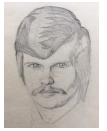
HOW TO DO YOUR COLOR PROJECT 1



Computational Aesthetics
CS4497/6497
Project 1: Color ramp, loop, area
In different color spaces

Learning Objectives

Learn some Processing implementations for **geometry** and **GUI**

Learn about **color spaces** (RGB,XYZ,Lab,Lch): Conversion, Comparison, Issues

Learn about simple **parametric curves** (interpolation and approximation)

Practice **creative** exploration of **ambiguous** specifications

What to I want?

Given 3 input colors, C1, C2, C3, provided by an artist:

1. Define and show a nice **average** of these (which I will use as background)
2. Define and show a nice **approximating ramp** from C1 to C3, that passes near C2
3. Define and show a nice **interpolating ramp** from C1, through C2, to C3
4. Define and show a nice **approximating loop** through C1, C2, and C3
5. Define and show a nice **interpolating loop** through C1, C2, and C3
6. Define and compute a nice **accent** (i.e., highlight) color for these and their ramps
7. Define and show a nice **sampling** of colors in the triangle (C1,C2,C3)

Questions?

Proposals?

Discussion?

What do I mean by "nice"?

Given 3 input colors, C1, C2, C3, provided by an artist:

1. Define and show a nice average of these (which I will use as background)
2. Define and show a nice approximating ramp from C1 to C3, that passes near C2
3. Define and show a nice interpolating ramp from C1, through C2, to C3
4. Define and show a nice approximating loop through C1, C2, and C3
5. Define and show a nice interpolating loop through C1, C2, and C3
6. Define and compute a nice accent (i.e., highlight) color for these and their ramps
7. Define and show a nice sampling of colors in the triangle (C1,C2,C3)

What do I mean by "nice"?

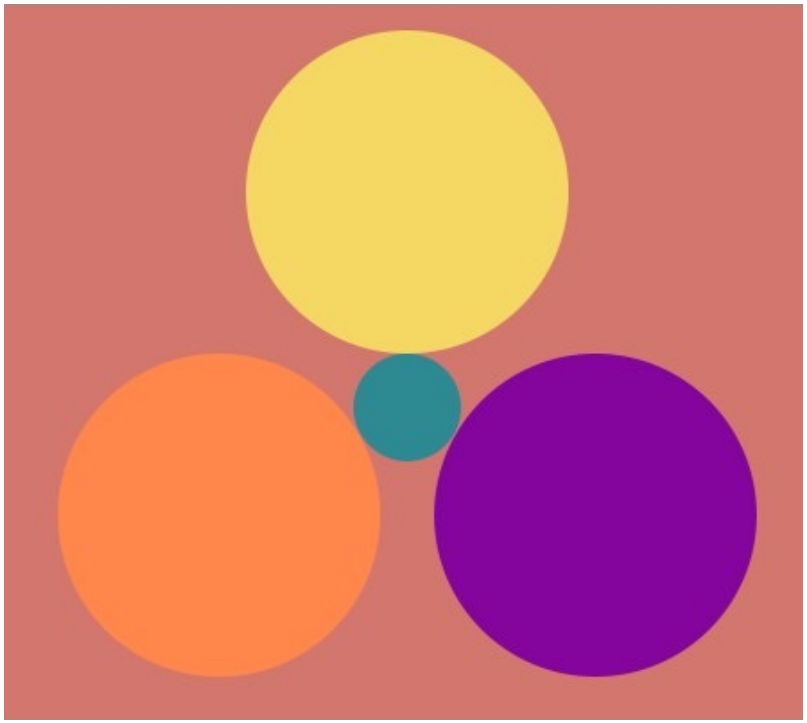
Given 3 input colors, C1, C2, C3, provided by an artist:

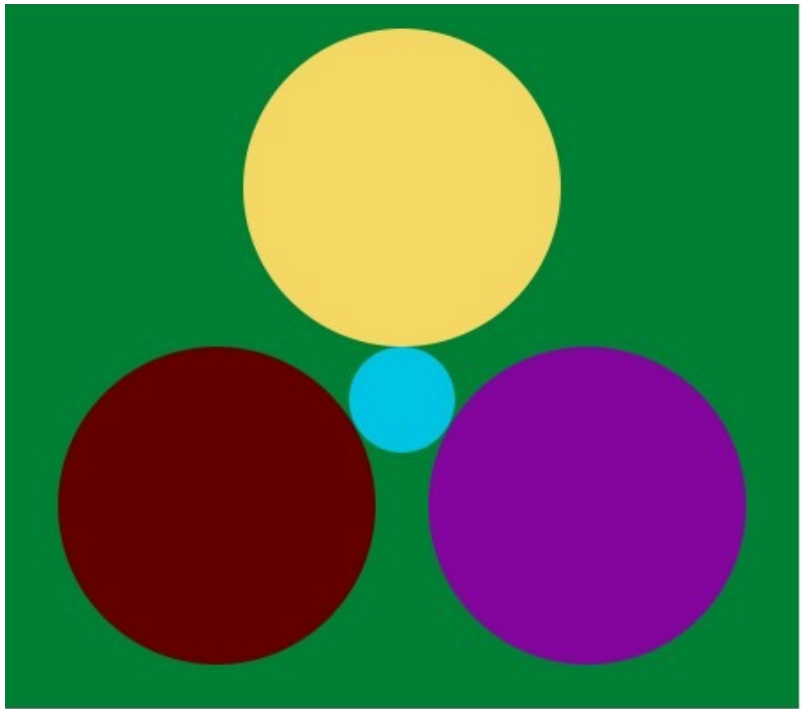
1. Define and show a nice average of these (which I will use as background)
2. Define and show a nice approximating ramp from C1 to C3, that passes near C2
3. Define and show a nice interpolating ramp from C1, through C2, to C3
4. Define and show a nice approximating loop through C1, C2, and C3
5. Define and show a nice interpolating loop through C1, C2, and C3
6. Define and compute a nice accent (i.e., highlight) color for these and their ramps
7. Define and show a nice sampling of colors in the triangle (C1,C2,C3)

Will please (not offend) the viewer

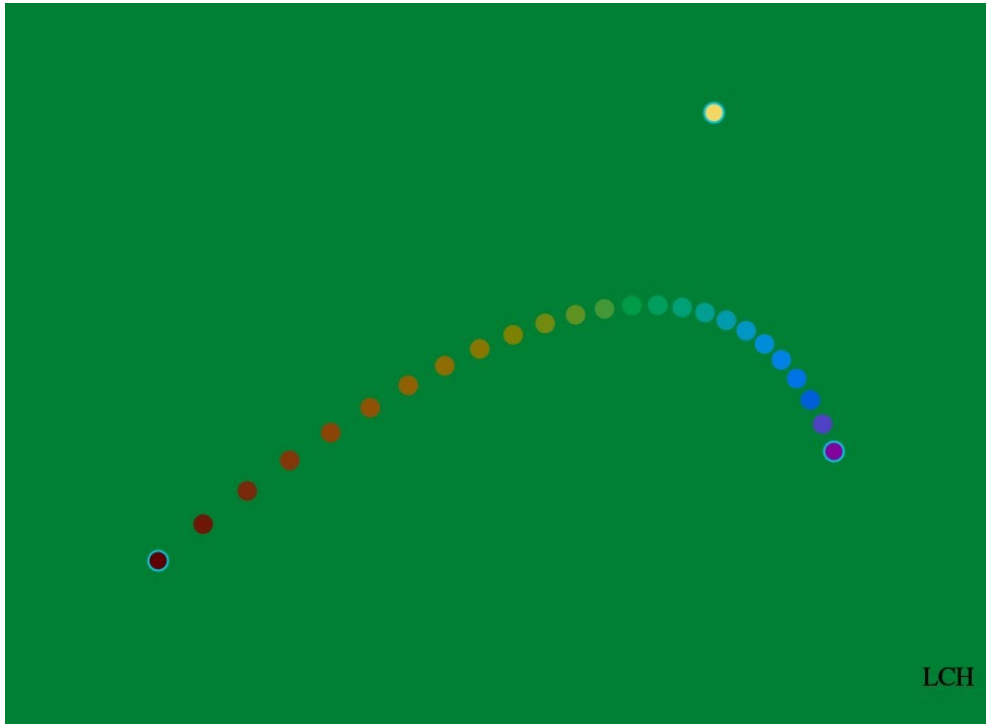
1. No surprising artifacts (accelerations?) in ramps/loops/samplings
2. Appears to perfectly fit the input colors
3. May be easily controlled by the input colors (effect of changing a color: not surprising)
4. Input colors play symmetric roles (except for ramp)

Concrete example of average

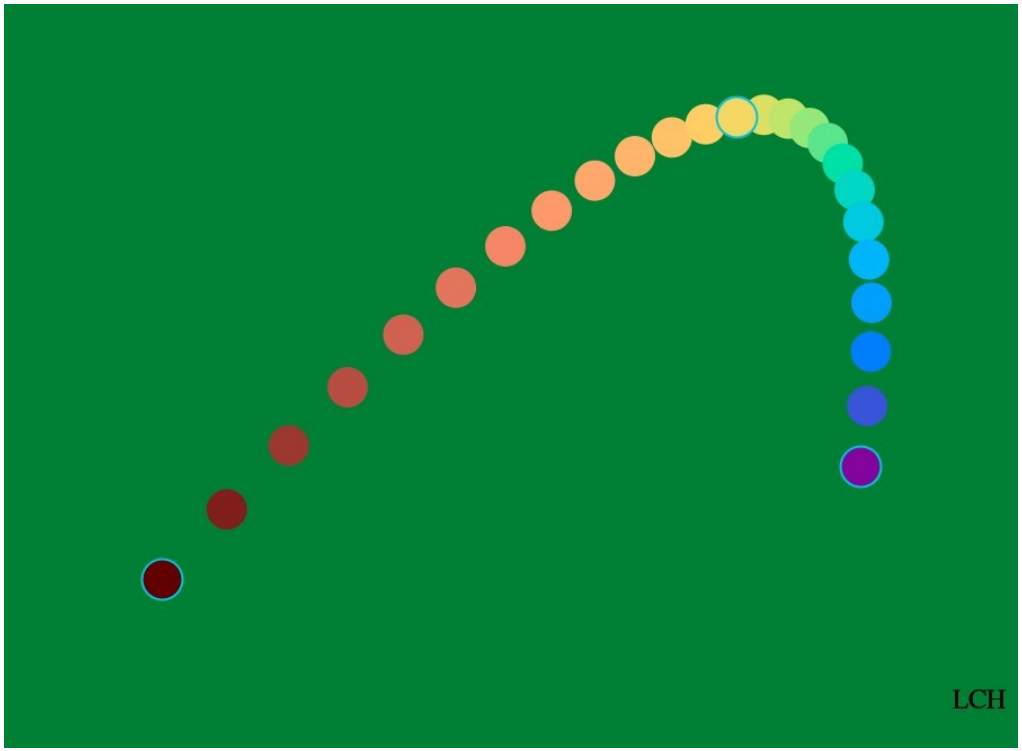




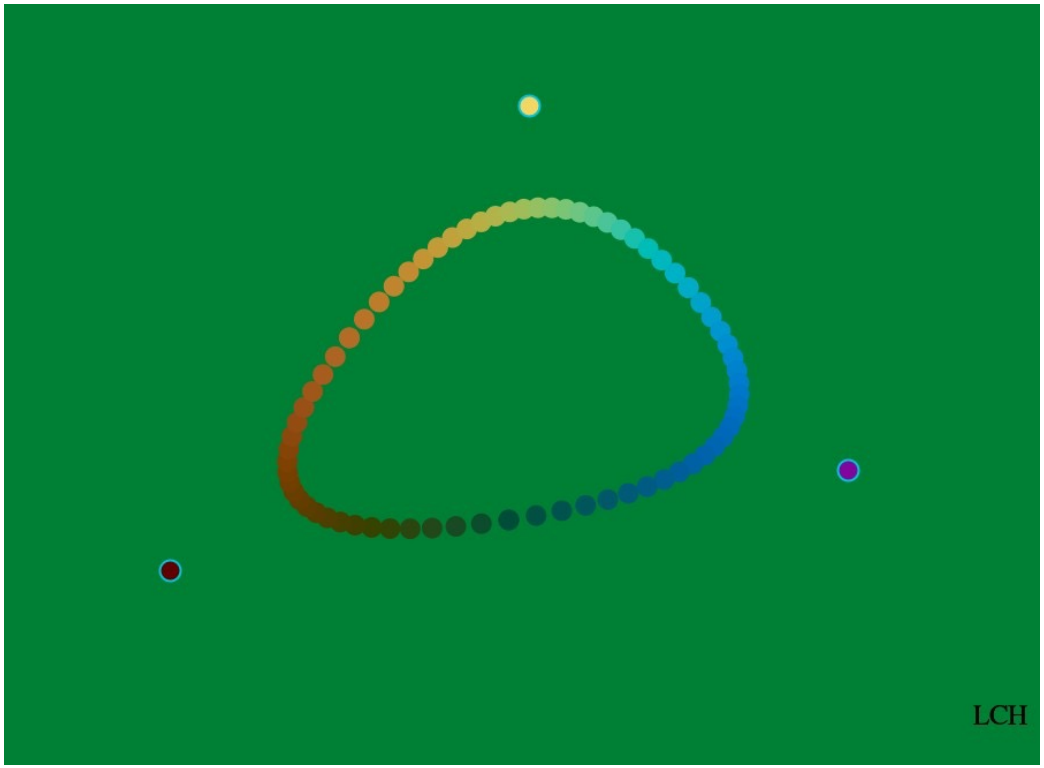
Concrete example of approximating ramp



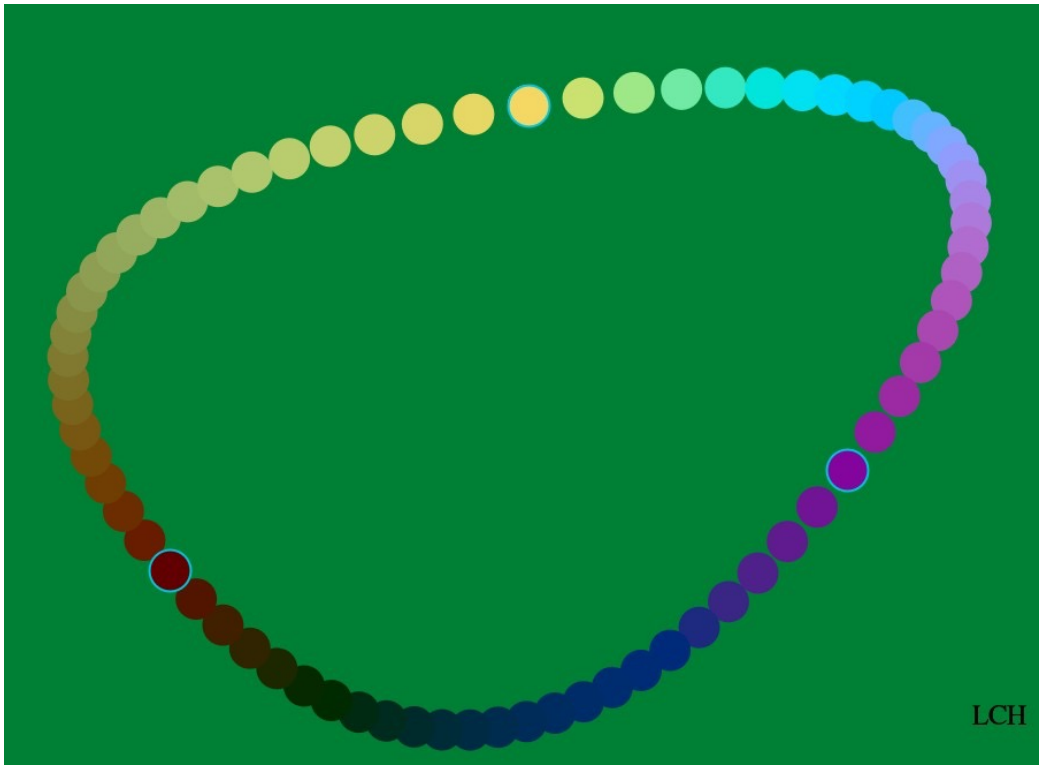
Concrete example of interpolating ramp



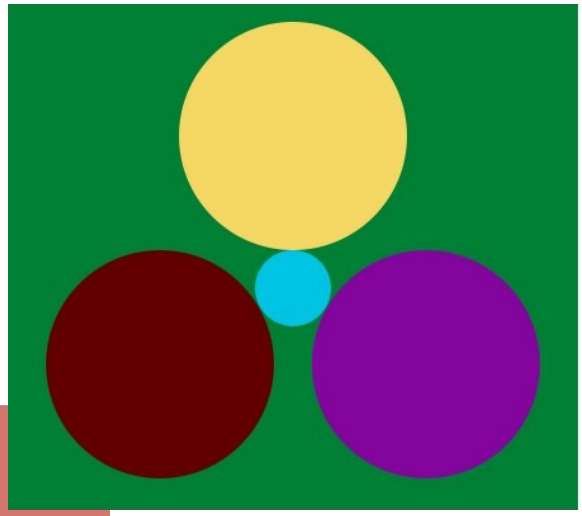
Concrete example of approximating loop

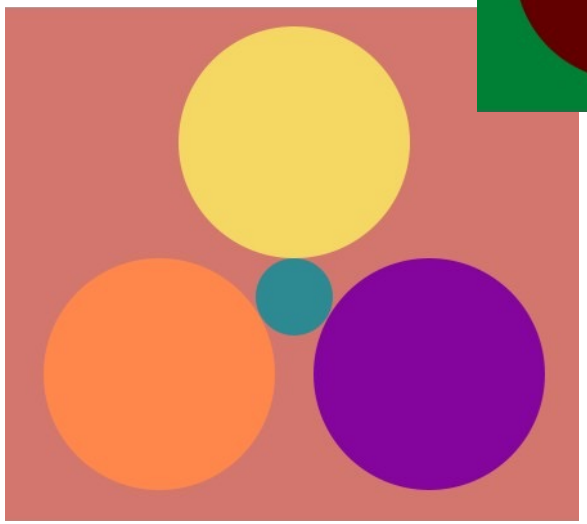


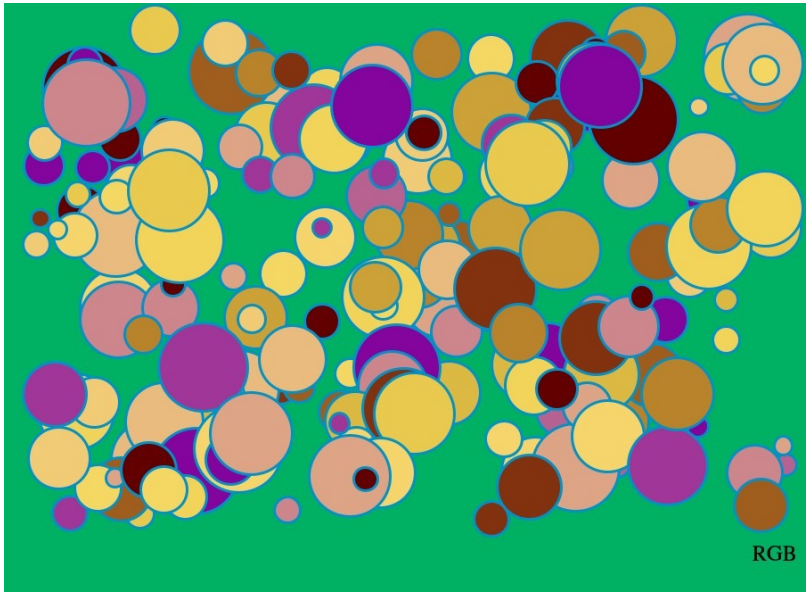
Concrete example of interpolating loop



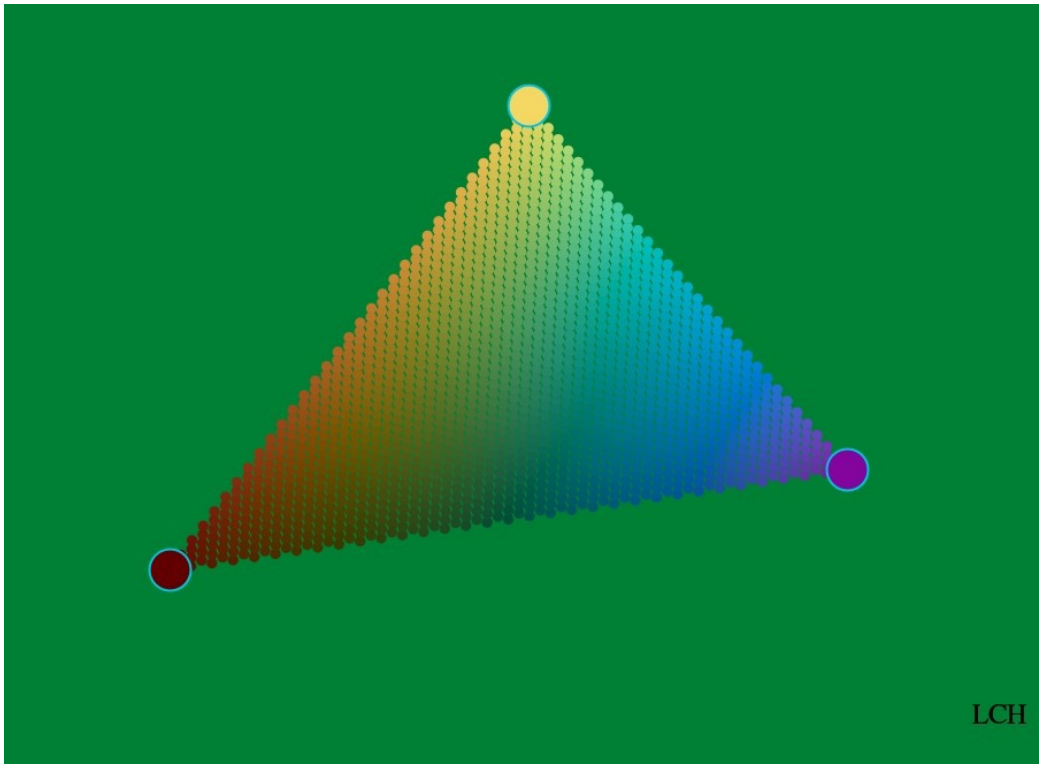
Concrete example of accent







Concrete example of sampling



What does “color” mean?

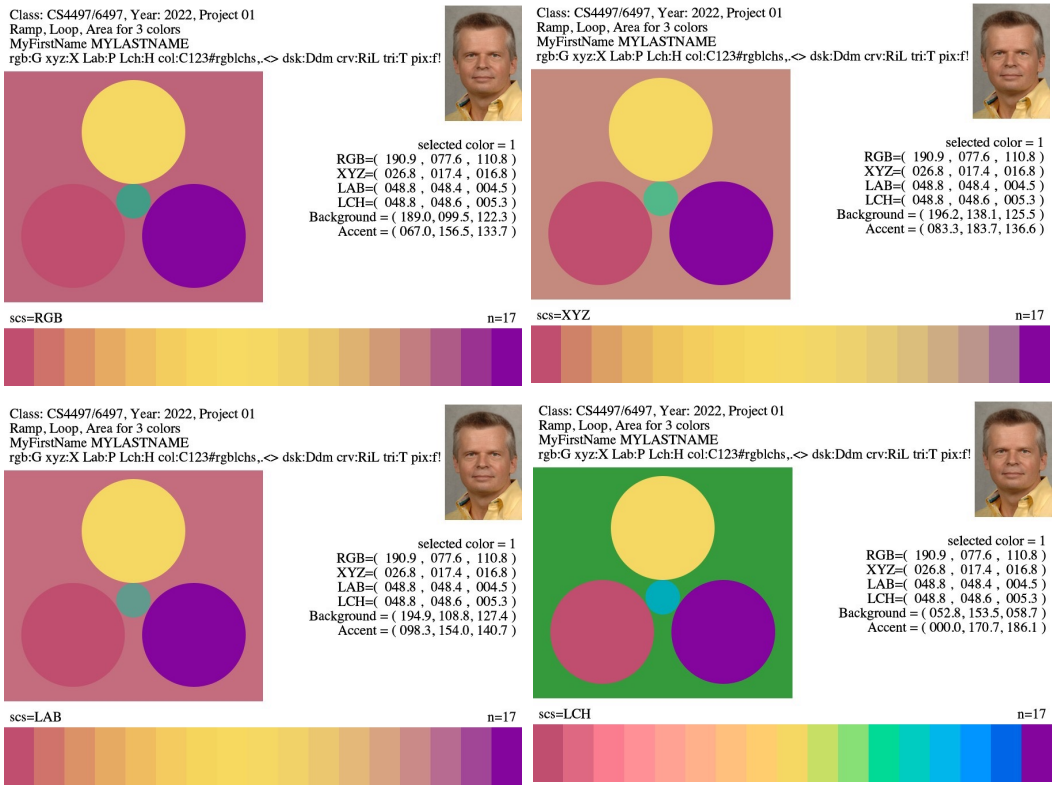
Suggestions?

Questions?

Discussion?

What is a reasonable strategy?

- Compute average, accent, ramps, loops, & samples in different **color spaces**
- Compare the results for different input sets
- Discuss benefits/limitations and recommend which space is best for what



How to navigate these color spaces?

Color spaces are **three-dimensional** (see human perception of colors)

A color may be represented as a **point** in one of these **spaces**

I provide you with mappings (**transformations**) between these color spaces

1. RGB (red, green, blue)
2. XYZ
3. Lab
4. Lch (lightness, chroma, hue)

We will discuss these before the project starts

Compare the results on all 4 of these for different input sets

Discuss benefits/limitations and recommend which space is best for what

For extra credit, you may add a new color space that looks more promising

What does “accent” mean?

Suggestions?

Discussion?

What does “accent” mean?

You should show some related discussions and examples, but do not expect to find the answer online

- if you do, include reference and
- use critical thinking to show that you understand its limitations and ambiguities

Enjoy **creative** exploration of an **ambiguous** specification

To construct your accent (point in 3D) out of the plane through the input colors, you may have to define and implement functions for

- Defining points and vectors
- Combining them as weighted averages, sums, and differences
- Computing their dot product, norm, cross product

We will discuss these before the project starts

Big picture questions / discussions?

Why is computing color averages, accents, ramps, loops… palettes important?

Why do we need to worry about several color spaces?

Why is this 3D? Why not just the color wheel?

Big picture questions / discussions?

Why is computing color averages, accents, ramps, loops… palettes important?

Does color matter (in life, in business)?

Why should Georgia Tech students (rather than artists) worry about this?

Why do we need to worry about several color spaces?

Why is this 3D? Why not just the color wheel?

Answer these questions concisely, but clearly and concretely, in your report!

Processing sketch (baseCode) provided

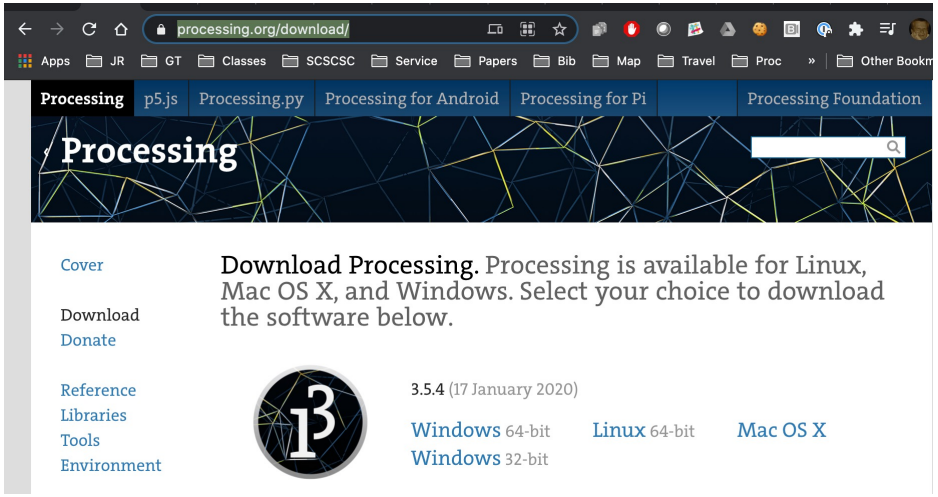
You do not HAVE to use it, but must implement all the functionalities discussed above

Download Processing

Download recent version from:

https://processing.org/download/

https://processing.org/download/



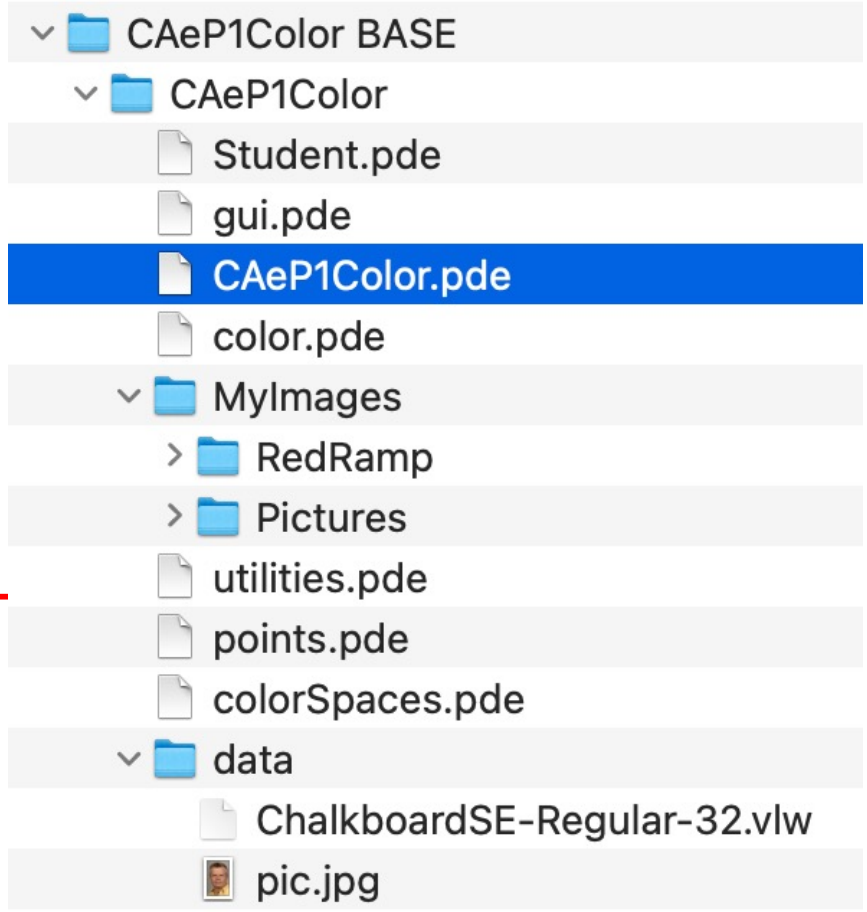
Download and unzip baseSketch1 for project

Download CAeP1ColorBASE.zip from Canvas

Unzip

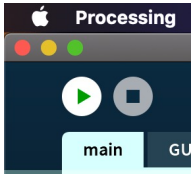
Start Processing

Open CAeP1Color.pde

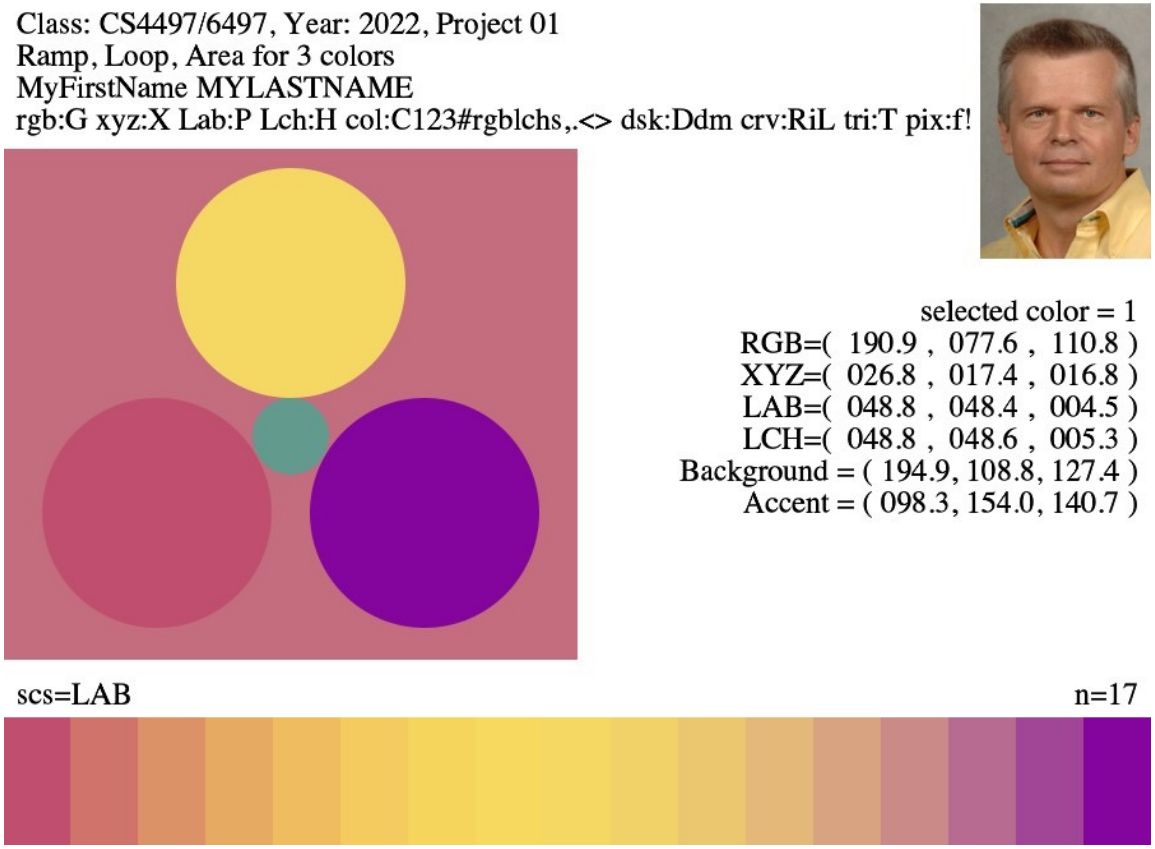


Check that it runs

Press the run button (green arrow)



You should see this:



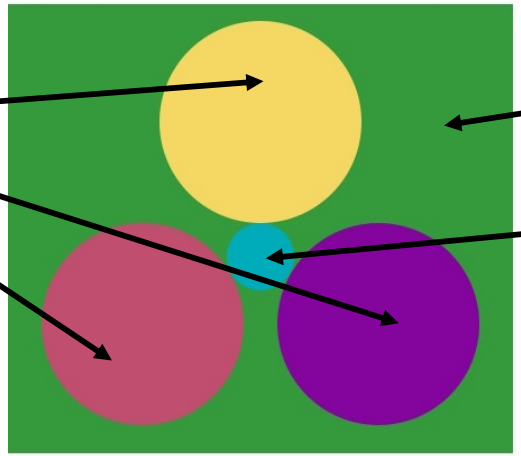
COVER MODE ('C')

Change to your name and picture

Class: CS4497/6497, Year: 2022, Project 01
Ramp, Loop, Area for 3 colors
MyFirstName MYLASTNAME
rgb:G xyz:X Lab:P Lch:H col:C123#rgblchs,.<> dsk:Ddm crv:RiL tri:T pix:f!

Input colors, C1, C2, C3

Average



selected color = 1
RGB=(190.9 , 077.6 , 110.8)
XYZ=(026.8 , 017.4 , 016.8)
LAB=(048.8 , 048.4 , 004.5)
LCH=(048.8 , 048.6 , 005.3)
Background = (052.8, 153.5, 058.7)
Accent = (000.0, 170.7, 186.1)

Accent

Selected Color Space 'G'=RGB, 'X"=XYZ, 'P'=Lab, 'H'=Lch

Number of colors in ramp: change with ',', '.', '<', '>' keys

scs=LCH

n=17

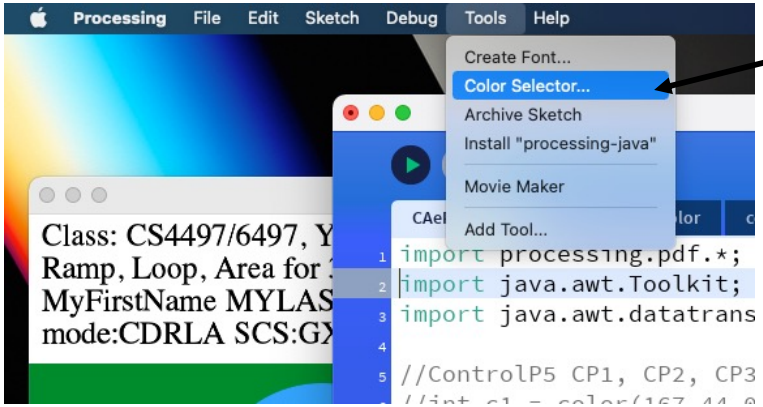
Color ramp of n samples through C1, C2, C3

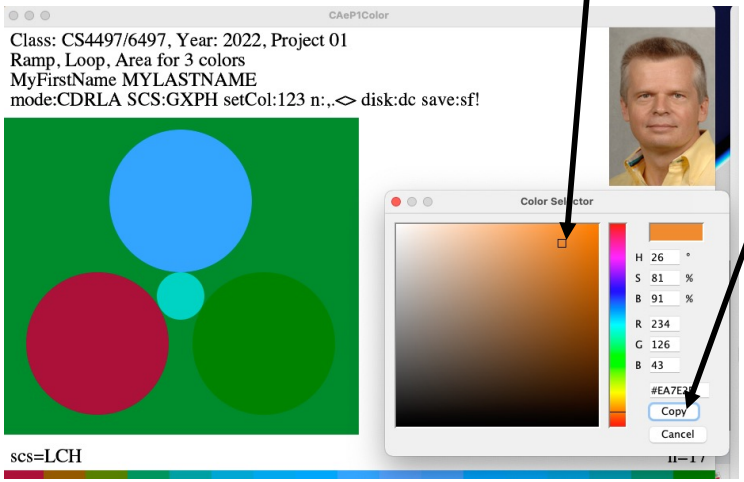
Change input color 1

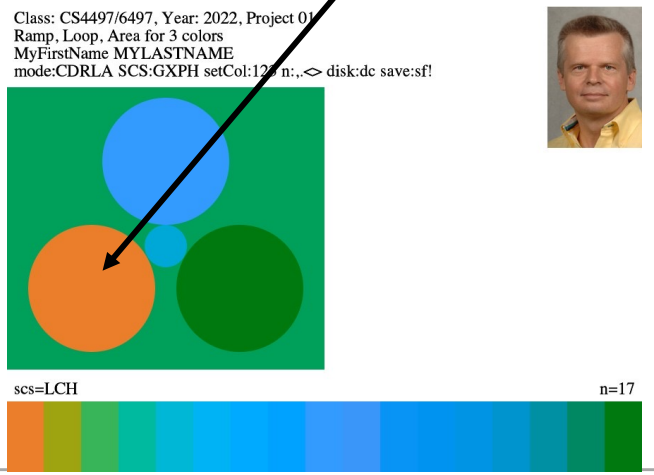
In editor, start color selector

In selector,
pick color, press copy

In sketch, press ‘1’





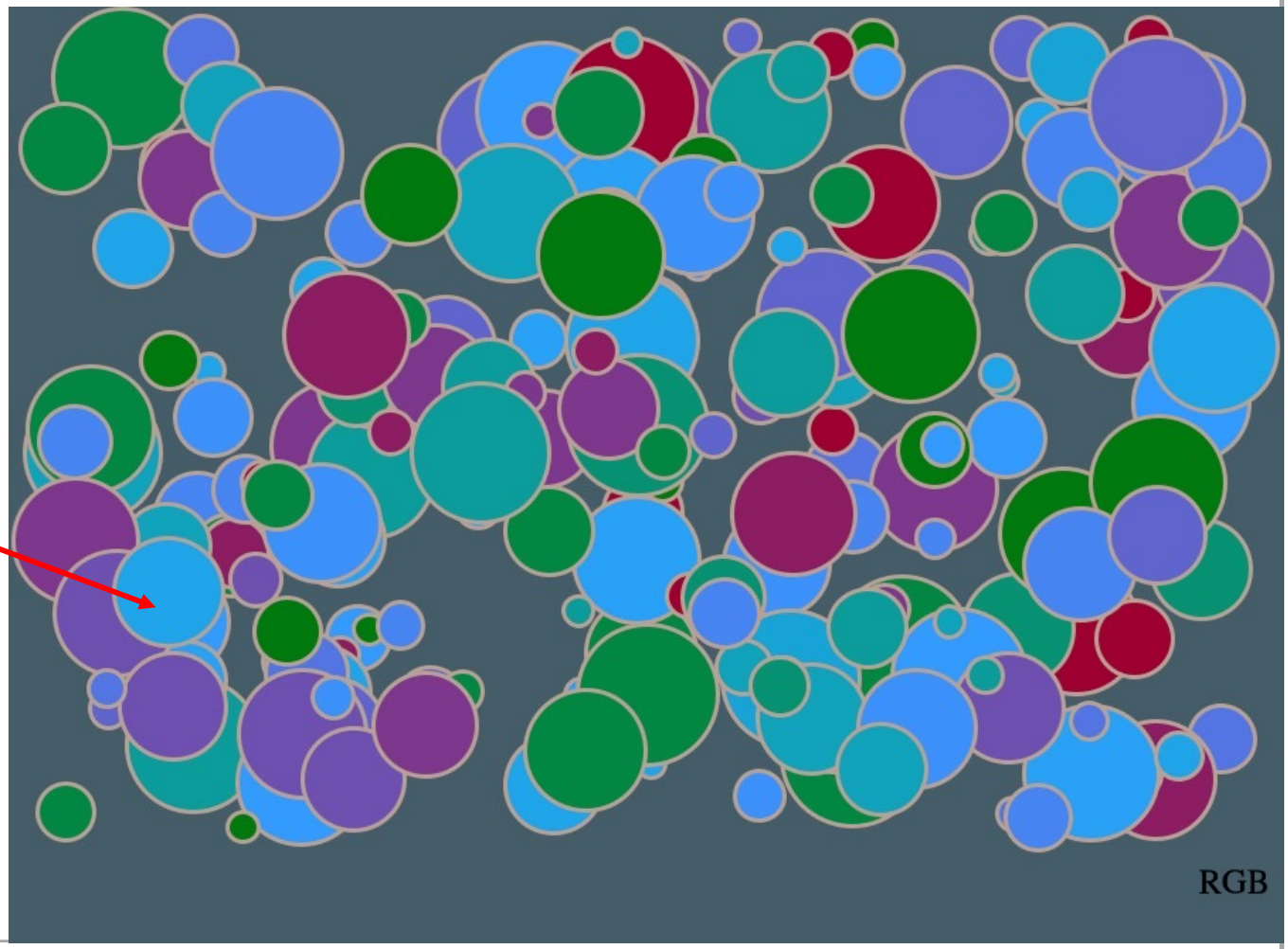


DISK MODE ('D')

Random disks
colored randomly
using samples from
your ramp

'c' to reassign colors

'd' to recreate disks



RAMP MODE ('R')

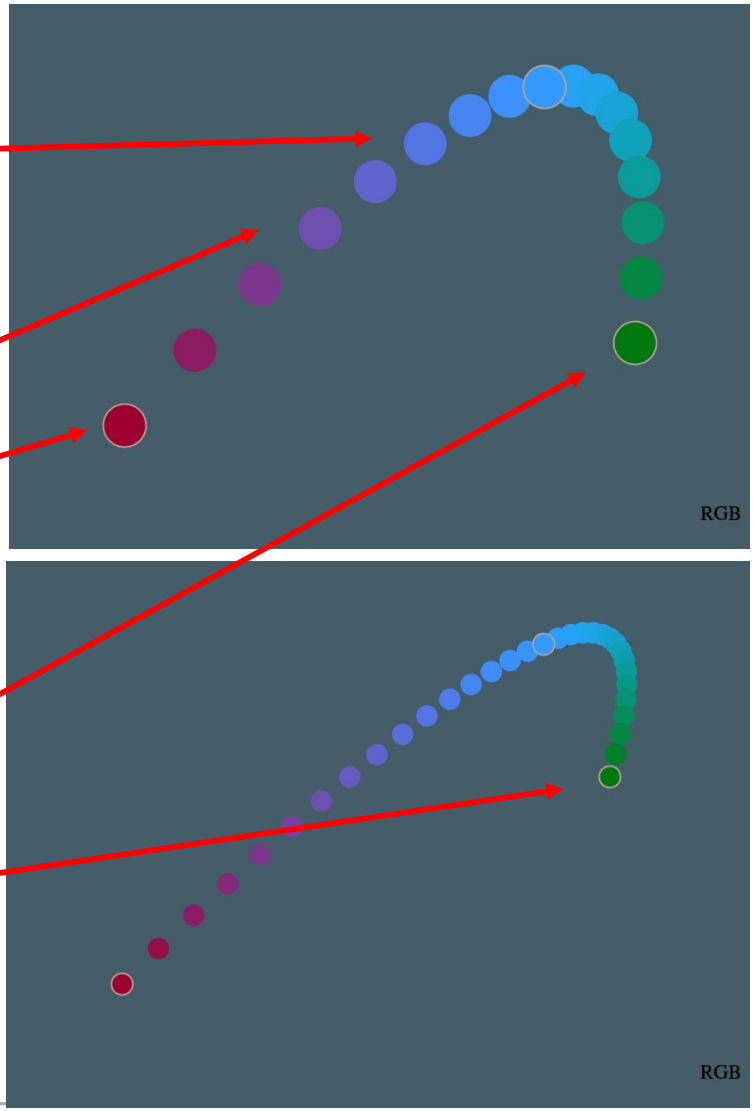
Interpolating ramp Show disk samples using interpolating position and ramp color

Background = average in SCS

Circle color = highlight in SCS

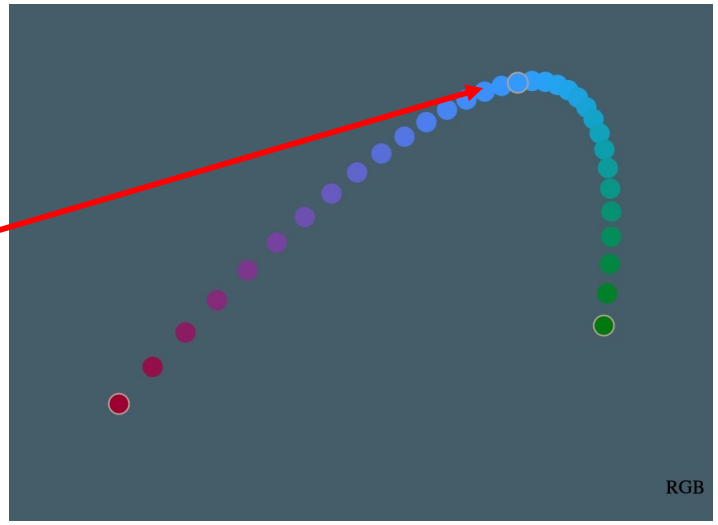
Press mouse button near one of the 3 control disks to select it

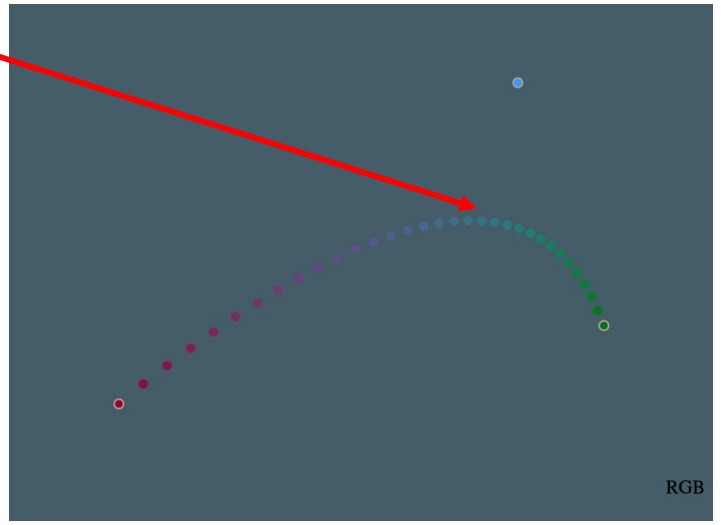
Drag it
Then release the mouse button
This does not affect the colors



Approximating RAMP MODE ('R')

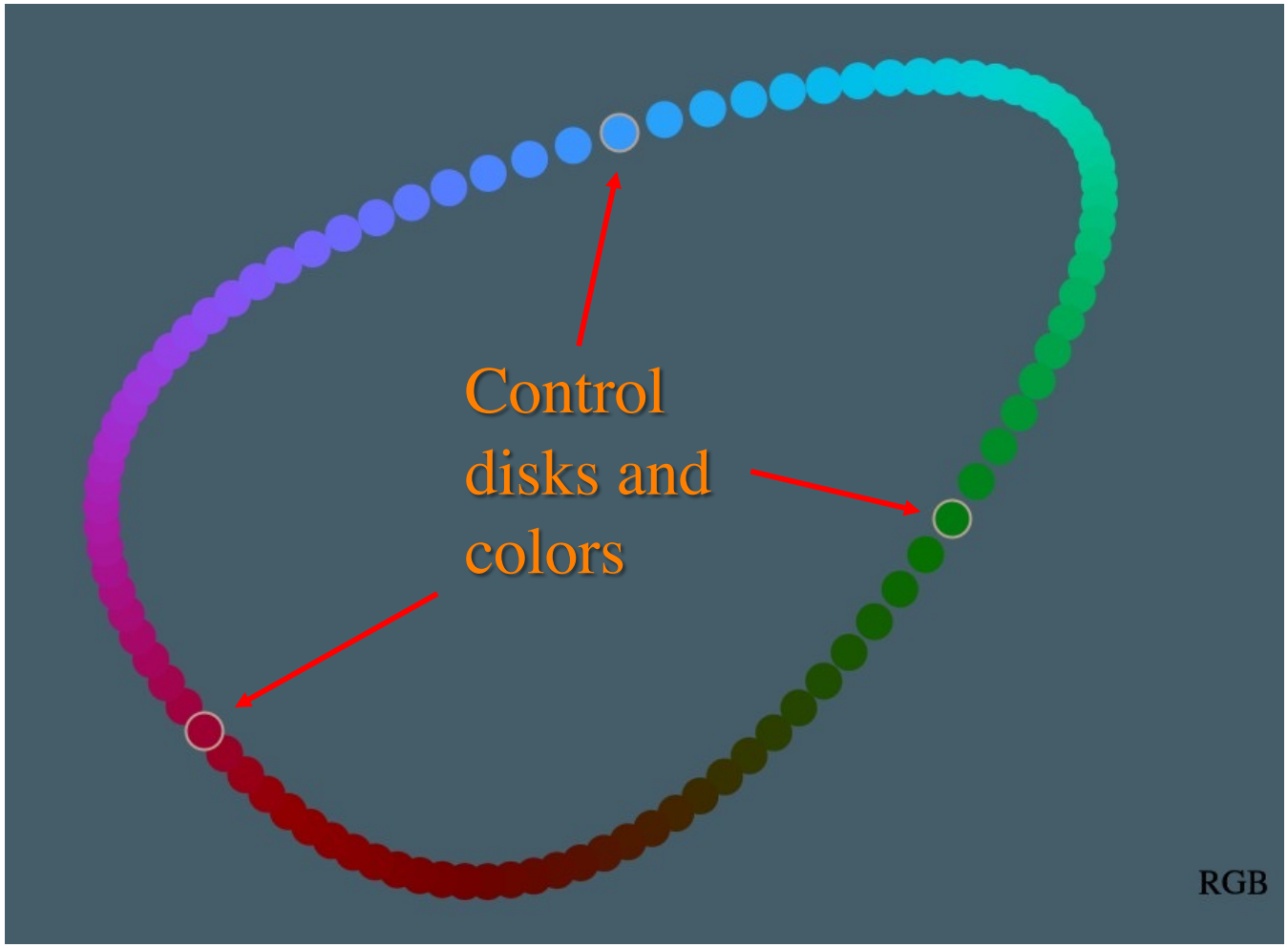
Use 'I" to toggle between interpolating and approximating modes for disks and for colors





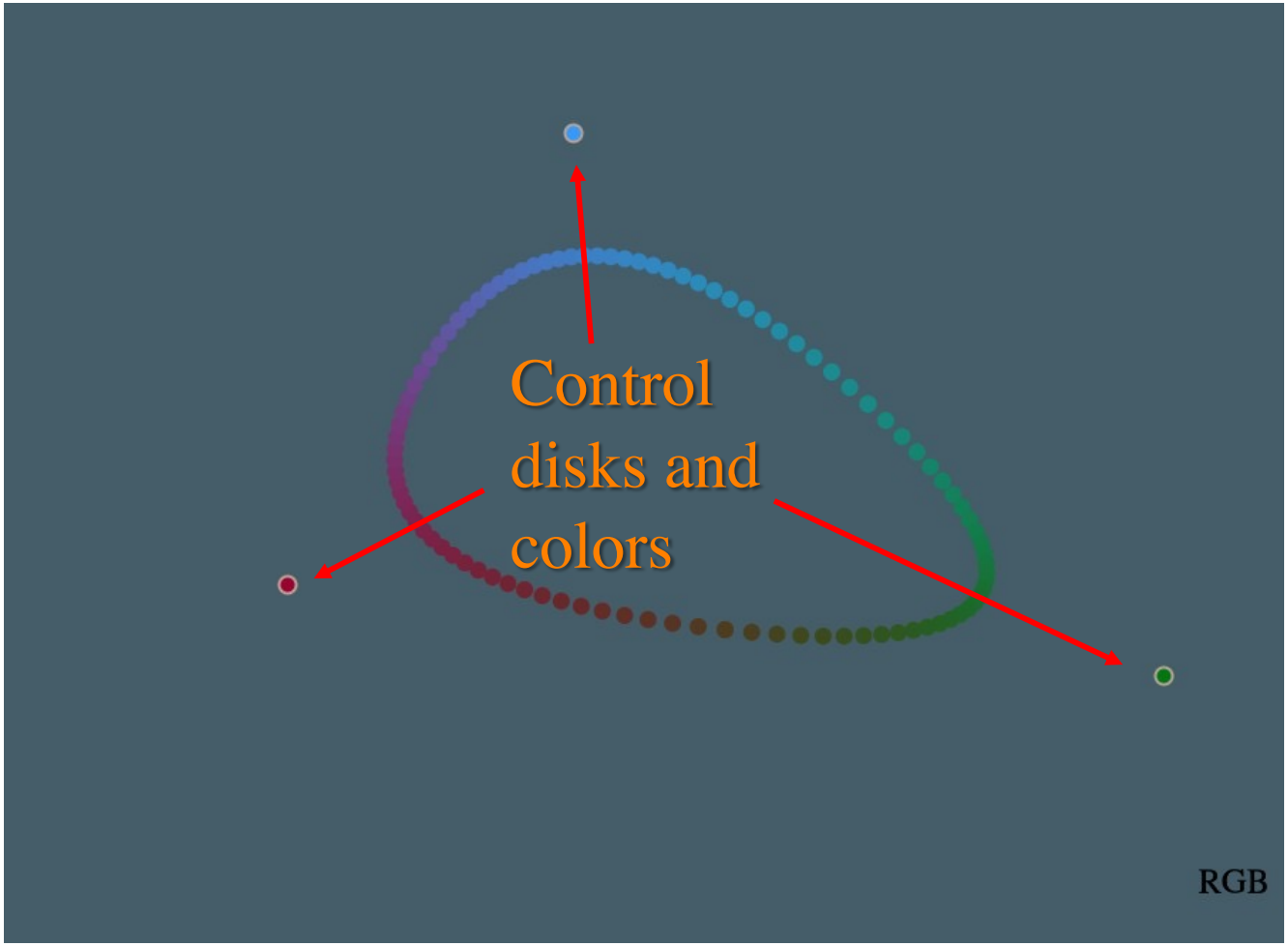
Interpolating LOOP MODE ('L')

Implement the interpolation in the plane and in the SCS



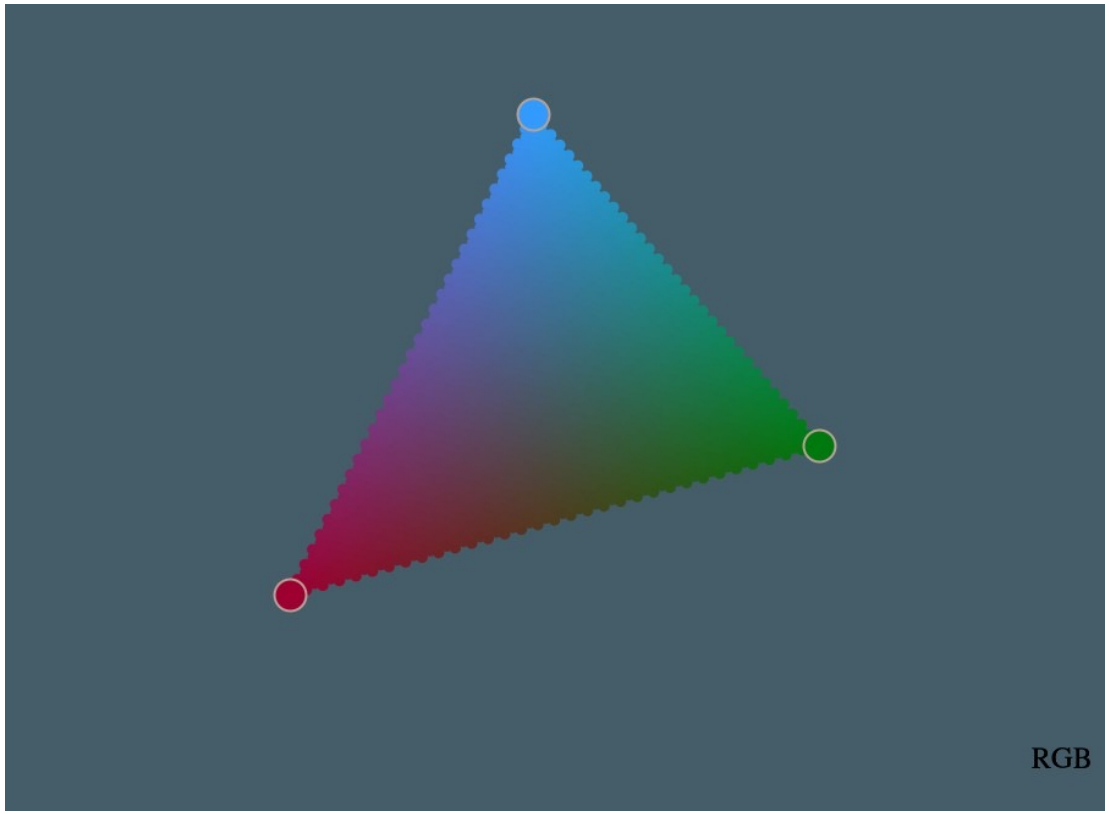
Approximating LOOP MODE ('L')

Implement the approximation in the plane and in the SCS

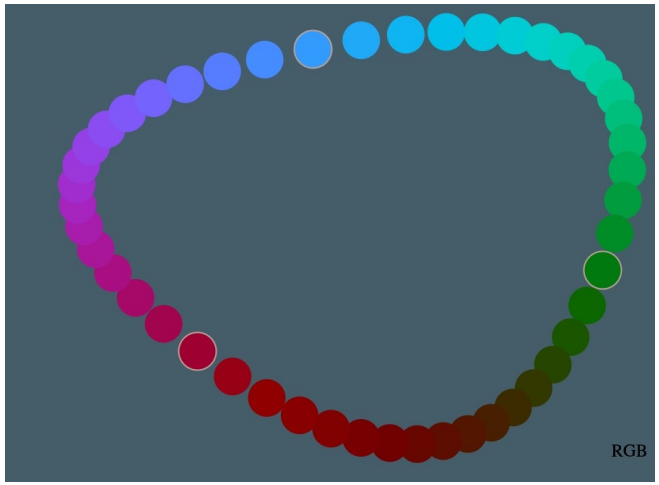


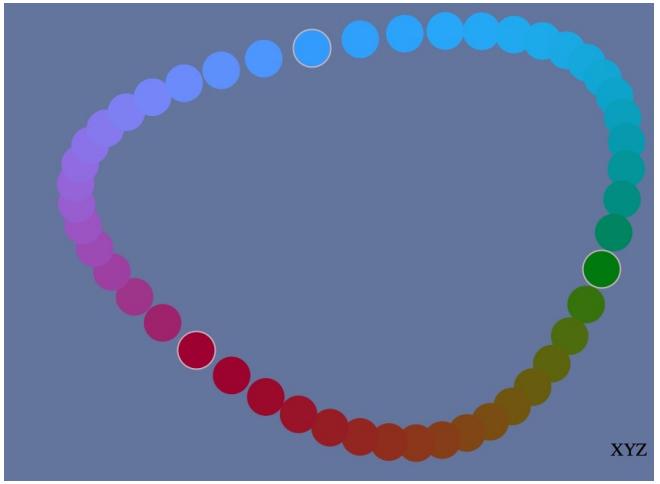
TRIANGLE MODE ('T')

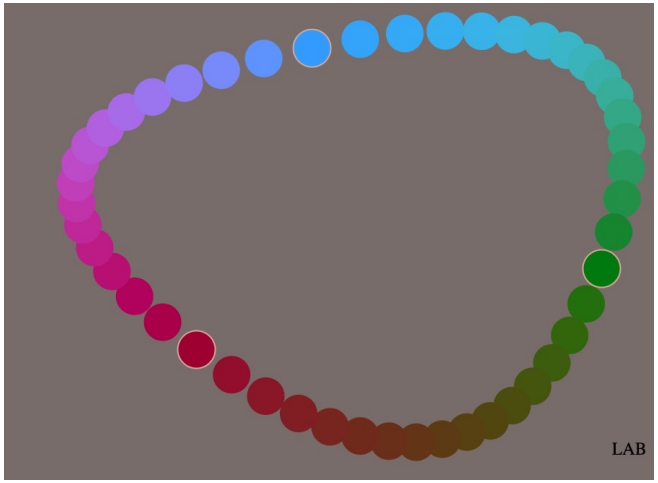
Implement the sampling of triangle-area and the proper coloring of the samples in the SCS



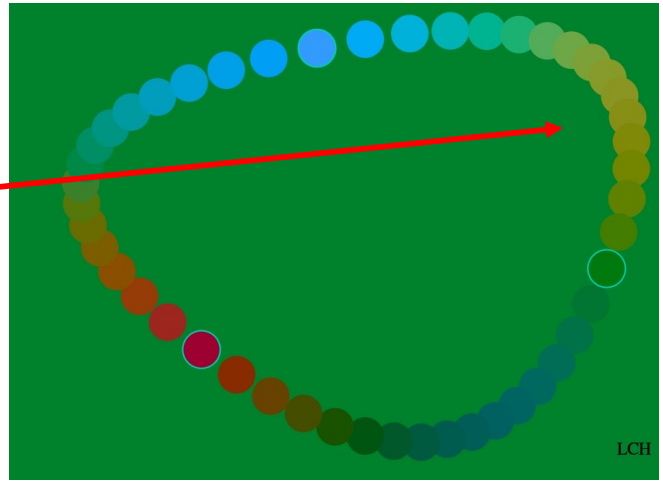
Compare the four SCS





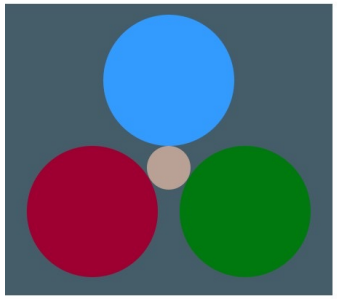


?



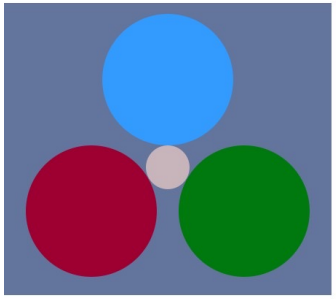
Compare SCS benefits using ramp smoothness

Class: CS4497/6497, Year: 2022, Project 01
Ramp, Loop, Area for 3 colors
MyFirstName MYLASTNAME
mode:CDRLA SCS:GXPH setCol:123 n:,.<> disk:dc save:sf!



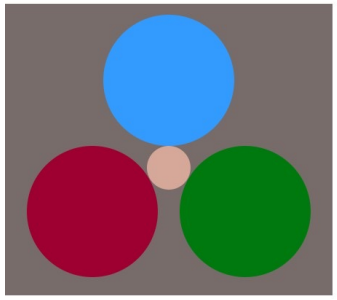
scs=RGB n=33

Class: CS4497/6497, Year: 2022, Project 01
Ramp, Loop, Area for 3 colors
MyFirstName MYLASTNAME
mode:CDRLA SCS:GXPH setCol:123 n:,.<> disk:dc save:sf!

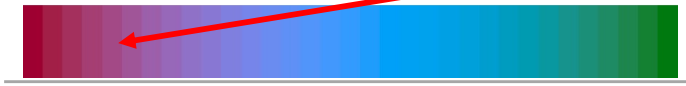


scs=XYZ n=33

Class: CS4497/6497, Year: 2022, Project 01
Ramp, Loop, Area for 3 colors
MyFirstName MYLASTNAME
mode:CDRLA SCS:GXPH setCol:123 n:,.<> disk:dc save:sf!

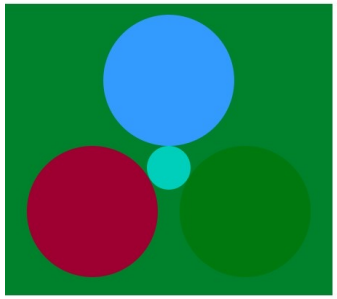


scs=LAB n=33



?

Class: CS4497/6497, Year: 2022, Project 01
Ramp, Loop, Area for 3 colors
MyFirstName MYLASTNAME
mode:CDRLA SCS:GXPH setCol:123 n:,.<> disk:dc save:sf!



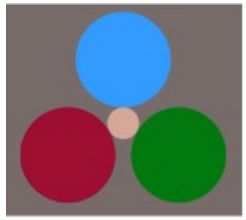
scs=LCH n=33

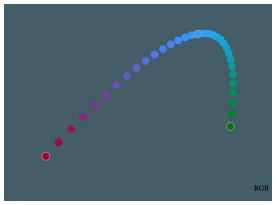
Project deliverables 1: Implementation

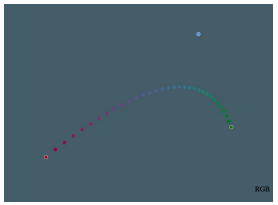
Use base sketch or write your own code from scratch or use other color pickers/tools

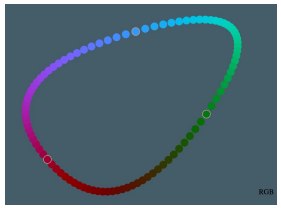
Given 3 input colors, your code should compute and show:

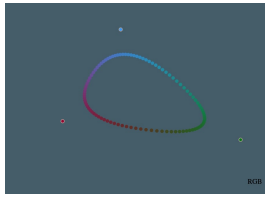
1. The average (background) and highlight (border, center disk)
2. A ramp of n samples that interpolates/approximates them
3. A loop of n samples that interpolates/approximates them
4. Random disks with highlight border over average background
5. A sampling of the interior of the triangle

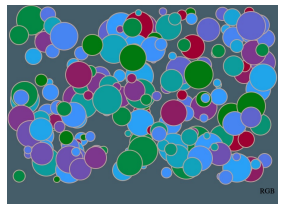


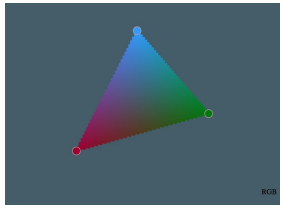












Project deliverables 2: Report

PDF with a short report (compliant with the 7Cs of good technical writing)

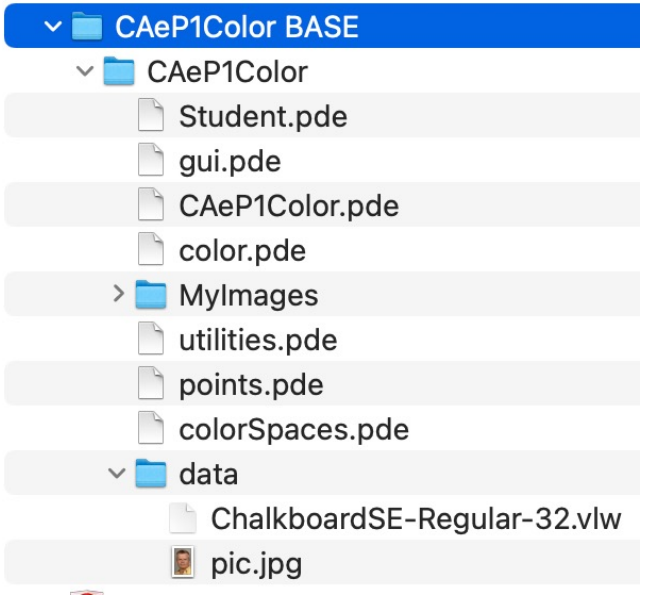
Title, name of course/project, your name and small picture

Short sections:

1. Why is **color** important in life, industry, society?
2. What is a beautiful **color scheme**? Why is it important?
3. What is a **color space**? Why is it 3D? Pros/Cons of RGB, XYZ, Lab, Lch?
4. Pros/Cons using **average** as **background** ? How is it computed? In which color space?
5. How do you compute weights u(a,b,c,t), v(a,b,c,t), w(a,b,c,t) such that **interplant**
   P(t)=u(a,b,c,t)A + v(a,b,c,t)B + w(a,b,c,t)C satisfies P(a)=A, P(b)=B, P(c)=C
6. Pros/Cons of color spaces for computing the **ramp** (illustrate with examples)
7. Details of how you implemented the **loop approximation** and the **interpolation**?
8. How do you **sample the triangle area**?
9. What are common uses of **highlight**? How do you compute it?
10. Propose / demonstrate a **better highlight** (extra credit).

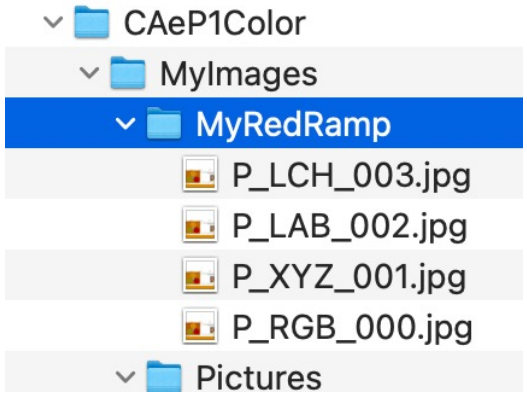
Base sketch: Folder

Change pic.jpg to your photo (150x200 pix)



Pressing '!' saves images into MyImages
Tags them with the SCS (selected color space)
and numbers them

Copying "MyRedRamp" into the clipboard and pressing 'f' will create a subfolder with that name and save your images there

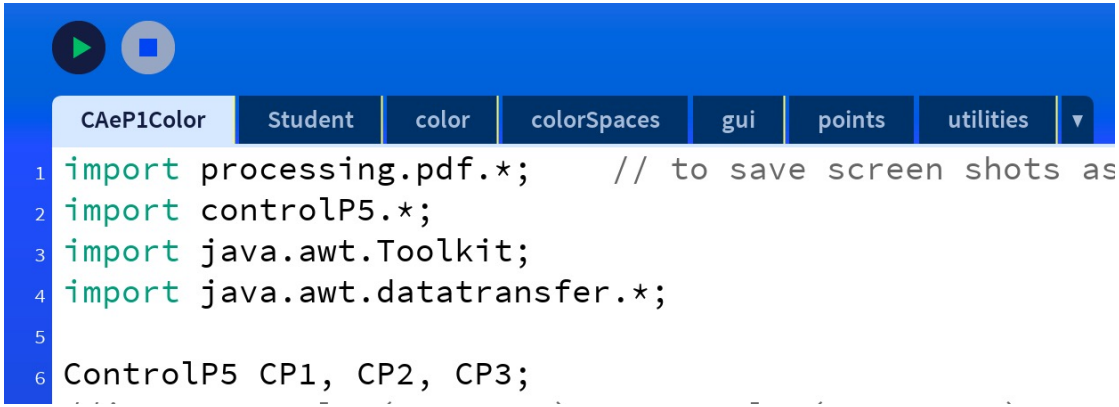


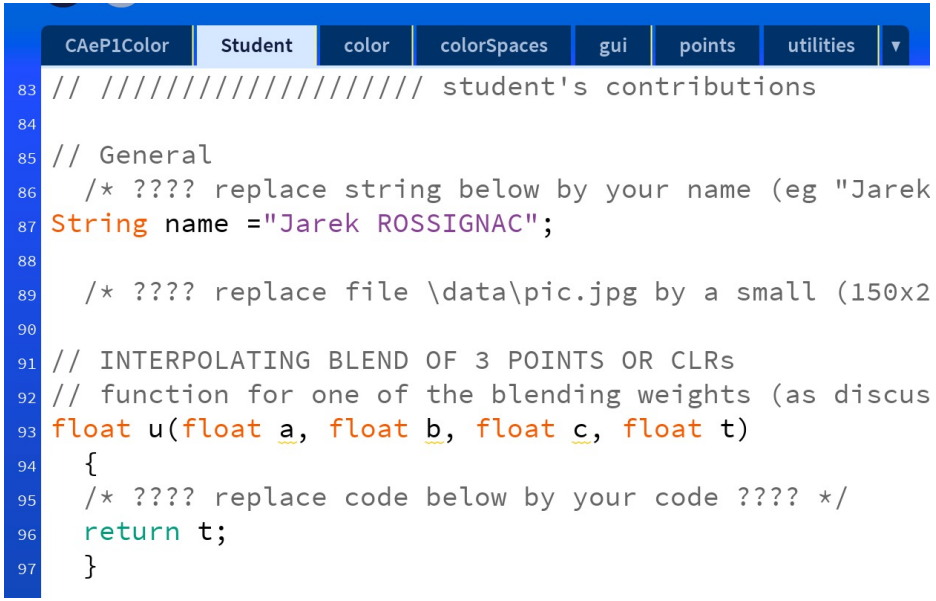
Base sketch: Structure

Sketch has several tabs

Main tab is CAeP1Color

You should edit the Student tab as suggested in the ????...????





Base sketch: gui

What key actions do

Picture: f !

Change n: , . < >

Change SCS: 1 2 3 4

Change mode: t d

Make disks / colors: D c

Save colors: s

- Into clipboard that you can paste
- Into CAeP1Color

CAeP1Color | Student | color | colorSpaces | gui | points | utilities

```
void keyPressed()
  {
  if(key=='!') {snapJPG=true; println("saved pix to: MyImages/"+PicturesF
  if(key==',') {n=max(3,n-1);  r=width/n/2;}
  if(key=='.') {n++; r=width/n/2;}
  if(key=='<') {n=int((n+1)/2);  if(n<3) n=3; r=width/n/2;}
  if(key=='>') {n=2*n-1; r=width/n/2;}
  if(key=='1') {SCSvalue=_RGB; SCSname = "RGB"; makeColors();}
  if(key=='2') {SCSvalue=_XYZ; SCSname = "XYZ"; makeColors();}
  if(key=='3') {SCSvalue=_LAB; SCSname = "LAB"; makeColors();}
  if(key=='4') {SCSvalue=_LCH; SCSname = "LCH"; makeColors();}
  if(key=='s') setClipboard("int c1 = color"+Si(C1)+", c2 = color"+Si(C2)
  if(key=='f') {PicturesFileName=getClipboard(); println("PicturesFileNam
  if(key=='d') demoDisks=!demoDisks;
  if(key=='c') makeColors();
  if(key=='D') makeDisks();
  if(key=='t') test=!test;
  }
String guide="SCS:1234 n:,.<> disk:dcD test:t save:sf!"; // help info
```

CAeP1Color | Student | color | colorSpaces | gui | points | utilities

```
int c1 = color(175,60,0), c2 = color(199,166,1), c3 = color(93,137,0);
// when running the sketch, to save the current 3 colors as initial colors, press 's'
```

Base sketch: points

Tools for defining/combining points and drawing disks/rectangles

Class PNT

Function P() in line 16 is in Student

You need to write it! Using u()…?

CAeP1Color | Student | color | colorSpaces | gui | points | utilities

```
92  // function for one of the blending weights (as discuss
93  float u(float a, float b, float c, float t)
94    {
95    /* ???? replace code below by your code ???? */
96    return t;
97    }
98
99  // Makes point at parameter s along pareabola through p
100 PNT P(float a, PNT A, float b, PNT B, float c, PNT C, f
101   {
102   /* ???? replace code below by your code ???? */
103   return B;
104   }
105
```

CAeP1Color | Student | color | colorSpaces | gui | points | utilities

```
1  class PNT
2    {
3    float x=0,y=0;
4    PNT (float px, float py) {x = px; y = py;}
5    }
6  PNT P() {return P(0,0); }
7  PNT P(float x, float y) {return new PNT(x,y); }
8  PNT P(PNT P) {return P(P.x,P.y); }
9  void disk(PNT P, float r, color c) {fill(c); ellips
10 float d2(PNT P, PNT Q) {return sq(Q.x-P.x)+sq(Q.y-P
11 float d(PNT P, PNT Q) {return sqrt(d2(P,Q)); }
12 PNT P(float a, PNT A, float b, PNT B) {return P(a*A
13 PNT P(float a, PNT A, float b, PNT B, float c, PNT
14 PNT P(PNT A, PNT B){return P(0.5,A,0.5,B);}
15 PNT P(PNT A, PNT B, PNT C){return P(1./3,A,1./3,B,1
16 PNT P(PNT A, PNT B, PNT C, float t){return P(0,A,0.
17
18 void                                w, float h, color c) {}
19
20
21
22
23
```

Cut
Copy
Copy as HTML
Paste
Select All

Comment/Uncomment
→ → Increase Indent
← ← Decrease Indent

Find in Reference
Show Usage...
Jump to Declaration
Rename

saved edRamp/P1000.jpg

Base sketch: color (CLR)

I made a CLR class

- Container for the 3 color values
- In any of the color spaces (unspecified)

I gave you constructors, extractors, and conversions

CAeP1Color | Student | color | colorSpaces | gui

```
2  class CLR
3    {
4    float x=0,y=0,z=0;
5    CLR (float px, float py, float pz)
6    }
7
8  ////////////////////////////////////
9  CLR K(float x, float y, float z) {ret
10 CLR K() {return K(0,0,0); }
11 CLR K(color c) {return K(red(c),greer
12 CLR K(CLR C) {return K(C.x,C.y,C.z);}
13 CLR I(CLR C) {return K(255-C.x,255-C.
14
15 ////////////////////////////////////
16 color c(CLR C) {return color(C.x,C.y,
17 String S(CLR C) {return "("+str(C.x)+
18 String Si(CLR C) {return "("+str(int(
19
20 ////////////////////////////////////
21 CLR RGBtoLAB(CLR C) {return KofD(RGBt
22 CLR RGBtoLCH(CLR C) {return KofD(RGBt
23 CLR RGBtoXYZ(CLR C) {return KofD(RGBt
```

Base sketch: Conversion between color spaces

My CLR converters are in tab colorSpaces and are using java code from

http://rsbweb.nih.gov/ij/plugins/download/Color_Space_Converter.java

CAeP1Color | Student | color | gui | points | utilities

```
// FROM: http://rsbweb.nih.gov/ij/plugins/download/Color_Space_Conver
// See also https://physics.stackexchange.com/questions/487763/how-ar

public float[] D65 = {95.0429, 100.0, 108.8900};
public float[] whitePoint = D65;
public float[][] Mi  =  {{ 3.2406, -1.5372, -0.4986},
                         {-0.9689,  1.8758,  0.0415},
                         { 0.0557, -0.2040,  1.0570}};
public float[][] M   =  {{0.4124, 0.3576,  0.1805},
                         {0.2126, 0.7152,  0.0722},
                         {0.0193, 0.1192,  0.9505}};

// lightness: 0<L<100, cyan-magenta: -100<a<100, blue-yellow: -100<b<
float L=74,a=5, b=5;
float L0=57, a0=-88, b0=-91;
float L1=74, a1=95, b1=65;

float rgb_max=256;
```

CAeP1Color | Student | color | gui | points

```
51 ////////////////////////////////////
52     // XYZfromRGB RGBfromXYZ
53     // LABfromRGB RGBfromLAB
54     // LCHfromRGB RGBfromLCH
55     // XYZfromLAB LABfromXYZ
56     // LABfromLCH LCHfromLAB
57
58
59 CLR XYZfromRGB(CLR RGB)
60   {
61   float R=RGB.x, G=RGB.y, B=RGB.z;
62   float r = R / rgb_max;
63   float g = G / rgb_max;
64   float b = B / rgb_max;
65   // assume sRGB
```

Base sketch: Student

You should not have to write more than ~50 LoC

1 LoC for each of: u(), P(), WAK()

10 LoC for showTri()

Not sure about KaccentInSCS()

CAeP1Color | Student | color | colorSpaces | gui | points | utilities

```
// INTERPOLATING BLEND OF 3 POINTS OR CLRs
// function for one of the blending weights (as discussed in the
float u(float a, float b, float c, float t)
  {
  /* ???? replace code below by your code ???? */
  return t;
  }

// Makes point at parameter s along pareabola through points A, I
PNT P(float a, PNT A, float b, PNT B, float c, PNT C, float t)
  {
  /* ???? replace code below by your code ???? */
  return B;
  }

// Makes color at parameter s along pareabola through CLRs A, B,
CLR WAK(float a, CLR A, float b, CLR B, float c, CLR C, float t)
  {
  /* ???? replace code below by your code ???? */
  return B;
  }

// Show colored disks inside triangle of points (A,B,C) with ass
void showTri(PNT A, PNT B, PNT C, CLR a, CLR b, CLR c, float r,
  {
  /* ???? change this to some 12 lines of code below to implemen
  disk(P(A,B,C),r,c(KblendInSCS(a,b,c)));
  }

// Improve the computation of the accent color
CLR KaccentInSCS(CLR A, CLR B, CLR C)
  {
```

ToDos check list for your first project

Make sure that you have done the following:

1. Swap my picture for yours
2. Enter your name
3. Implement the functions (????...????) in tab Student, debug them
4. Make useful pictures for your report
5. Write report and save as YourNAME.PDF (up to 4 pages in 10 point font)
   1. See deliverables above, include figures and references
6. Remove the folder MyImages
7. Put your PDF inside the folder CAeP1Color
8. Make sure that the **data** subfolder is there
9. Compress the folder CAeP1Color
10. Make sure that it decompresses, runs, and contains the report
11. Submit the compressed file

Due date:

Before class: 2 weeks (after project starts)

Start working on it now, make steady progress, plan to finish early

Grading

50% for running code with all functionalities running correctly

50% for nice report (Clear, Correct, Concrete, Complete, Concise, Convincing)

To get more than 'A', do the extra credit (and/or be Clever):

- Better highlight relative to a color scheme (c1, c2, c3)
- Insight about what a good color scheme palette is
- create an application (mock-up web page) using your best palette
- include overview of prior art for color schemes selection (web pages, interior, clothing)

Graduate students (CS6497):

- Higher standards in the write up
- Some interesting extension / application

Teamwork

Students may form teams of 2 or 3 partners (no more)

- TAs will provide an initial matching
- Feel free to opt to do this alone or to form a different team of 2 or 3 only

Your team members must be listed in your report

Team members are allowed to

- Exchange and discuss ideas,
- Compare each other's solutions,
- Proofread each other's write-up and point out where it is not C-compliant

Team members should not

- Exchange (or even show each other their code)
- Communicate with other teams