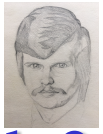
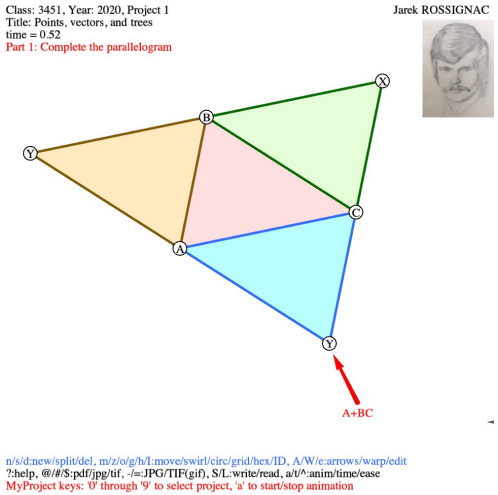
HOW TO DO YOUR PROJECT 1



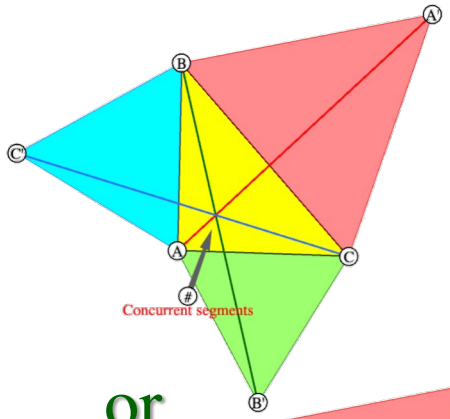
today

week 1

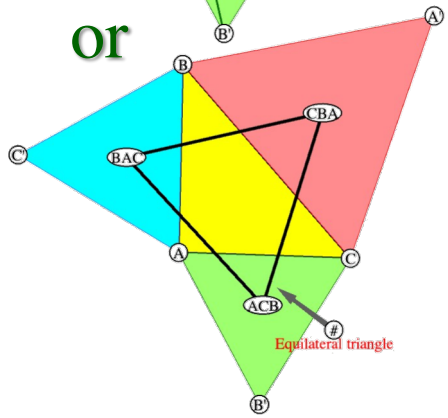
week 2



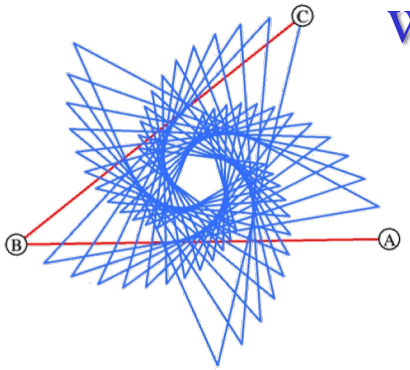
Phase 1: C



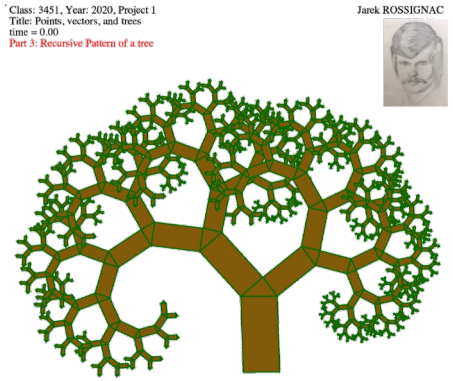
or



Phase 2: B



or…?



Phase 3: A…

CS3451 Fall 2020:

Playing with points and vectors

Learning Objectives

Learn how to download, edit, run the base Processing sketch for the project

Learn how to personalize it

Lear where and how to write you code for each phase of the project

Learn how to save pictures for your report

Learn how to make a gif animation for your submission

Learn how to program simple P+V geometric constructions

Learn how to display points, arrows, polygons

Learn how to use animation

Learn how to insert annotations

Learn how to add GUI functions

… and so much more

Soooooooo?

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

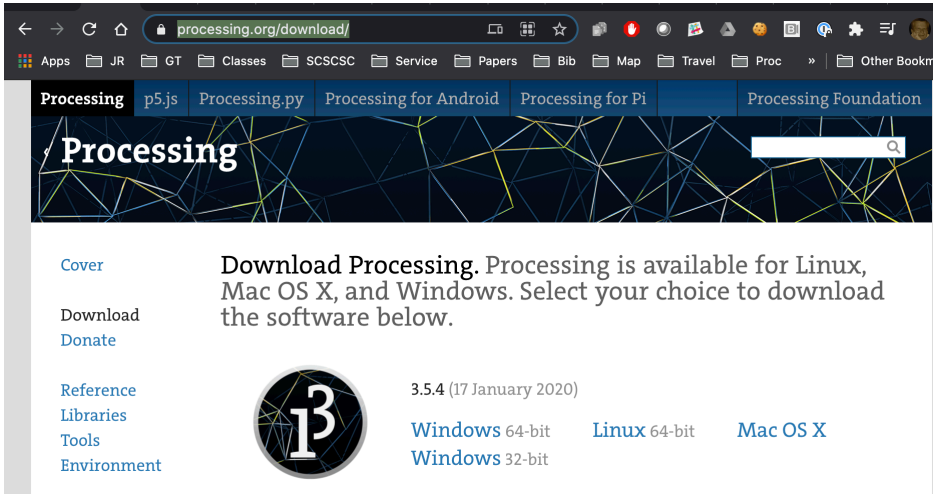
Practice working with others

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

Download Processing

Download recent version from: https://processing.org/download/

https://processing.org/download/



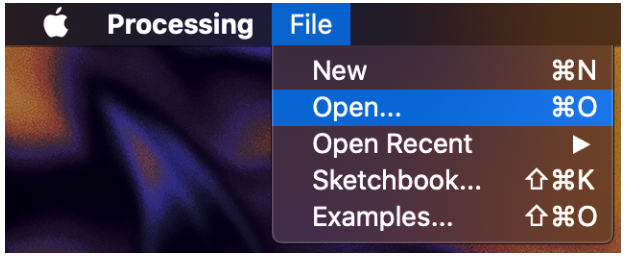
Download and unzip baseSketch1 for project

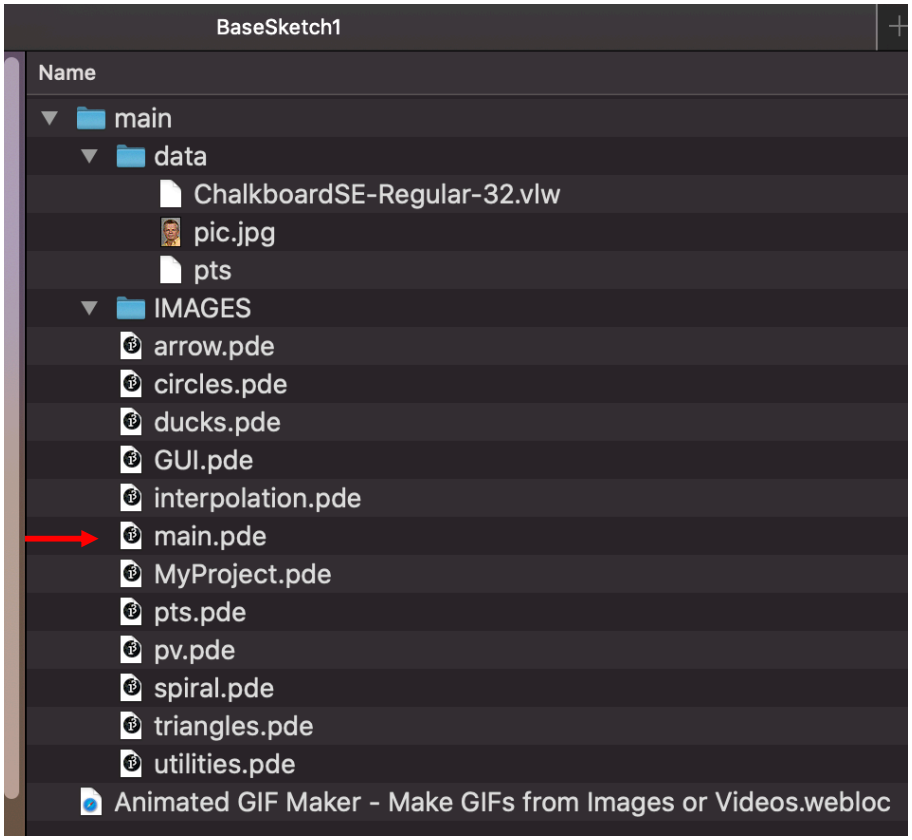
Download baseSketch1.zip from Canvas

Unzip

Start Processing

Load baseSketch1/main/main.pde





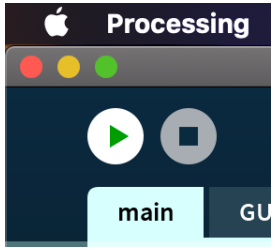
Self-guided demo

main | GUI | MyProject | arrow | circles | ducks | grid | interpolation | pts | pv

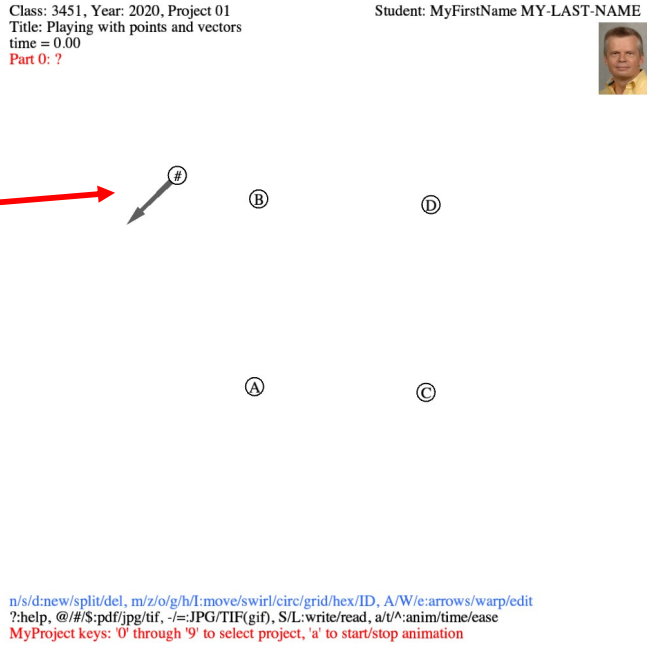
```
//************************** SELF-GUIDED DEMO  *************
/*
a           stop animation
#           save jpg picture in /IMAGES/PICTURES_JPG
t+drag      change time
a           restart animation
^ ^ ^       toggle ease in/out
-           save frames for GIF in /IMAGES/MOVIE_FRAMES_JPG
a           stop animation
drag        move whole arrow or end
e+drag      move arrow start or end
]           save control arrows to \data\arrows
W           hide face-warp
o           make 9 control arrows around a circle
+           use only 4 control arrows
[           load saved control arrows from \data\arrows
A           hide arrows
1           show demo of Phase 1
2           show demo of Phase 2
C           write content of clipboard on canvas
0           show sandbox
A           show arrows
m+drag      move all arrows
}           save control arrows to \data\arrows0
z           scale and rotation all arrows around screen center
}           load control arrows from \data\arrows0
3           show demo of Phase 3
a           stop animtion
t+drag      change time
*/
```

Check that it runs

Press the run button (green arrow)



Canvas should open and show this:



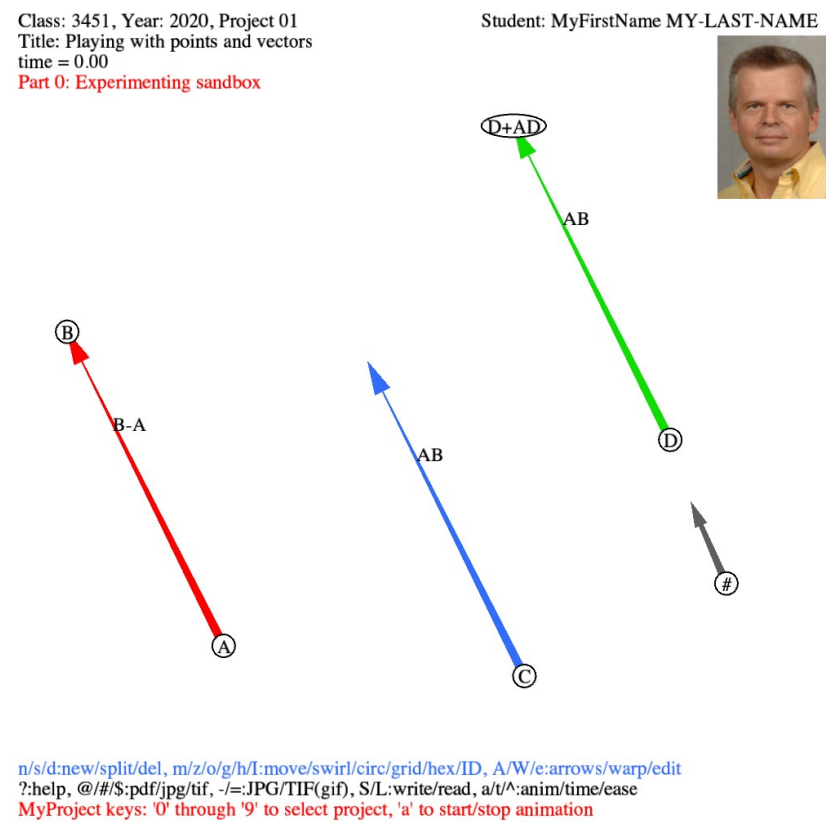
Move the mouse:

- The grey arrow should follow
- It shows when the mouse-key is pressed
- and which key is pressed

Check animation and arrows

Press ‘a’: you should see some of the four control points moving back&forth

- Press ‘a’ again to stop/restart animation
- Hold ‘t’, press & drag mouse to change time



Check animation and arrows

Class: 3451, Year: 2020, Project 01
Title: Playing with points and vectors
time = 0.42
Part 0: Experimenting sandbox

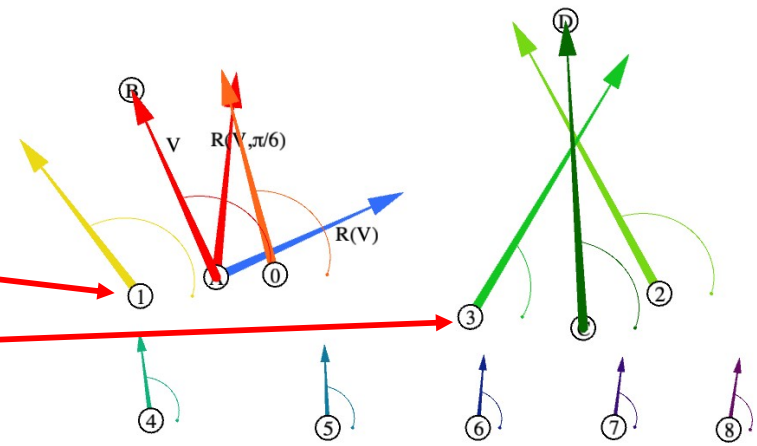
Student: MyFirstName MY-LAST-NAME

Press ‘A’ to see 9 control-arrows

The first 4 control two moving arrows

Press ‘a’ to see them move

- (A,B) oscillates between
  - Orange (0) and yellow (1)
- (C,D) oscillates between
  - Lime (2) and green (3)



n/s/d:new/split/del, m/z/o/I:move/swirl/circ/ID, A/W/e:arrows/warp/edit
?:help, @/#/$:pdf/jpg/tif, -/=:JPG/TIF(gif), {/}:write/read, a/t/^:anim/time/ease
MyProject keys: '0' through '9' to select project, 'a' to start/stop animation

Their endpoints (A,B,C,D) serve as control-points for some projects.

Learn to edit the endpoints on any control-arrow

To edit one of the four control-arrows

- Press & hold LMB (left mouse-button) near the start of an arrow and drag to translate the whole arrow
- Press & hold ‘e’, then press & hold LMB near the start of an arrow and drag to move the start of the arrow only
- Press & hold left mouse-button (LMB) near the end of an arrow and drag to move the end only

Learn to edit control-arrows

Class: 3451, Year: 2020, Project 01
Title: Playing with points and vectors
time = 0.31
Part 0: Experimenting sandbox

Student: MyFirstName MY-LAST-NAME

Mnemonic in blue:

m/z/o/I : move/swirl/circ/ID

to move (translate) all arrows:

keep ‘m’ key pressed and press&drag LMB

to rotate and scale the whole arrangement:

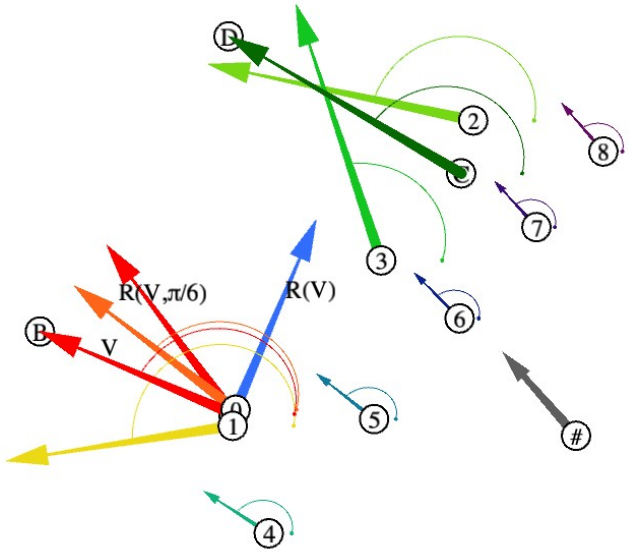
keep ‘z’ pressed …

to set the 9 arrows in a perfect ring:

press ‘o’

to show/hide arrow-IDs:

press ‘I’



n/s/d:new/split/del, m/z/o/I:move/swirl/circ/ID, A/W/e:arrows/warp/edit
?:help, @/#/$:pdf/jpg/tif, -/=:JPG/TIF(gif), {/}:write/read, a/t/^:anim/time/ease
MyProject keys: '0' through '9' to select project, 'a' to start/stop animation

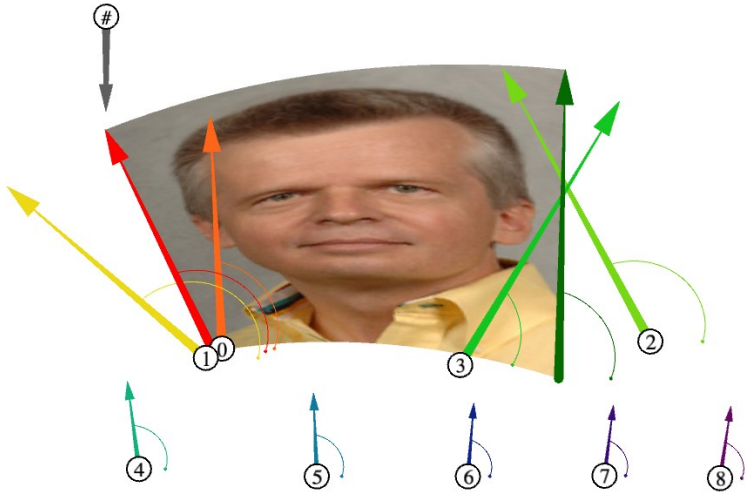
Check-out my face-warp

Press ‘W’ to start demo mode for face warping

- Press ‘a’ to start animation
- Press ‘A’ to see all arrows
- Hold ‘e’ down and edit the control arrows

Class: 3451, Year: 2020, Project 01
Title: Playing with points and vectors
time = 0.50
Part 0: Experimenting sandbox

Student: MyFirstName MY-LAST-NAME

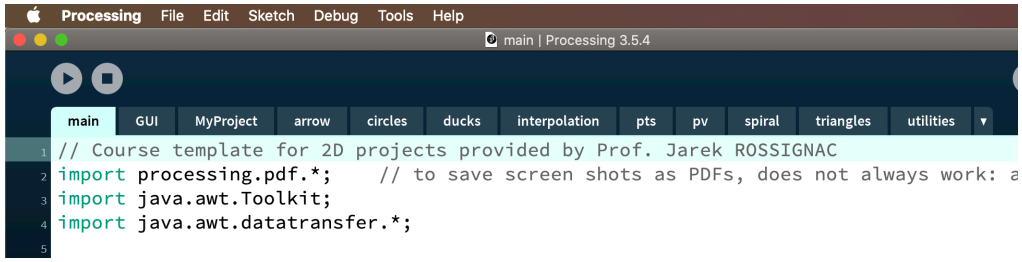


n/s/d:new/split/del, m/z/o/I:move/swirl/circ/ID, A/W/e:arrows/warp/edit
?:help, @/#/$:pdf/jpg/tif, -/=:JPG/TIF(gif), {/}:write/read, a/t/^:anim/time/ease
MyProject keys: '0' through '9' to select project, 'a' to start/stop animation

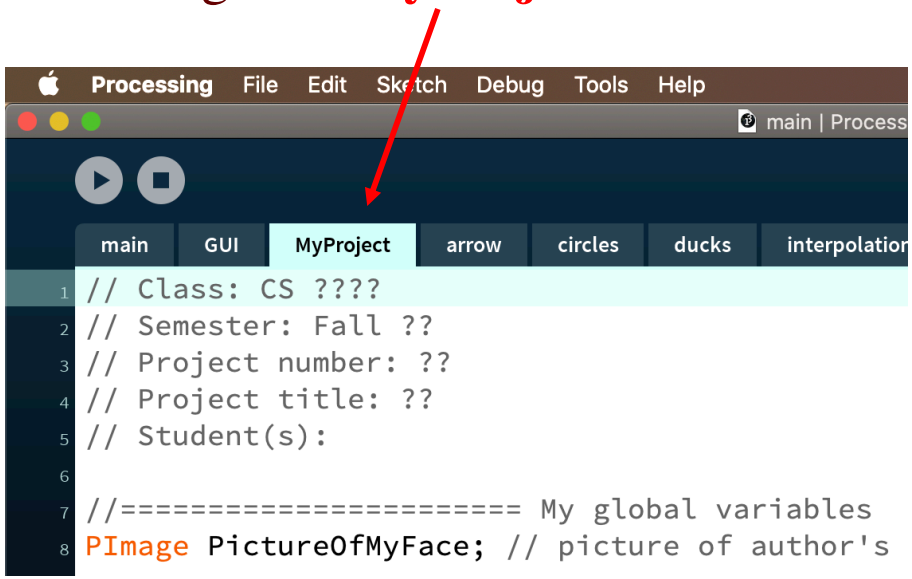
Press ‘W’ again to stop

Overview of the sketch

The sketch is organized into a dozen tabs



For your projects, you should only need to change the **MyProject** tab



Main

main defines

setup(), executed once,

which calls mySetup();

which is in MyProject

and which you can edit

```
main | GUI | MyProject
51  font = loadFont("(
52  mySetup();
53  } // end of setup
```

```
main | GUI | MyProject | arrow | circles | ducks | grid | interpolation | pts | pv | spiral | triangles | utilities
27 //====================== my setup, executed once at the beginning
28 void mySetup()
29   {
30   DrawnPoints.declare(); // declares all ControlPoints. MUST BE DONE BEFORE ADDING POINTS
31   SmoothenedPoints.declare(); // declares all ControlPoints. MUST BE DONE BEFORE ADDING POINTS
32   DrawnPoints.empty(); // reset pont list P
33   SmoothenedPoints.empty(); // reset pont list P
34   //initDucklings(); // creates Ducling[] points
35   }
```

draw() executed at each frame

It erases the canvas

```
main | GUI | MyProject | arrow | circles | ducks | grid | interpolation | pts | pv | sp
56 void draw()        // executed at each frame
57   {
58   int i=0;
59   background(white); // clear screen by painting white background
60
```

which computes the locations of the four animated points, A, B, C, D

and calls showMyProject(A,B,C,D);

which is in MyProject

and which you can edit

```
main | GUI | MyProject | arrow | circles | ducks | grid | inter
85
86  // Displays my project
87  PNT A = ArrowLeft.rP(),  B = ArrowLeft.rQ(),
88      C = ArrowRight.rP(), D = ArrowRight.rQ();
89  if(!showFaceWarp) showMyProject(A,B,C,D);
```

GUI

Defines what happens when a key or mouse is pressed and mouse moved/dragged

It calls:

myKeyPressed(); defined in MyProject

Etc.

main | GUI | MyProject | arrow | circles | ducks | interpolation

```
125 //   println("key pressed = "+key+" = "+int(key));
126     myKeyPressed(); // defined in TAB MyProject
127     }
128
```

pv tab: Points and Vectors

You will use functions, provided in tab pv, for creating points and vectors and for computing distances, angles, etc.

There are many.. for your convenience. Do not worry!

We will explore a few of these at a time.

main | GUI | MyProject | arrow | circles | ducks | interpolation | pts | pv | spiral | triangles | utilities

```
//*************************************************************************
// TITLE:          GEOMETRY UTILITIES IN 2D
// DESCRIPTION:    Classes and functions for manipulating points, vectors, edges, t
// AUTHOR:         Prof Jarek Rossignac
// DATE CREATED:   September 2009
// EDITS:          Revised July 2017
//*************************************************************************
//**********************************************************************
//**** POINT CLASS
//**********************************************************************
class PNT
  {
  float x=0,y=0;

  // CREATE
  PNT () {}
  PNT (float px, float py) {x = px; y = py;};

  // MODIFY
  PNT setTo(float px, float py) {x = px; y = py; return this;};
  PNT setTo(PNT P) {x = P.x; y = P.y; return this;};
  PNT translate(VCT V) {x += V.x; y += V.y; return this;}
  PNT translate(float s, VCT V) {x += s*V.x; y += s*V.y; return this;}
```

Pars 1, 2, 3 of your project and grading scheme

The project is divided into three parts; 1, 2, and 3

Grading

Part 1 (if correct, you get at least a 'C'= 4/10)

Parts 1 & 2 (if **also** correct, you get at least a 'B'= 6/10)

Parts 1 & 2 & 3 (if **also** correct, you get at least an 'A'= 8/10)

You get the credit for a part if all the following are true:

1. Your solution is **correct**
2. You deliver a clear, correct, and concise **write-up** explaining your solution
   - using point+vector constructions and pseudocode
3. You deliver a clean (readable) **source code** that works correctly
   - and instructions (if any are needed): see "guide" below
4. You deliver a short **GIF animation** (created using the control-point motion provided)

Requirements for PART 1 include personalization

1. Fill in the header of MyProject
2. Adjust PartTitles
3. Swap my picture for yours
4. Edit title (course, year, project number), name, subtitle

Header for Project 1

Of course, you should use the course number for your class and your name, instead of mine

main | GUI | MyProject | arrow | circles | ducks | interpolation

```
// Class: CS 3451
// Semester: Fall 2020
// Project number: 1
// Project title: Playing with Points and Vectors
// Student(s): Jarek Rossignac
```

Adjust PartTitle Strings

The user can choose which part to explore/animate by pressing ‘0’…’9’

Leave Part 0 for doing your **experiments**. It will not be graded.

Focus on **Part 1, 2,** and **3**.

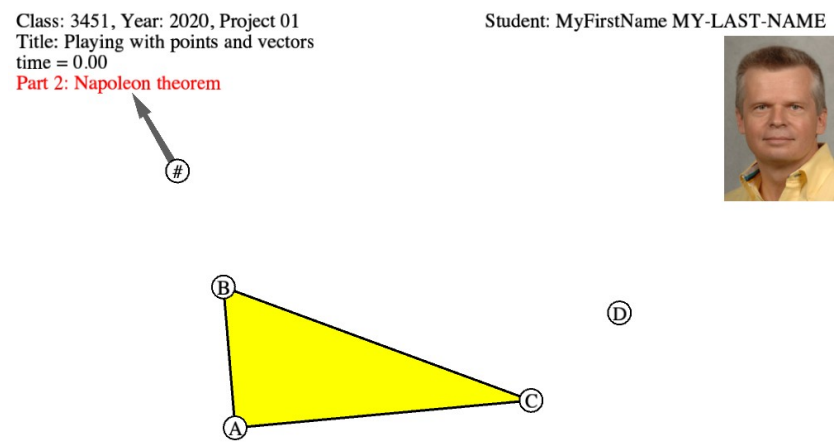
Edit the PartTitle[i]

I decided to do Napoleon Th for part 2

```
void showPart2(PNT A, PNT B, PNT C, PNT D) //
  {
  PartTitle[2] = "Napoleon theorem"; // https:
  cwf(black,3,yellow);  showLoop(A,B,C);
```

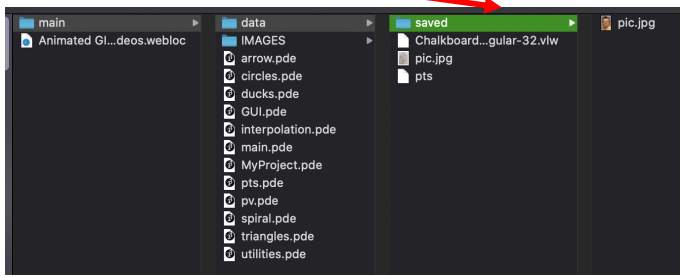
Run the sketch to make sure

Press ‘1’, ‘2’, ‘3’ to check



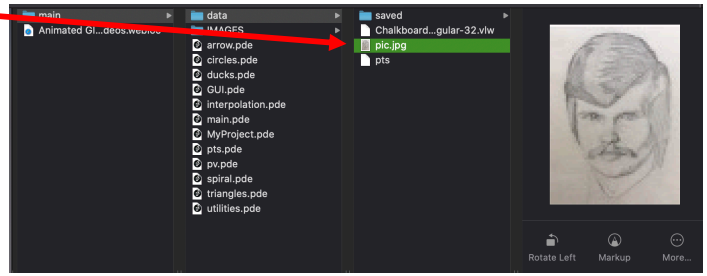
Use your picture … instead of mine

In main/data, create a folder "saved" and move my pic.jpg there



Copy yours in main/data

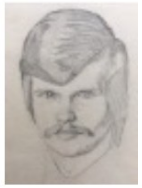
- Picture of your face only
- Front view



Run sketch to check

Class: 3451, Year: 2020, Project 01
Title: Playing with points and vectors
time = 0.00
Part 0: My experiments

Student: MyFirstName MY-LAST-NAME



Edit title, name, subtitle

Check/edit class, year, project number

Remove “Student:” and enter your First and LAST (all caps) name

Fell free to tweak the title of the project, maybe to reflect

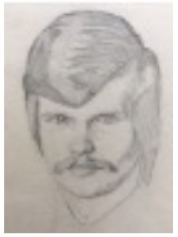
main | GUI | MyProject | arrow | circles | ducks | grid | interpolation | pts | pv | spiral | triangles | utilities

```
//***************************** My text  *****************************
String title ="Class: 3451, Year: 2020, Project 01",
       name ="Student: MyFirstName MY-LAST-NAME";
String subtitle = "Playing with points and vectors";
String guide="MyProject keys: '0' through '9' to select project parts, 'a' to start/stop animation ";
```

Test it!

Class: 3451, Year: 2020, Project 1
Title: Points, vectors, and trees
~~time = 0.00~~
Part 0: My experiments

Jarek ROSSIGNAC



ToDos check list for your first project

Make sure that you have done the following:

1. Fill in the header of TAB MyProject
2. Adjust PartTitles
3. Swap my picture for yours
4. Edit title (course, year, project number), name, subtitle

Implement Part 1: function to draw a triangle

The goal here is to compute the fourth corner of a parallelogram, assuming that the first 3 are given.

I will use A, B, and C as the given corners.

As a reference, I will draw a triangle that has them as vertices.

Tools for this are in the tab pv

I could use 3 calls: show(A,B); show(B,C); show(C,A);

Instead, I will use one call to showLoop (A,B,C);

```
void show(PNT P, PNT Q) {line(P.x,P.y,Q.x,Q.y); };                                                    //
void show(PNT A, PNT B, PNT C)  {beginShape();  A.v(); B.v(); C.v(); endShape();}                  // rend
void show(PNT A, PNT B, PNT C, PNT D)  {beginShape();  A.v(); B.v(); C.v(); D.v(); endShape();}      // re
void showLoop(PNT A, PNT B, PNT C)  {beginShape();  A.v(); B.v(); C.v(); endShape(CLOSE);}
void showLoop(PNT A, PNT B, PNT C, PNT D)  {beginShape();  A.v(); B.v(); C.v(); D.v(); endShape(CLOSE);}
void arrow(PNT P, PNT Q) {show(P,V(P,Q)); }                                                           /
```

Drawing a triangle

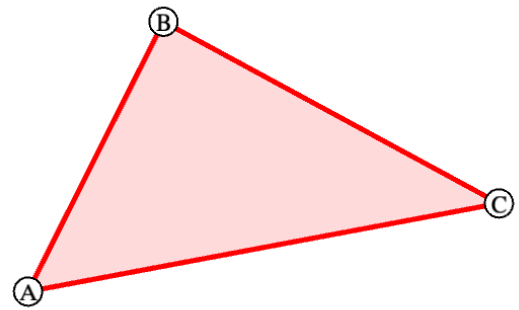
Just type at the beginning of showPart1()

```
cwfo(dred,5,pink,70);
showLoop(A,B,C);
```

Try it

- Don't forget to press '1'
  - Or change the initialization of partShown to 1
- Try the animation ('a')

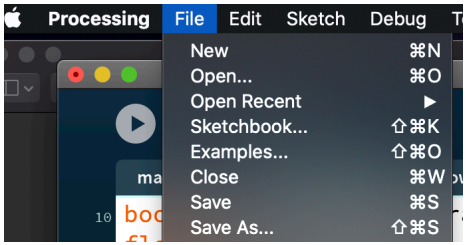
```
boolean lerp=true, spir
float b=0, c=0.5, d=1;
int partShown = 1;
```



Save the points (press 'S')

SAVE YOUR CODE! (command + s on a mac)

- Save your work often!!!



Fill the color the triangle

main | GUI | MyProject | arrow | circles | ducks | interpolation | pts | pv | spiral | triangles | utilities

```
113 void cwf(color c, int w, color f) {stroke(c); strokeWeight(w); fill(f); } // color, width, fill
114 void cWf(color c, color f) {stroke(c); strokeWeight(1); fill(f); } // color, width, fill
```

Use cwf(c,w,f,o) in utilities

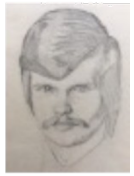
- c = border color
- w = border width
- f = fill color
- o = opacity (0=transparent, 100=opaque)

I will use cwfo(dred,5,pink,70);

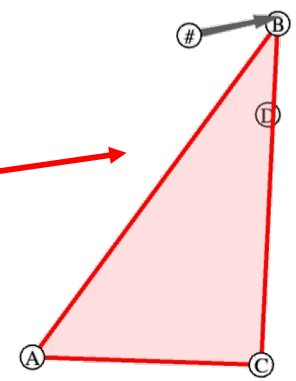
- I temporarily placed D.writeLabel("D");
- before drawing the triangle to show the opacity

Class: 3451, Year: 2020, Project 1
Title: Points, vectors, and trees
time = 0.00
Part 1: Complete the parallelogram

Jarek ROSSIGNAC



Try it



Colors

I use the HSB color interface

- On a MAC
- You can select a Processing function call
- and press shift+command+f to
- see its definition in a browser

I have defined a few colors in tab utilities

- You can tweak these and define more
- Using color selector

main | GUI | MyProject | arrow | circles

```
44  colorMode(HSB, 360, 100, 100);
```

Processing | p5.js | Processing.py | Processing for Android | Processing for Pi | Processing Fou

Processing

Language
Libraries
Tools
Environment

This reference is for Processing 3.0+. If you have a previous version, use the reference included with your software in the Help menu. If you see any errors or have suggestions, please let us know. If you prefer a more technical reference, visit the Processing Core Javadoc and Libraries Javadoc.

Name colorMode()

Examples

```
noStroke();
colorMode(RGB, 100);
for (int i = 0; i < 100; i++) {
  for (int j = 0; j < 100; j++) {
    stroke(i, j, 0);
    point(i, j);
  }
}
```

```
noStroke();
colorMode(HSB, 100);
for (int i = 0; i < 100; i++) {
  for (int j = 0; j < 100; j++) {
    stroke(i, j, 100);
    point(i, j);
  }
}
```

main | GUI | MyProject | arrow | circles | ducks | interpolation | pts | pv | spiral | triangles | utilities

```
// ************************************************** COLORS
color  // set more colors using Menu >  Tools > Color Selector
   black=#000000, grey=#5F5F5F, white=#FFFFFF,
   pink=#FF8B8F, red=#FF0000, dred=#FF0000,
   lime=#9EFF6F, green=#10DE02, dgreen=#076A01,
   cyan=#00FDFF, blue=#336DFC, dblue=#021FAA,
   yellow=#FEFF00, orange=#FFB217, brown=#815B09,
   magenta=#CE5DFF, dmagenta=#650190,
   metal=#C5DEE3, dmetal=#7A8D90,
   banana=#C7FC1C, dbanana=#77980E,
   gold=#FFE812, dgold=#988A0B,
   rose=#E37C6C, drose=#9D5449
   .
```

Color Selector

H 167 °
S 77 %
B 56 %
R 32
G 144
B 121
#209079
Copy
Cancel

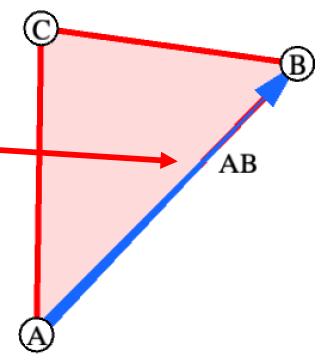
Make and show vector AB between points A and B

I create a vector AB

And draw it (in blue)

Starting at A

- I also commented out
- The display of point D




```
cwfo(dred,5,pink,70);
showLoop(A,B,C);
VCT AB = V(A,B);
show(A,AB,blue,"AB");
```

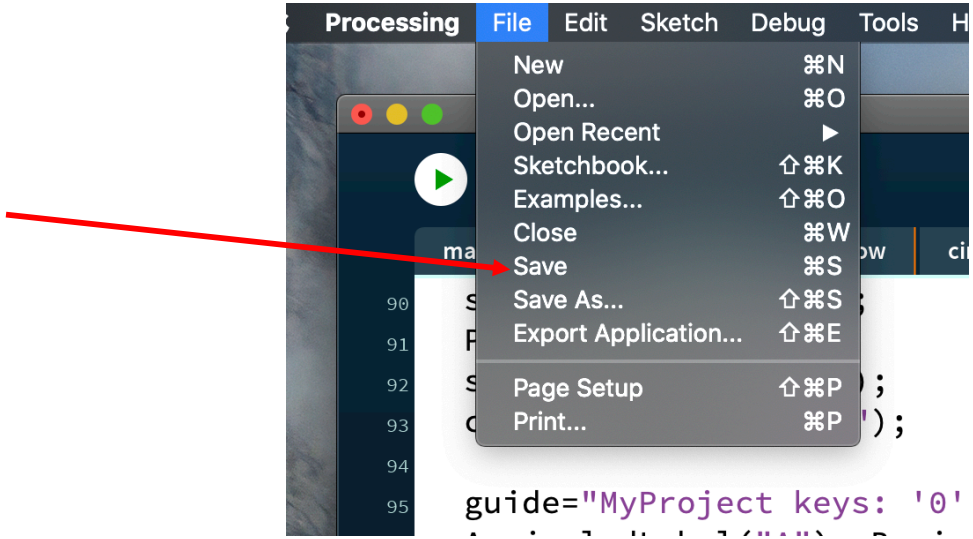
Show the same vector AB from another point

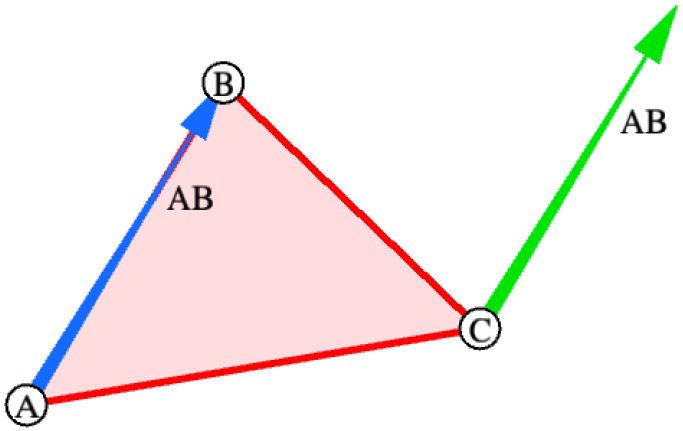
Now I show AB in green

Starting from C

Try it. Animate to check.

Don't forget to save your code, often !!!

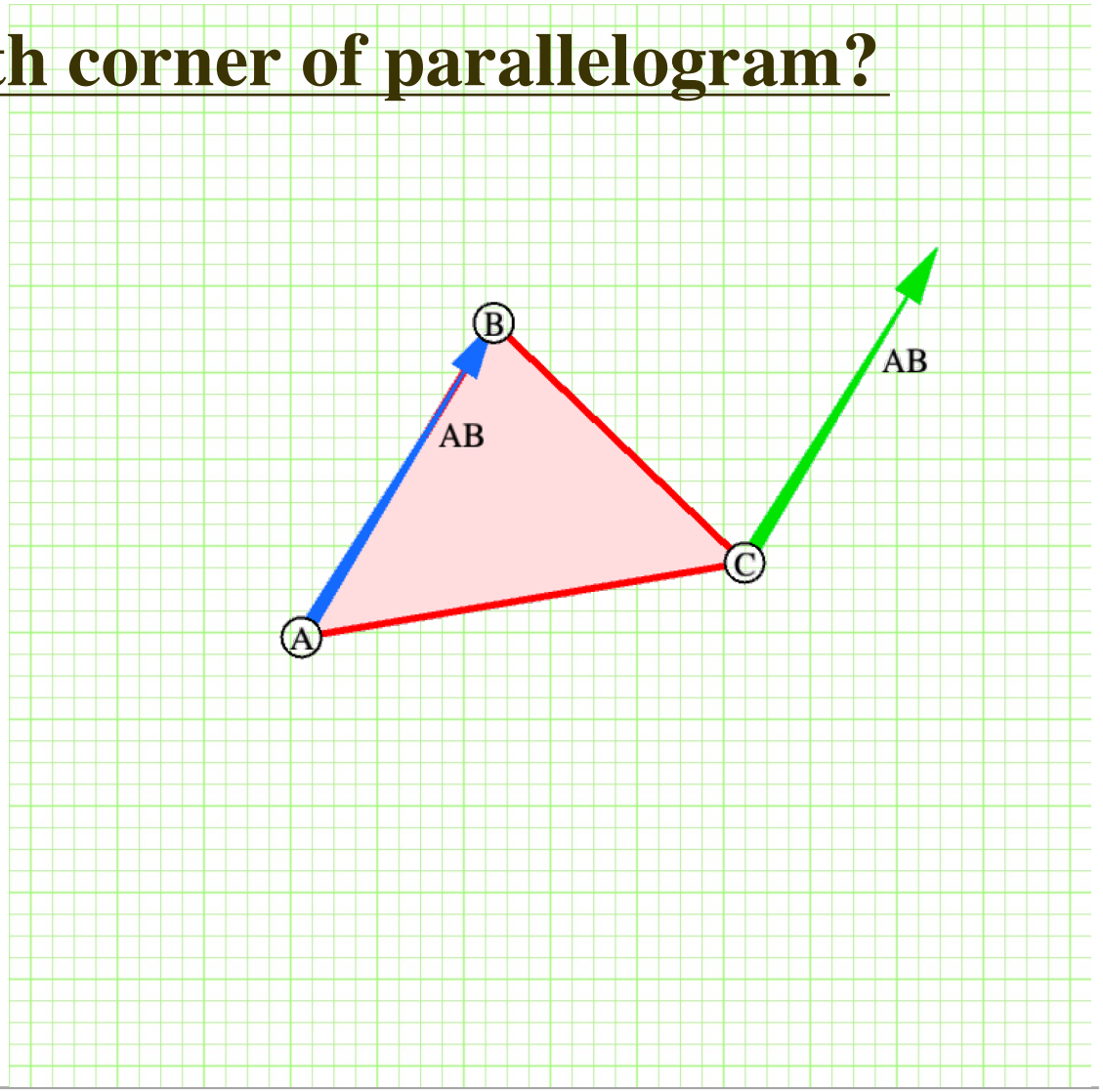





```
cwfo(dred,5,pink,70);
showLoop(A,B,C);
VCT AB = V(A,B);
show(A,AB,blue,"AB");
show(C,AB,green,"AB");
```

How to compute the fourth corner of parallelogram?

Write it in English.

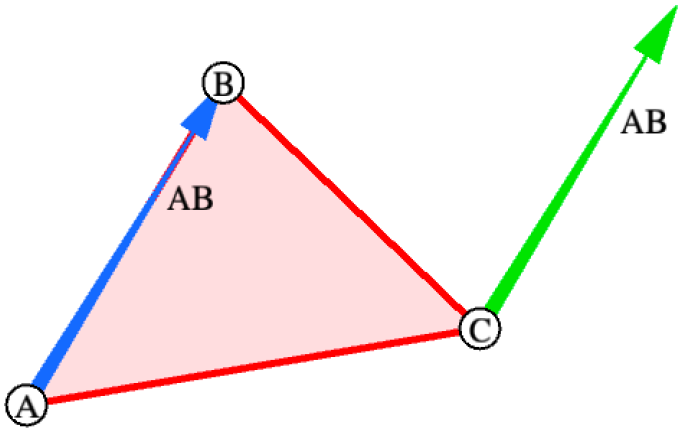


How to compute the fourth corner of parallelogram?

Here is a hint.

The fourth corner X of parallelogram with corners A, B, C, is obtained by displacing point ? by vector ?

Replace the "?"s:



If you do not remember what a parallelogram is, check here:
https://en.wikipedia.org/wiki/Parallelogram

Writing the solution in math form

As part of your deliverables need to provide a mathematical expression, construction, or algorithm for each part. You should do this using the point & vector notation learned in class. When simple, you can type it as a comment in your code.

\\ The fourth corner X of parallelogram with corners A, B, C, is C+AB

I omit the arrow above the vector

To implement it, I use the P(P,V); function in tab pv

main | GUI | MyProject | arrow | circles | ducks | interpolation | pts | pv

```
PNT P(PNT P, VCT V) {return P(P.x + V.x, P.y + V.y); }
PNT P(PNT P, float s, VCT V) {return P(P,W(s,V)); }
```

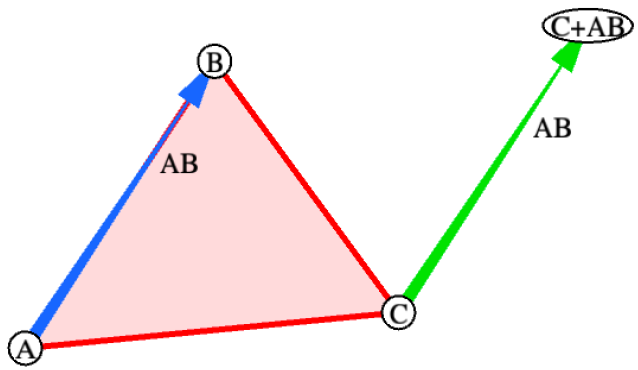
Try declaring, creating, and displaying point X yourself

And test your result before looking at the next slide

Completing PART 1

P(C,AB) returns C+AB,

Where AB is returned by V(A,B);




```
cwfo(dred,5,pink,70);
showLoop(A,B,C);
VCT AB = V(A,B);
show(A,AB,blue,"AB");
show(C,AB,green,"AB");
PNT X = P(C,AB);
X.circledLabel("C+AB");
```

Cleaning it up

Display and fill triangle [BCX]

To complete the parallelogram.

Notice that the arrows and labels

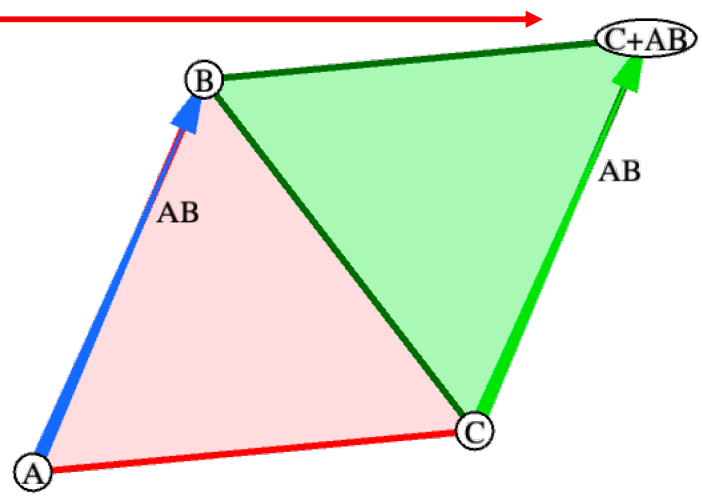
at the end

to avoid occluding them

```
cwfo(dred,5,pink,70);
showLoop(A,B,C);
VCT AB = V(A,B);
PNT X = P(C,AB);
cwfo(dgreen,5,green,70);
showLoop(C,B,X);
show(A,AB,blue,"AB");
show(C,AB,green,"AB");
X.circledLabel("C+AB");
```

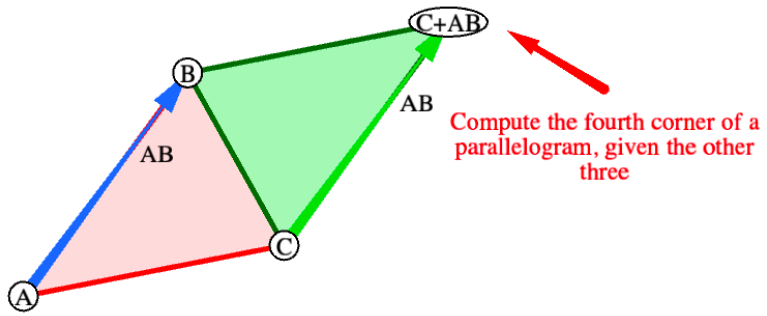
Notice the label C+AB

which provides the math



Adding text from clipboard

If you copy some text (here I used "Compute the fourth corner of a parallelogram, given the other three") into the **clipboard**, place the cursor-arrow where you want to point to, and press 'C', the text will be displayed on the canvas below the start of the arrow.



Have the arrow point **upward** to avoid obstructing the text with it.

(I was lazy, sorry.)

Are we done with PHASE 1?

No!

Why not?

….

Why is our solution incorrect?

Write your answer in English

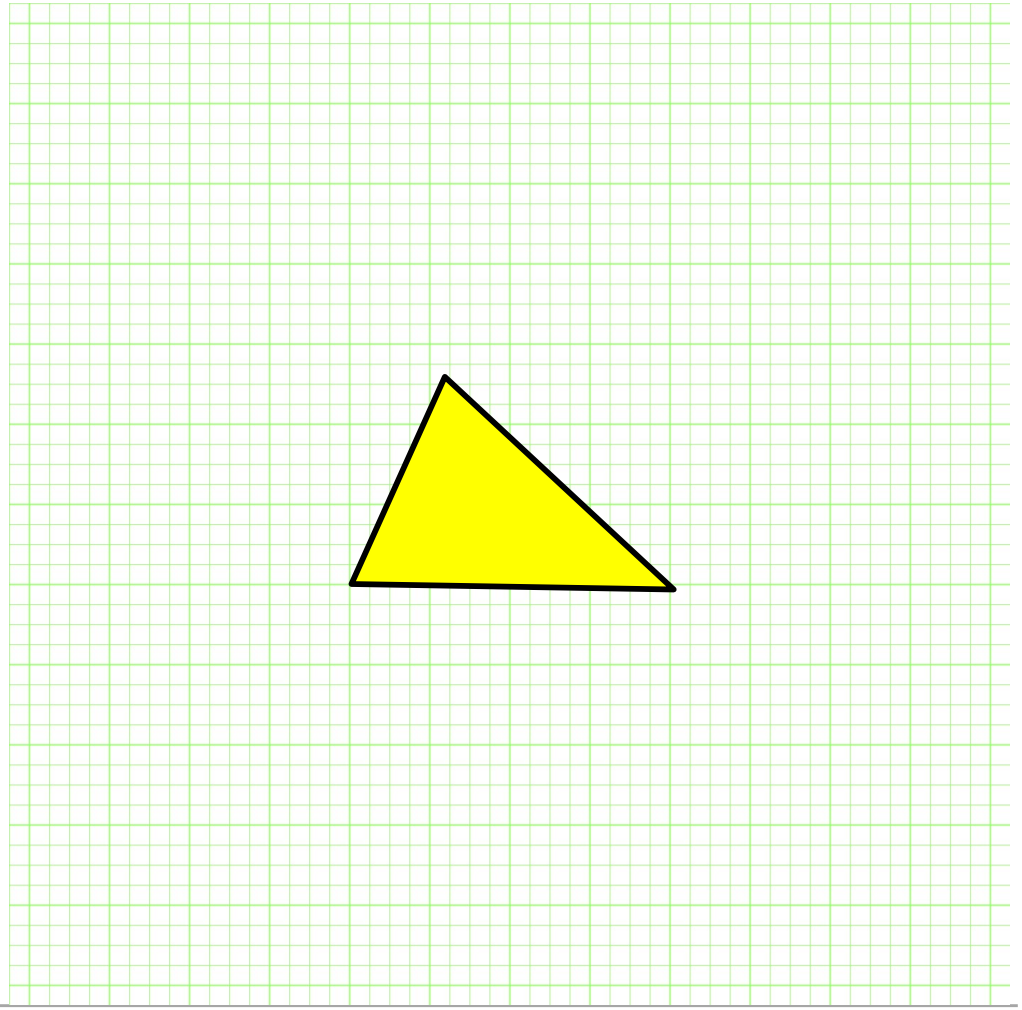
Why is our solution incorrect?

Because the assignment was **ambiguous**.

Do you see why?

How many solutions?

Draw them here



There are several solutions

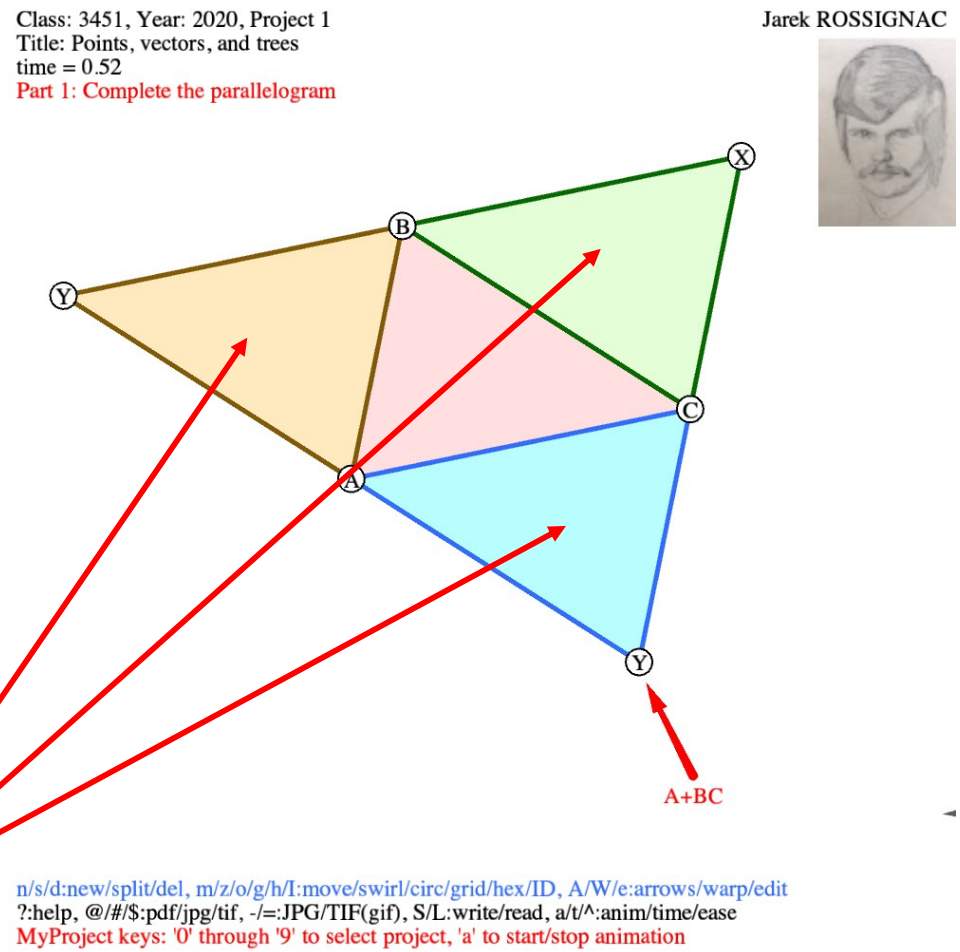
It is your responsibility to detect such ambiguities.

In your report, clearly state how you chose to disambiguate

- Make sure that your choice is reasonable/acceptable

If in doubt, clarify them during class or via piazza questions.

- A good strategy here is to compute and show all the solutions or to explain that you assumed A, B, C, X are in cw or ccw order.



How to add key-controlled actions

```
//if(key=='X') ;  // used in MyProject
//if(key=='Y') ;  // used in MyProject
//if(key=='Z') ;  // used in MyProject
```

I want to use ‘X’, ‘Y’, ‘Z’ keys to toggle these solutions

In GUI, check that these keys are free and mark them as used

Declare flags

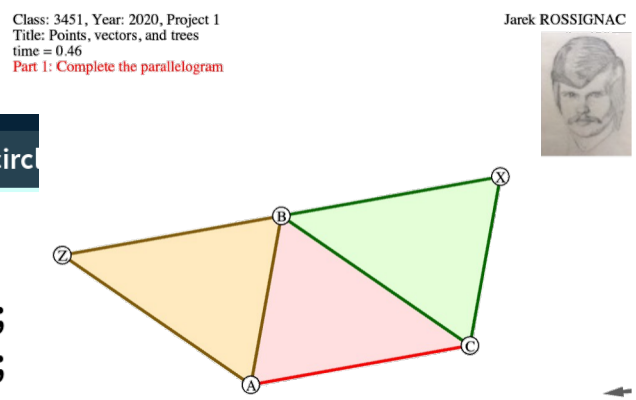
use them in your solution

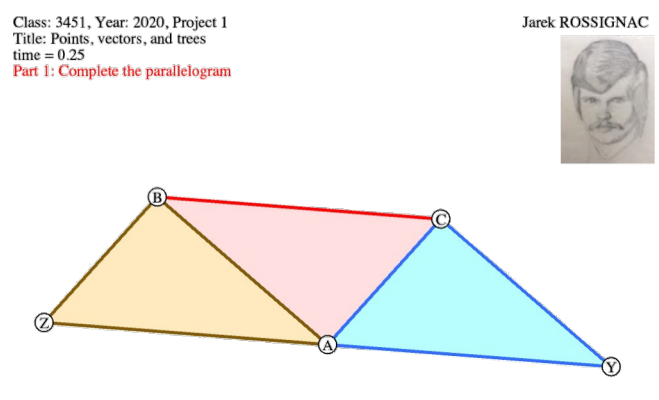
and advertise in the guide

Toggle in myKeyPressed()

```
boolean showX=true, showY=false, showZ=false;
void showPart1(PNT A, PNT B, PNT C, PNT D) //
  {
  cwfo(dred,5,pink,70); showTriangle(A,B,C);
  if(showX) {PNT X = P(C,V(A,B)); cwfo(dgreen,5,lime,70); showTriangle(C,B,X); X.writeLabel("X");} // C+AB
  if(showY) {PNT Y = P(A,V(B,C)); cwfo(blue,5,cyan,70); showTriangle(A,C,Y); Y.writeLabel("Y");} // A+BC
  if(showZ) {PNT Z = P(B,V(C,A)); cwfo(brown,5,orange,70); showTriangle(B,A,Z); Z.writeLabel("Z");} // B+CA
  guide="X/Y/Z:show/hide solutions";
  A.writeLabel("A"); B.writeLabel("B"); C.writeLabel("C"); // D.writeLabel("D");
  }
```

```
void myKeyPressed()
  {
  if(key=='X') showX=!showX;
  if(key=='Y') showY=!showY;
  if(key=='Z') showZ=!showZ;
```



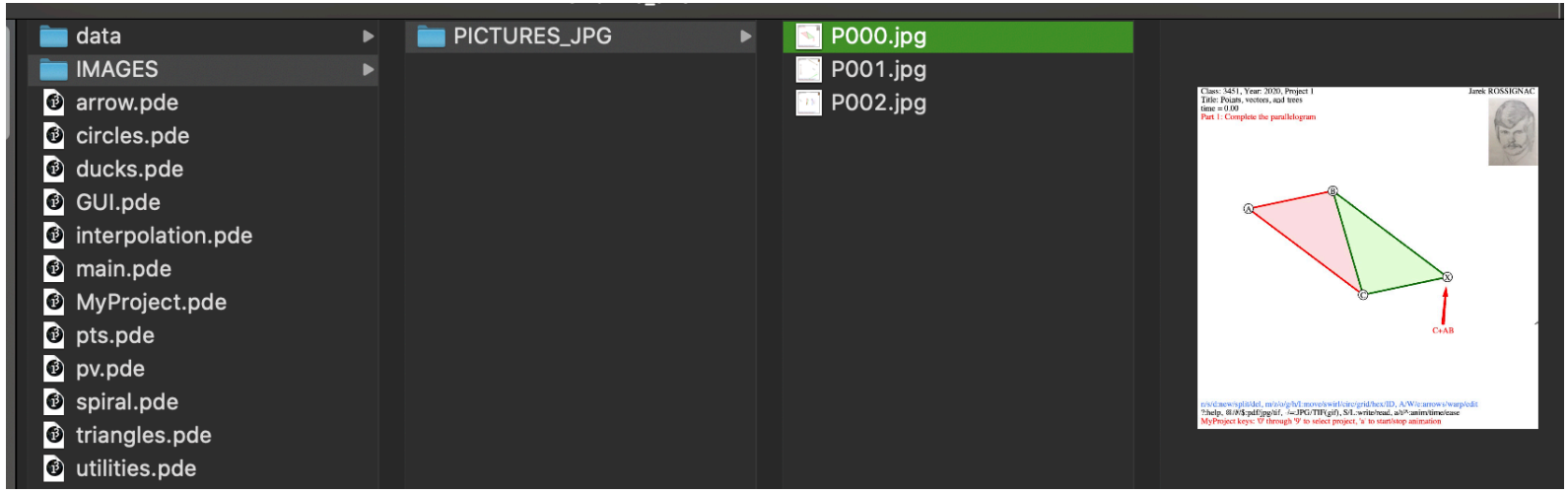


How to make a picture of the screen for your report

Being able to make pictures will useful for explaining more complex solutions and also for asking questions (to the TAs or me).

Doing so is easy. Just press '#'. Try it

- The pictures are in
- IMAGES/PICTURES_JPG
- They are numbered automatically



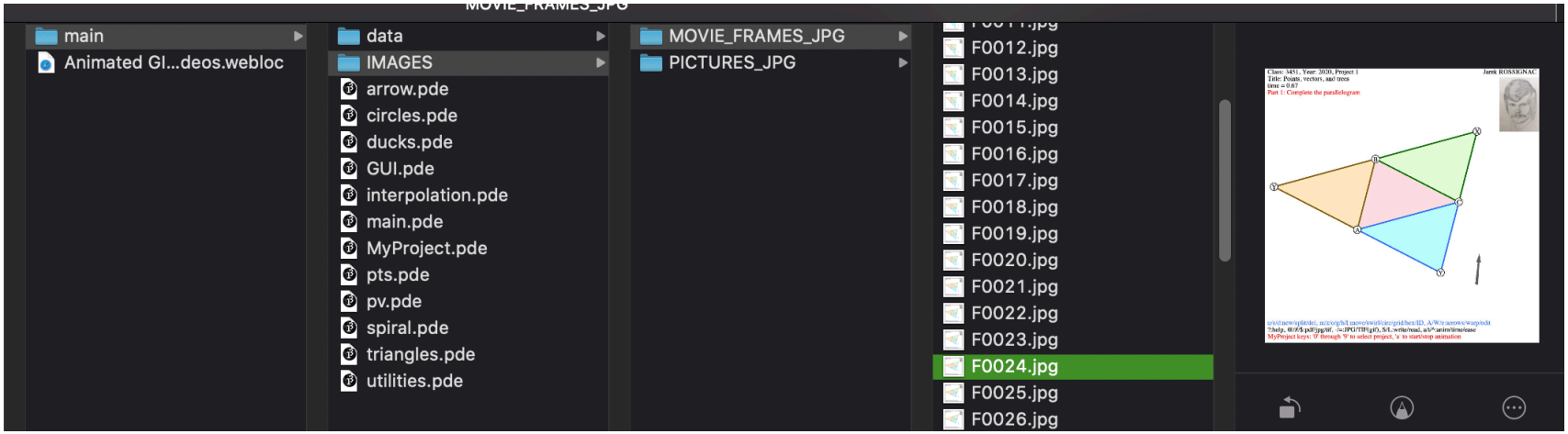
How to save numbered frames for a GIF animation

Press ‘a’ to start animation

Press ‘-’ to start and stop recording

Keep it **short** less than 4 seconds often suffice

Numbered frames are saved in main/IMAGES/MOVIE_FRAMES_JPG

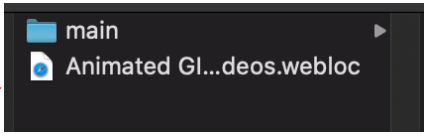


Save your chose control-arrow configurations

You can do this for each phase (0..9)

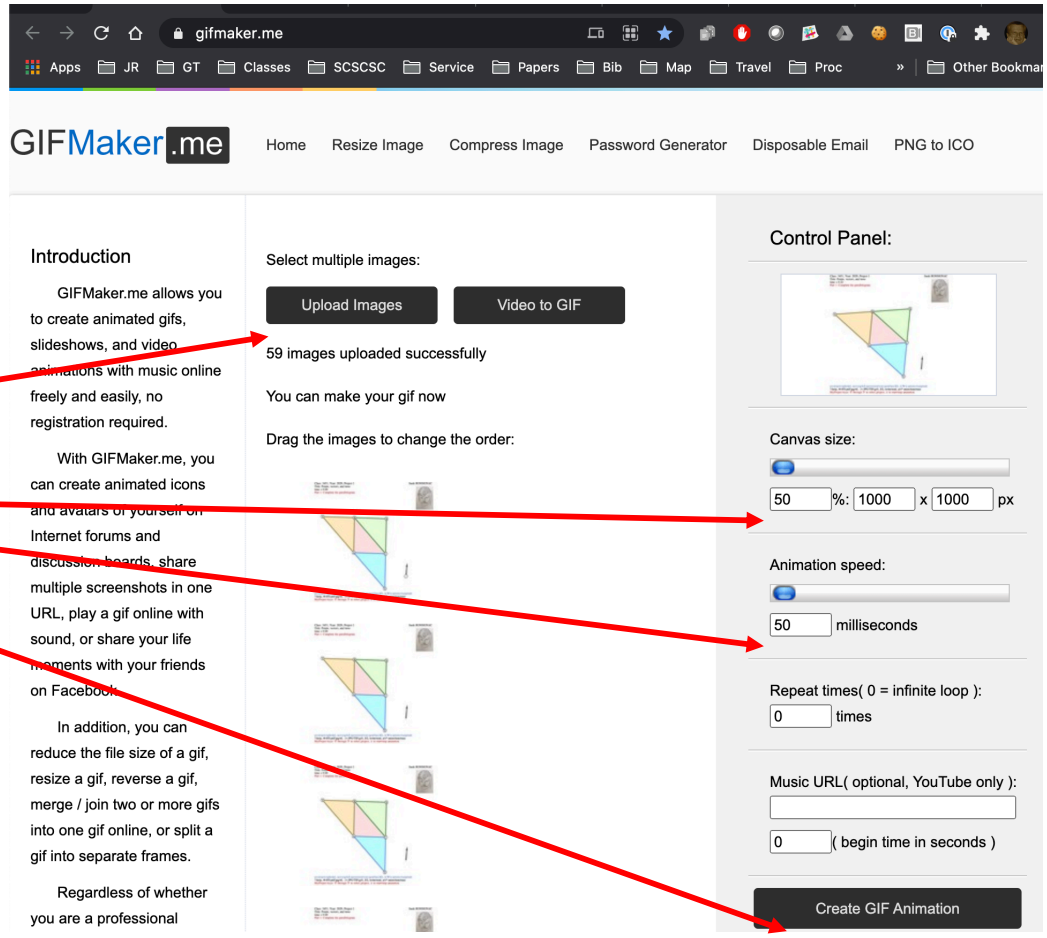
- Select the phase (press ‘0’…’9’)
- Show the control-arrows (press ‘A’)
- Edit their endpoints with the mouse
  - Hold ‘e’ when you do this if you want to translate an arrow by dragging its start
  - Hold ‘m’ or ‘z’ to edit them all
- Check the resulting animation (‘a’)
- Keep editing till you like the result
- Save the configuration (press ‘{‘)
  - This will not affect what you saved for the other phases

How to make an animated GIF

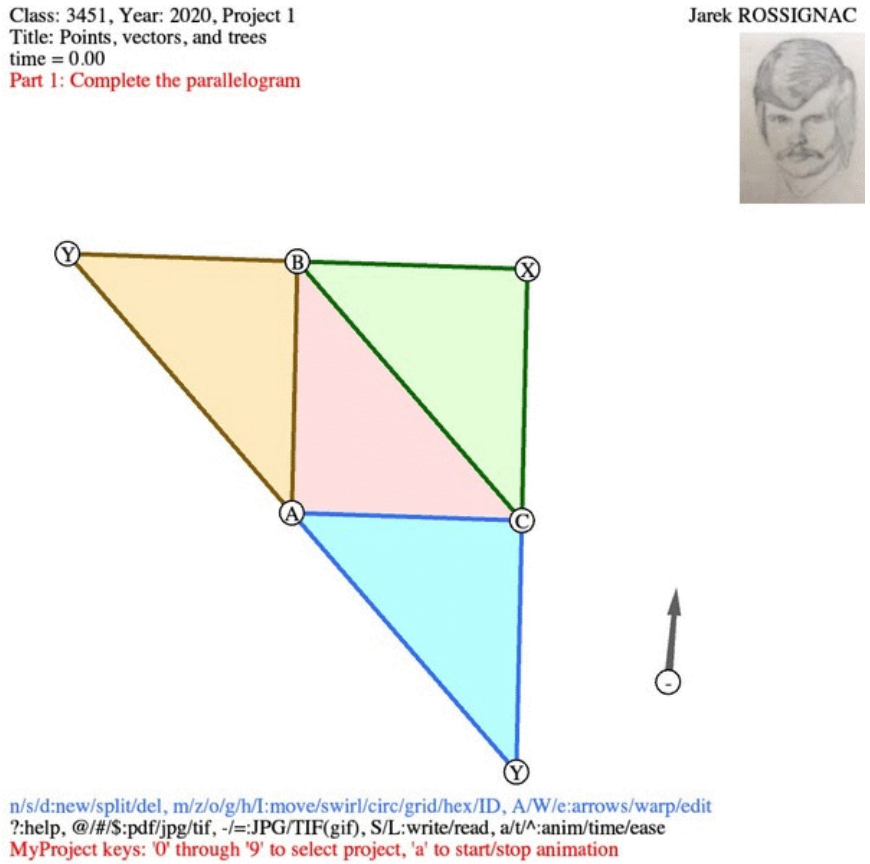


Use the link in the sketch folder
to https://gifmaker.me/
https://gifmaker.me/

Open **gifmaker**
Select and upload all the frames
Set the parameters as shown
Press [Create GIF Animation]
Wait till it is done
Download (option appears below Create GIF)
Rename file to: Proj1Phase1NAME.gif
where NAME is your last name in caps
Submit



GIF for Phase 1 (uploaded as Proj1Phase1ROSSIGNAC.gif)

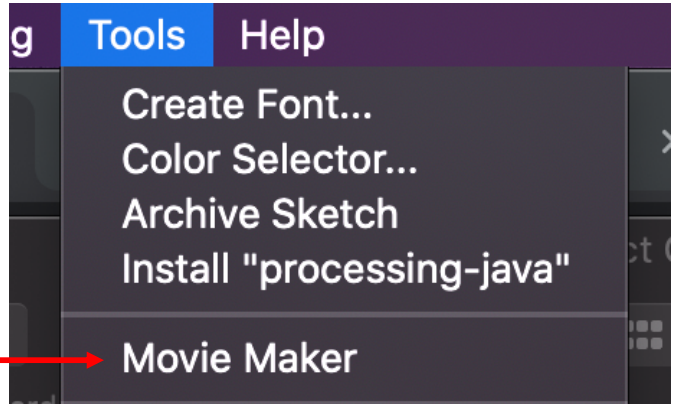


How to make a movie

Create the movie frames as you did for the GIF

But using '=' instead of '-'

Start Movie Maker



Follow instructions to select folder with frames

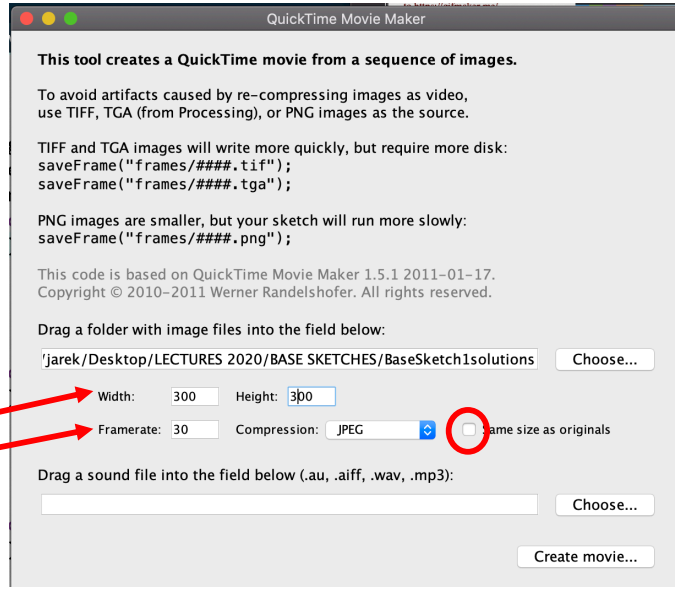
In main/IMAGES/MOVIE_FRAMES_TIF

and where to save the result

into file: Proj1Phase1NAME.mov

Replace NAME by your Last Name in CAPS

Set parameters to **small resolution**



About phase 2: Pick one of these two

Implement animated validation of:

Napoleon theorem of triangle ABC

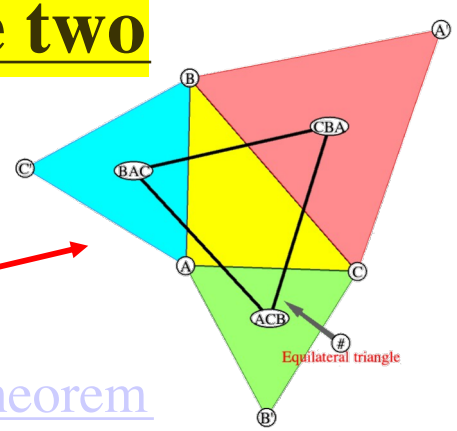
https://en.wikipedia.org/wiki/Napoleon%27s_theorem

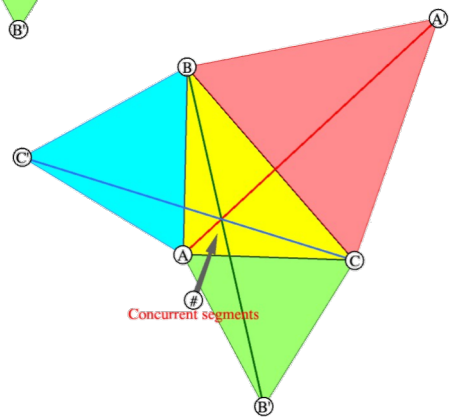
or

Fermat point of triangle ABC

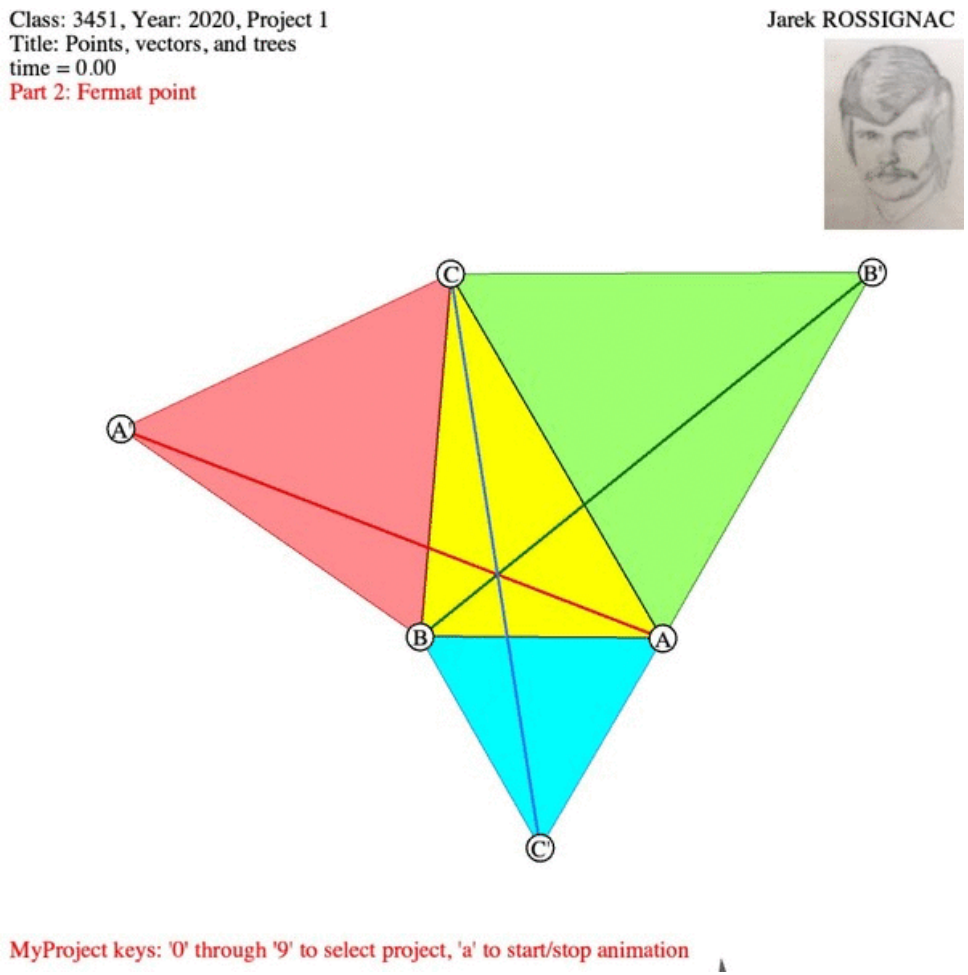
https://en.wikipedia.org/wiki/Fermat_point

by growing equilateral triangles on its sides





GIF for phase 2: Fermat center



Phase 3: Your choice—be creative

But your solution

must be **invented**, **designed** and **implemented** by you

- Do not copy code written by others

AND

needs to meet some requirements:

- Must use **nontrivial** recursion or iteration
- The math must be fully expressed using the point and vector operators covered so far
- The implementation must use the point and vector functions covered so far
  - Do not use coordinates
  - Do not use PVector or other libraries

Some examples of acceptable solutions to get full marks

I include 3 below

I have Pinterest folders with examples that may inspire you

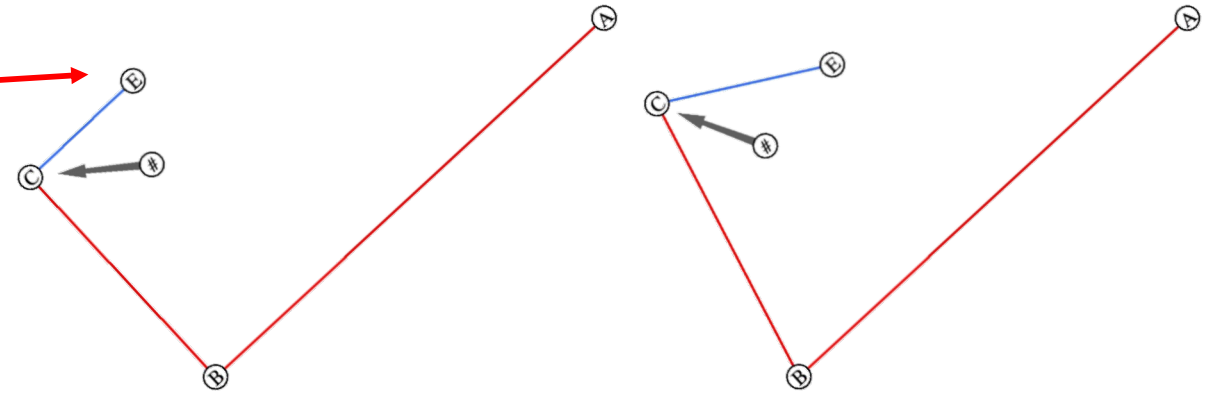
- Patters: https://pin.it/1QJXCLP
- Structures: https://pin.it/5ejpTrH
- Motions: https://www.pinterest.com/jarekrossignac/motion/

Recursion for PHASE 3: log spiral (A)

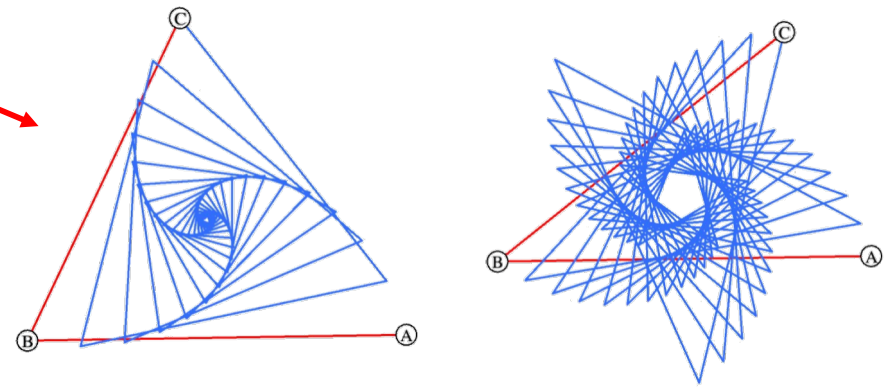
Given animated points A, B, C

Compute E st ABC~BCE

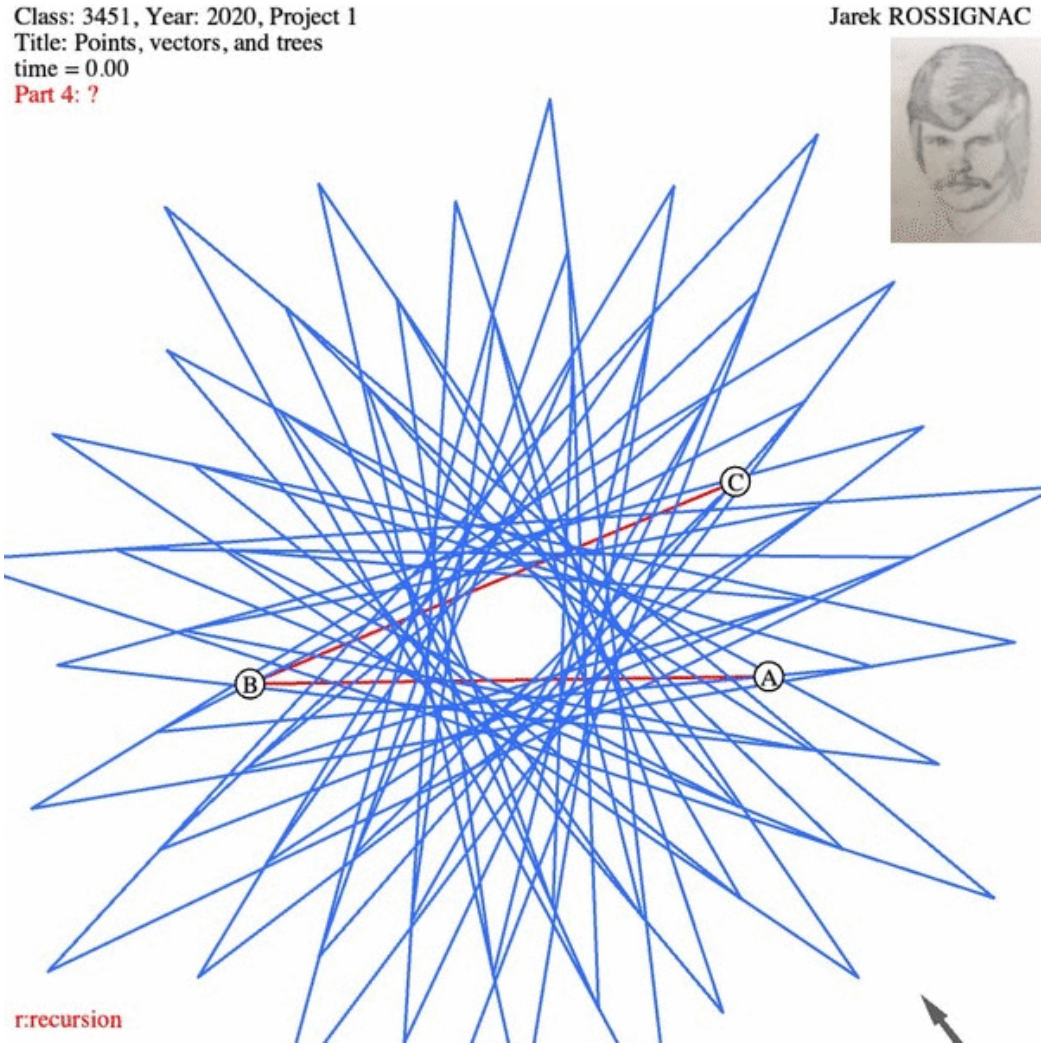
display edge(C,E)



Recurse (A,B,C) → (B,C,E)



GIF for PHASE 3: Spiral



Recursion for PHASE 3: Tree (A+)

Controlled by animated points A, B, C, D designed to simulate growth

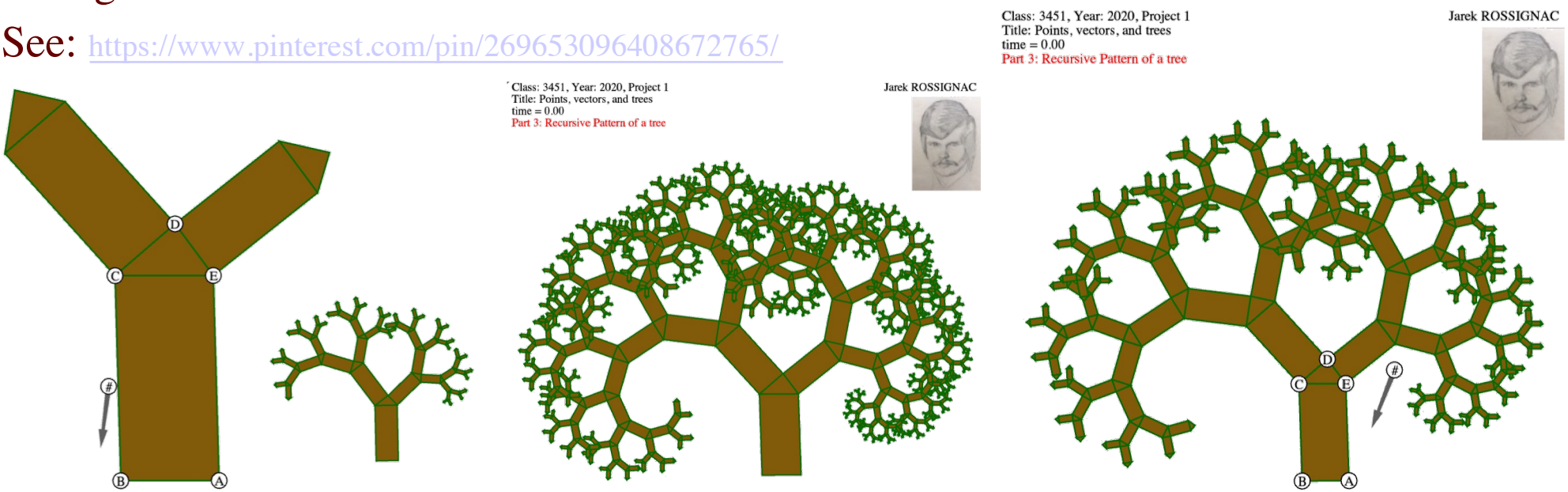
Compute E to complete parallelogram

Measure ratios and angles to compute C and D from A and B

Recurse (A,B,…) ➔ (D,C,…) and (E,D,…)

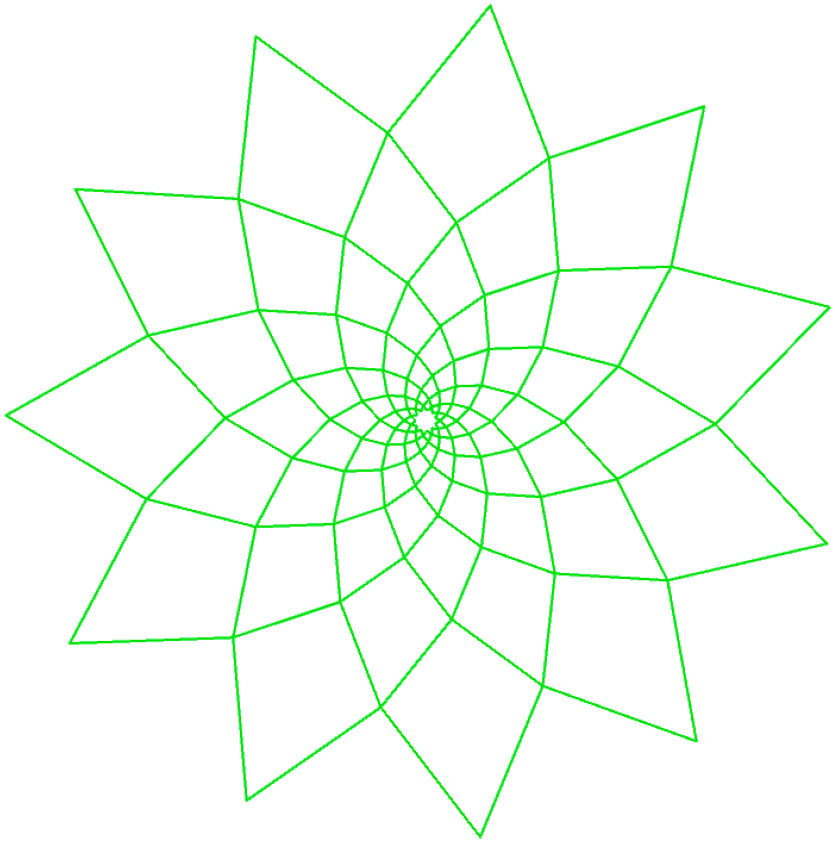
Change recursion with time

See: https://www.pinterest.com/pin/269653096408672765/



GIF for PHASE 3: tree

Other ideas for PHASE 3

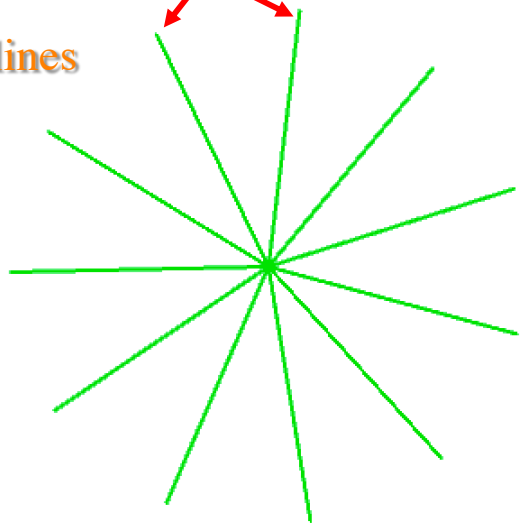


To get you started:

main | GUI | MyProject | arrow | circles | ducks | interpolation | pts

```
void showPart5(PNT A, PNT B, PNT C, PNT D) //
  {
  PartTitle[5] =   "Michell truss";
  int k=11, r=11;
  PNT[][] T = new PNT[r][k];
  PNT O = ScreenCenter();
  VCT V = V(O,A);
  for(int i=0; i<k; i++) T[0][i]=P(O,R(V,TWO_PI*i/k));
  cw(green,2);
  for(int i=0; i<k; i++) show(O,T[0][i]);
```

You don't need these two lines



Only 14 lines of code in total

Q&As

Please submit clear **questions** within 24 hours after class

So that we can address them right away or discuss during the next lecture

- Or on Piazza

Include images if appropriate

Use private questions if they include part of your code or solution

Use public questions that may help others better understand what needs to be done or understand how to do some things in Processing

My slides, the project description, or my discussion may (and will) contain **errors**.

Please point them out, even if you are not sure.

t

s

t

s

t

s