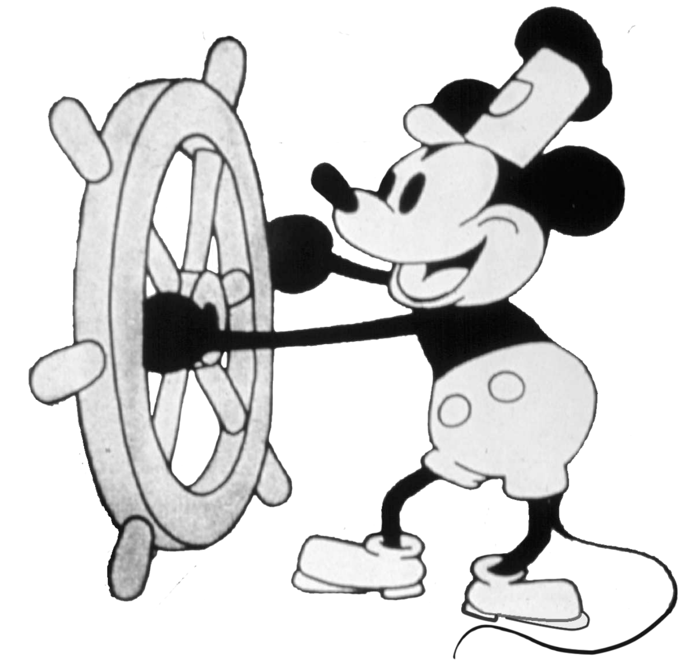


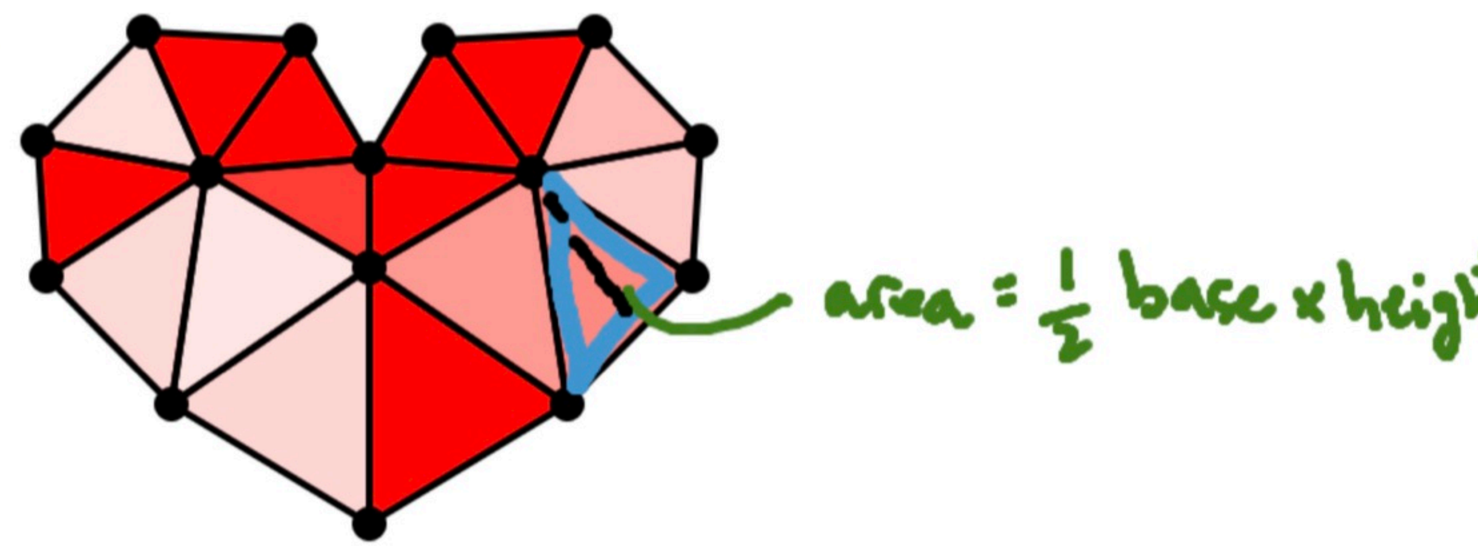
CSCI 461: Computer Graphics

Middlebury College, Spring 2025

Lecture 2A: Linear Algebra (Vectors)

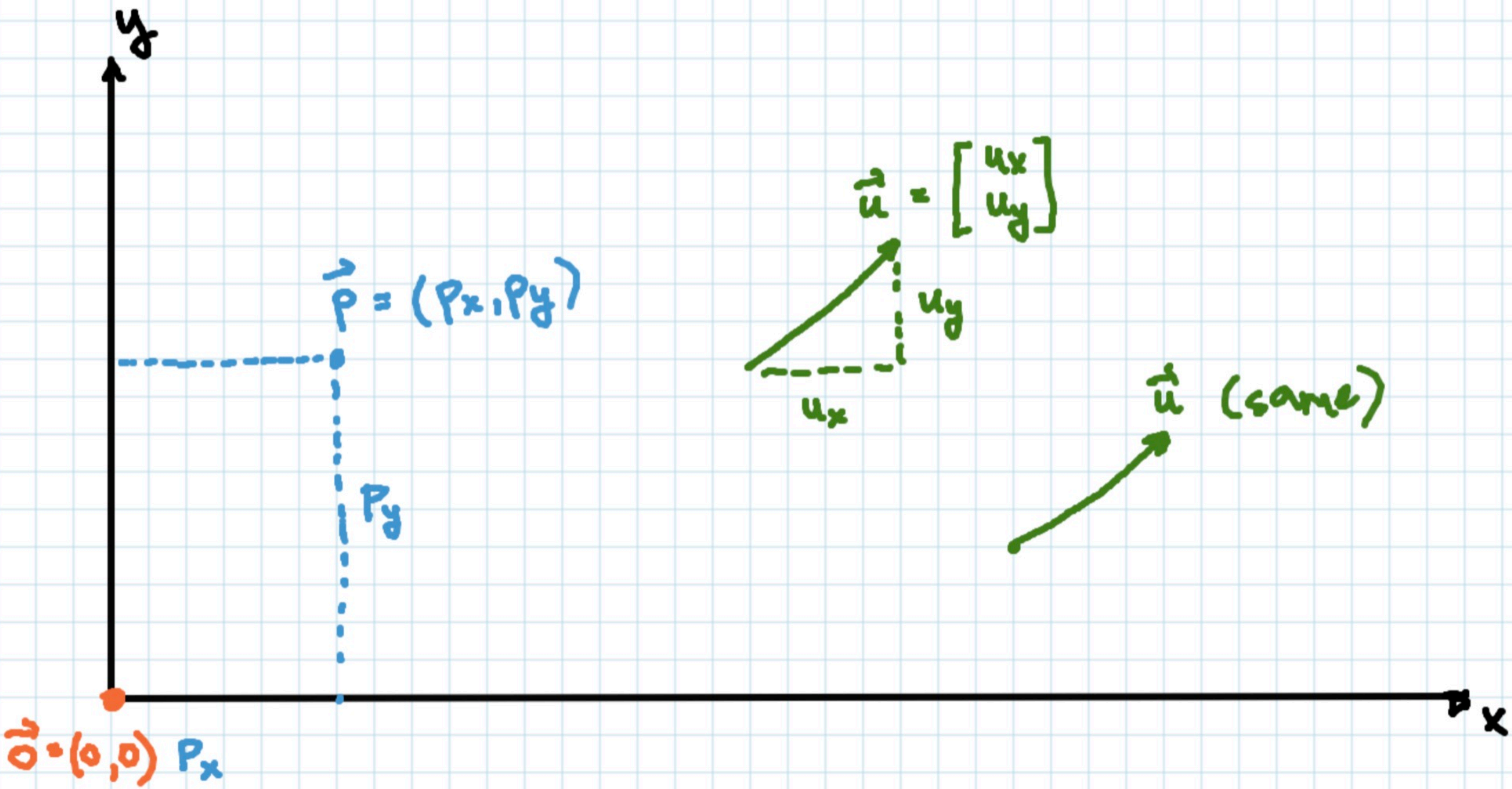
By the end of today's lecture, you will be able to:

- perform operations on vectors such as **addition, subtraction, scaling**,
- calculate the **length** of a vector and compute **unit vectors**,
- compute the **dot product** and **cross product** of two vectors,
- calculate the **area** of a triangle using the cross product,
- **represent lines and planes** using vector notation,
- use `glMatrix` to do everything mentioned above.

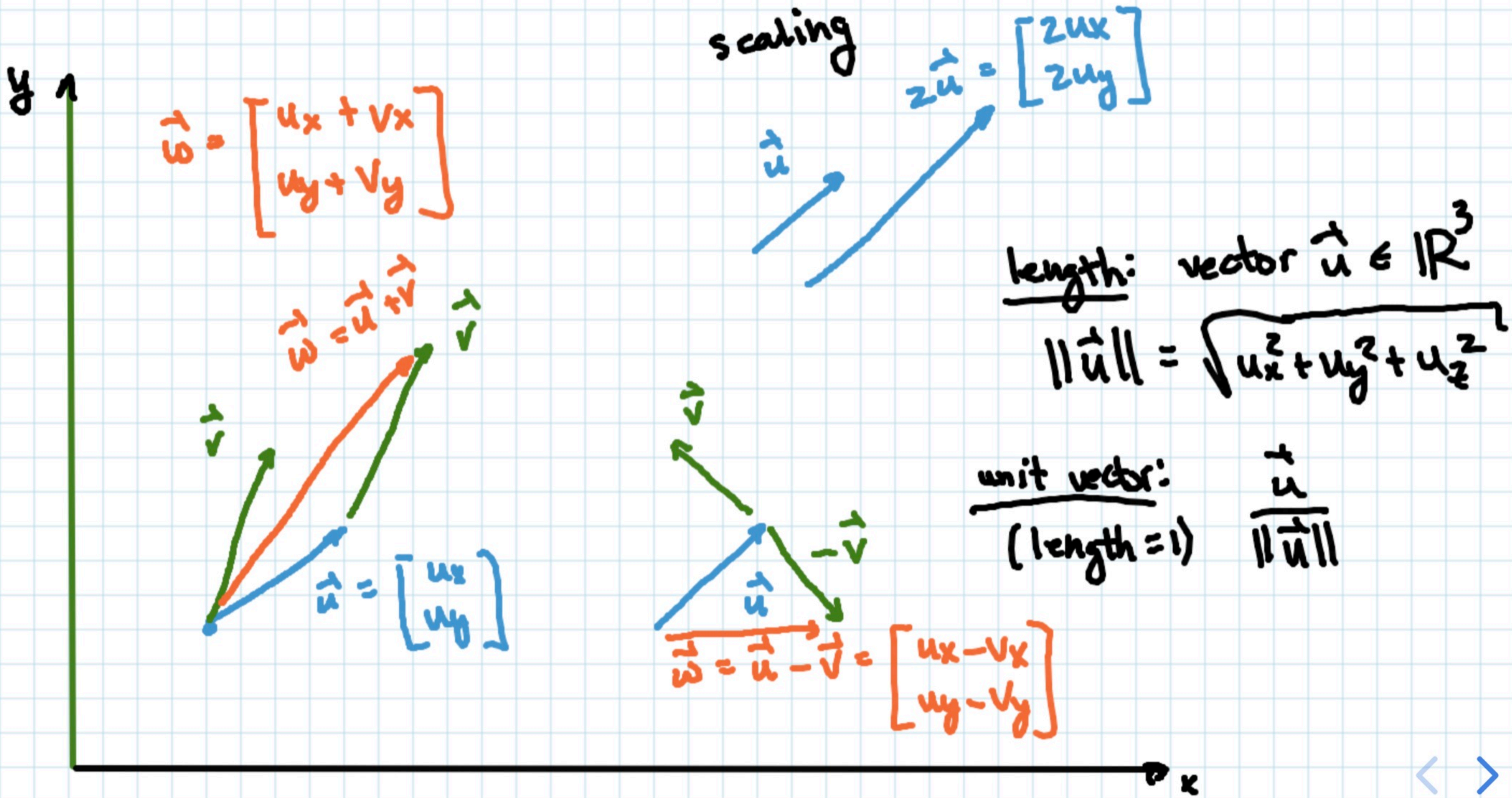


Building blocks: points and vectors.

We'll use an arrow on top of symbols to denote points ($\vec{p}$) and vectors ($\vec{u}$).

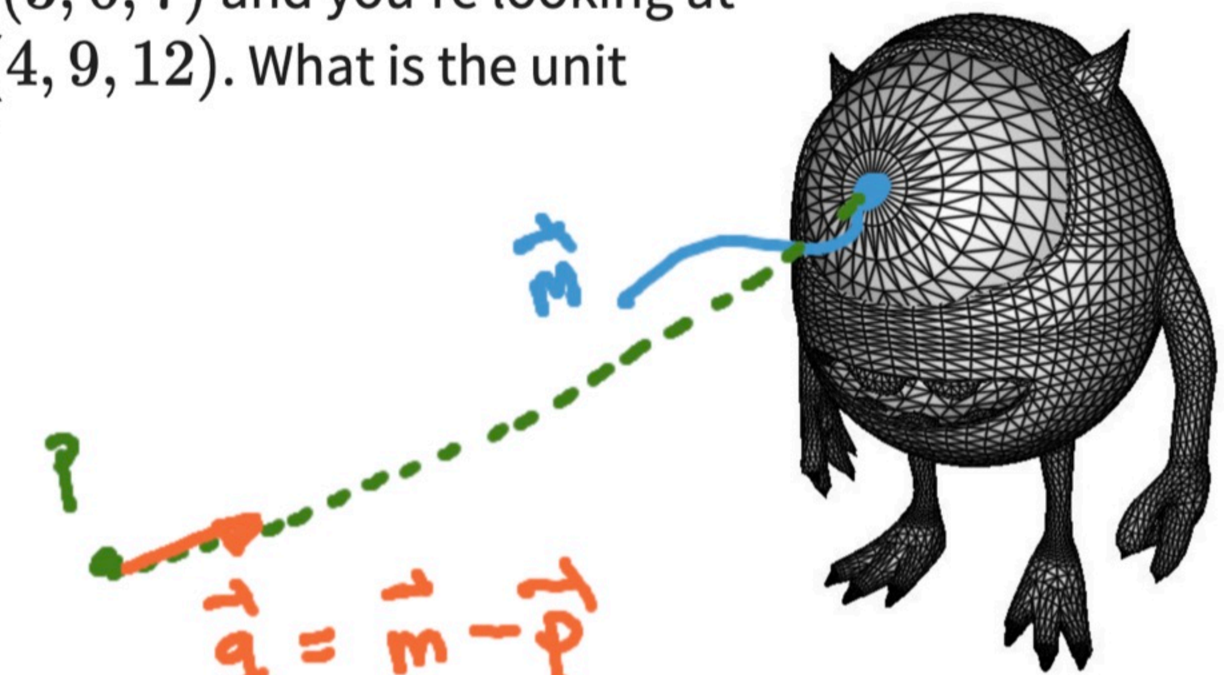


Vector addition, subtraction, scaling, unit vectors.



Example: calculating a unit gaze direction.

Imagine you're standing at a point $\vec{p} = (3, 6, 7)$ and you're looking at Mike Wazowski who is located at $\vec{m} = (4, 9, 12)$. What is the unit vector pointing in the direction of Mike?

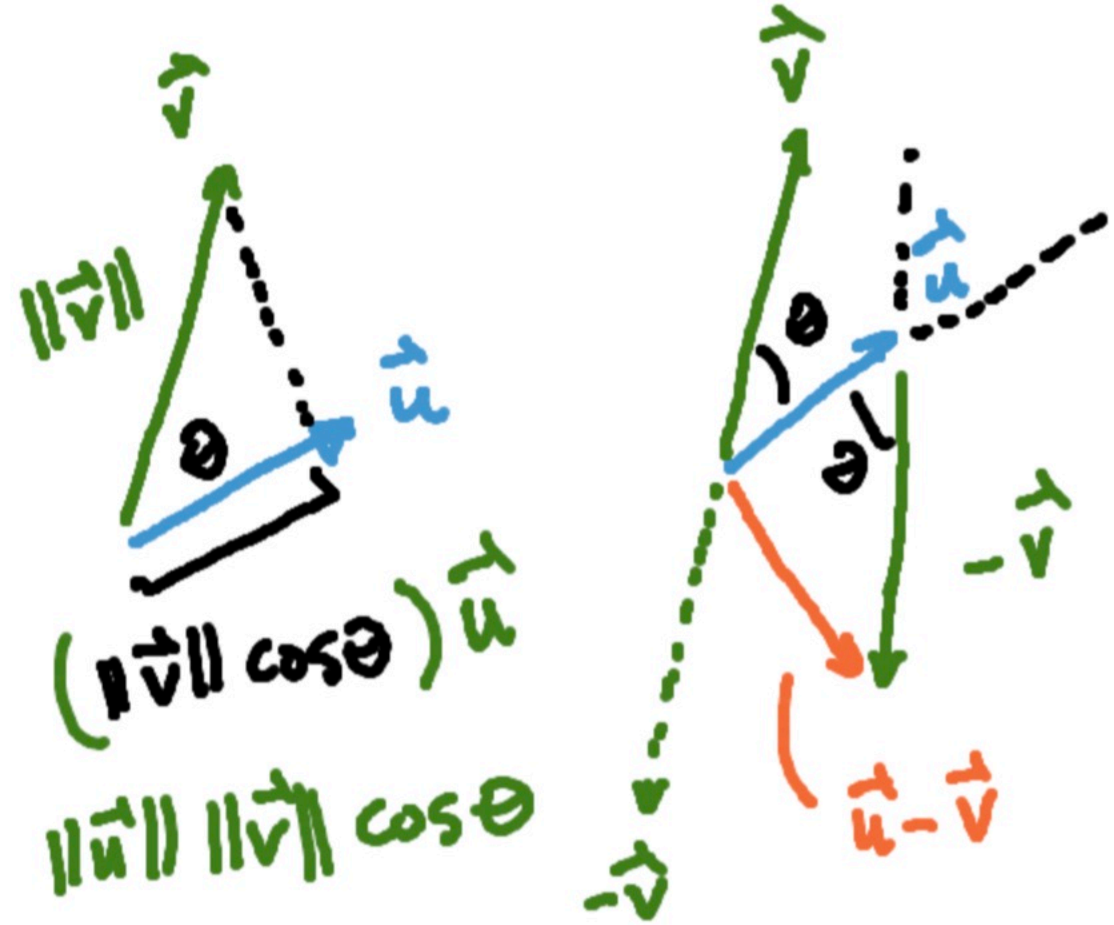


$$\vec{g} = \frac{\vec{m} - \vec{p}}{\|\vec{m} - \vec{p}\|}$$

$$\vec{g} = \frac{\begin{bmatrix} 4 \\ 9 \\ 12 \end{bmatrix} - \begin{bmatrix} 3 \\ 6 \\ 7 \end{bmatrix}}{\| \cdots \|} = \frac{\begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}}{\| \cdots \|} = \frac{\begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}}{\sqrt{1^2 + 3^2 + 5^2}} = \frac{\begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}}{\sqrt{35}}$$

Derivation of dot product in relation to angles.

$c^2 = a^2 + b^2 - 2ab\cos\alpha$



cosine law

$$\|\vec{u}-\vec{v}\|^2 = \|\vec{u}\|^2 + \|\vec{v}\|^2 - 2\|\vec{u}\|\|\vec{v}\|\cos\theta$$

$$\|u\|^2 + \|\vec{v}\|^2 - 2u_xv_x - 2u_yv_y - 2u_zv_z = \|\vec{u}\|^2 + \|\vec{v}\|^2 - 2\|\vec{u}\|\|\vec{v}\|\cos\theta$$

$$u_xv_x + u_yv_y + u_zv_z = \|\vec{u}\|\|\vec{v}\|\cos\theta$$

(dot product)

$$\vec{u}\cdot\vec{v} = \sum_i u_iv_i = \|\vec{u}\|\|\vec{v}\|\cos\theta$$

if $\theta = 90°$ $(\pi/2)$

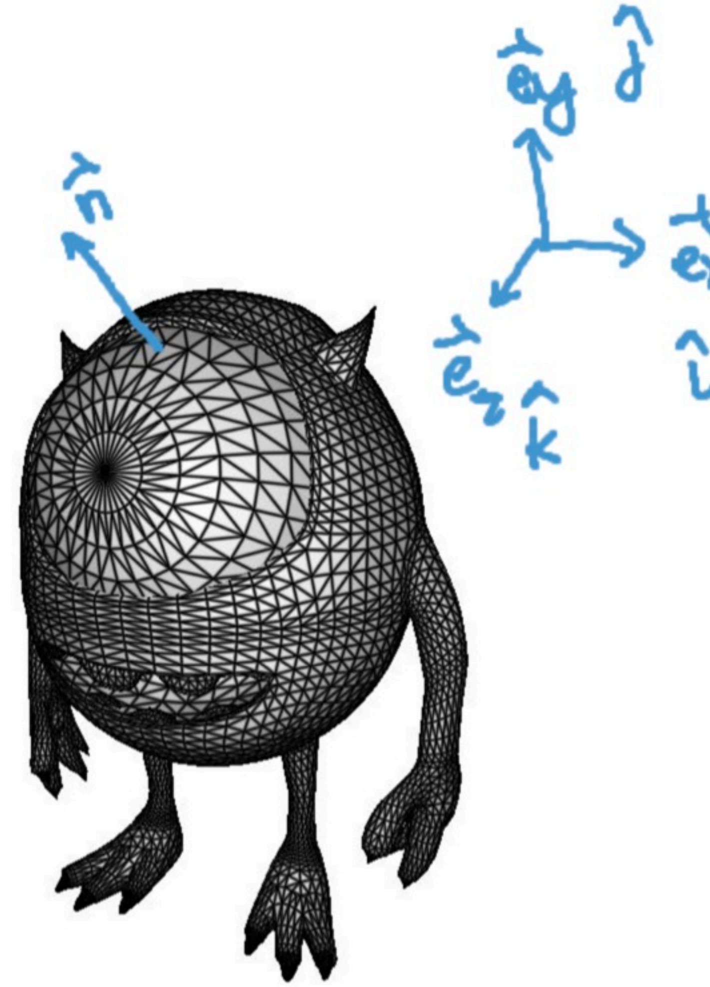
$\vec{u}\cdot\vec{v} = 0$ perpendicular/orthogonal.

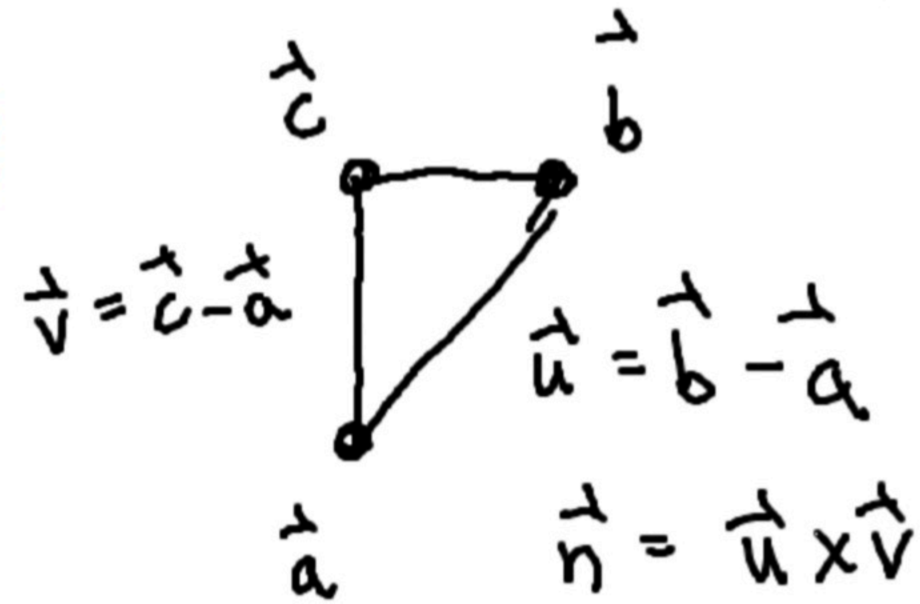
Definition of the cross product.

⊥ perpendicular.

$$\vec{u} \times \vec{v} = (u_y v_z - u_z v_y)\vec{e_x} - (u_x v_z - u_z v_x)\vec{e_y} + (u_x v_y - u_y v_x)\vec{e_z}$$

order matters!





area of a triangle $= \frac{1}{2} \| \vec{u} \times \vec{v} \|$

Exercise: show that $\vec{u} \cdot (\vec{u} \times \vec{v}) = 0$.

$$\vec{u} \times \vec{v} = (u_y v_z - u_z v_y)\vec{e}_x - (u_x v_z - u_z v_x)\vec{e}_y + (u_x v_y - u_y v_x)\vec{e}_z$$

Use:

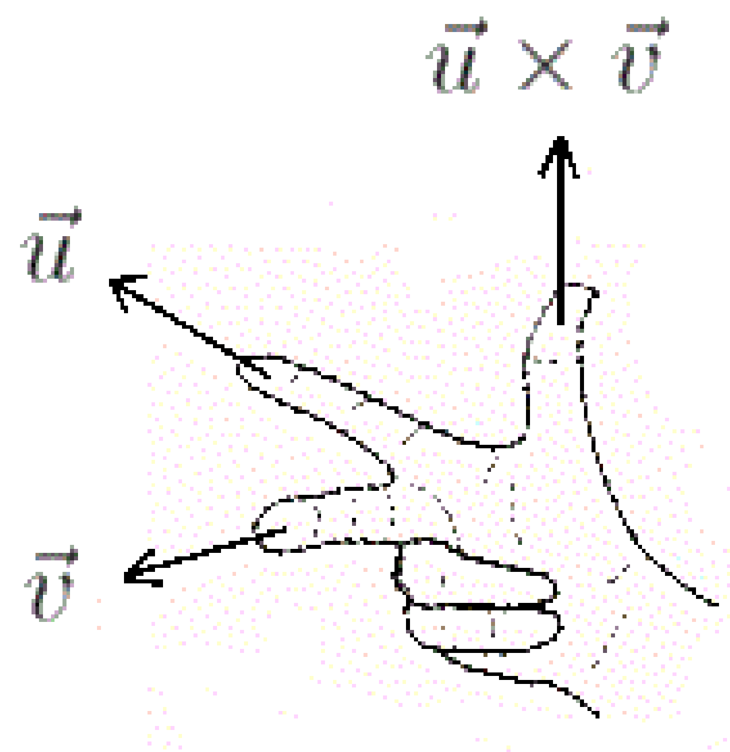
$$\vec{u} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad \vec{v} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

Solution: first we calculate $\vec{u} \times \vec{v} = \begin{bmatrix} 12 - 15 \\ -(6 - 12) \\ 5 - 8 \end{bmatrix} = \begin{bmatrix} -3 \\ 6 \\ -3 \end{bmatrix}$.

Then verify $\vec{u} \cdot (\vec{u} \times \vec{v}) = (1)(-3) + (2)(6) + (3)(-3) = -3 + 12 - 9 = 0$.

Visualizing the cross product.

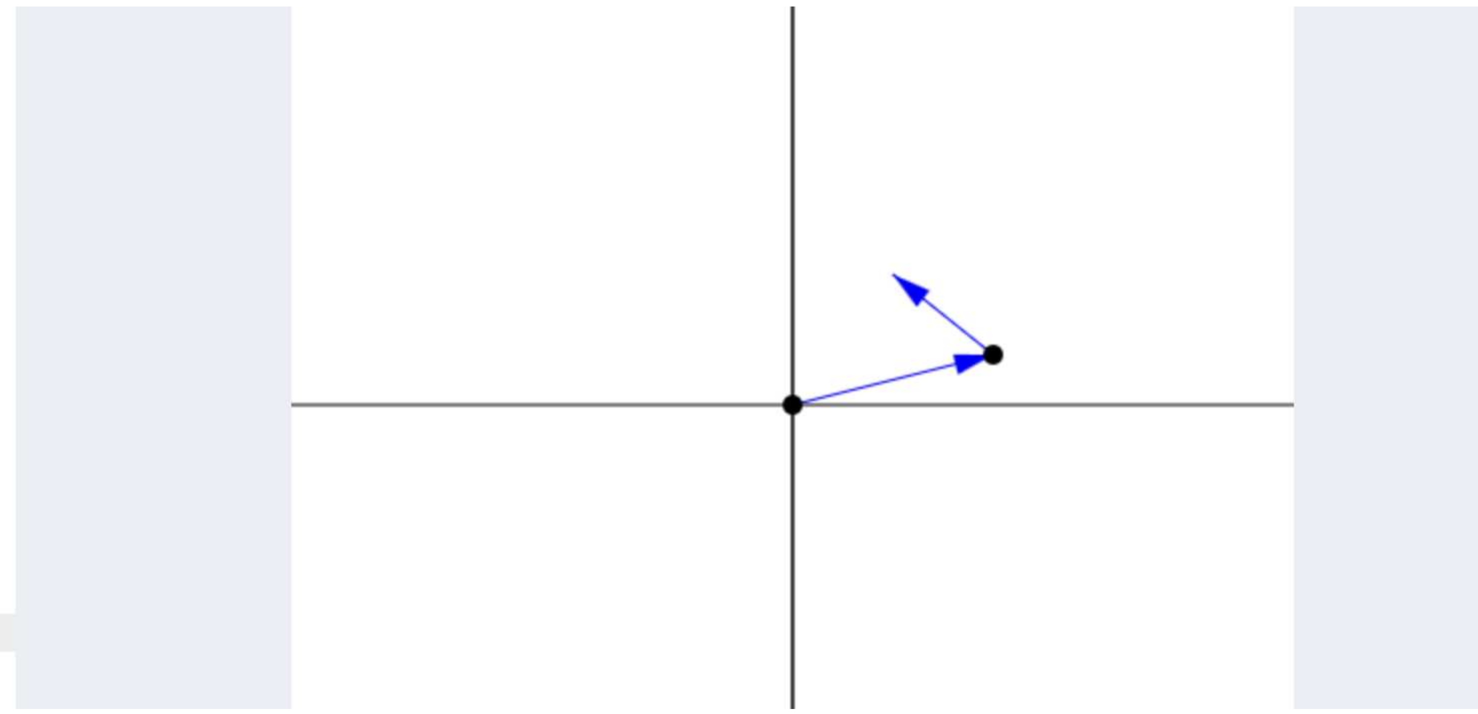
Using **RIGHT** hand: align index finger with $\vec{u}$, then align middle finger (or all other fingers) with $\vec{v}$ (moving towards your palm). Thumb will point in direction of $\vec{u} \times \vec{v}$.



Exercises: using `glMatrix`.

Look up documentation at: https://glmatrix.net/docs/.

```

const origin = vec2.fromValues(0, 0);
const u = vec2.fromValues(100, 25);
const v = vec2.fromValues(-50, 40);

const x0 = vec2.add(vec2.create(), origin, u);

// array of points to plot as dots
let points = [origin, x0];

// array of vectors to plot
let vectors = [u, v];

// array of vector tails (optional)
// set an entry to 'undefined' to use the origin
let tails = [undefined, points[1]];

```

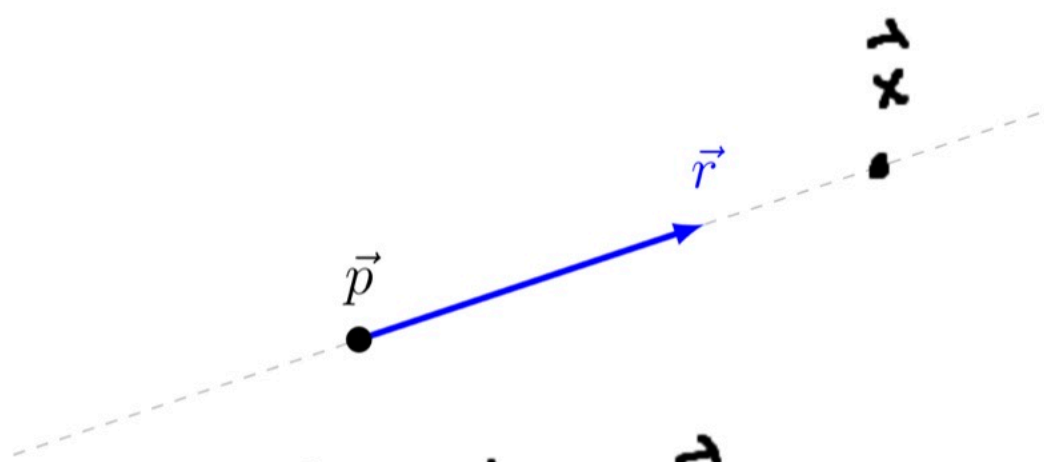


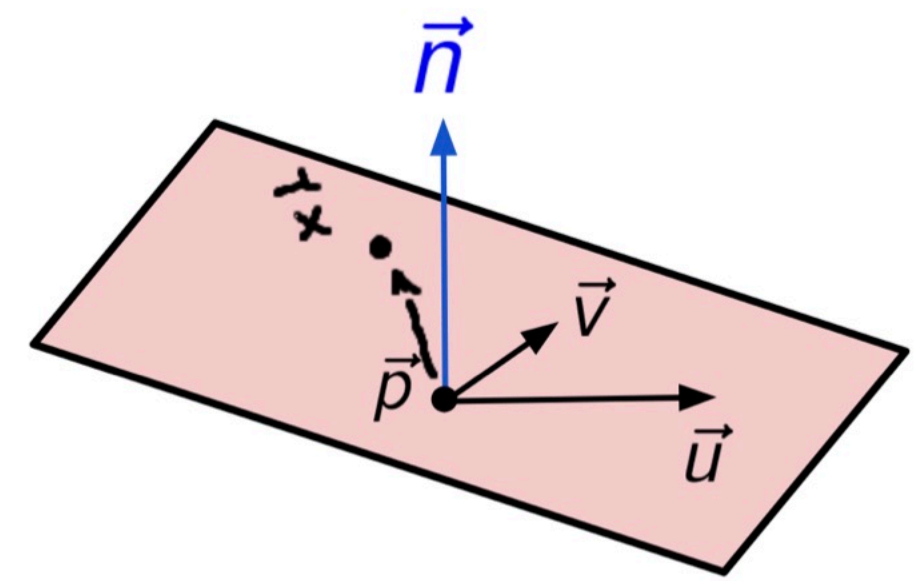
- Calculate and visualize the vector $\vec{w}_1 = \vec{u} + \vec{v}$. Use `console.log` to print the result, and verify (on paper).
- Calculate and plot $\vec{x}_1 = \vec{x}_0 + \vec{v}$.
- Visualize $-\vec{v}$.
- Calculate and visualize the vector $\vec{w}_2 = \vec{u} - \vec{v}$.
- Calculate the length $\ell = ||\vec{w}_2||$ and use `console.log` to print the result.
- Normalize $\vec{w}_2$ to produce $\vec{w}_3 = \frac{1}{\ell}\vec{w}_2$.
- Use $\vec{w}_3$ to plot the end of the vector $\vec{u} - \vec{v}$ using `vec2.scaleAndAdd`\`.
- Calculate and visualize a vector perpendicular to $\vec{w}_1$ and use `.dot` to verify the vector is indeed perpendicular to `w1` (use `console.log` to print the result).

Possible implementation.

```
const origin = vec2.fromValues(0, 0);
const u = vec2.fromValues(100, 25);
const v = vec2.fromValues(-50, 40);

const x0 = vec2.add(vec2.create(), origin, u);

const w1 = vec2.add(vec2.create(), u, v);
const x1 = vec2.add(vec2.create(), x0, v);
const mv = vec2.negate(vec2.create(), v);
const w2 = vec2.subtract(vec2.create(), u, v);

const l = vec2.length(w2);
console.log(l);
const w3 = vec2.normalize(vec2.create(), w2);
console.log(w3);
console.log(vec2.scale(vec2.create(), w2, 1.0/l));
const x2 = vec2.scaleAndAdd(vec2.create(), origin, w3, l);

const n = vec3.cross(vec3.create(), vec3.fromValues(0, 0, 1), vec3.fromValues(w1[0], w1[1], 0));

console.log(vec2.dot(n, w1));

let points = [origin, x0, x1, x2];
let vectors = [u, v, w1, mv, w2, n];
let tails = [undefined, points[1], undefined, x0];
```

Representing lines and planes.





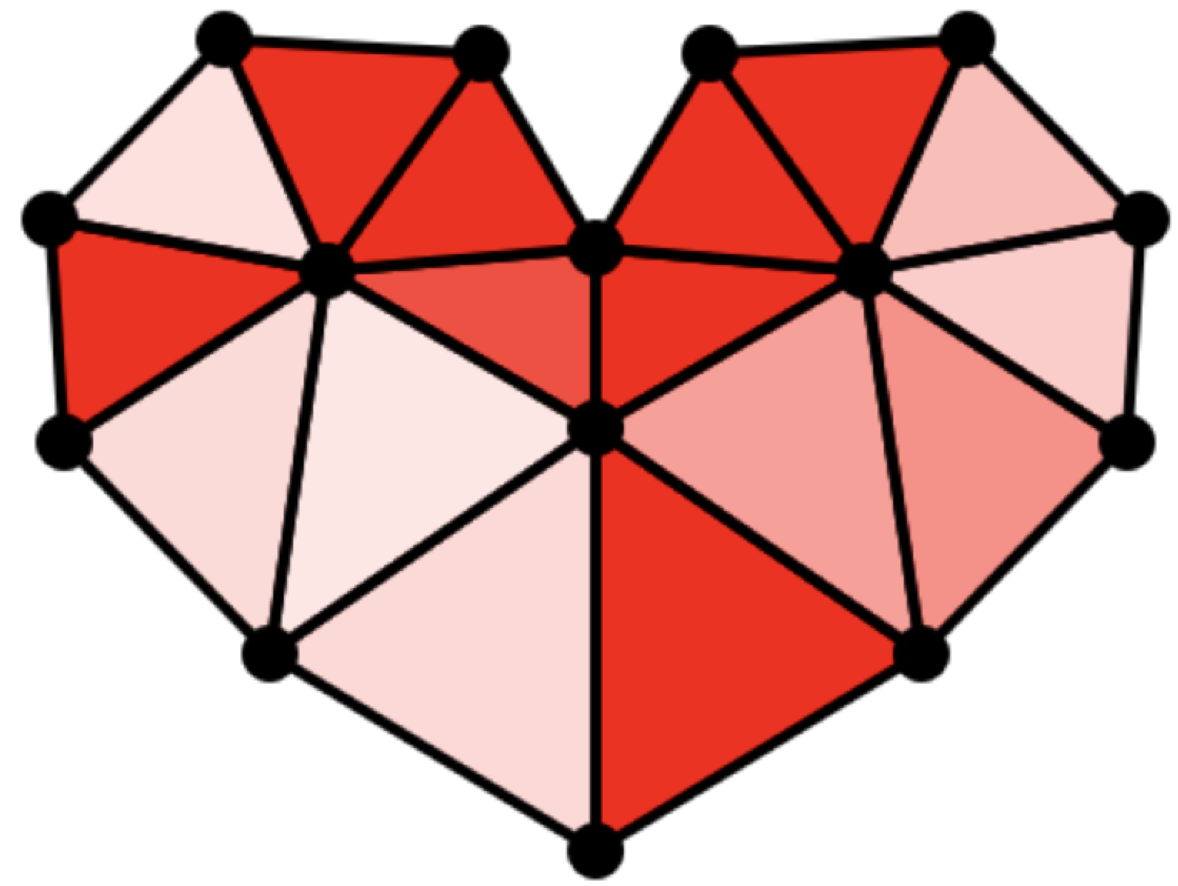
$$\vec{x} = \vec{p} + t\vec{r}$$

$$t \in \mathbb{R}$$

$$\vec{x} = \vec{p} + s\vec{u} + t\vec{v}$$

$$\vec{n} \cdot \underbrace{(\vec{x} - \vec{p})}_{\text{vector in plane}} = 0$$

Back to our heart example.



- **Left-half of the room:** find the area of the **left** side of the heart.
- **Right-half of the room:** find the area of the **right** side of the heart.
- Split up the work and add up the result to estimate the *total* area of the heart.
- *Hint:* use the cross product. Hover over the dots in the notes to get triangle vertex coordinates (doesn't need to be exact).

Summary

- Dot product will be useful for calculating diffuse component of lighting model.
- Cross product useful for calculating perpendicular vectors (and areas).
- Intersections useful for figuring out what we can see (and also what is in a shadow).
- Please try out the `example` assignment on the Setup page.
- **Office hours :** (please come say hi!)
  - Tuesdays: 2pm - 3:30pm
  - Thursdays: 2pm - 3:30pm
  - Fridays: 11am - 12pm
- Next class: **matrices**.