Three paradigms for rendering:
1) Points, 2) Lines, 3) Triangles (surfaces)

1) Points

\[ \mathbf{x} = \langle 2, 1 \rangle = ( \hat{2} ) \]

Point \rightarrow pixel - can be given any color.

2) Lines

JavaScript:

```
moveTo(2, 1)
lineTo(2, 2)
lineTo(0, 2)
lineTo(12, 2)
```

```
penup();
moveTo(2, 1)
pendown();
moveTo(22);```

Vector Graphics
③ Drawing Triangles

```cpp
float verts[3][2] = {{2, 1}, {2, 2}, {0, 2}};
verts[0][1] = 1;
verts[2][0] = 0;

Draw Arrays (GL_TRIANGLES, 0, 3);
```

Front face

(with vertices are in counterclockwise order)

(CCW)
GPU – Graphics Processing Unit.

Shader programs

Vertex shader

Fragment shader (Compute shaders)

CUDA, OpenCL alternative methods for programming GPUs

3 vertices

pixel (fragment)
In addition to GL_TRIANGLES
there are GL_TRIANGLE_FAN
GL_TRIANGLE_STRIP

Vents array: set up similar to before

Vents holds point $v_0, v_1, v_2, v_3, v_4, v_5$, for a triangle fan

For a triangle strip

Allow not repeating vertices so offer.
Animation

Hidden Surfaces

Front buffer
Back buffer

Back faces
Calling the transfer outside the view