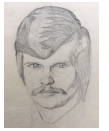
SIMPLE COMBINATIONS OF POINTS AND VECTORS



Please… focus

Silence / **close all sources of distraction** and apps not related to this class

✍ Take lots of notes and drawings by hand on paper

This requires summarization, helps understanding, reveals confusion, helps memorization

? Ask lots of questions, offer your answers, and share your comments

Do not be afraid, I like weird questions and wrong answers

Participate actively in individual and team exercises

Practice working with others

Review your notes and understanding and be ready to take a pop quiz

Q&As

Click “Raise Hand” in BlueJeans

Feel free to share a whiteboard or image.

Errors

I will make mistakes

- Some will be deliberate to see whether you are paying attention
  - Please watch out for them and point them out (raise a virtual hand)
- Some will be typos or genuine errors
  - Please watch out for these and point them out (raise a virtual hand)
  - If you are not sure, ask anyway
    - A nice way to ask is "I understand that …, but I do not understand why…"

Learning Objectives

- Understand the semantic difference between **points** and **vectors**
- Understand how to **invent**, **write**, and **implement expressions** that compute new points or vectors from known ones
- Learn the **notation convention** that I will use for points, vectors, and operators on them
- Learn how to use a few simple **Processing functions** (provided) that create new points and vectors from others, that measure angles or distances, or that displays them.

Motivation: Geometric primitives

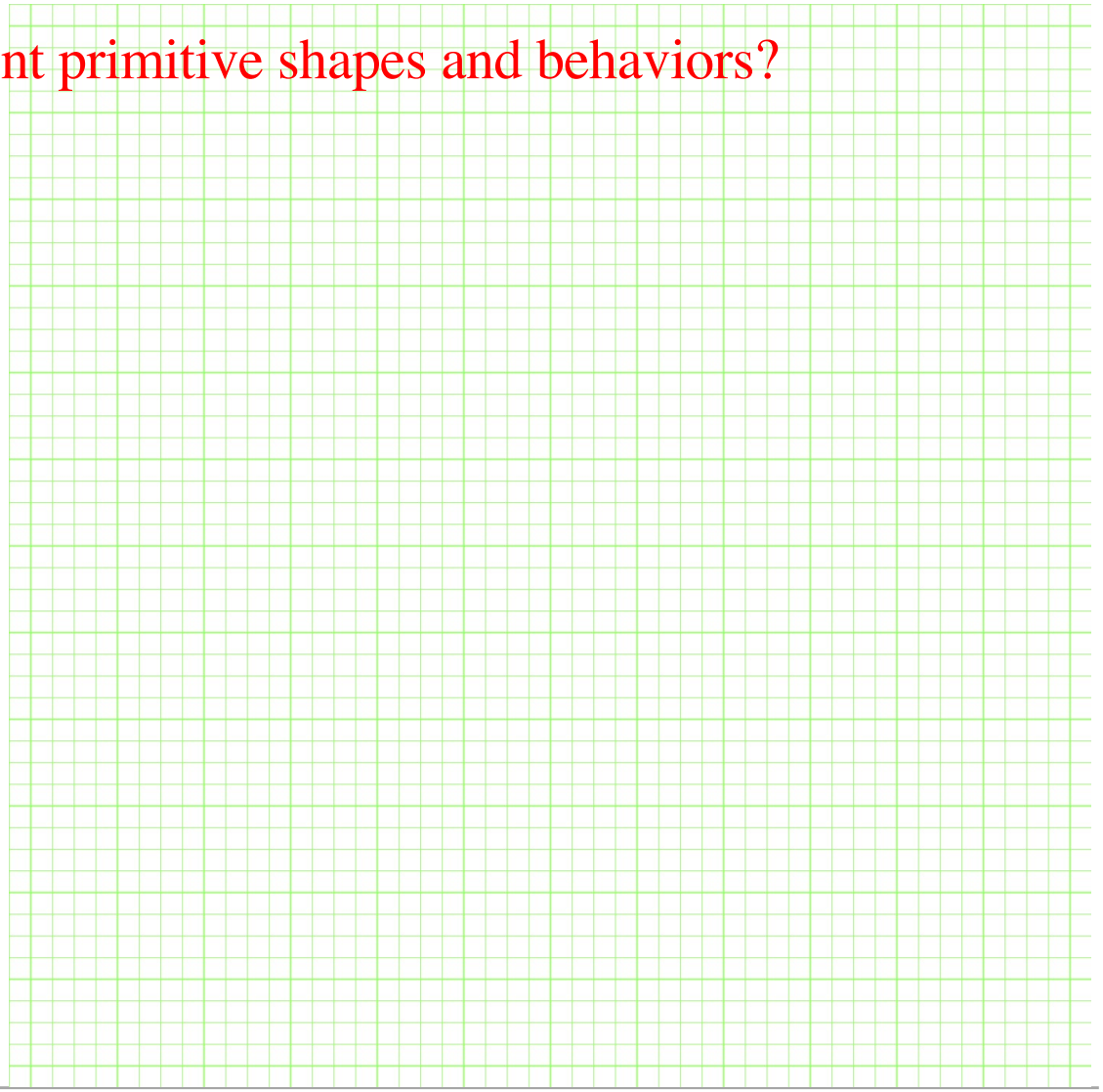
In the first part of this course, we will first focus on **planar** things:

shapes, **animations**, **user-interaction**

We will discuss primitive **shapes** (circles, edges, triangles, simple curves, rays) and primitive **behaviors** (motions, morphs, warps)

What **mathematical concepts** (abstractions) do we need to represent them?

What **concepts** do we need to represent primitive shapes and behaviors?



Primitive geometric abstractions in 2D

We will use

- **Scalar**: real number representing distance d, time t, area a
- **Point**: location on the canvas of center C of a circle or the vertex P of an edge
- **Vector**: velocity V, displacement AB that brings point A to point B, force W
- **Arrow**: situated vector <P,V> representing velocity V at a point P (*NEW*)
- **Map**: transformation that moves (translate, rotate, dilate) points, vectors, arrows
- **Frame**: a representation of a map (for ex: <P,U,V>)

It is essential that you internalize these concepts and never confuse them.

Explain the difference between a **vector** and a **vertex** of an edge

Explain the difference between a **point** and a **vector**

List and draw the 6 primitive geometric abstractions in 2D that we will deal with

-
-
-
-
-
-

Others?

-
-
-
-

What is the relation between a **frame** and an **arrow**?

-
-

Scalar (real or integer)

Real numbers: 0, .5 -4.5, a, b, c, d, x, y, u, v, w, s, t, π, α (lowercase *not in italics*)

Floating point numbers may represent:

- Time values
- Distances, lengths, magitudes
- Areas
- Cartesian coordinates: x, y
- Polar Coordinates: magnitude m, angle w

Integers: 3, i, j, k, n

- May represent indices, subscripts, whole powers of transformations, rational numbers (2/3)

Operators:

- Arithmetic: st+u, $3x^2+4$, (t-a)/(c-b)
- Exponent: $\mathbf{R}^x$
- Modulo: i%2 is 0 or 1

Understanding "modulo": Write 0/1 toggle using module

Assume i is either 0 or 1

Consider the function t()

int t(int i) {if(i==0) return 1; else return 0;}

Write it using % without the if()

int t(int i) {return ;}

What is a point? (Write a formal definition)

Point

A location on the plane of the canvas (may be outside of the canvas)

We will **visualize** it as small **disk** around that location

- Filled or
- With the letter name of that point

Math **notation**:

- **Uppercase** letters
- Typically we will use letters: A, B, C, D, E, F, G, O, P, Q
- Sometimes with subscript indices P_i

Distinguish the concept from its representation

Point (**concept**) is a location

There are many possible and used **representations** of a point:

- Cartesian coordinates wrt some **frame**
- Polar coordinates wrt some **frame**
- **Intersection** of two given lines or curves (possible issue of uniqueness!)
- A **parametrized curve** C(t) and a particular value t_0 of that parameter
- The place where two moving objects will touch at first collision
- The left-most vertex of a polygon

What is a point? (Write a formal definition)

Vectors

Displacement:

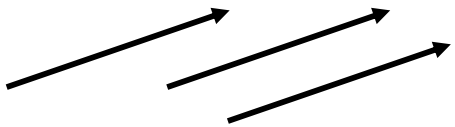
- Often **between two points** or per unit of time (velocity)
- Other uses: direction (tangent, normal), acceleration, spring force

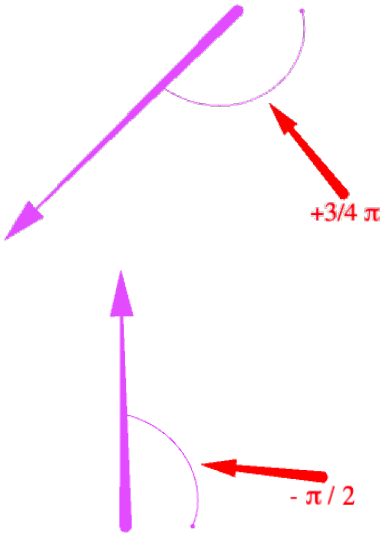
Math notation:

- Uppercase letters (notice that I do not use overhead **arrow** : $\vec{V}$)
- Typically I, J, K, N, T, U, V
- Sometimes with subscript indices V_i

Is **not situated**:

- Has a length ("**magnitude**") and
- Has a **direction**: angle $V^{\angle}$ measured **clockwise** from X-axis
- Can be drawn anywhere on the plane (same vector)





What is the difference between a point and a vector?

Write your answers:

A point represents a :

A vector represents a :

How do we distinguish between points and vectors?

In text or formulae:

From the context (I do not use overhead arrows, except in some papers)

In code:

Because they are variables of different types

Idea: We could have the convention that all point-variable start with P and all vector variables start with V

Q&As

Click “Raise Hand” in BlueJeans

Feel free to share a whiteboard or image.

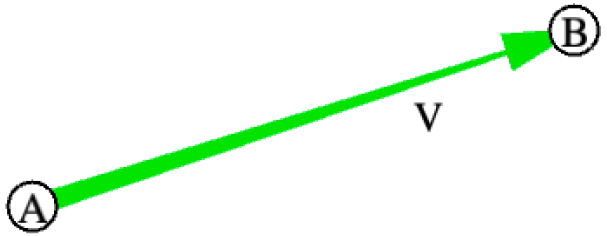
Creating vectors from points and vice versa

AB denotes a **vector** that takes point A to point B

Difference-of-points expression

A+V denotes a **point** obtained by translating A by V

Point+vector expression



Quiz: What is A + AB ?

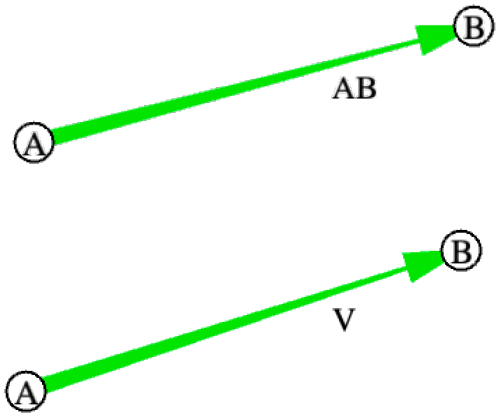
-
-
-
-
-
-
-
-

Convenient but semantically incorrect notation

Some colleagues write B – A instead of AB

This difference-of-points notation is semantically *incorrect*

If A is the center of Amsterdam and B the center of Brussels,
What could B – A mean?



But it is **convenient** and sometimes intuitive

If V = B – A, then A + V = A + B – A = A

I try to avoid it (often), and so should you, for clarity (unless you are trying to simplify an expression or make symmetry apparent).

Why does this incorrect notation actually work?

If you represent points and vectors by their Cartesian coordinates:

If A = (A.x, A.y) and B = (B.x, B.y),

then AB = < B.x–A.x , B.y–A.y >

Notice that I use (x,y) for points and <x,y> for vectors to emphasize the difference

Hence, it makes sense to use B – A for AB

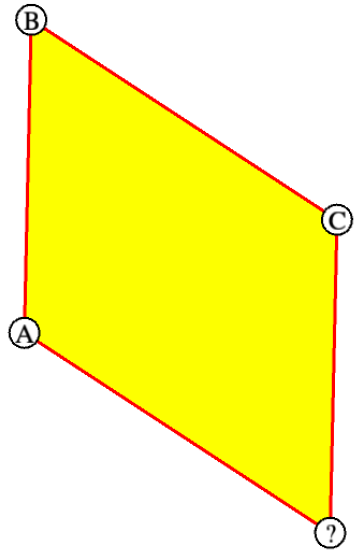
Notice that the order of the letters in B-A and in AB are reversed

Challenge

Write the expression for the fourth point of a parallelogram given 3 of its corners:
A, B, C

Write it using the proper point+vector notation:
D =

Write it now using the incorrect notation:
D =



Implementing this is PART 1 of your first project.
How hard can that be?

D = ?

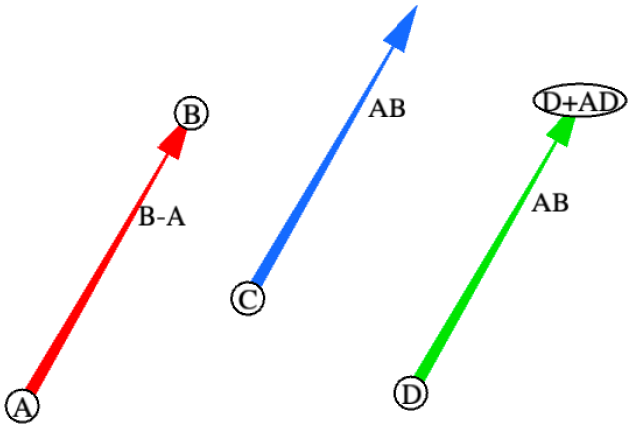
Q&As

Click “Raise Hand” in BlueJeans

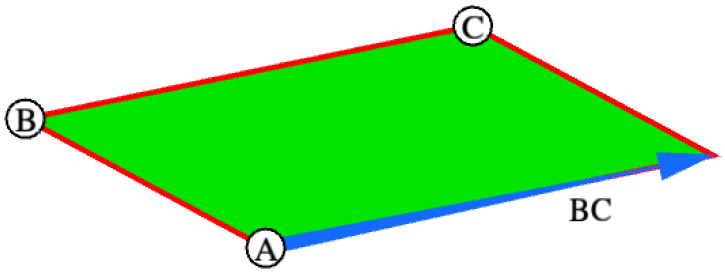
Feel free to share a whiteboard or image.

Implementation and display of points and vectors

I have implemented functions for creating, using, and displaying points and vectors.
I will show you how to use them when we look at the base code that I gave you.




```
VCT AB = V(A,B);
show(A,AB,red,"B-A");
show(C,AB,blue,"AB");
PNT P = P(D,AB);
show(D,AB,green,"AB");
circledLabel(P,"D+AD");
```

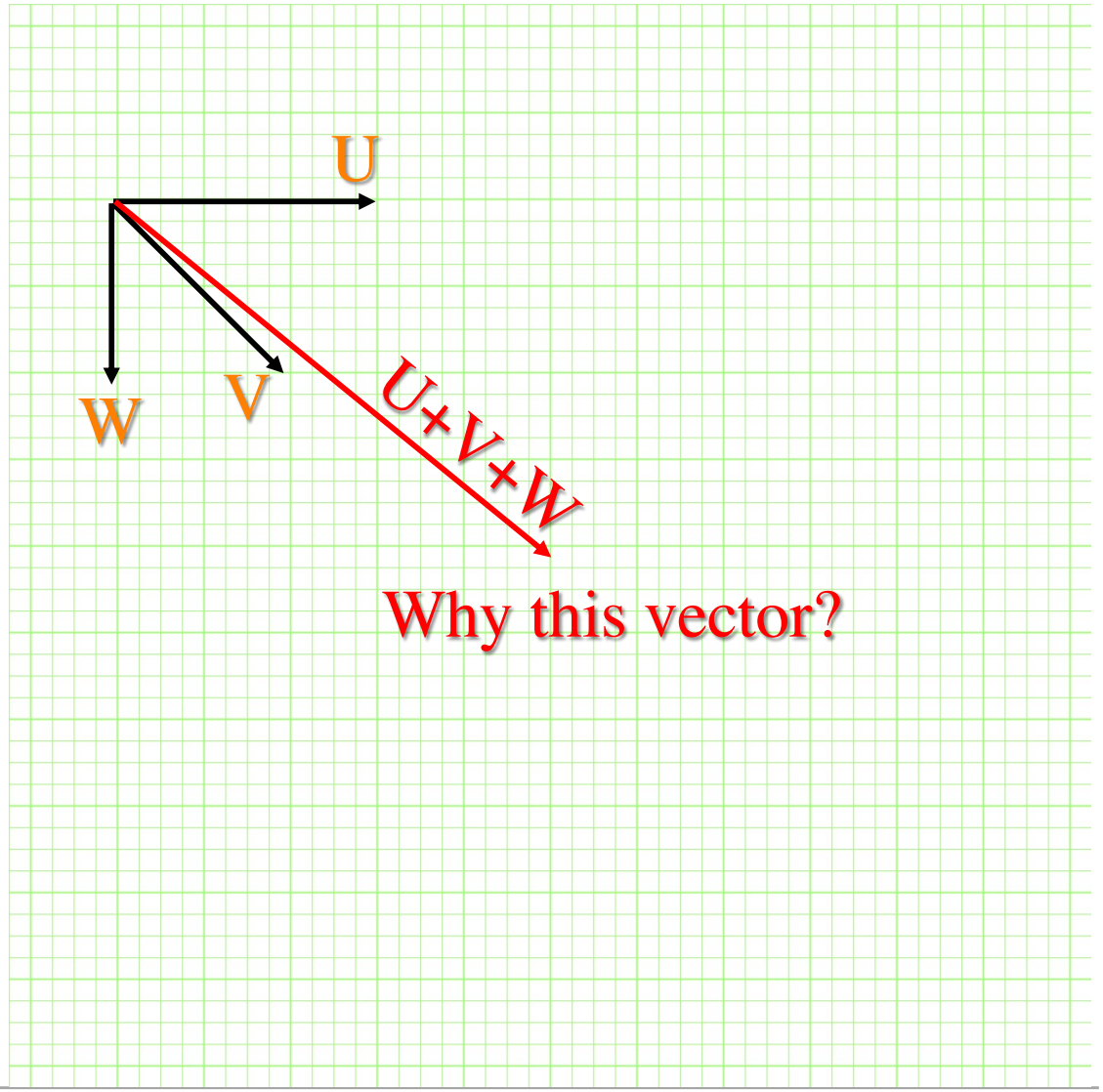



```
cwf(red,4,green);
showLoop(A,B,C,P(A,V(B,C)));
show(A,V(B,C),blue,"BC");
```

Adding vectors

Given vectors U, V, and W, you can create new vectors as linear combinations of these, such as:

- U+V+W



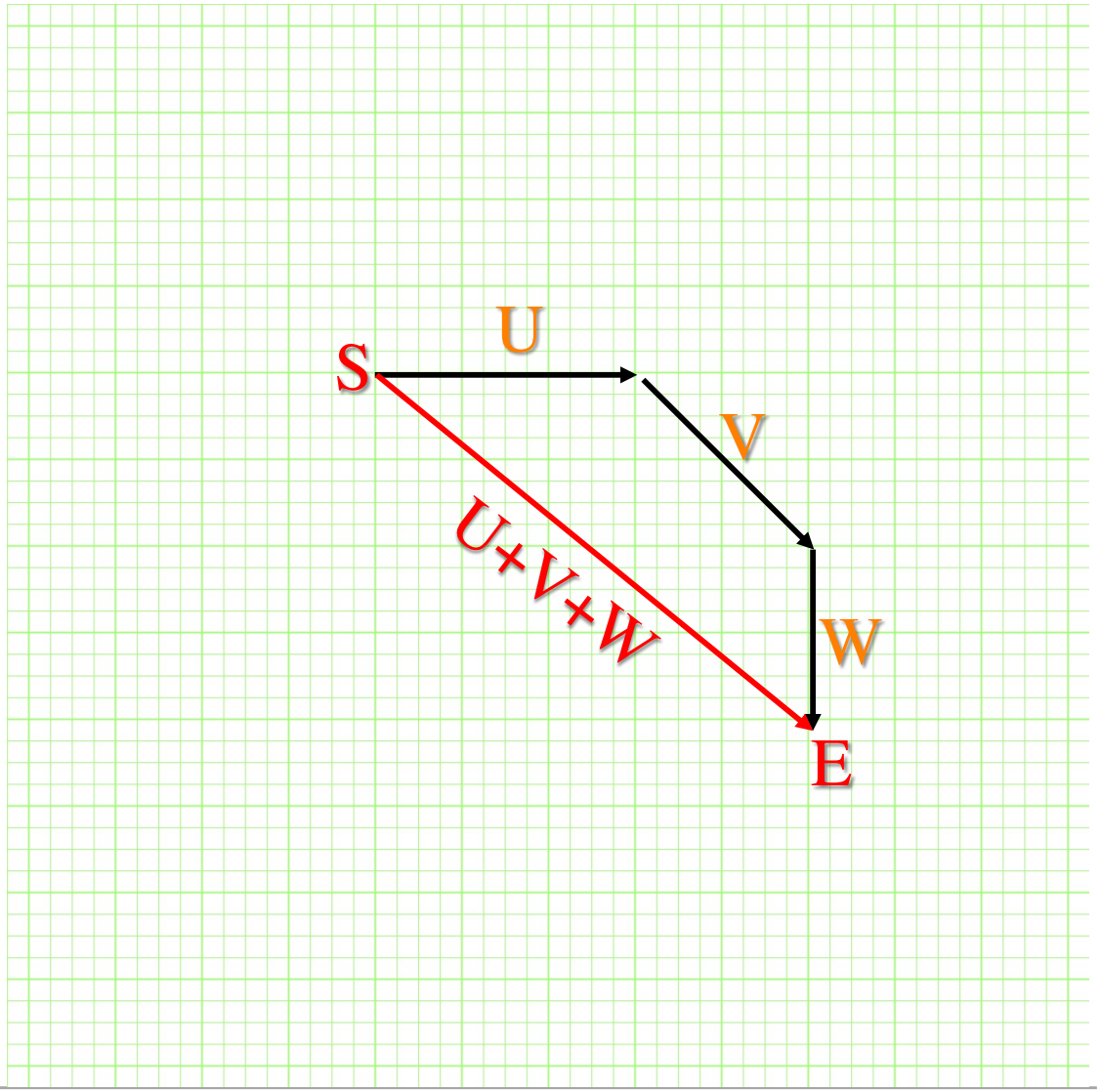
Adding vectors

Given vectors U, V, and W, you can create new vectors as linear combinations of these, such as:

- U+V+W

By making a chain in any order, starting from any point.

The result is the vector from the start point S to the end point E

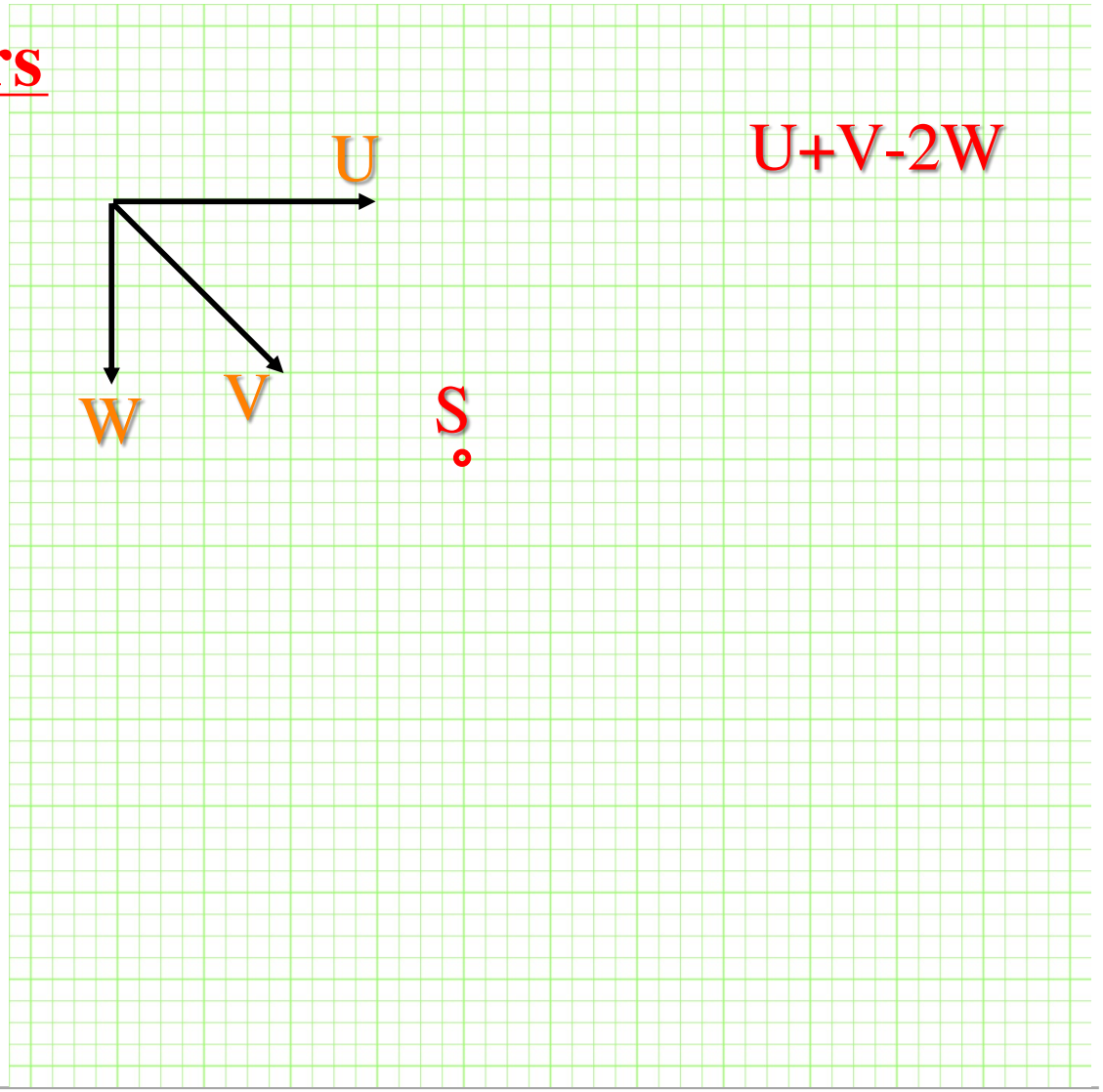


Scaling and adding vectors

Scaling a vector scales it length
2V, 1.5V, -3V, -V

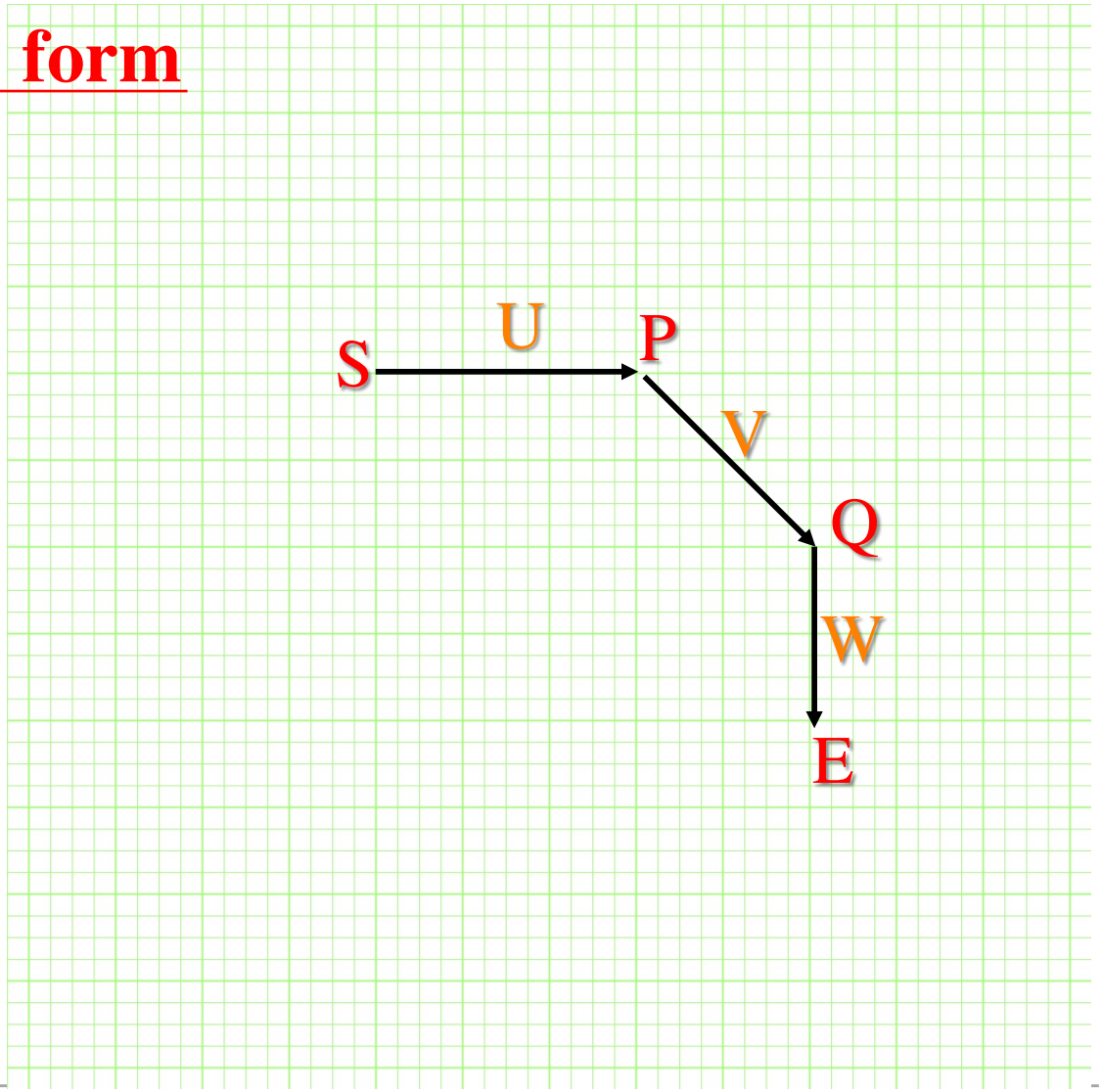
I sometimes abuse and write
V/2 for ½V

Given vectors U, V, and W,
draw U+V-2W as an arrow,
starting from point S



Simplifying vectors in PQ form

Can you simplify SP+PQ+QE?

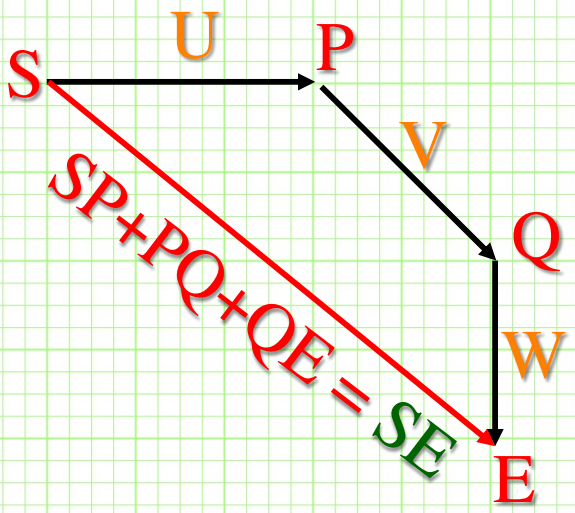


Simplifying vectors in PQ form

Can you simplify SP+PQ+QE?

It simplifies to SE

Why?



Simplifying vectors in PQ form

Can you simplify SP+PQ+QE?

It simplifies to SE

Why?

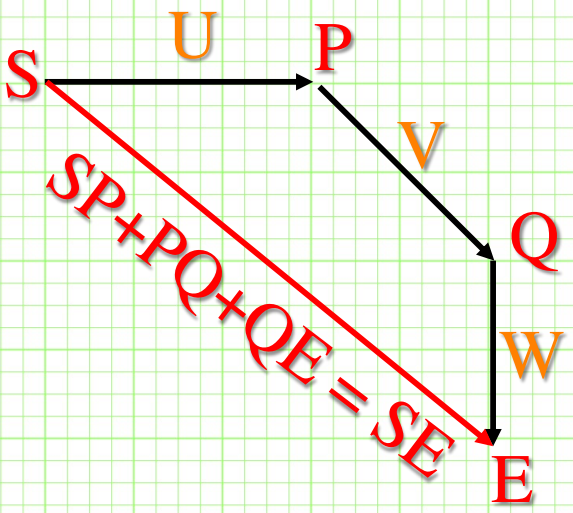
SP = P – S

SP+PQ+QE = P–S+Q–P+E–Q

= ~~P~~–S+~~Q~~–~~P~~+E–~~Q~~

= E–S

= SE



Q&As

Click “Raise Hand” in BlueJeans

Feel free to share a whiteboard or image.

Affine combination (weighted-sum) of points

What does P–S+Q–P+E–Q represent?

What does A + B – A represent?

Weighted-sum-of-points

What does P–S+Q–P+E–Q represent?

- A vector

What does A + B – A represent?

- A point

Weighted-sum-of-points

What does P–S+Q–P+E–Q represent?

- A vector

What does A + B – A represent?

- A point

What is going on?

- Is any linear combination of points valid?
- Which ones yield a point?
- Which ones yield a vector?

Weighted-sum-of-points

Is any **linear combination** of points valid?

- **No**, only those that yield a point or a vector!

Which ones yield a **point**?

- Those for which the coefficients sum up to 1, I call them **average**-of-points
  - A + C – B = A + BC

Which ones yield a **vector**?

- Those for which the coefficients sum up to 0, I call them **difference**-of-points
  - P–S+Q–P+E–Q = SE

Q&As

Click “Raise Hand” in BlueJeans

Feel free to share a whiteboard or image.

What have we learned so far?

- The difference between **points** and **vectors**
  - Point = , vector =
- How to create (write) the **vector between two points**
  - Two methods: or
- How to create a vector as a linear combination of vectors
  -
- How to create **points as point+vector**
  -
- Which linear combinations of points are valid
  -
- Which linear combinations of points create a vector and which create a point
  -
  -

Q&As

Click “Raise Hand” in BlueJeans

Feel free to share a whiteboard or image.

Two more things: Rotation and magnitude

To get a vector obtained by **rotating** a vector V by w, write $V^{o}w$

Nonstandard notation

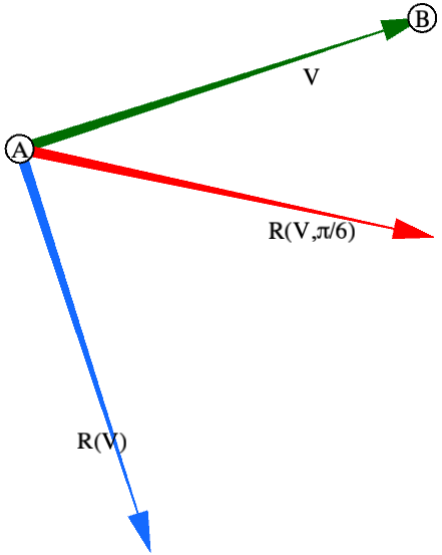
Short cut: V^{o} stands for $V^{o}(\pi/2)$, rotate 90 degrees **clockwise**

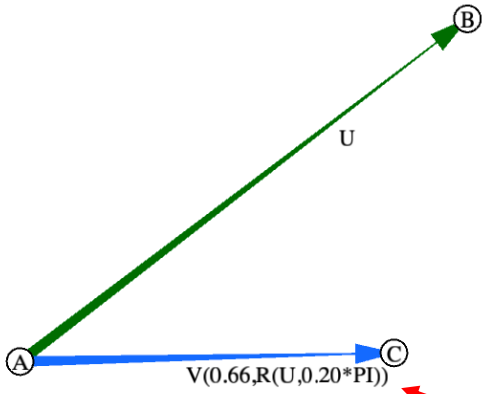
To obtain the scalar **magnitude** (i.e. length) of vector V, write $|V|$

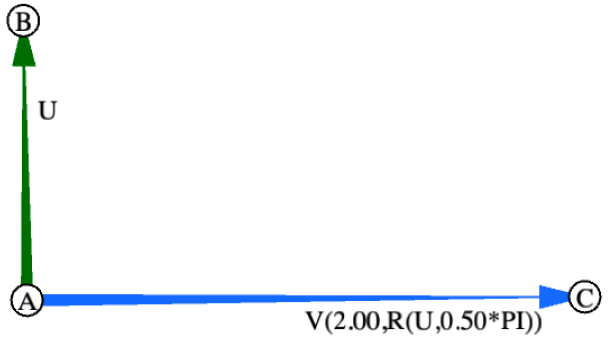
Nonstandard, many use $\|V\|$

Rotation, angle and magnitude

```
cwf(red,4,green);
VCT V = V(A,B);
show(A,V,dgreen,"V");
VCT R = R(V);
show(A,R,blue,"R(V)");
VCT W = R(V,PI/6);
show(A,W,dred,"R(V,π/6)");
```







Write on canvas code for the blue vector

```
VCT U = V(A,B);
VCT V = V(A,C);
show(A,U,dgreen,"U");
float w = angle(U,V);
float m = n(V)/n(U);
show(A,V,dred,"V("+nf(m,1,2)+",R(U,"+nf(w/PI,1,2)+" π))");
VCT W = V(m,R(U,w));
show(A,V,blue);
```

Q&As

Click “Raise Hand” in BlueJeans

Feel free to share a whiteboard or image.

Distance between points?

How would you obtain/write the distance between points A and B?

Using previously discussed operators

Distance between points?

How would you obtain the distance between points A and B?

Using previously discussed operators

|AB|

Sometimes I will write it as d(A,B)

Others may write it as || B – A ||

Complimentary Processing functions provided (1/4)

Name	Code	Math	Meaning
Vector reference	VCT V;	V	Declares **pointer** to yet unknown VCT object
Copy of vector	W=V(U);	W=U	Makes a **copy** of object U and set W to point to it
Coordinates V.x, V.y	V.x, V.y		in pixels, x=right, y=down
Vector from coord	V(u,v)	<u,v>	Makes VCT objects with coordinates <u,v>
Vector rotated 90°	R(U)	U°	Makes VCT version of U rotated 90° cw
Vector rotated w	R(U,w)	U°w	Makes VCT version of U rotated cw by w radians
Inversed vector	M(V)	-V	Makes opposite vector
Scaled vector	V(s,U)	sU	Makes new VCT sU, does NOT scale U
Vector sum V(U,W)	U+W		Makes VCT equal to sum
Scaled vector sum	V(U,t,W)	U+tW	Makes VCT U + (W scaled by t)
Scaled vector sum	V(s,U,t,W)	sU+tW	Makes VCT sU+tW
Divided vector	S(1./d,U)	U/d	Makes new VCT U/d
Norm of vector	n(U)	|U|	Returns scalar magnitude (length) of vector U
Normalized vector	U(V)	$\underline{V}$	Makes new VCT V/|V|, does not change V
Angle of vectors	angle(U,W)	U^W	angle in [-π, π] cw from direction of U to the one of W

Q&As

Click “Raise Hand” in BlueJeans

Feel free to share a whiteboard or image.

Complimentary Processing functions provided (2/4)

Name	Code	Math	Meaning
Vector LERP	L(U,t,W)	L(U,t,W)	Makes VCT as linear interpolation (1-t)U+tW
Vector LPM	S(U,t,W)	S(U,t,W)	Makes Log Polar Morph m^tU° (tw) with m=|W|/|U|, w=U^W
Dot product dot(U,W)	U•W		scalar |U| |W| **cos**(U^W)
Det product det(U,W)	U:W		scalar |U| |W| **sin**(U^W) = U° •V
Point pointer	PNT Q;	Q	Makes pointer to yet unknown PNT object
Coordinates Q.x, Q.y	Q.x, Q.y		in pixels, dx=right, dy=down
Make point PNT	Q=P(x,y);		Makes PNT object (x,y) and PNT pointer Q to it
Point object P(u,v)		(u,v)	Makes PNT objects with coordinates (u,v)
Change point	A.setTo(B);		Sets coordinates of A to those of B
Copy of point	Q=P(R);		Makes a copy of object R and set Q to point to it
Vector between pts	V(A,B)	AB	Makes VCT object "B-A" from A to B
Normalized vector	U(A,B)	$\underline{AB}$	Returns $\underline{AB}/\|\underline{AB}\|$, unit vector from A towards B
Point+vector	P(A,U)	A+U	Makes PNT A+U
Point+s*vector	P(A,s,U)	A+sU	Makes PNT A+sU

Q&As

Click “Raise Hand” in BlueJeans

Feel free to share a whiteboard or image.

Complimentary Processing functions provided (3/4)

Name	Code Math	Meaning
Move by vector	A.translate(V);	Map, changes point A to A+V
Dilate from F	A.dilateWrtPNT(s,F);	Map, changes point A to F+sFA
Rotate about F	A.rotateWrtPNT(w,F);	Map, changes point A to F+FA°w
Point LERP L(A,t,B)	L(A,t,B)	Returns A+tAB
Timed LERP	L(a,A,b,B,t)	Returns A+(t-a)/(b-a)AB
Mouse	Mouse()	Returns PNT for current mouse location
Previous mouse	PMouse()	Returns PNT for mouse location at previous frame
Screen center	ScreenCenter()	Returns PNT for center of the screen

Q&As

Click “Raise Hand” in BlueJeans

Feel free to share a whiteboard or image.

Complimentary Processing functions provided (4/4)

Name	Code	Meaning
Label point	A.writeLabel("A");	Displays “A” in an ellipse
Draw circle	show(C,r);	Draws (filled) circle with center C and radius r
Draw edge	show(A,B);	Draws edge [A,B]
Draw hat	show(A,B,C);	Draws edges [A,B] & [B,C]
Draw triangle	showLoop(A,B,C);	Draws triangle [A,B,C]
Draw quad	showLoop(A,B,C,D);	Draws quad [A,B,C,D]
Draw segments	showPolyline(A,B,C);	Draws [A,B] & [B,C]
Draw segments	showPolyline(A,B,C,D);	Draws [A,B] & [B,C] & [C,D]
Draw as *arrow* PV	show(P,V);	Draws V as arrow starting from point P
Draw as *arrow* PsV	show(P,s,V);	Draws sV as arrow starting from point P
Draw as *arrow* PVcS	show(P,V,c,S);	Draws V as arrow from P with color c & text S

Q&As

Click “Raise Hand” in BlueJeans

Feel free to share a whiteboard or image.

What we did not cover yet

Arrows

The code "Draw arrow PV: show(P,V);" uses a point and a vector, not an arrows

Maps

Well, a rotation (remember **V**°w ?) is an example of a map

Can you think of other simple maps?

Frames

These are local coordinate systems

Q&As

Click “Raise Hand” in BlueJeans

Feel free to share a whiteboard or image.

t

s

t

S

t

s