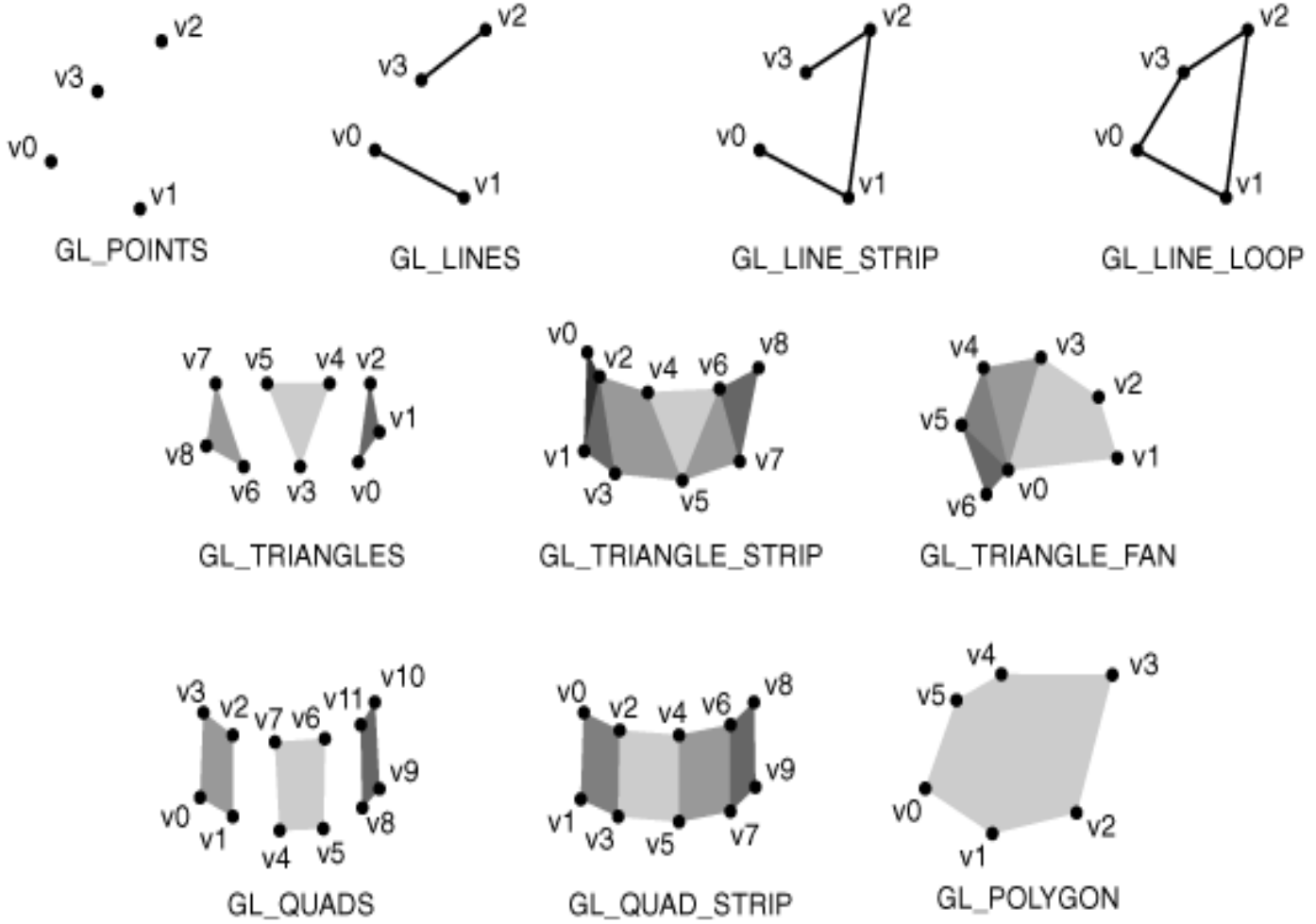
CS475m - Computer Graphics

Lecture 2 : OpenGL Drawing

What is OpenGL?

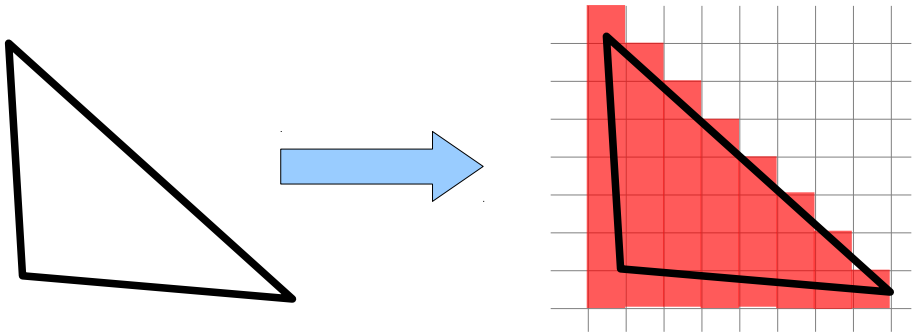
- **Open G**raphics **L**ibrary
- *API* to specify geometric objects in 2D/3D and to control how they are *rendered* into the framebuffer.
- A software interface to graphics hardware.
- Cross language, cross platform, open source
- Alternatives – Direct3D (Microsoft)

OpenGL Primitives



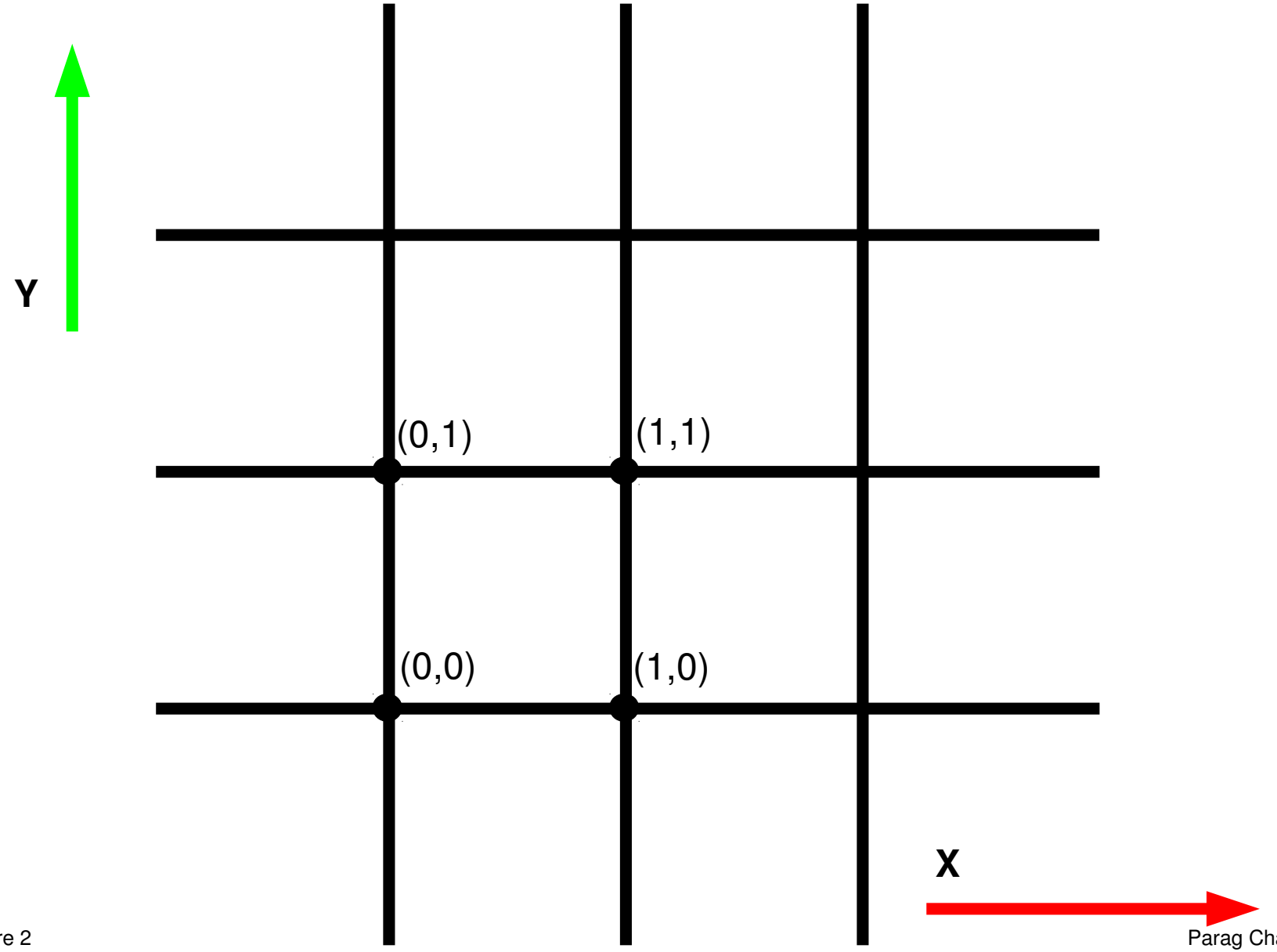
OpenGL Fragments

- A ***fragment*** is a pixel with a lot of other information:
  - Location
  - Color
  - Normal
  - Depth
  - Opacity
  - ....

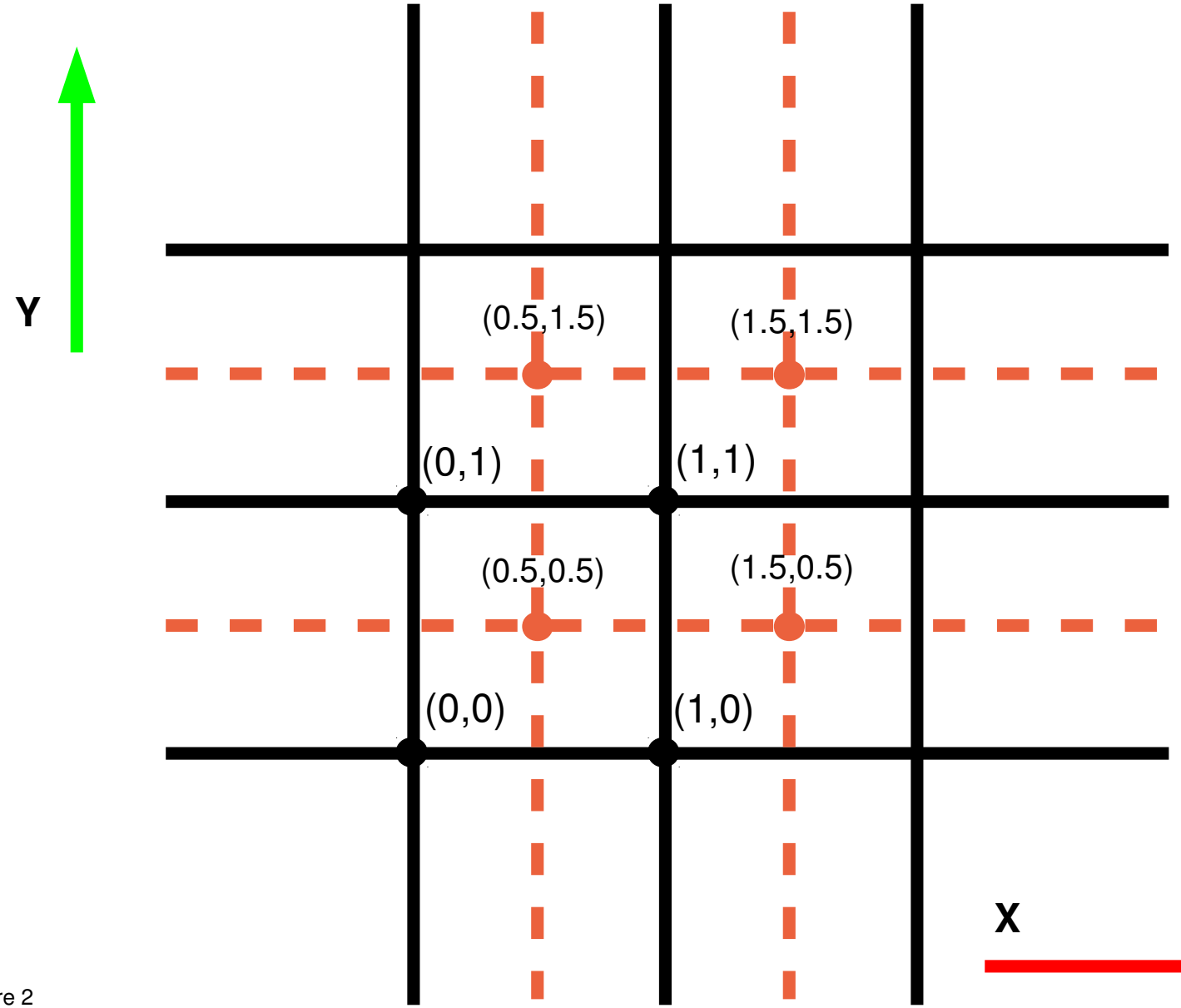


OpenGL rasterizes primitive shapes and outputs fragments.

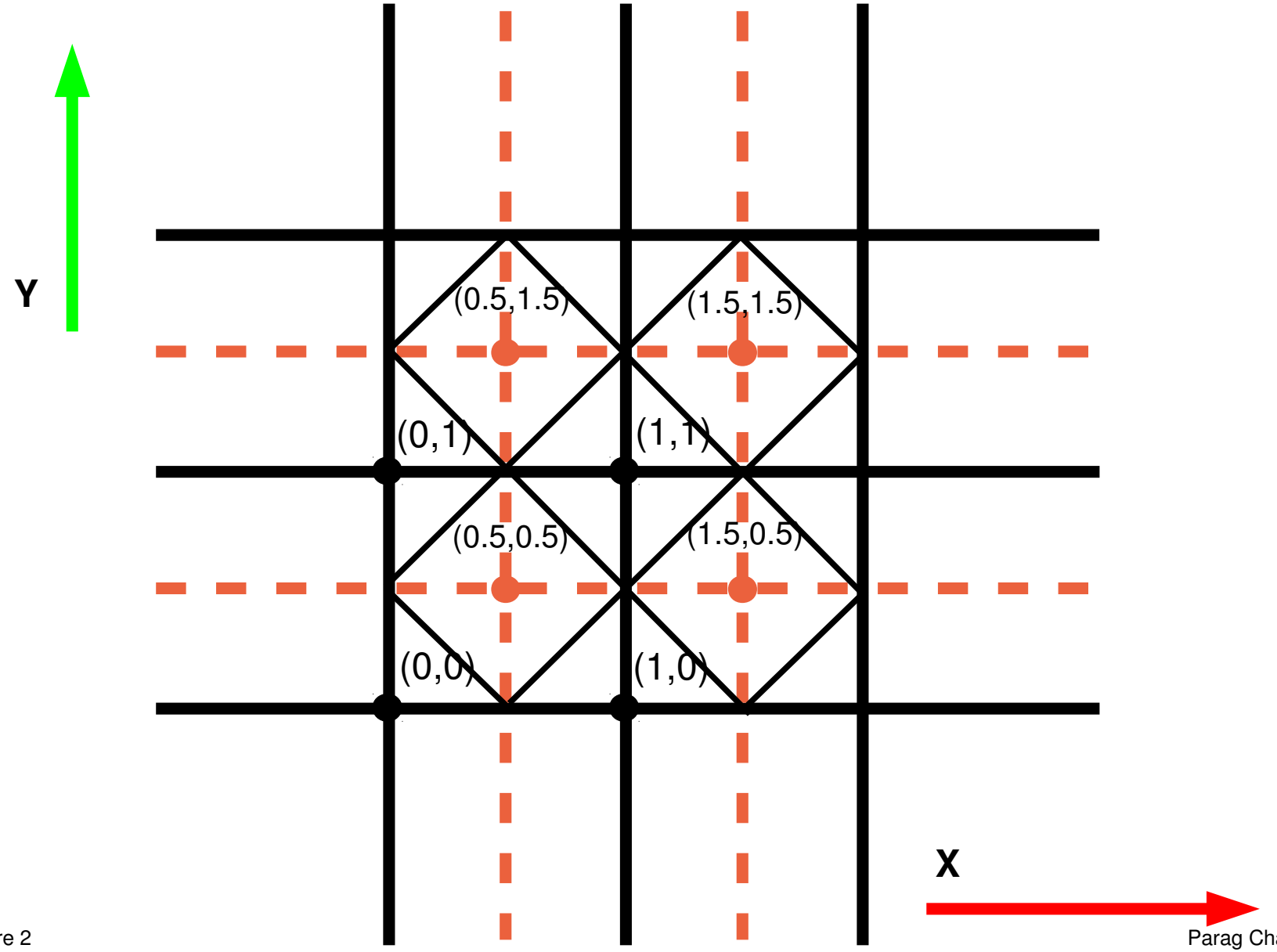
OpenGL Rasterization



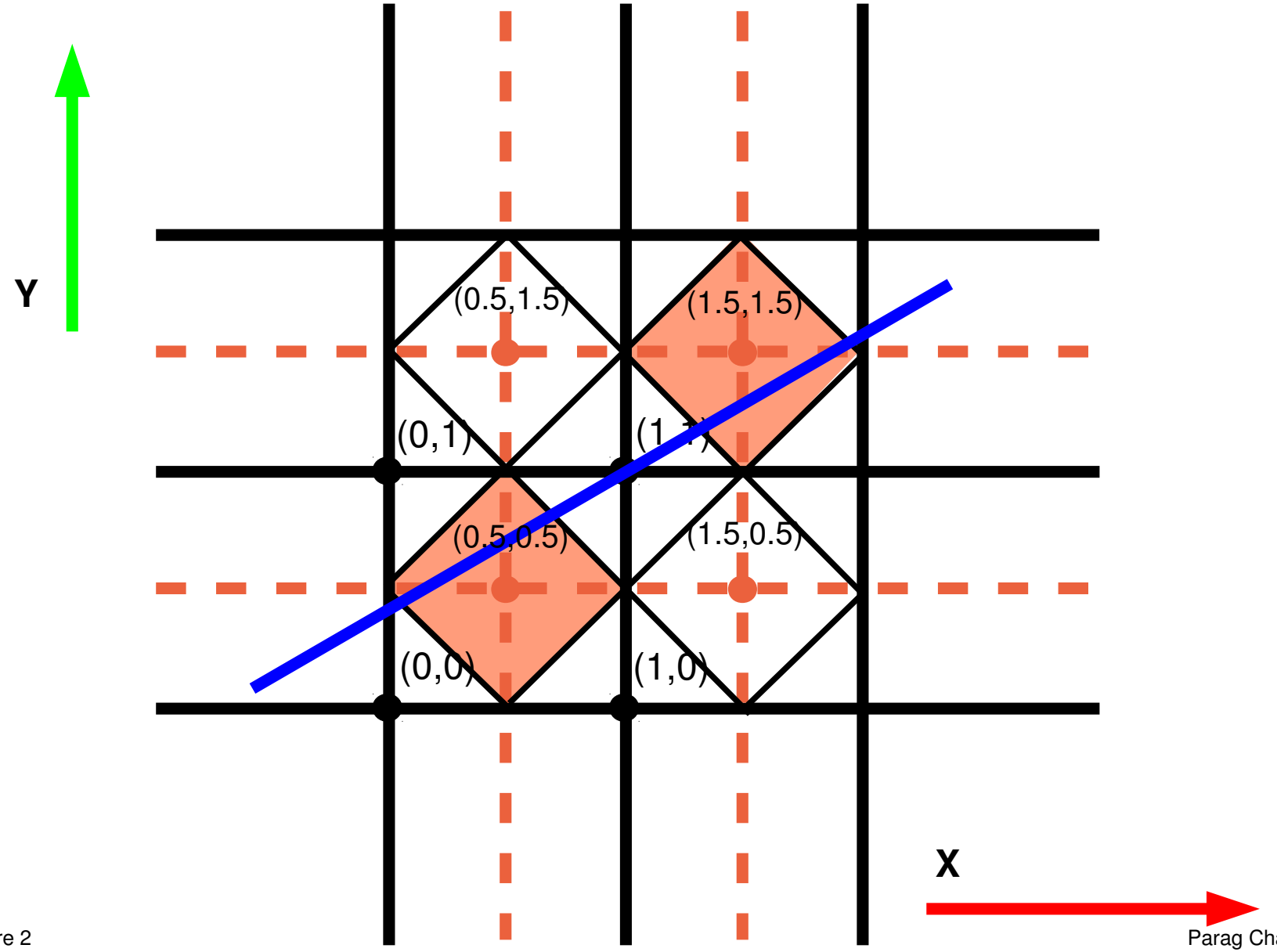
OpenGL Rasterization



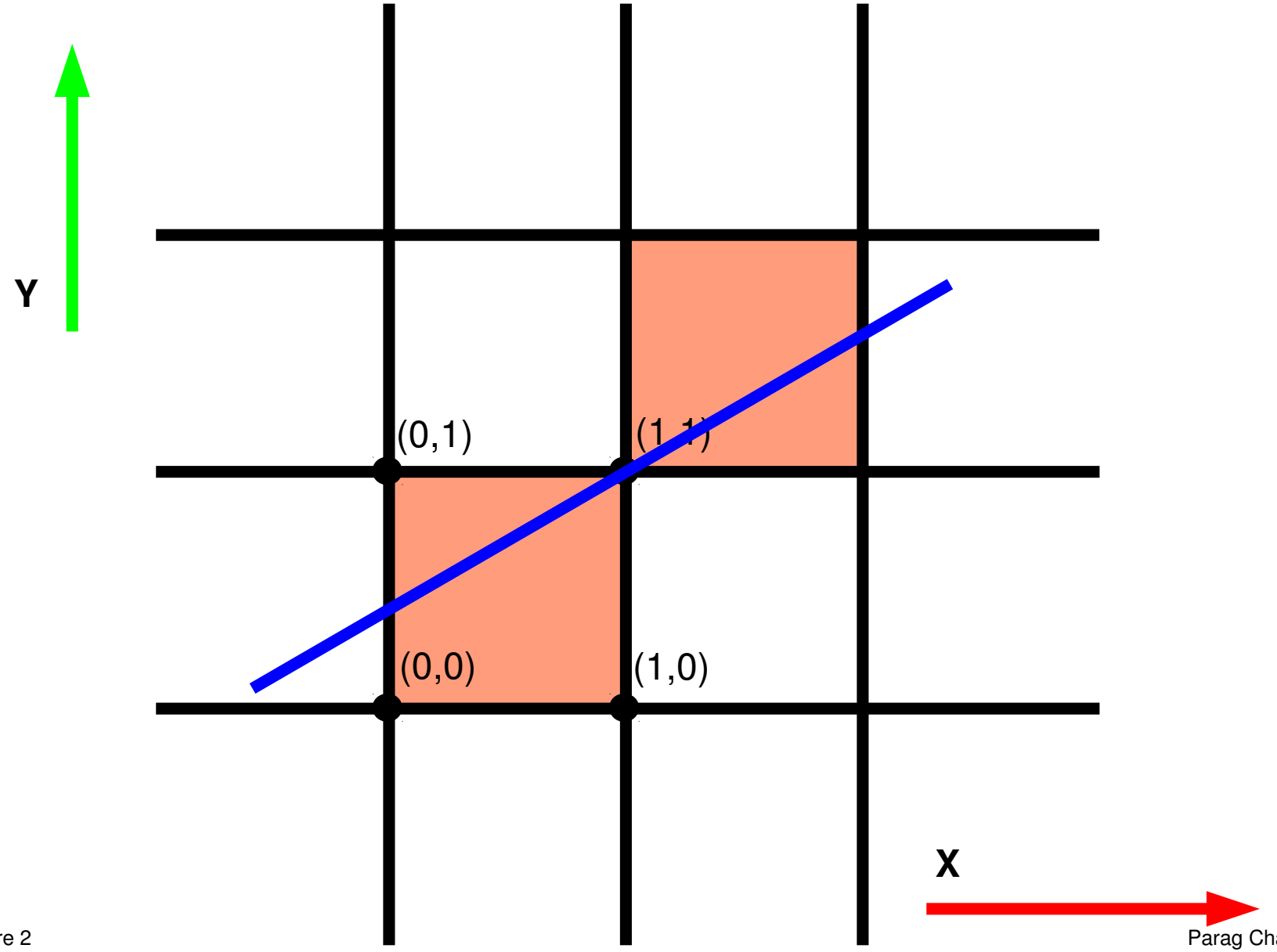
OpenGL Line Rasterization



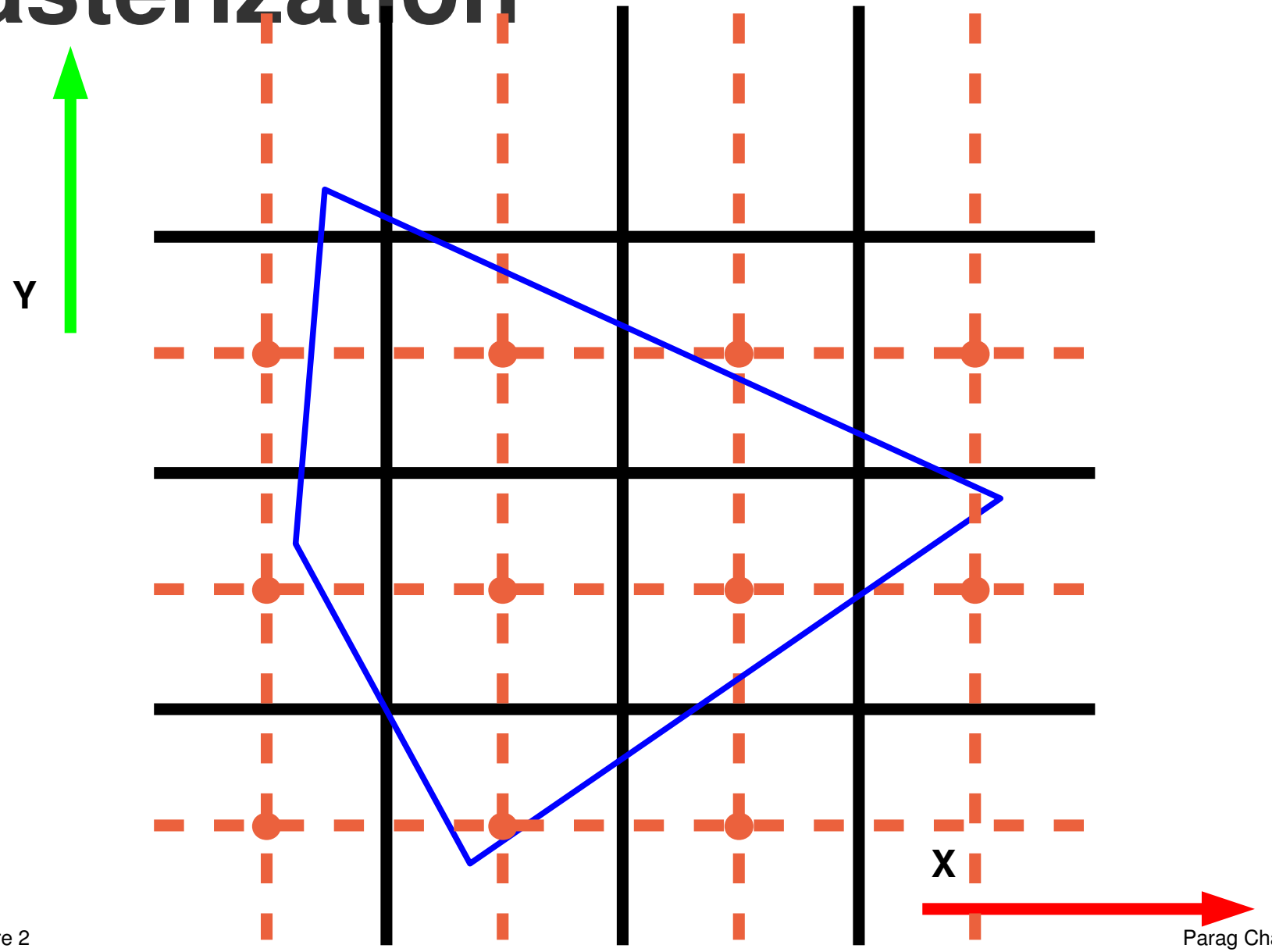
OpenGL Line Rasterization



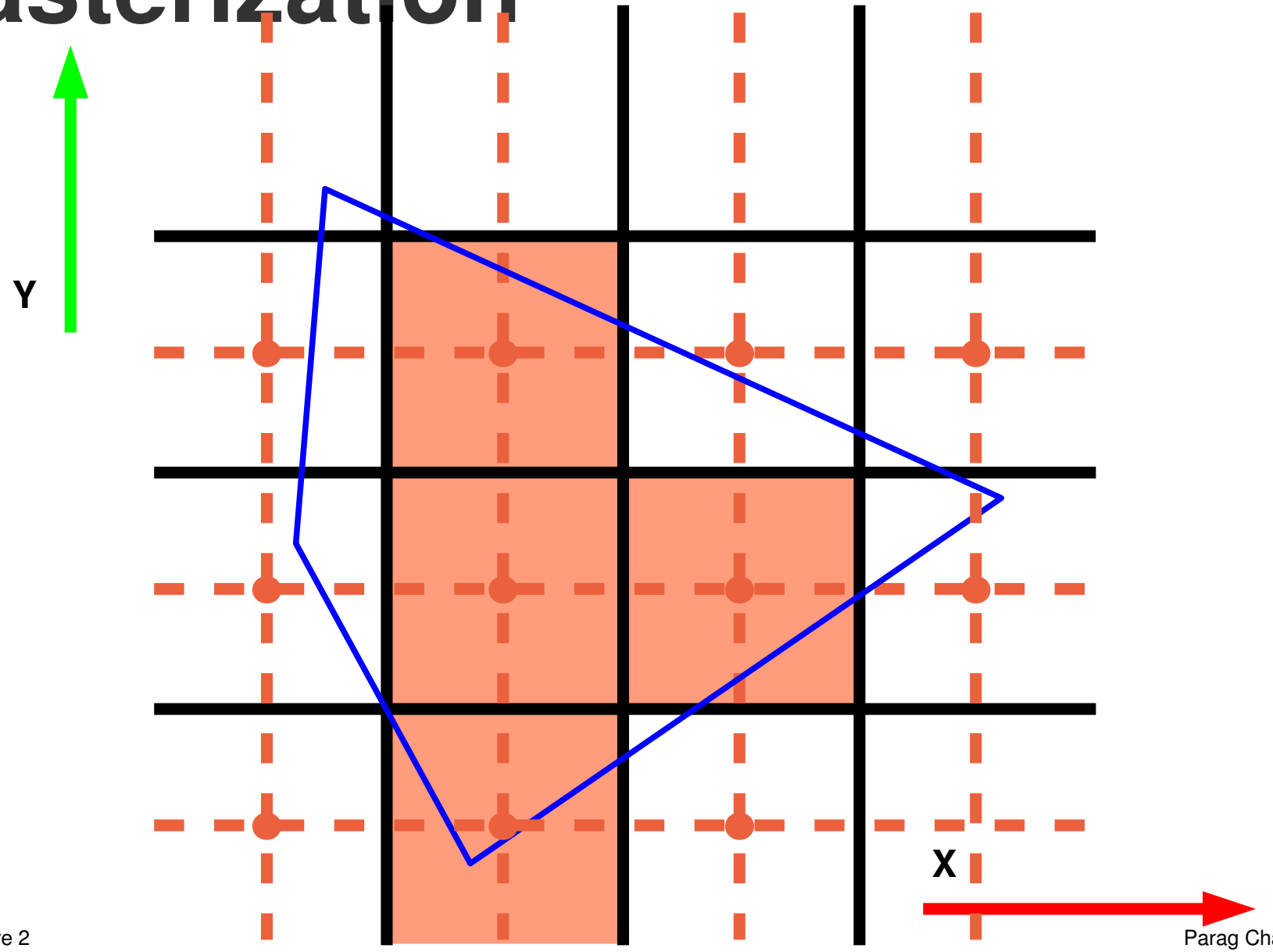
OpenGL Line Rasterization



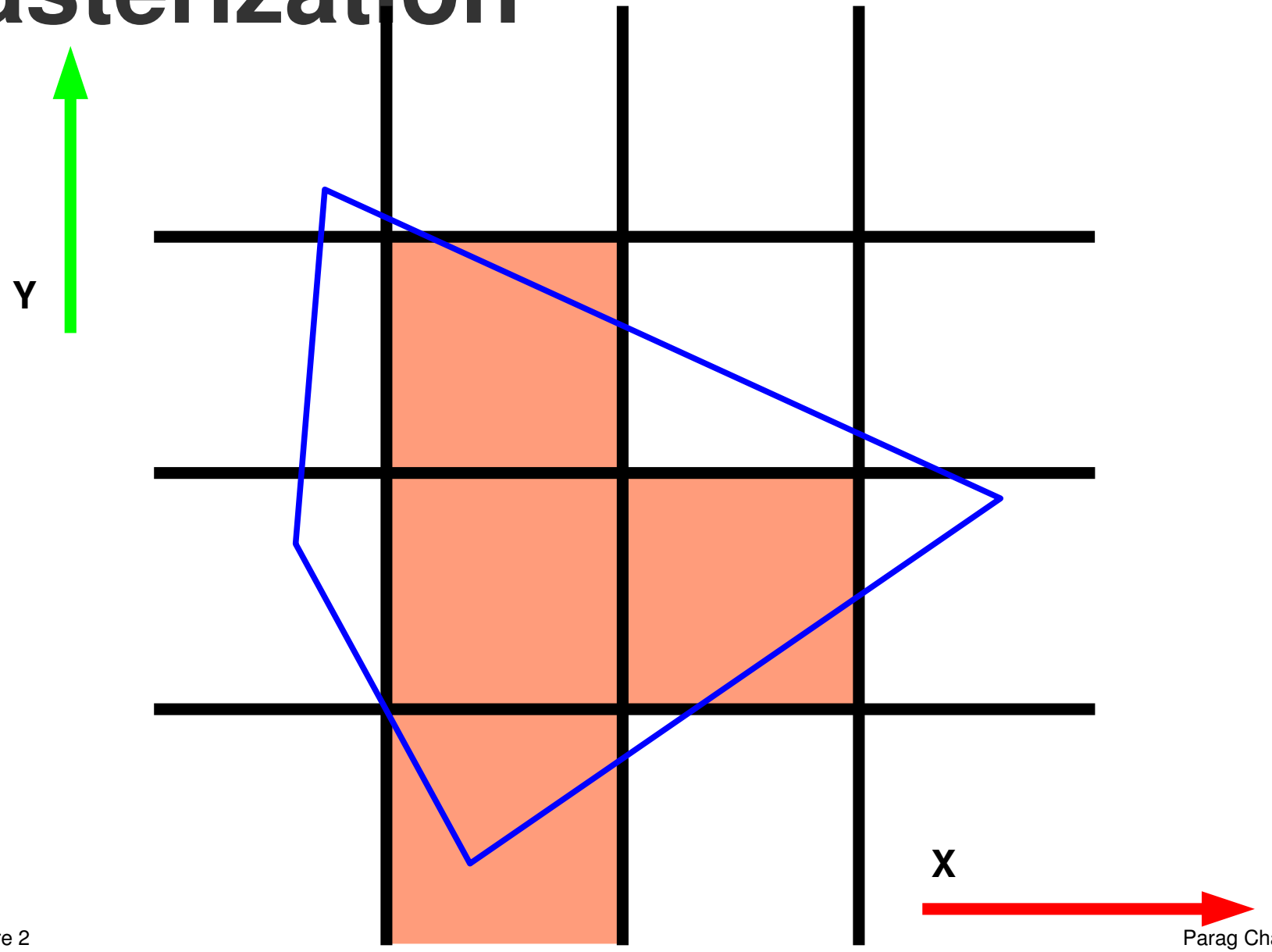
OpenGL Polygon Rasterization



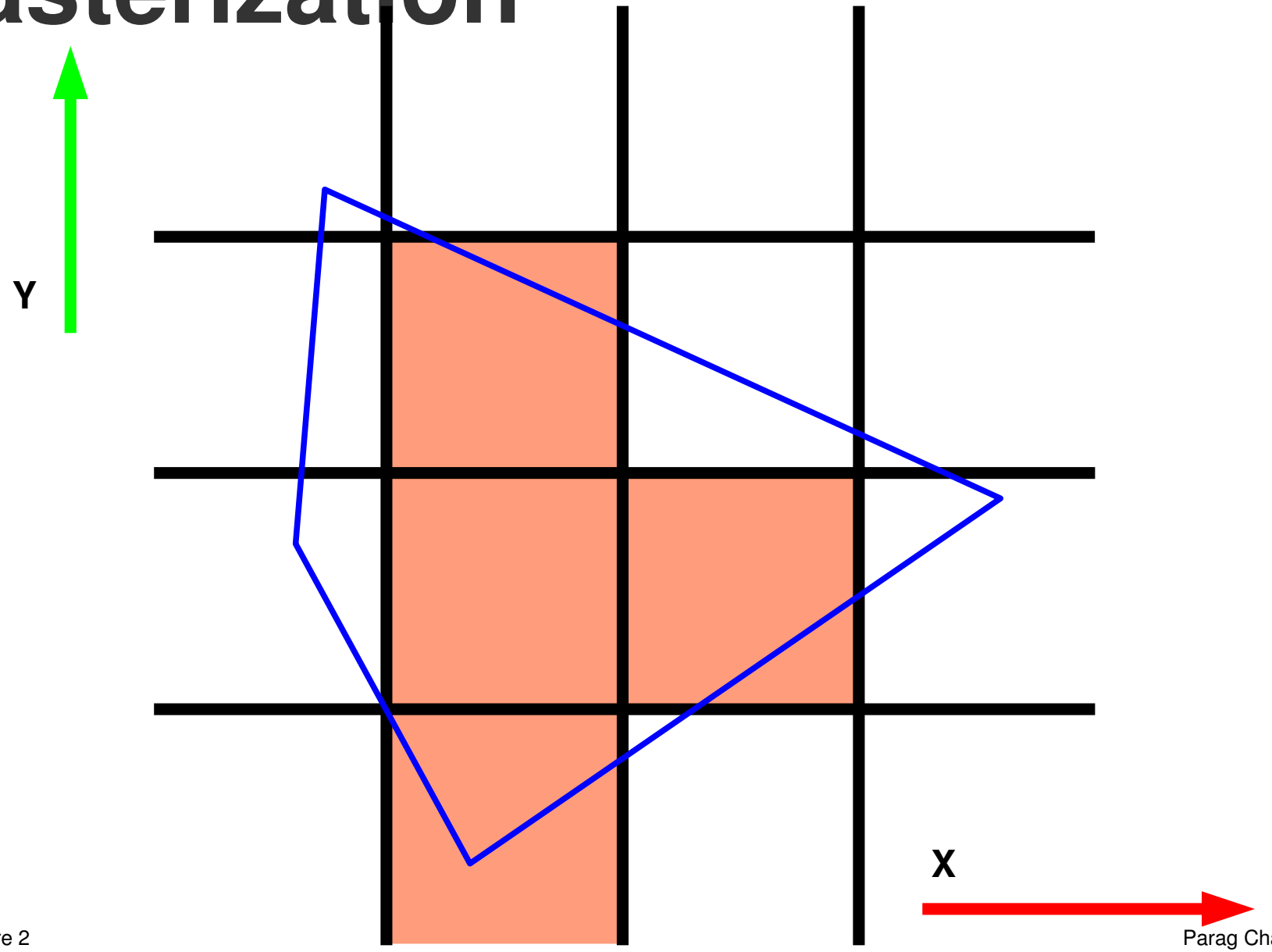
OpenGL Polygon Rasterization



OpenGL Polygon Rasterization



OpenGL Polygon Rasterization



OpenGL State Machine

- Primitive data flows through the state machine, gets rendered according to current state – does not alter the state – only vertices and normals specifications.
- Almost everything else changes state and state changes are usually expensive.

Example

```
glBegin(GL_TRIANGLES);
    glVertex2f(0.0, 0.0);
    glVertex2f(1.0, 0.0);
    glVertex2f(0.0, 1.0);
glEnd();
```

Example

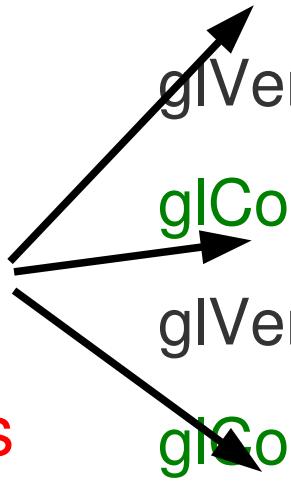
```
glBegin(GL_TRIANGLES)
    glVertex2f(0.0, 0.0);
    glVertex2f(1.0, 0.0);
    glVertex2f(0.0, 1.0);
glEnd();
```

Change state

Do not change state

Example

```
glBegin(GL_TRIANGLES);
    glColor3f(1.0, 0.0, 0.0);
    glVertex2f(0.0, 0.0);
    glColor3f(1.0, 0.0, 0.0);
    glVertex2f(1.0, 0.0);
    glColor3f(1.0, 0.0, 0.0);
    glVertex2f(0.0, 1.0);
glEnd();
```



Bad Programming

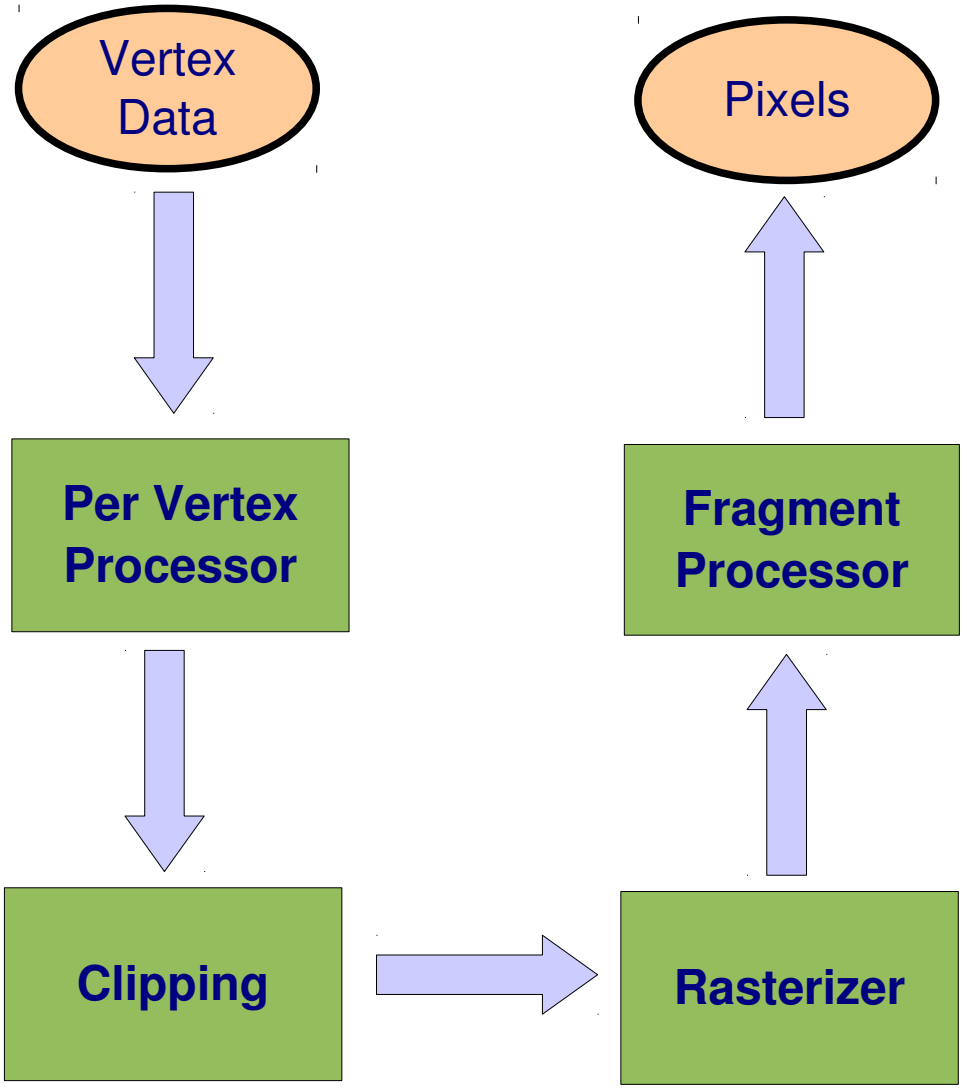
* when color is the same for all vertices!

Example

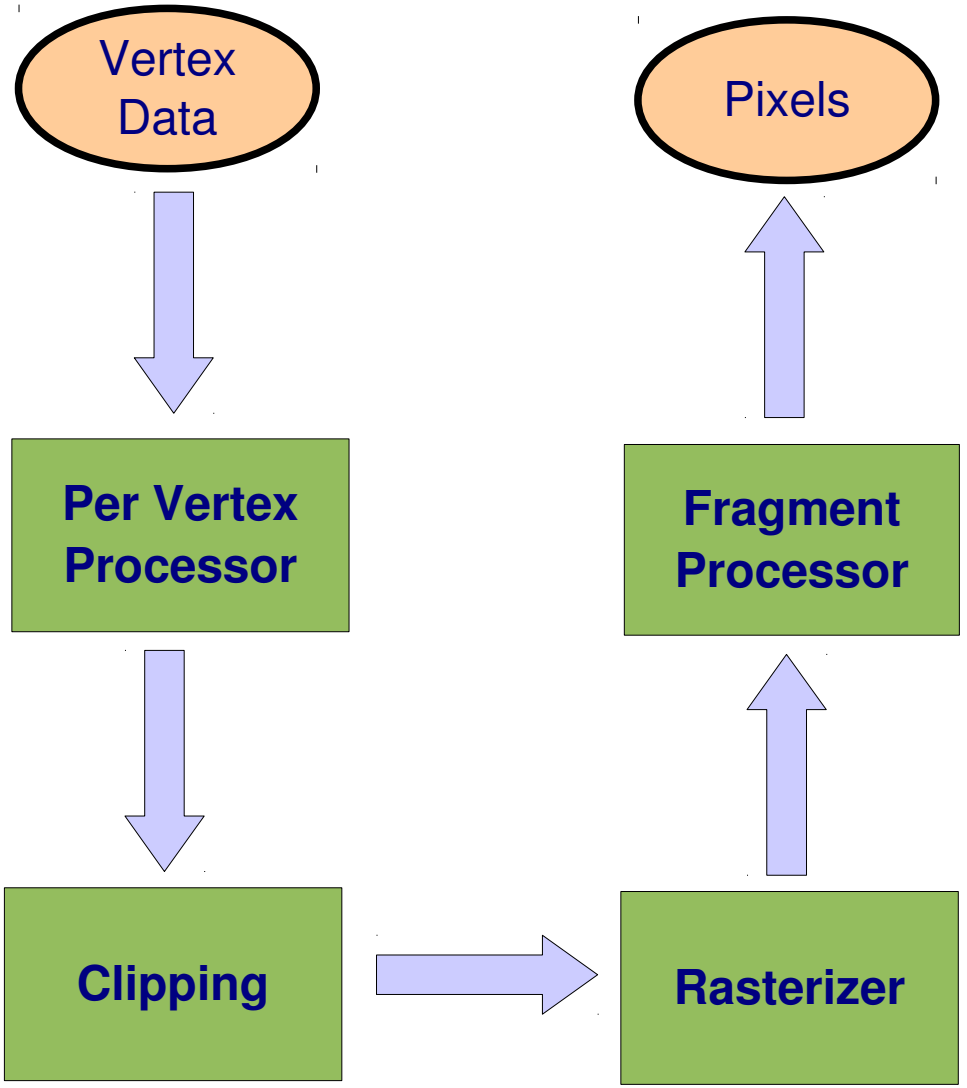
```
glColor3f(1.0, 0.0, 0.0);
glBegin(GL_TRIANGLES);
    glVertex2f(0.0, 0.0);
    glVertex2f(1.0, 0.0);
    glVertex2f(0.0, 1.0);
glEnd();
```

This is better.

The Graphics Pipeline



The Graphics Pipeline



Why pipeline?

GLUT: Event driven programming

- glutMainLoop() – Infinite Loop
- Callbacks
  - Display – Called whenever something is to be drawn. Register using glutDisplayFunc().
  - Resize – Called whenever the window is resized.Register using glutReshapeFunc().
  - Keyboard, Mouse – Called whenever there is input. Register using glutKeyboardFunc().
  - Idle – Called whenever nothing else is being called. Register using glutIdleFunc().

Demo