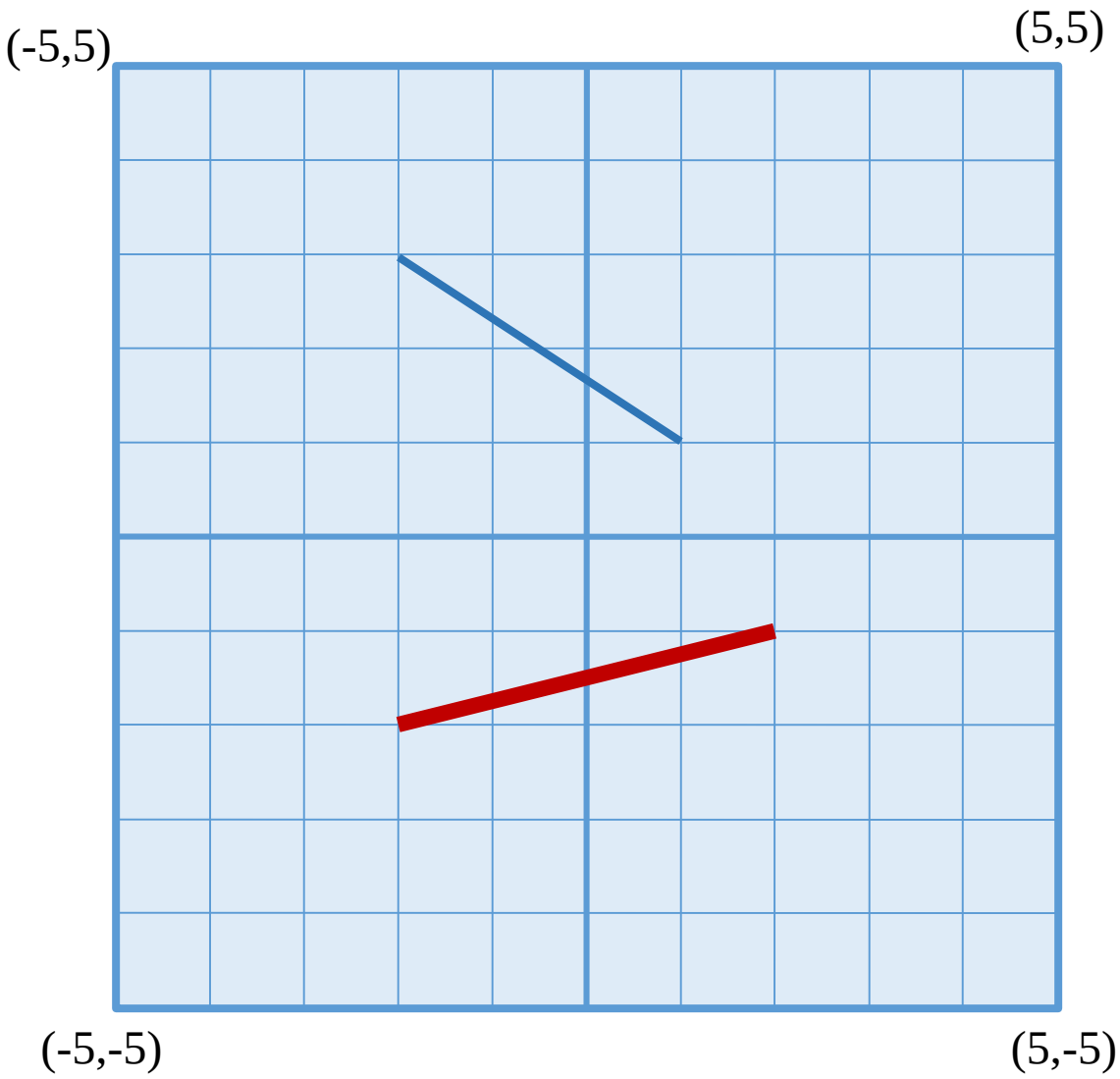
- glBegin(GL_LINES);
- glVertex2f(-2, -2);
- glVertex2f(2, -1);
- glEnd();



Some Things Cannot Be Done Inside glBegin/glEnd

- glLineWidth(2.0);
- glBegin(GL_LINES);
- glVertex2f(-2, -2);
- glVertex2f(2, -1);
- glEnd();

- glLineWidth(1.0);
- glBegin(GL_LINES);
- glVertex2f(-2, 3);
- glVertex2f(1, 1);
- glEnd();



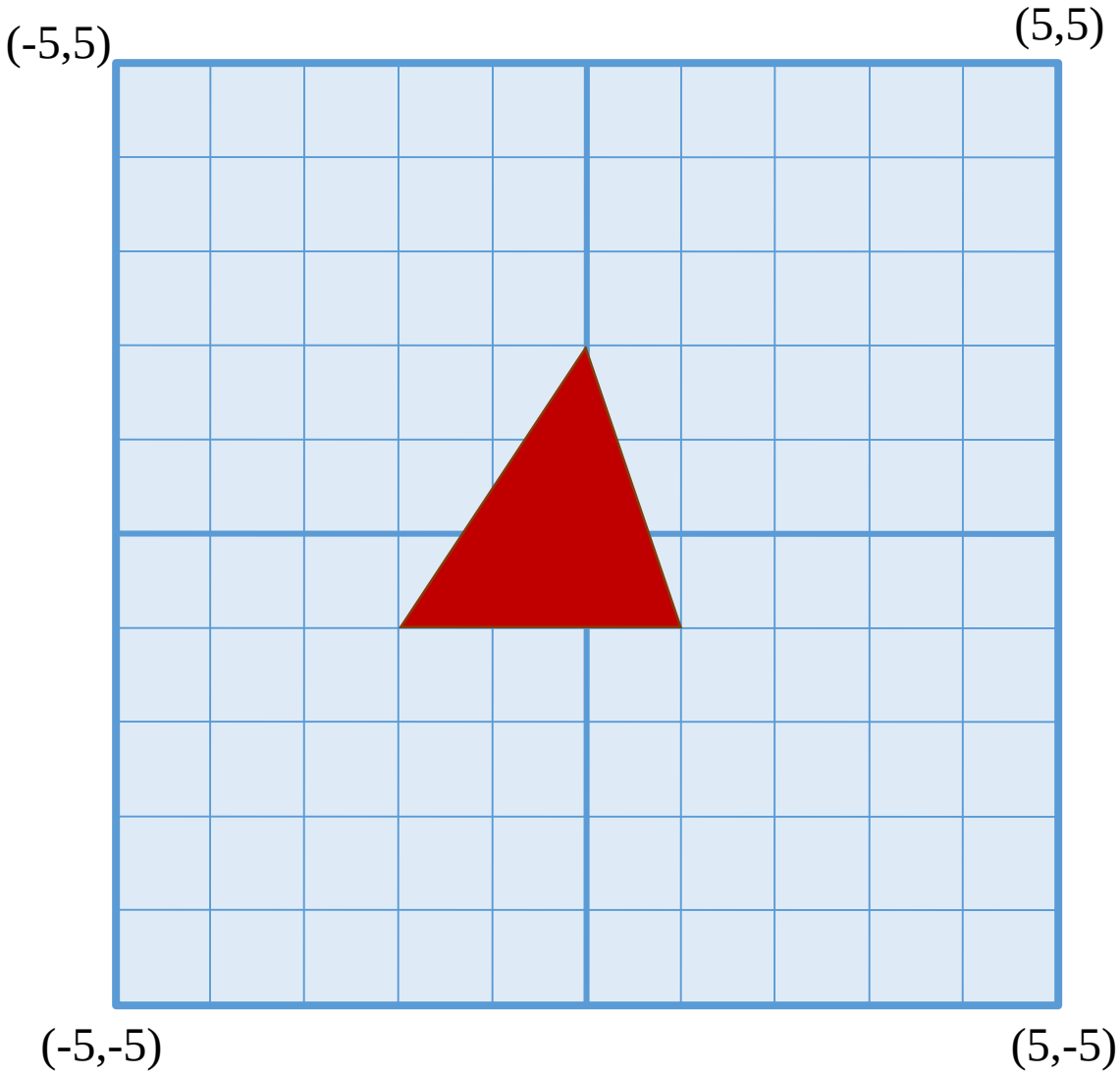
Color in OpenGL

Setting Color

Takes in RGB

- glColor3f(1, 0, 0);
- glBegin(GL_TRIANGLES);
- glVertex2f(-2, -1);
- glVertex2f(1, -1);
- glVertex2f(0, 2);
- glEnd();

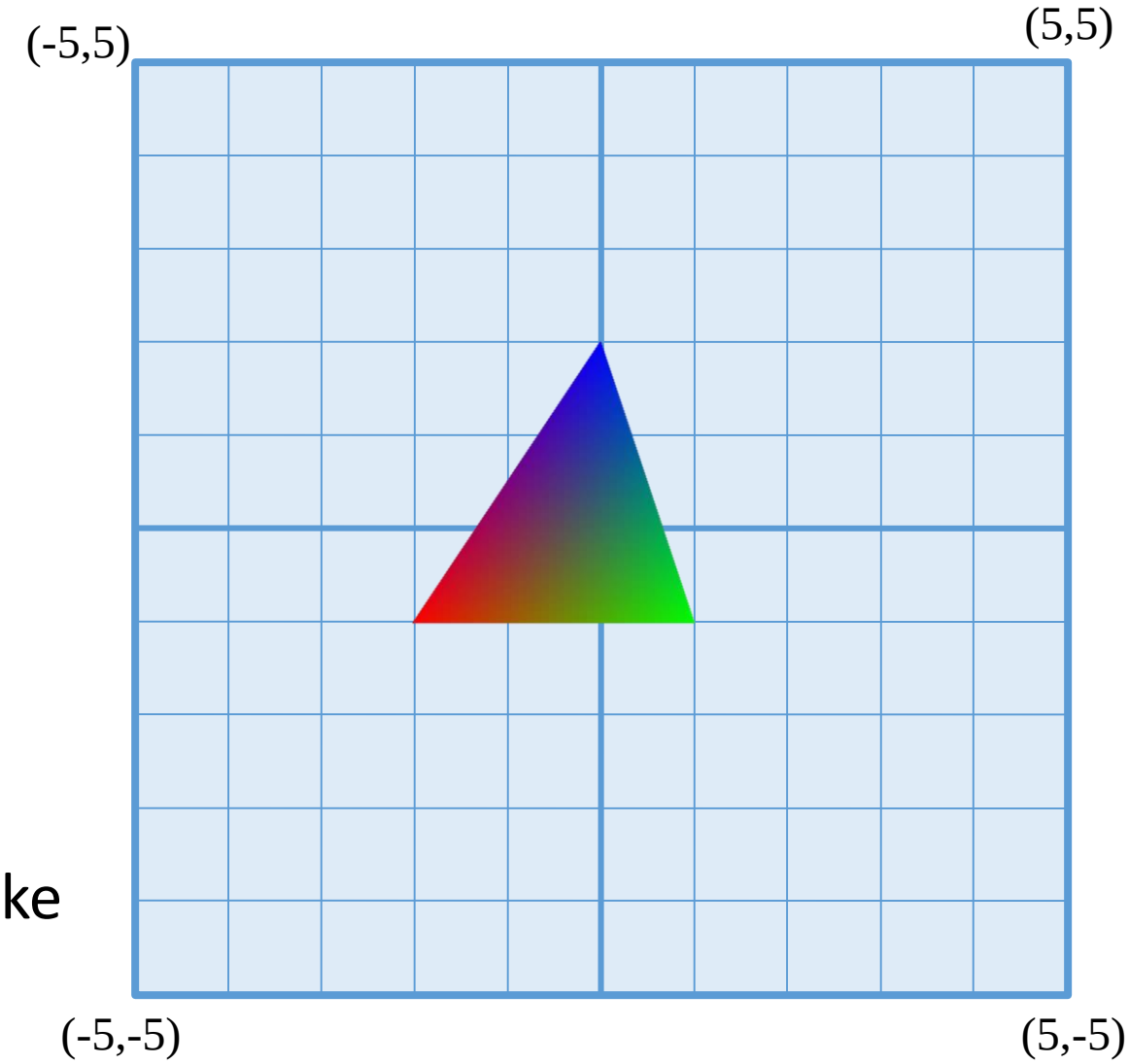
- OpenGL is a state machine.



Color Per Vertex

- glBegin(GL_TRIANGLES);
- glColor3f(1, 0, 0);
- glVertex2f(-2, -1);
- glColor3f(0, 1, 0);
- glVertex2f(1, -1);
- glColor3f(0, 0, 1);
- glVertex2f(0, 2);
- glEnd();

- Why does the triangle color look like this?

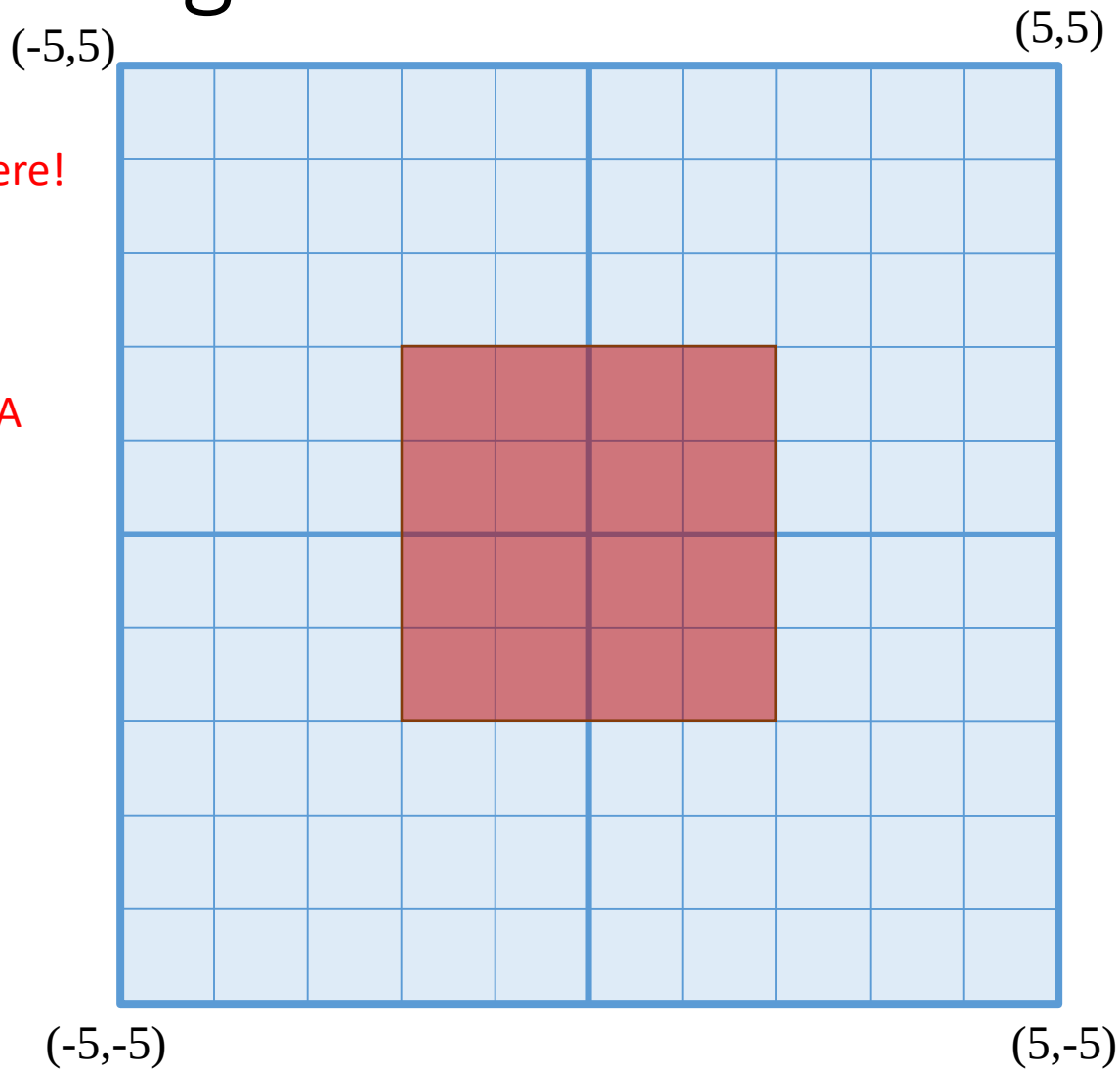


Transparency and Blending

- glEnable(GL_BLEND);
- glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
- glColor4f(1, 0.5, 0.5, 0.5);
- glBegin(GL_QUADS);
- glVertex2f(-2, -2);
- glVertex2f(2, -2);
- glVertex2f(2, 2);
- glVertex2f(-2, 2);
- glEnd();

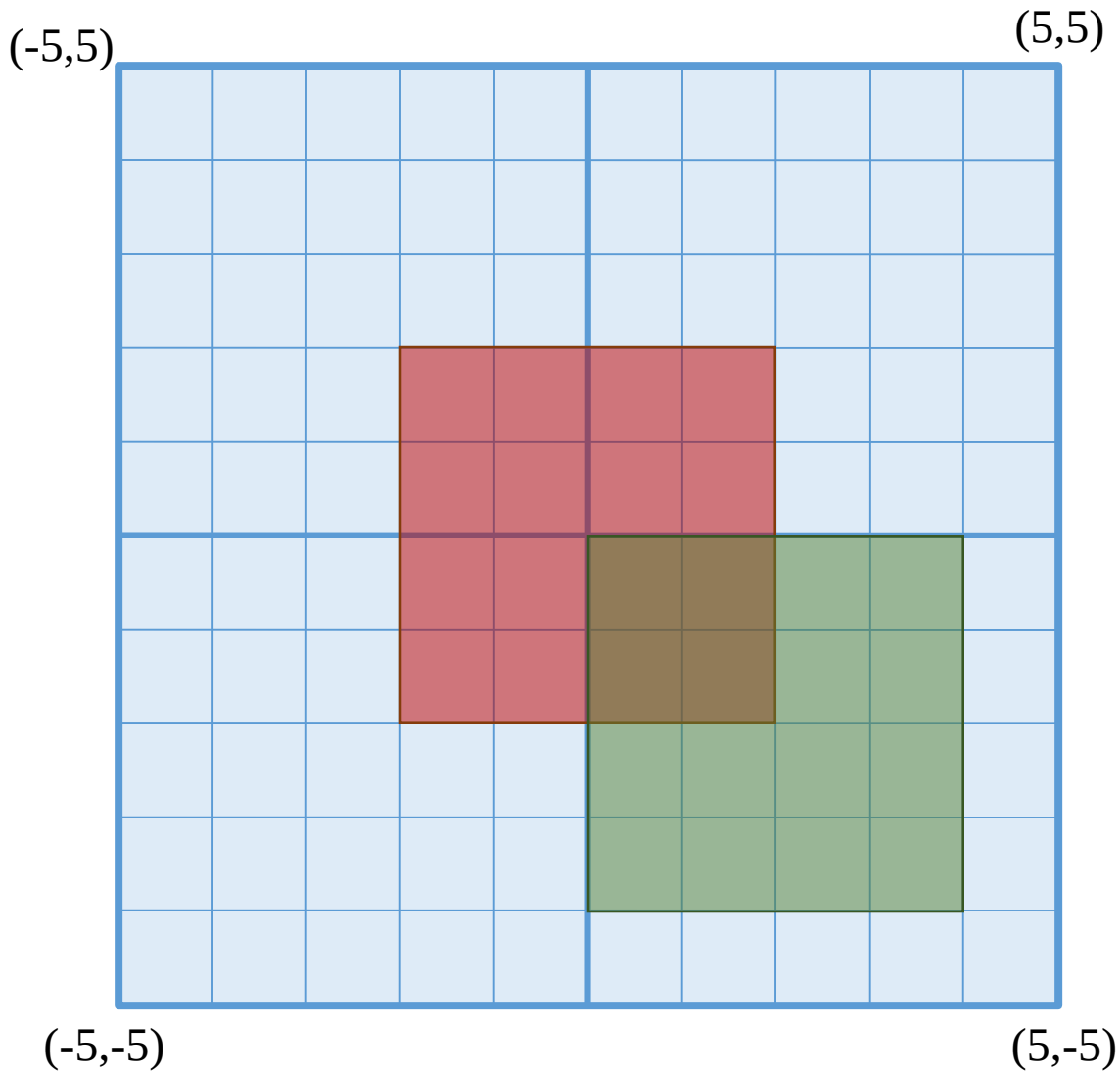
Many possible blend modes here!

Takes in RGBA



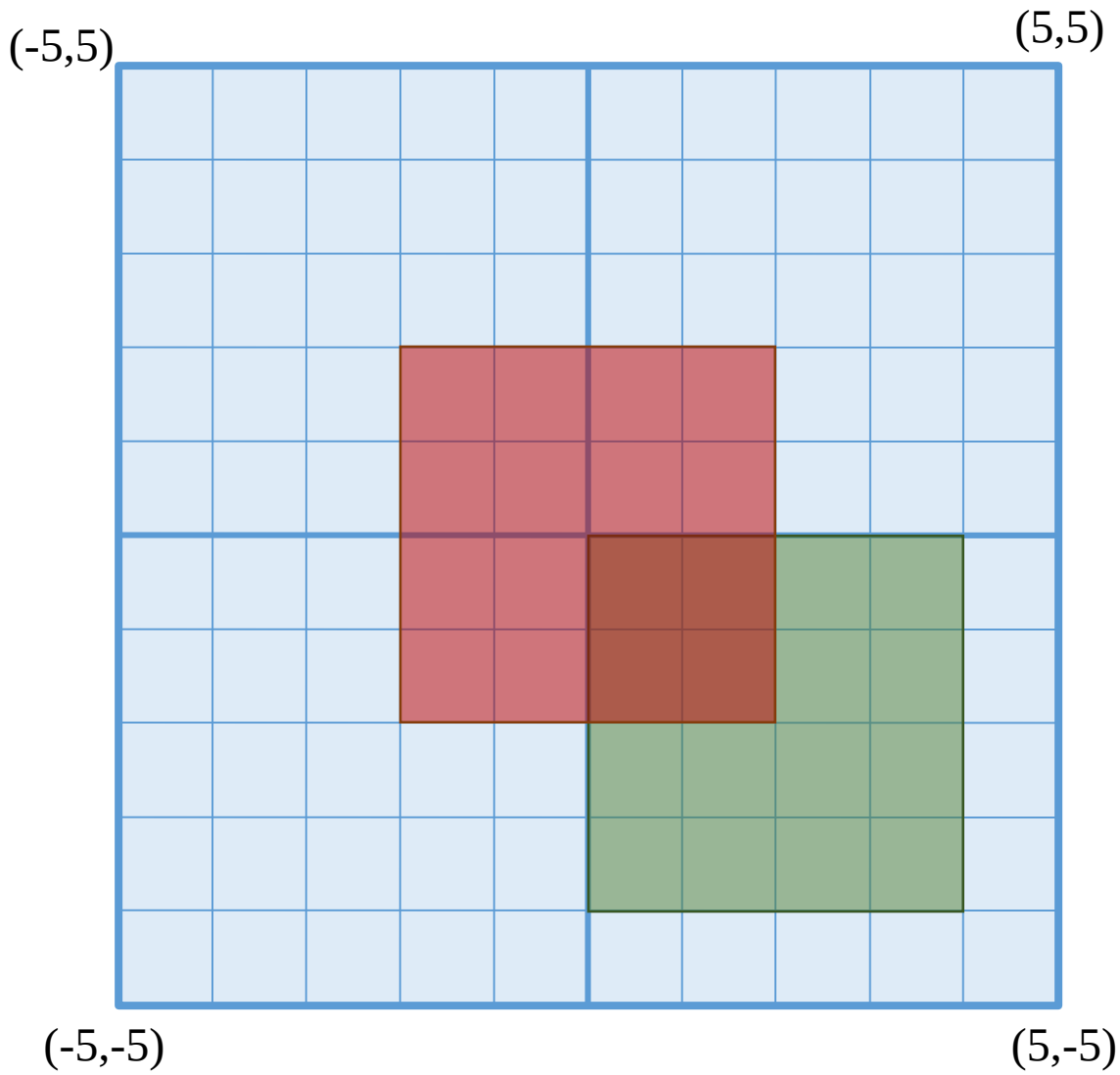
Transparency and Blending: Drawing Order

- glEnable(GL_BLEND);
- glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
- drawRedSquare();
- drawGreenSquare();



Transparency and Blending: Drawing Order

- glEnable(GL_BLEND);
- glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
- drawGreenSquare();
- drawRedSquare();

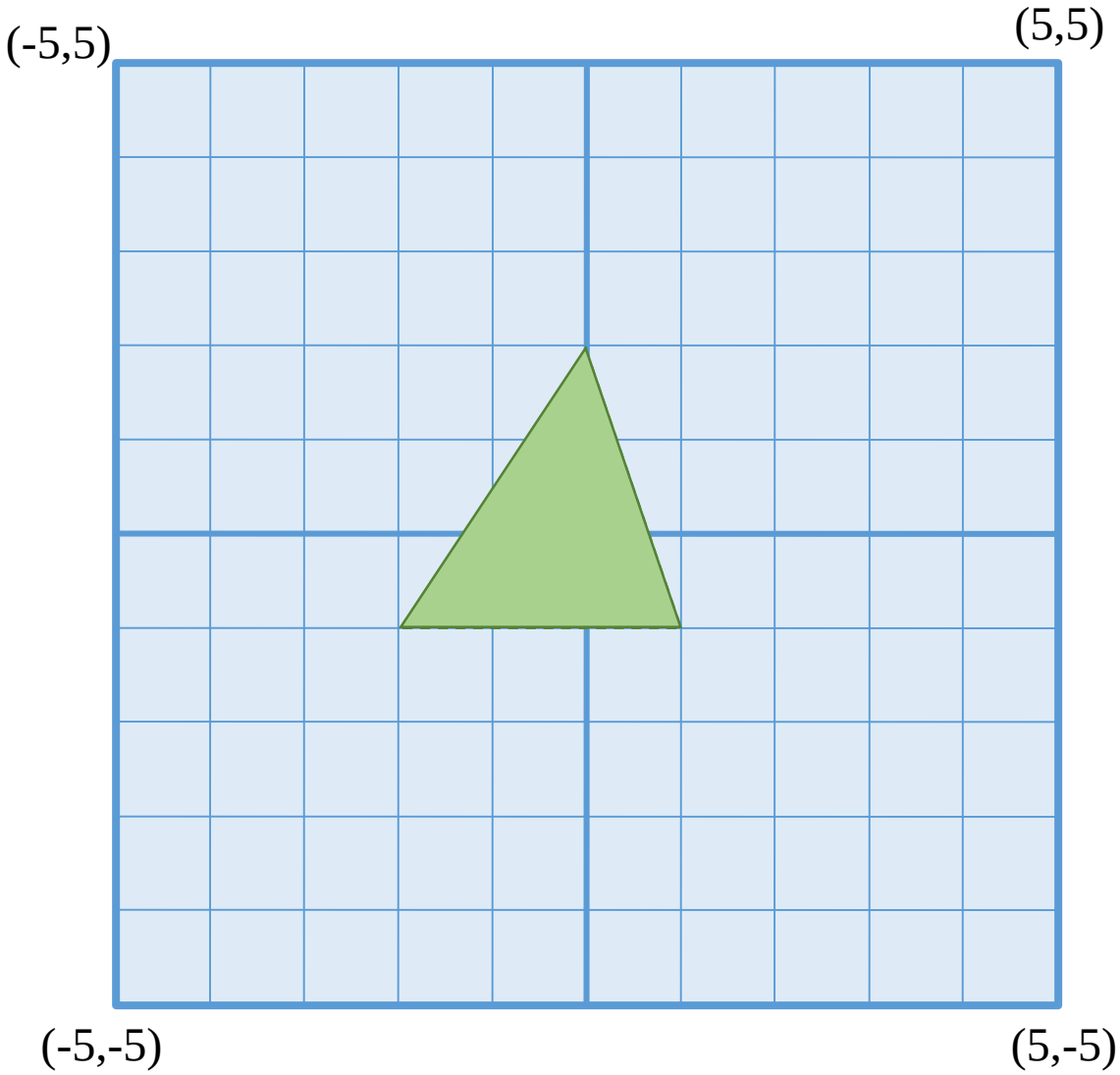


Transformations

Translation

Notice it comes before the triangle

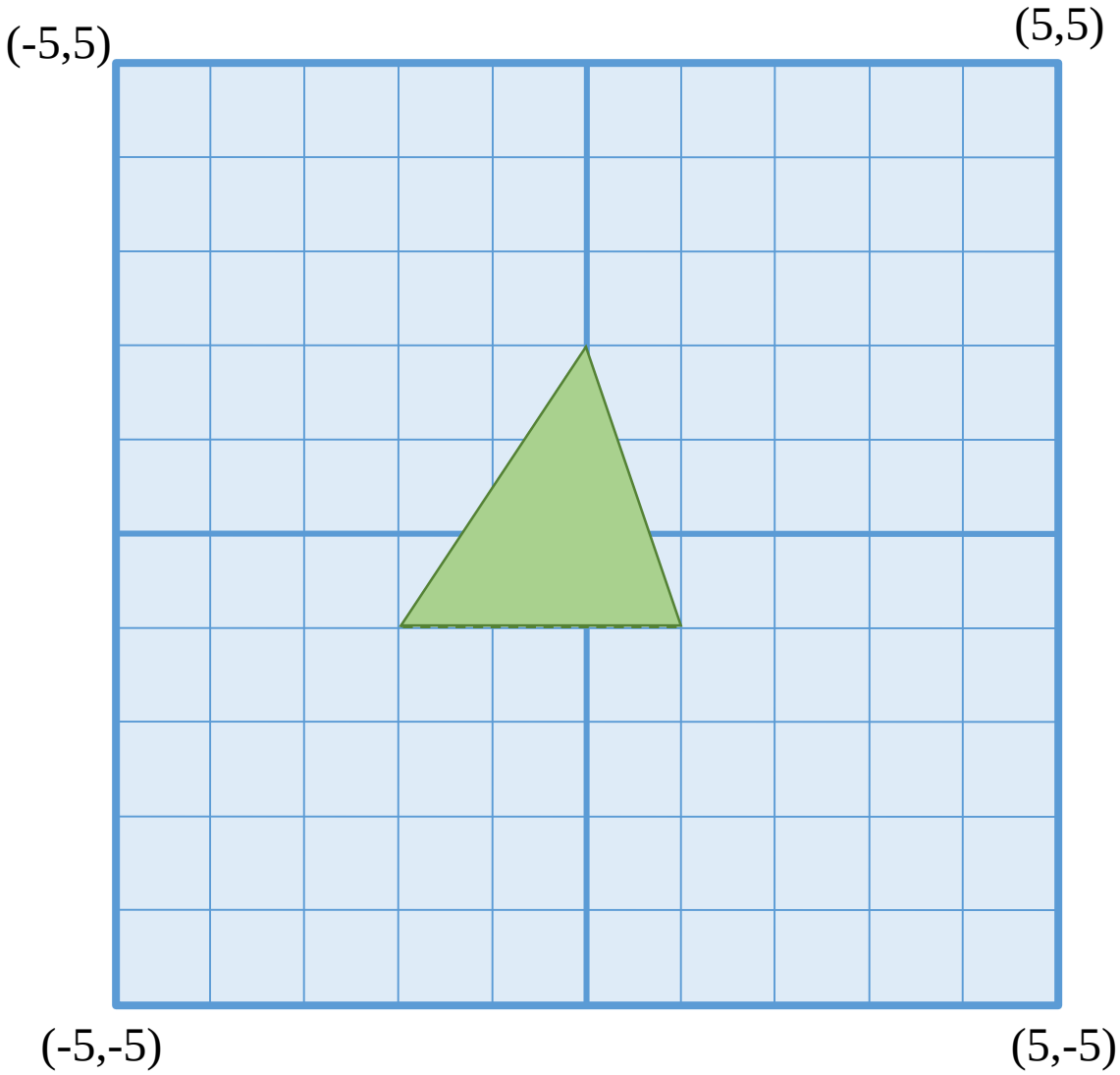
- glTranslatef(-2, -2, 0);
- glBegin(GL_TRIANGLES);
- glVertex2f(-2, -1);
- glVertex2f(1, -1);
- glVertex2f(0, 2);
- glEnd();



Rotation

Notice it comes before the triangle

- glRotatef(90, 0, 0, 1);
- glBegin(GL_TRIANGLES);
- glVertex2f(-2, -1);
- glVertex2f(1, -1);
- glVertex2f(0, 2);
- glEnd();



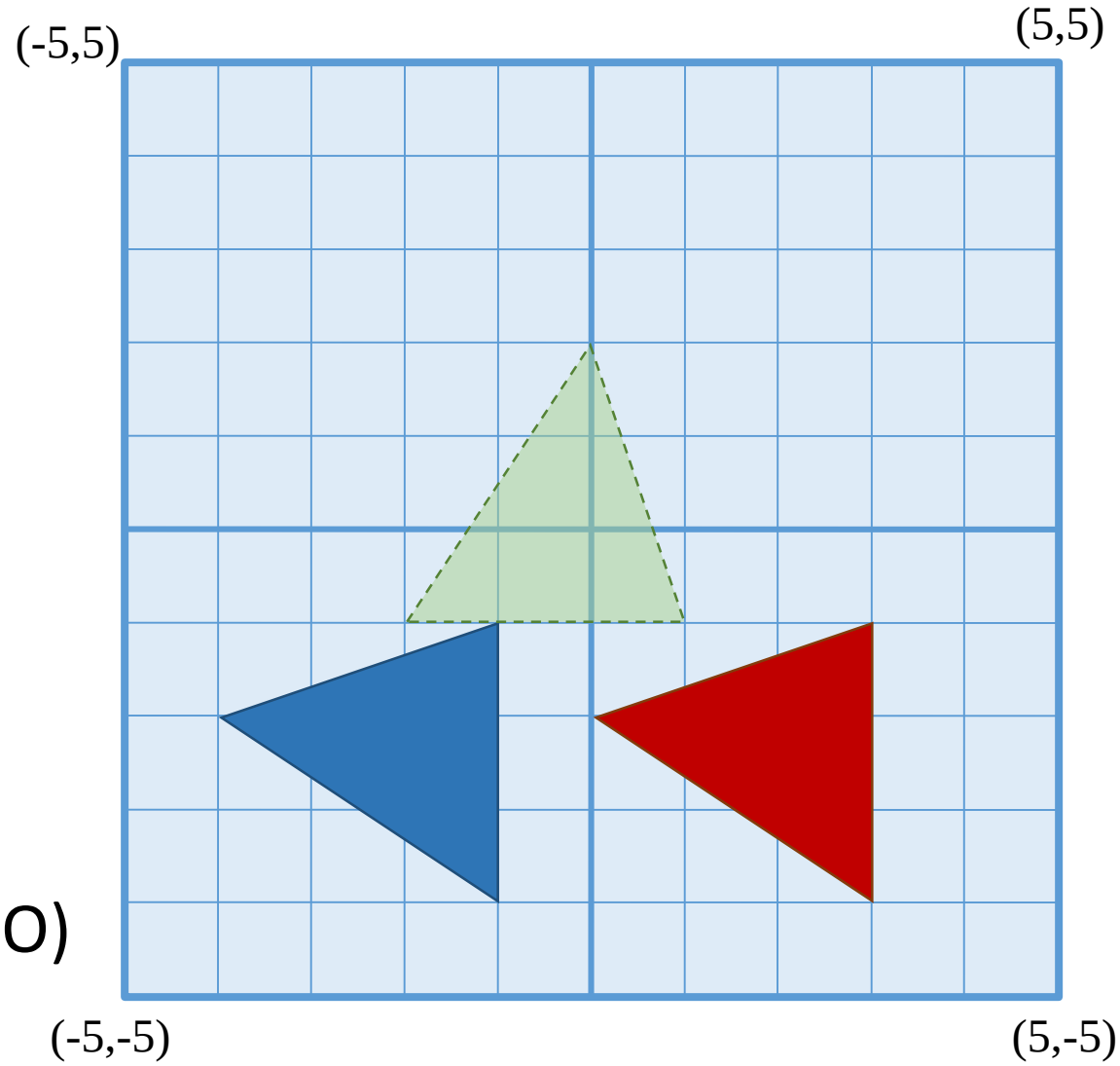
Transformations Are NOT Commutative

- glRotatef(90, 0, 0, 1);
- glTranslatef(-2, -2, 0);

- vs.

- glTranslatef(-2, -2, 0);
- glRotatef(90, 0, 0, 1);

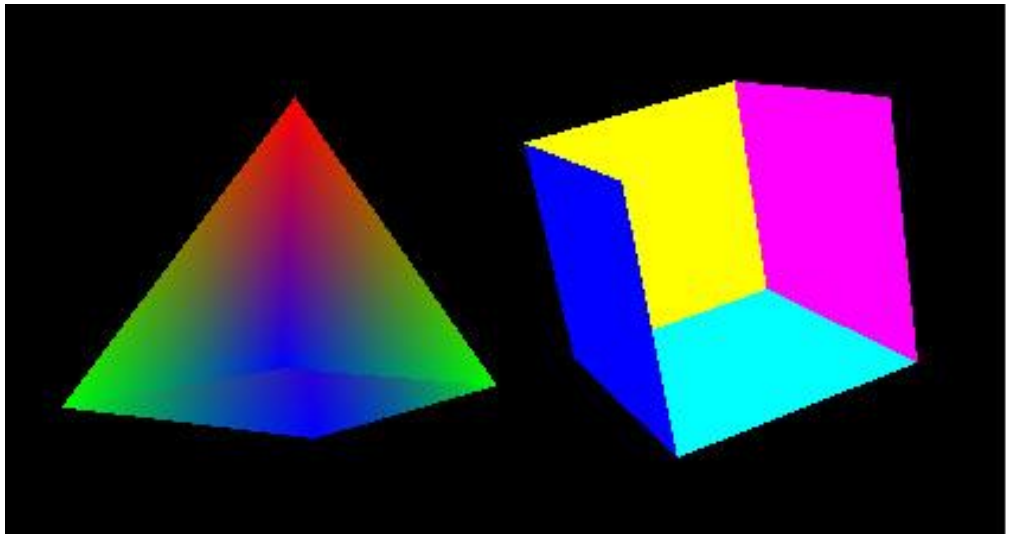
- Transformations are stacked (LIFO)

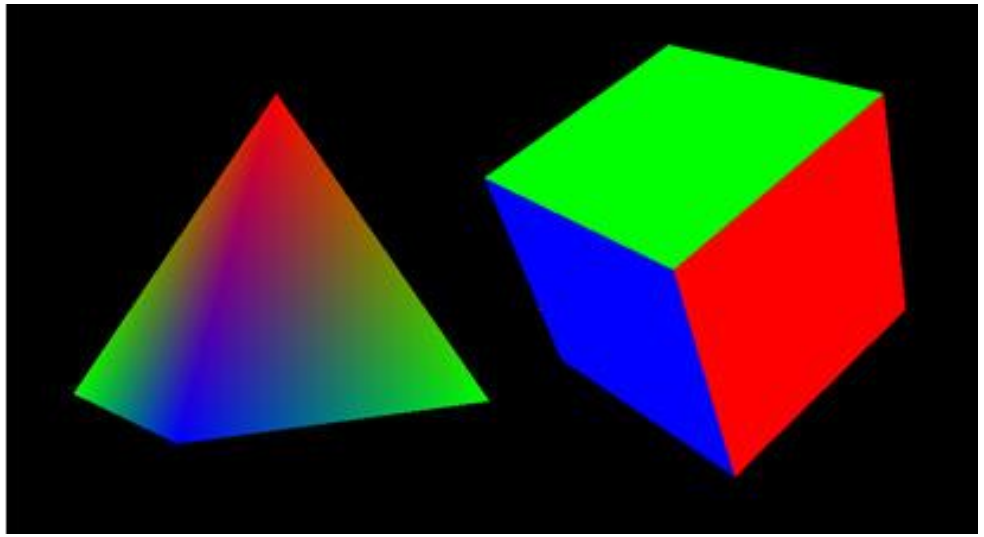


Going to 3D

Depth Test

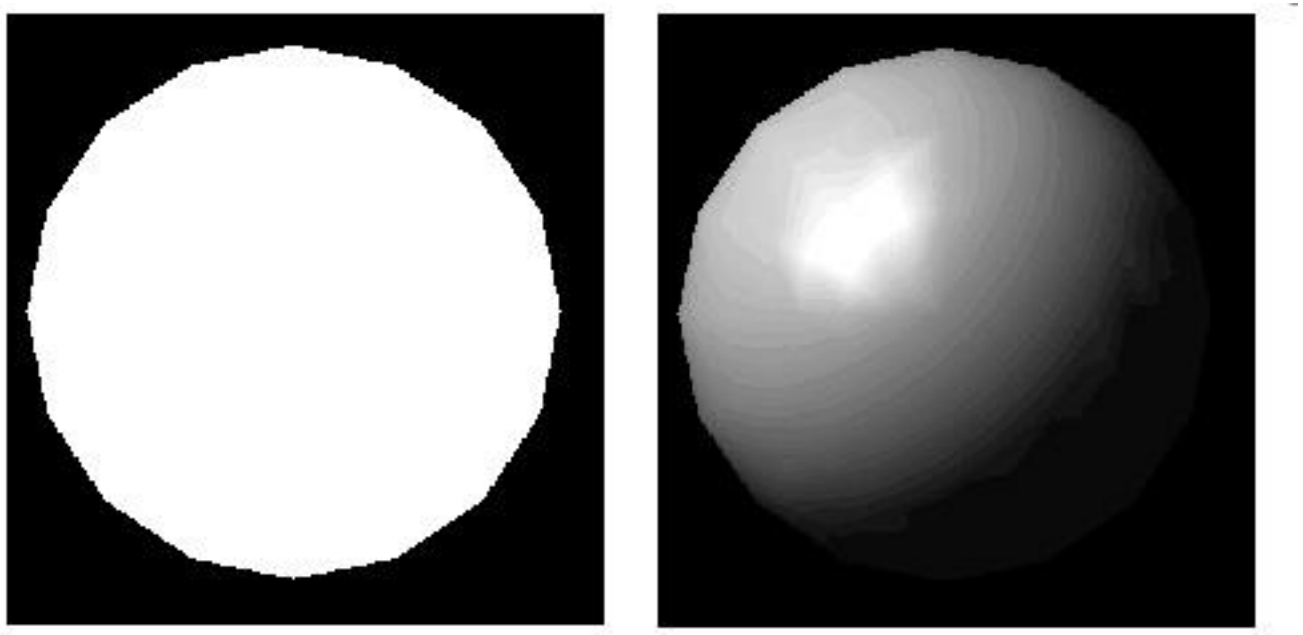
- Makes sure objects in front cover objects in back
- See glEnable(GL_DEPTH_TEST)





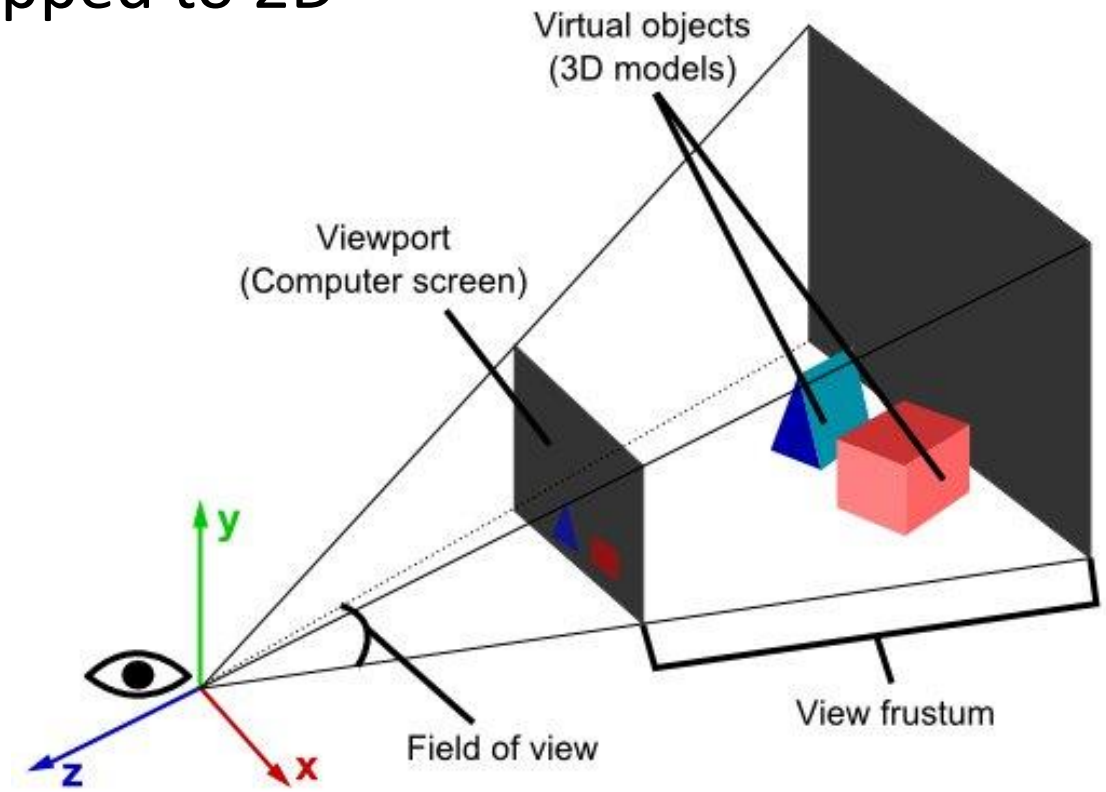
Lighting

- Colors primitives based on light and surface normal
- See glEnable(GL_LIGHTING) and glNormal



Projection

- Controls how 3D coordinates get mapped to 2D
- See glMatrixMode(GL_PROJECTION)



Source: http://www.real3dtutorials.com/tut00002.php

More Resources

- Official OpenGL Documentation
  - https://www.opengl.org/wiki/OpenGL_Reference
  - Or “man glVertex” on Linux/Mac

- Legacy OpenGL Tutorials
  - NeHe (http://nehe.gamedev.net/tutorial/lessons_01__05/22004/)
  - Programming Techniques GLUT Tutorial (http://www.programming-techniques.com/2011/12/glut-tutorial-drawing-basic-shapes.html)

- Modern OpenGL Tutorials
  - OpenGL-Tutorial (http://www.opengl-tutorial.org/)
  - OpenGL-Introduction (https://open.gl/)