

Basic Course Information

Title:	CTE Generative Art/Design Plus Unity AB
Transcript abbreviations:	
Length of course:	Full Year
Subject area:	Visual & Performing Arts (F) / Visual Arts
UC honors designation?	No
Prerequisites:	CTE Computer Science 1 AB: 60-40-09/10 (a-g) G or CTE AP Computer Science A (Java) AB (a-g) C
Co-requisites:	None
Integrated (Academics / CTE)?	Yes: Software and Systems Development
Grade levels:	10th, 11th, 12 th
Course learning environment:	Classroom Based

Course Description

Course overview:

Students will learn fundamental art/design topics – color, shape/form and light – typically taught in an introductory traditional studio art course, but as they relate to the concerns of *GENERATIVE ART* and design. Other topics are specific to "marks" that can only be generated by computer. The *medium* comprises instructions to a computer graphics rendering system, encoded in a computer program written in the programming language Processing, and applied to 2D and 3D virtual spaces.

Section 1 of the course will comprise several topics from Part 2 of the book *Generative Design: Visualize, Program, and Create with Processing*, Authors: Bohner et al. 2012 . Other units and material will come from other similarly-themed books and sources. There are 9 Processing units: (1) Color; (2) Shape; (3) Light; (4) Randomness and noise; (5) Oscillation figures; (6) Formulated bodies; (7) Attractors; (8) Data Visualization; (9) Music Visualization. Videos/Images for many of the exercises in these units can be viewed at: <http://www.generative-gestaltung.de/1/code>.

Units 1, 2 and 3 (**color, shape/form and light**) also implicitly comprise principles and criteria of **composition**. Additionally, the shape/form unit includes a multi-week study of traditional pen and ink calligraphy. The light unit includes a month-long unit in photography. Teaching of these two traditional Visual Arts (fine arts) along with Generative Art (media arts) provides students connections / reference points between these two broad categories.

Units 4-8 introduces students to **techniques** for designs that can only be rendered programmatically. After completion of all five units, students are asked to design a composition using one or a combination of these techniques.

In Units 9-10, students study two examples for how to address visual design problems for (a) representation of large datasets and (b) a visual counterpart for experiencing music simultaneously as it plays in real time.

Section 2 is an introduction to Unity, a powerful cross-platform 3D "gaming" engine that can create Virtual/Augmented Reality Design spaces. The culminating project is a model/simulation of a Rubik's Cube. By section's end, students should be capable of passing the **Unity Certified User: VR Developer** certification exam.

Students entering the course will need to know the fundamentals of programming in Processing, a simplified form of Java with graphics capabilities. The programming prerequisite is either the UCOP-approved Computer Science 1 introductory programming course (or an equivalent course) or APCS-A.

As an integrated 2nd year programming course whose context and focus is art, it simultaneously allows students to (a) expand their programming knowledge and skill, and (b) apply these to design problems in the visual arts.

CA Arts Standards Addressed

VA/MA:Cr1 Generate and Conceptualize Artistic Ideas and Work

VA/MA:Cr2 Organize and Develop Artistic Ideas and Work

VA/MA:Cr3 Refine and Complete Artistic Work

VA/MA:Cr4 Select, Analyze, and Interpret Artistic Work for Presentation

VA/MA:Cr5 Develop and Refine Artistic Techniques and Work for Presentation

VA/MA:Cr6 Convey Meaning Through the Presentation of Artistic Work

VA/MA:Cr7 Perceive and Analyze Artistic Work

VA/MA:Cr8 Interpret Intent and Meaning in Artistic Work

VA/MA:Cr9 Apply Criteria to Evaluate Artistic Work

VA/MA:Cr10 Synthesize and Relate Knowledge and Personal Experiences to Make Art

Course content:

Unit 1: Color

Unit Summary

Students will review the RGB (Red,Green,Blue) additive model for the 16 million (2^{24}) possible colors one can generate programmatically. They will also explore color definitions based on the HSB model (Hue, Saturation, Brightness/Luminance).

Students will learn principles of Color Contrasts - the juxtaposition of color regions - as elaborated by Bauhaus artists

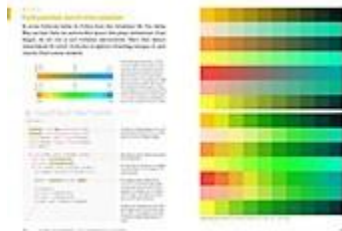
Jonannes Itten and Josef Albers. Per Itten's The Elements of Color, students will explore the following 6 categories: (a) Saturation; (b) Light/Dark; (c) Extension / Proportion / Region-Size; (d) Complements / Perceptual-Opposites; (e) Boundaries (Simultaneous); (f) Hue; and (g) Warm/Cool. Per Albers' The Interaction of Color, students will explore the overarching idea that perception of a color is subjective, affected by its neighboring colors.

For a related historical connection, students will read a brief history of the Bauhaus movement, and learn about its influence, particularly on architecture.

Unit Assignment(s)

One Key Assignment: Color palettes through interpolation. Students write 2 programs – using RGB and HSB – that create color gradients/paths from source color to target color using internal interpolation. Students compare / contrast the wildly different intermediate colors generated for RGB and HSB gradients. (MA:Cr1, MA:Cr2, MA:Cr3)

Video example of a color gradient: http://www.generative-gestaltung.de/1/P_1_2_1_01



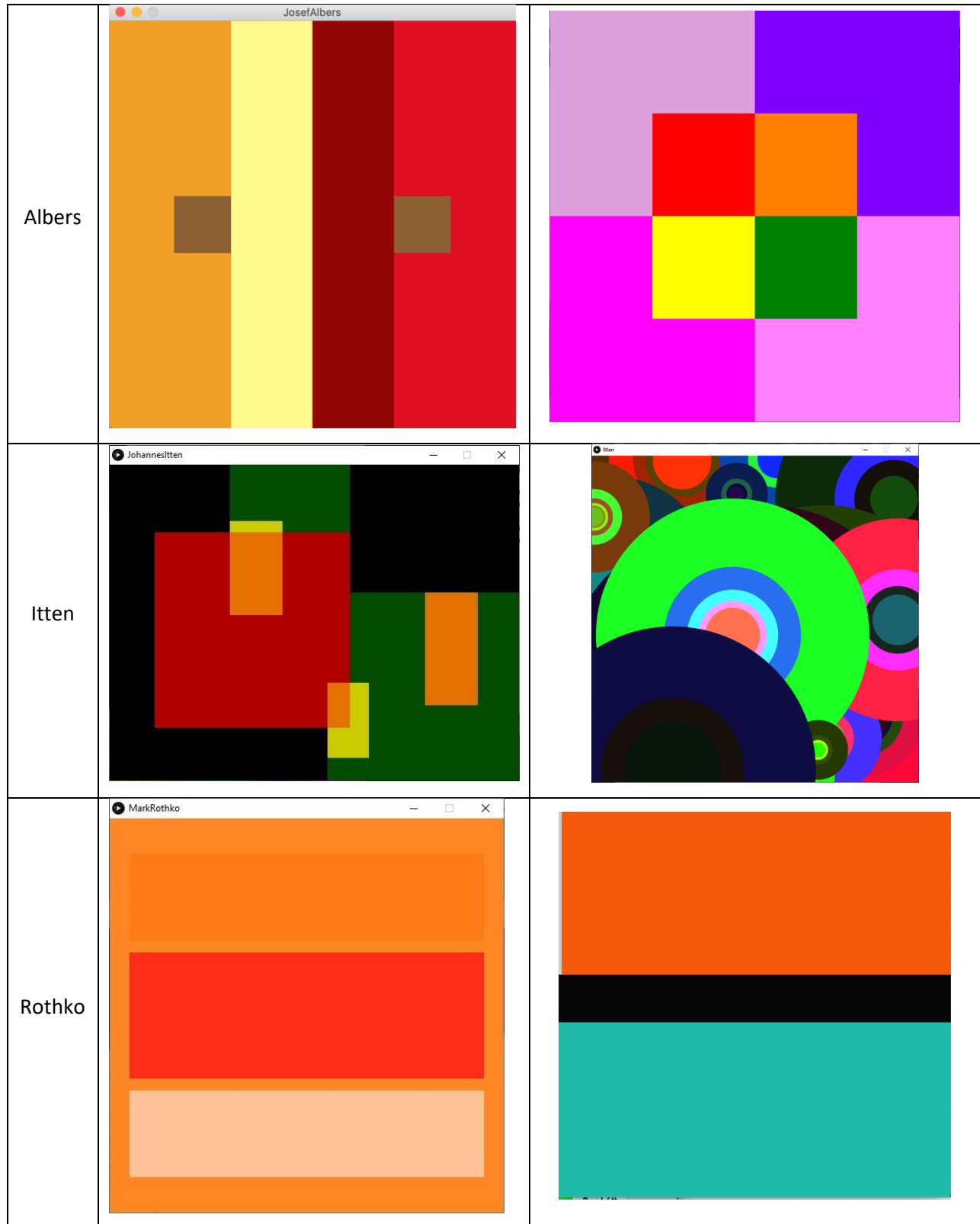
Questions:

1. How does one go about "sorting" colors, i.e. placing them in an order that forms a continuous gradient? What is the algorithm? (MA:Cr2, MA:Cr7)
2. Can any patterns be discerned in the gradients? (MA:Cr6, MA:Cr8)
3. Do the palettes "work", i.e. do they have a desired aesthetic effect, either harmonious or disharmonious? (MA:Cr4, MA:Cr7)

As with assignments in all units: Students "publish" their work on the classroom server and do a Virtual Gallery walk, followed by whole class discussion with each work-under-discussion shown on the large screen via the LCD projector and/or projected on each student's workstation using LanSchool. (MA:Cr4, MA:Cr5)

Examples of student work:

Using HSB colors, generate Programs where you both imitate – per your own interpretation of – the styles of color artists Albers, Itten and Rothko. (MA:Cr5, MA:Cr6, MA:Cr7, MA:Cr9, MA:Cr10)



Unit 2: Form / Shape

Unit Summary

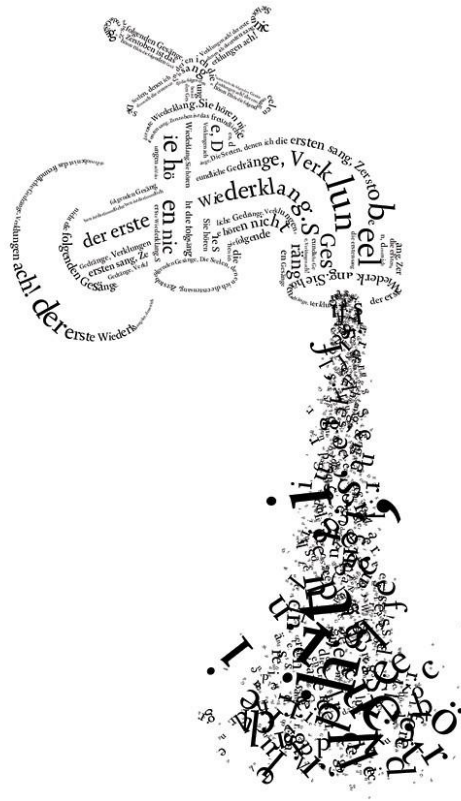
This rather large unit is an exploration of the visual effects of three distinct principles as applied to shapes or shape families: (1) Repetition using a grid; (2) Iteration using dumb and intelligent agents; and (3) Interaction via user drawing. Such compositions can be viewed on (at least) two levels of design building, much like in Pointillism, individual points of color comprise patterns that coalesce as recognizable representational objects. **The unit focuses specifically on patterns involving type / lettering.**

Students learn about the decorative characteristics of various Typefaces, Font Families and Calligraphic styles as literal text, design elements and/or textures that can be used in compositions. Before considering type, students practice **traditional Western pen nib and ink Calligraphy**, learning **Italic**, and either **Uncial** or **Fraktur** (a curved Blackletter font). Hand drawing Calligraphic letters teaches students to draw the shapes they see, rather than an interpretation of the shape. Students learn to manipulate and control Calligraphic pen and ink flow, and to become aware of the shapes of both: the marks needed to craft the letters and the white negative space those marks make. (VA:Cr1, VA:Cr2, VA:Cr3, VA:Cr4, VA:Cr5)

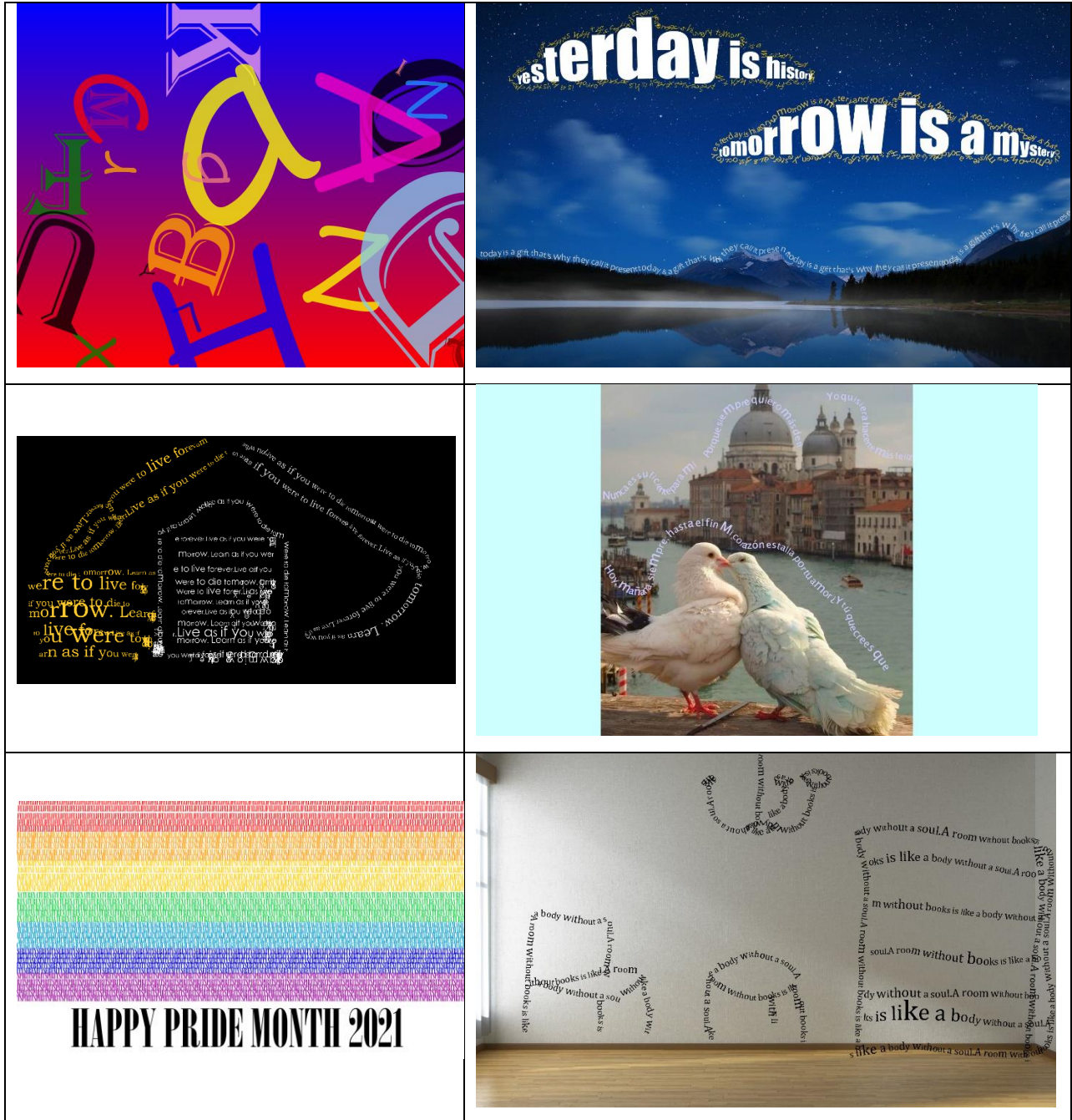
Unit Assignment(s)

One Key Assignment: Drawing with Animated Brushes (Principle 3, above): Drawing with Type (Letters). The program used is a much-enhanced version of the very basic program provided with the curriculum's resources. As the mouse moves, in its wake it leaves letters on the screen surface. The font family, size, angle and color of the letters can be controlled by various menu settings. Students draw three compositions: (1) Representational; (2) Abstract; and (3) a Word Cloud, where the composition takes word meanings into account.

Example of a representational composition (from curriculum):



Examples of student work (MA:Cr1-1, MA:Cr1-2, MA:Cr1-3, MA:Cr1-4, MA:Cr1-5, MA:Cr1-6, MA:Cr1-7, MA:Cr1-8, MA:Cr1-9, MA:Cr1-10):



Unit 3: Light

Unit Summary

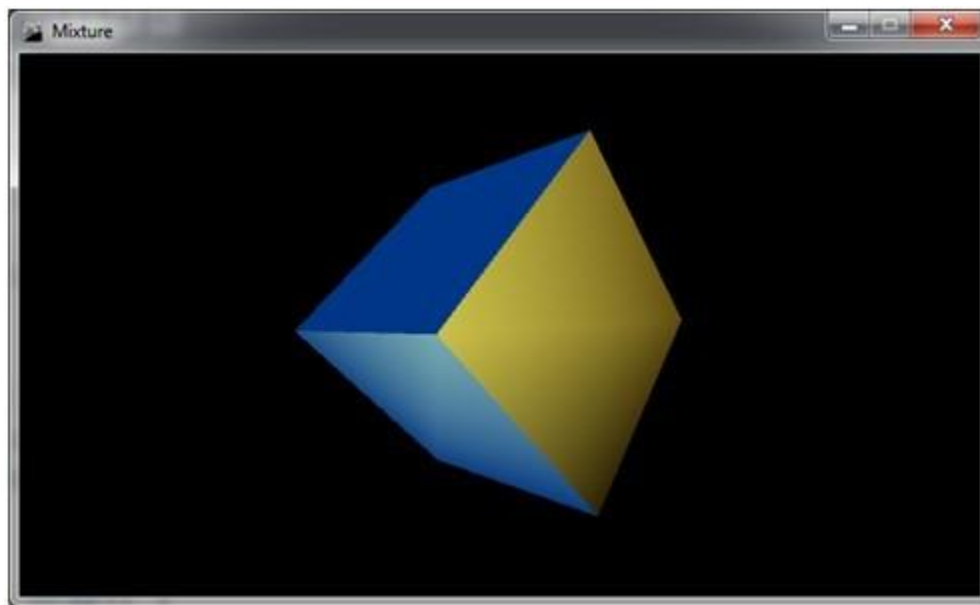
Students will study chiaroscuro (1) as an historical technique, and (2) as it can be rendered programmatically on 3-D virtual spaces to give the illusion of volume. Examples by Renaissance artists (Da Vinci, Caravaggio and Rembrandt) will be contrasted with those of pre-Renaissance artists (Giotto). Other techniques for shading dependent on medium (e.g. hatching or parallel lines for drawings, different colored tone blocks for woodcuts, extreme lighting for photography) will also be examined. (MA:Cr1-1, MA:Cr1-2, MA:Cr1-3, MA:Cr1-4, MA:Cr1-5)

This unit begins with a month-long introduction to **photography**, with an emphasis on light and composition. Student assignments include portraits, landscape, architectural interiors and exteriors, and street photography. Students consider the work of historic and contemporary photographers, e.g. Robert Capa, Gordon Parks, Cindy Sherman, Vivien Maier, Walker Evans, Ansel Adams, William Wegman, Diane Arbus, Dorothea Lange, Robert Frank, Lewis Hine and a host of others.

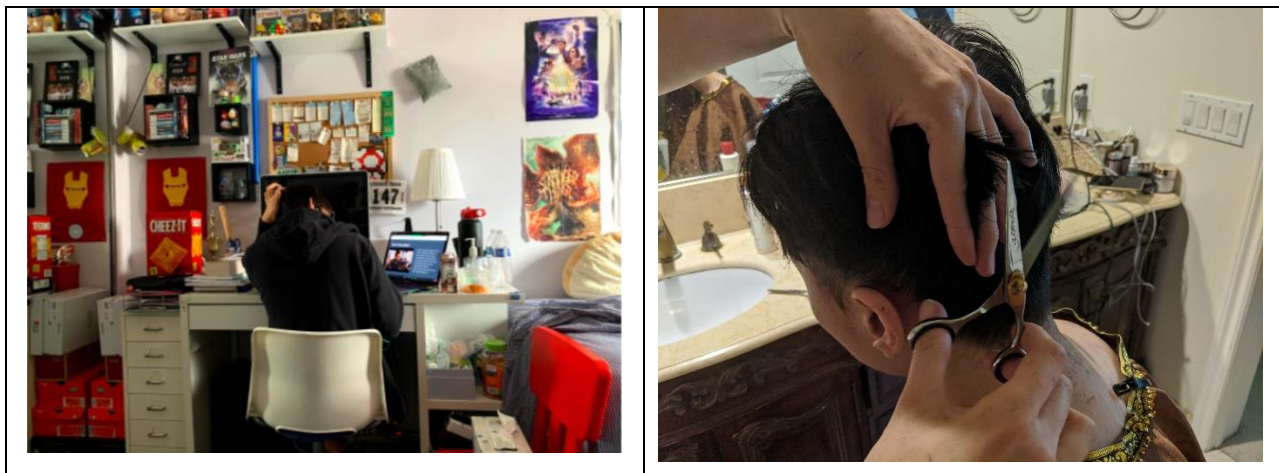
Unit Assignment(s)

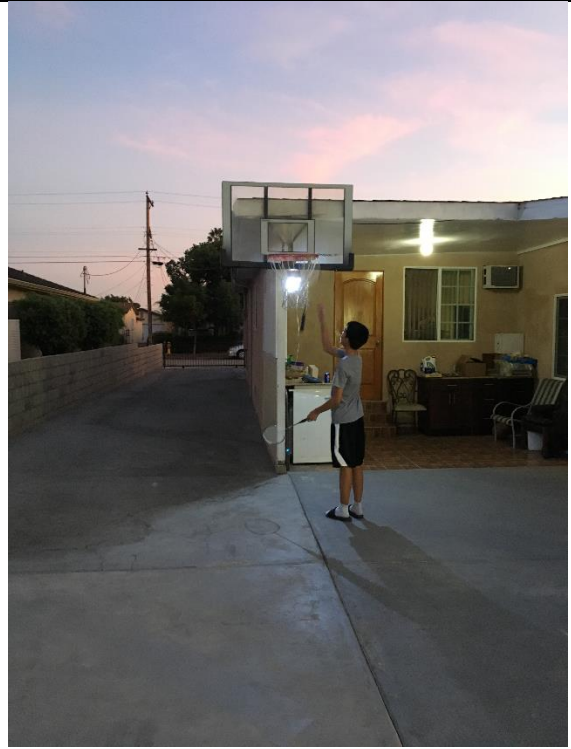
One Key Assignment: In their first explorations, students will dynamically position one or more light sources around colored spheres and cubes, as well as dynamically move the camera around such idealized objects. They will study the five examples that come with Processing's Basic Lights category (Directional, Mixture, Mixture-Grid, Reflection, Spot) and the two that come with Basic's Camera category (MoveEye, Perspective), as well as divvy up examples in other categories (Geometry, Motion, Shaders, Simulate) for class presentations.

Students will create 3D compositions (a) where the camera travels through a 3D virtual landscape to show how light falls on objects, (b) where light sources can travel over a static landscape, and (c) where camera and light are static, but objects rotate or undulate.



Examples of student work (photography) (VA:Cr1, VA:Cr2, VA:Cr3, VA:Cr4, VA:Cr5, VA:Cr6, VA:Cr7, VA:Cr8, VA:Cr9, VA:Cr10):





Unit 4: Randomness and noise

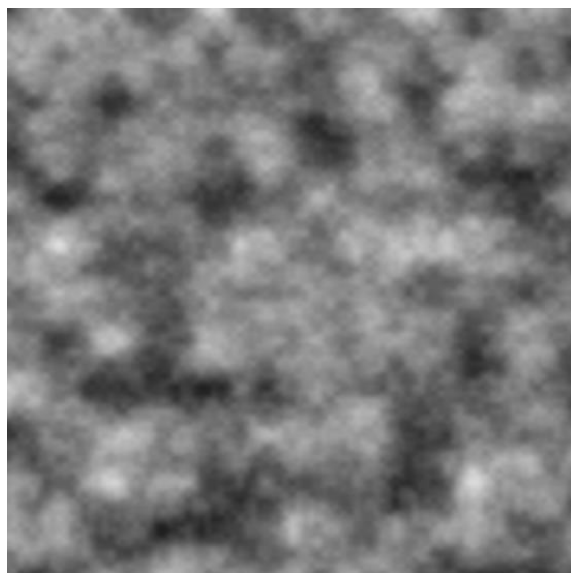
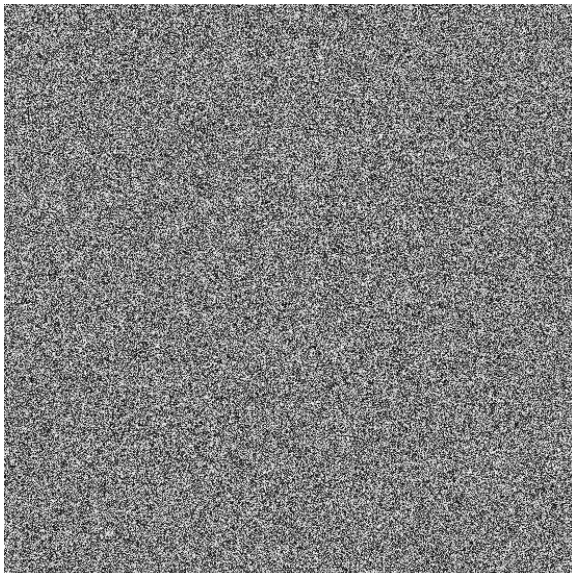
Unit Summary

Students learn the difference between data created by RANDOM generators, whose values are uniformly distributed over time, and NOISE, where values are not completely random, but are always close to those of their immediate neighbors. Noise occurs in natural phenomena, where exact data predictions cannot be forecast. Students will learn about Processing's pseudo-random generator, where transitions between neighbors can swing wildly, and Processing's noise() function, where smooth transitions can be generated over one, two or three dimensions. Most of the unit investigates the use of noise to create compositions mimicking natural formations, such as clouds, rippling water, mountains, hair, smoke, and so on. ([MA:Cr1-1](#), [MA:Cr1-2](#), [MA:Cr1-3](#), [MA:Cr1-4](#), [MA:Cr1-5](#), [MA:Cr1-9](#))

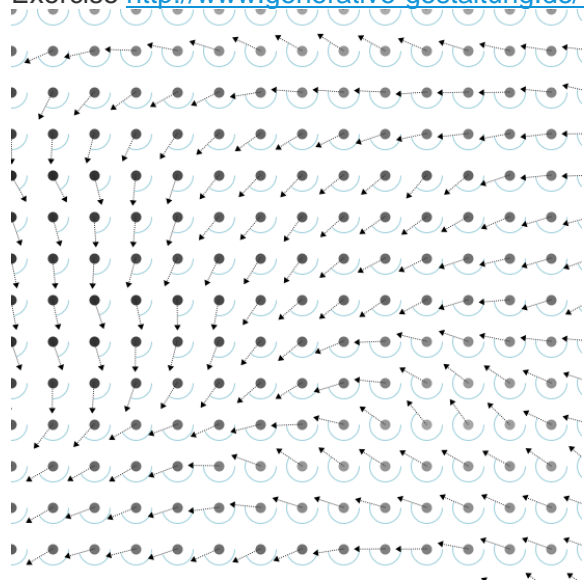
Unit Assignment(s)

One Key Assignment: As an engagement exercise, students examine the output from two programs for generating textures:

- the random program produces a uniform gray texture (http://www.generative-gestaltung.de/1/M_1_3_02), each pixel unrelated in magnitude to its neighbors (below, left).
- the noise program generates cloud-like patches with smooth transitions between regions (http://www.generative-gestaltung.de/1/M_1_3_03), effected by small changes in magnitude between neighboring pixels (below, right).

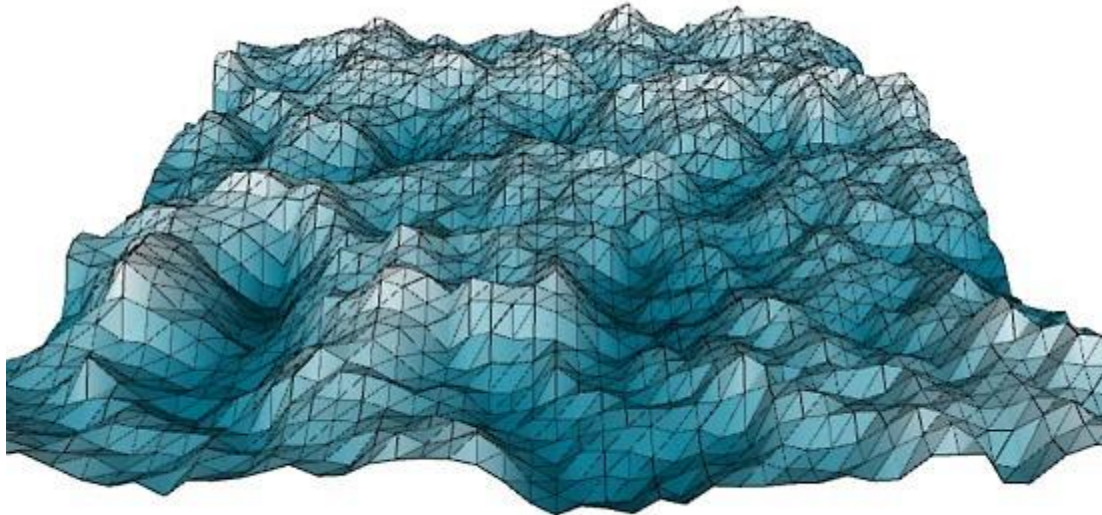


Exercise http://www.generative-gestaltung.de/1/M_1_5_01



is a graphic illustration of the effect of various parameters on the noise() function, using a grid of arrows and the effect of noise on the directions of the arrows. This includes (a) the “seed” value used to initialize the function; and (b) the effect of “octave” and “falloff” attributes that one can pre-set in noiseDetail(): octaves being similar to their role in harmonics in physics; and falloff defaulting to 50% of the immediately lower octave (0.5), but with legal values in the range $0 \leq \text{falloff} < 1.0$ for adjusting the impact of each successively higher octave on the generated noise() value.

These initial exercises culminate in a sketch that draws fantasy-like models of landscapes dotted with mountains and valleys:



Unit 5: Oscillation figures

Unit Summary

Students study forms known in mathematics as Lissajous figures. A Lissajous figure is constructed by mixing the x-coordinates of one sine wave with the y-coordinates of another, which creates a closed curve. There are several ways to add complexity resulting in modulated patterns: (a) superimpose figures with different frequencies; (b) extend figures into the 3rd dimension; (c) connect each point in a Lissajous curve to every other point in the curve using semi-transparent lines (to avoid a black clump). (MA:Cr1-1, MA:Cr1-2, MA:Cr1-3, MA:Cr1-4, MA:Cr1-5, MA:Cr1-9)

Unit Assignment(s)

One Key Assignment: (1) Three-dimensional Lissajous figures and (2) Lissajous figures where each point is connected to every other point.

Video examples of resultant sketches can be seen, respectively, at: http://www.generativegestaltung.de/1/M_2_4_01 and http://www.generativegestaltung.de/1/M_2_5_01.

A variation on the 3D algorithms is to fill in the surfaces of the figure with triangular sectors, with the boundaries of each sector defined by the origin point connecting to two points on the Lissajous path.

Students will create two sets of sketches, corresponding to the two types/families of Lissajous figures above.

Students will search the internet for images and videos using sine waves in other art media, and present to the class, for example:

Images (GIFs) <http://giphy.com/search/sine-wave>

Kinetic Sculptures <https://www.youtube.com/watch?v=dkkm518EsNo>
<https://www.youtube.com/watch?v=ICixCazf6-k>

Pendulum Art <https://www.youtube.com/watch?v=V87VXA6gPuE>

Unit 6: Formulated Bodies

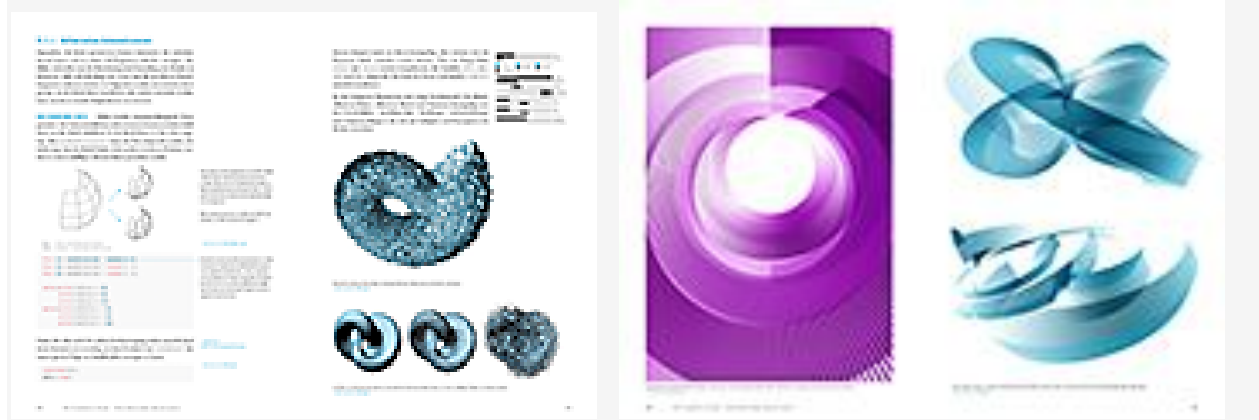
Unit Summary

3-dimensional regular figures, such as spheres and cylinders, can be described by classical mathematics formulas. One way to morph such structures is to begin with a 2-dimensional grid of points, each of which is projected into the 3rd dimension by application of a mathematical function/formula. This process produces a mesh whose surface is continuous.

Ultimately, a mesh can be “deconstructed” / distorted by the addition of random values to the x, y and z coordinates. The following video link illustrates the possibilities with a hollow cylinder and the alteration of various parameters involved in its construction:

http://www.generative-gestaltung.de/1/M_3_4_01_TOOL

(MA:Cr1-1, MA:Cr1-2, MA:Cr1-3, MA:Cr1-4, MA:Cr1-5, MA:Cr1-9)



Unit Assignment(s)

One Key Assignment:

A Mesh class is provided in the Generative Library that can generate 22 starting forms, such as PLANE, TUBE, SPHERE, TORUS, ELLIPTICTORUS, PARABOLOID, BOW, HORN, SHELL, KIDNEY, and other 3-D shapes. Students will write a subclass of the Mesh class that generates a hybrid of two such forms. This is accomplished by interpolating (x,y,z) values for differing intervals between the two forms. An example of possible hybrid forms of STEINBACHSCREW and BOW can be seen in the video at: http://www.generative-gestaltung.de/1/M_3_5_01

Unit 7: Attractors

Unit Summary

Attractors are virtual magnets – points in space that attract or repel the points around them, their power diminishing much like gravity lessens as the distance increases. The sketches generated are dynamic as follows: nodes (points that can move) are either attracted or repelled by attractors, and as nodes move incrementally, with each iteration, new attracting / repelling forces are calculated. Attractors need not generate simple movement, e.g. an attractor could cause a rotation effect on the points near it; the effect in 3D is as if one is bending and folding portions of a malleable plastic surface.

See http://www.generative-gestaltung.de/1/M_4_3_01_TOOL for a video demonstrating this type of 3D distortion).

Generally speaking, attractors are useful when one wants to either cluster or diminish density in particular areas of a sketch.

(MA:Cr1-1, MA:Cr1-2, MA:Cr1-3, MA:Cr1-4, MA:Cr1-5, MA:Cr1-9)

Unit Assignment(s)

One Key Assignment:

Students examine classical gravitation and other natural attractive/repulsive forces using various sketches in **Processing's Examples** that illustrate and isolate particular aspects of classical attraction/repulsion phenomena in physics, e.g. Spring, Springs, SoftBody, GravitationalAttraction3D, AccelerationWithVectors. Such sketches can be found in the **Simulate** and **Vectors** categories.

Students will modify parameters dealing with force, mass, acceleration and particularly damping (which determines how oscillations decay) in order to learn how these factors affect the motion.

Students will also examine the examples in **Processing's Topics** category **Interaction**, which are far simpler cases where the cursor attracts or repels the tip of a single flexible object.

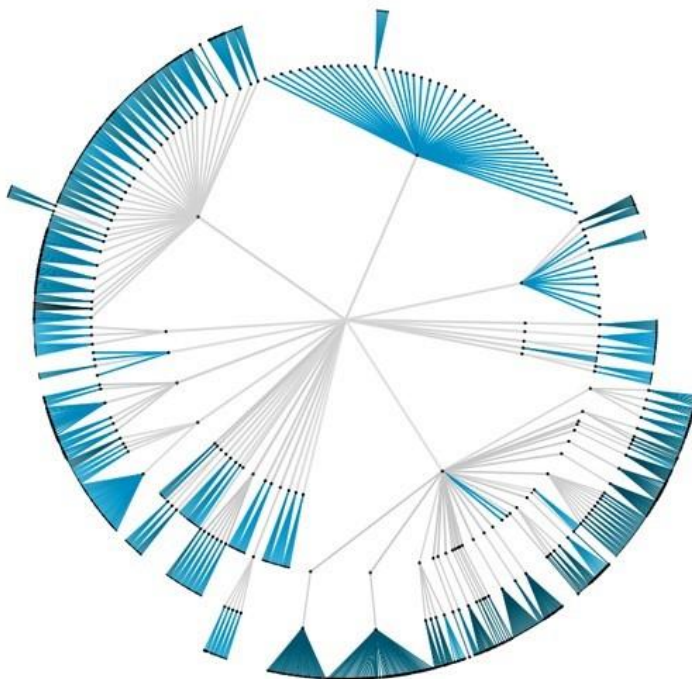
Students will write their own version of one of these programs.

Unit 8: Data Visualization

Unit Summary

The unit introduces the topic of data visualization, a field within computer science that concerns itself with information representation as a graphic display of entire datasets – particularly ones that are non-visual or do not readily lend themselves to visual representations – in such a way that elucidates non-obvious relationships in the data or its subsets. The unit presents three specific data visualization schemes:

- A sunburst diagram – which is well suited for representing the relative proportions of hierarchically structured (i.e. a tree structure) data – displays the real-time contents of the file system on the student's PC. The sunburst diagram will also be dynamic, displaying information about a folder or file in text form as the cursor hovers over any part of the diagram. A video showing the construction of a sunburst form as the hard drive is being read can be seen at: http://www.generativegestaltung.de/1/M_5_5_01_TOOL



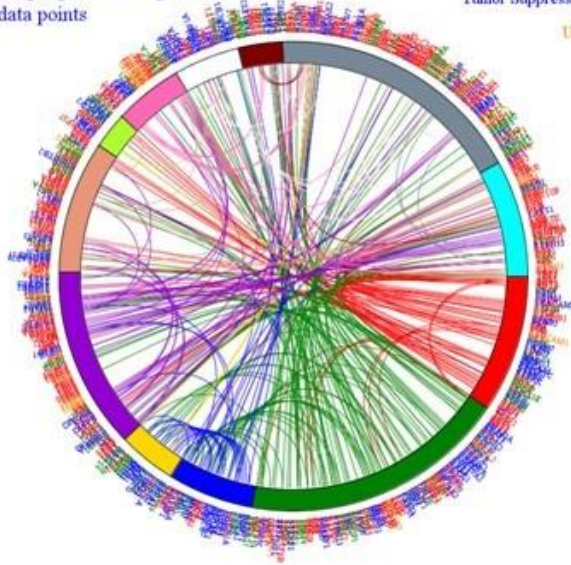
- A force-directed layout/graph – depicting the links between Wikipedia articles – employs arrows to represent the direction of the nodes' relationships to one another. The algorithm employs both (a) the repulsion of the nodes and (b) the attraction of their connections to optimally position – over time – the nodes so that they do not overlap. A video of the dynamic process by which the graph is constructed can be seen at: http://www.generative-gestaltung.de/1/M_6_4_01_TOOL
- One can use a radial set as an alternative to a Venn diagram when representing larger numbers of overlapping sets. The sets are proportionally dispersed along the circumference of a circle, and Bezier-curved lines connect intersecting points/pairs in different sets.

(MA:Cr1-1, MA:Cr1-2, MA:Cr1-3, MA:Cr1-4, MA:Cr1-5, MA:Cr1-6, MA:Cr1-8, MA:Cr1-9)

Functions ALL genes

339 unique genes, 12 regions
525 data points

Oncogene
Tumor Suppressor Gene
Other
Unknown



Unit Assignment(s)

One Key Assignment:

Students will focus on Bezier curves/lines, as they are used in both the sunburst diagram and the radial set diagram. As an intermediate step during the incremental building of the sunburst diagram, students construct a sunburst tree. The last step is using Bezier curves to connect lines from parent to children elements when these lines need to bend. Two control points off the curve, but along their paths, determine both the position and severity of the inflection regions. Students will modify values for the control points and observe how these affect the rendering of the curve. Students will similarly, but independently, construct the Bezier curves connecting the intersecting points in the radial set diagram.

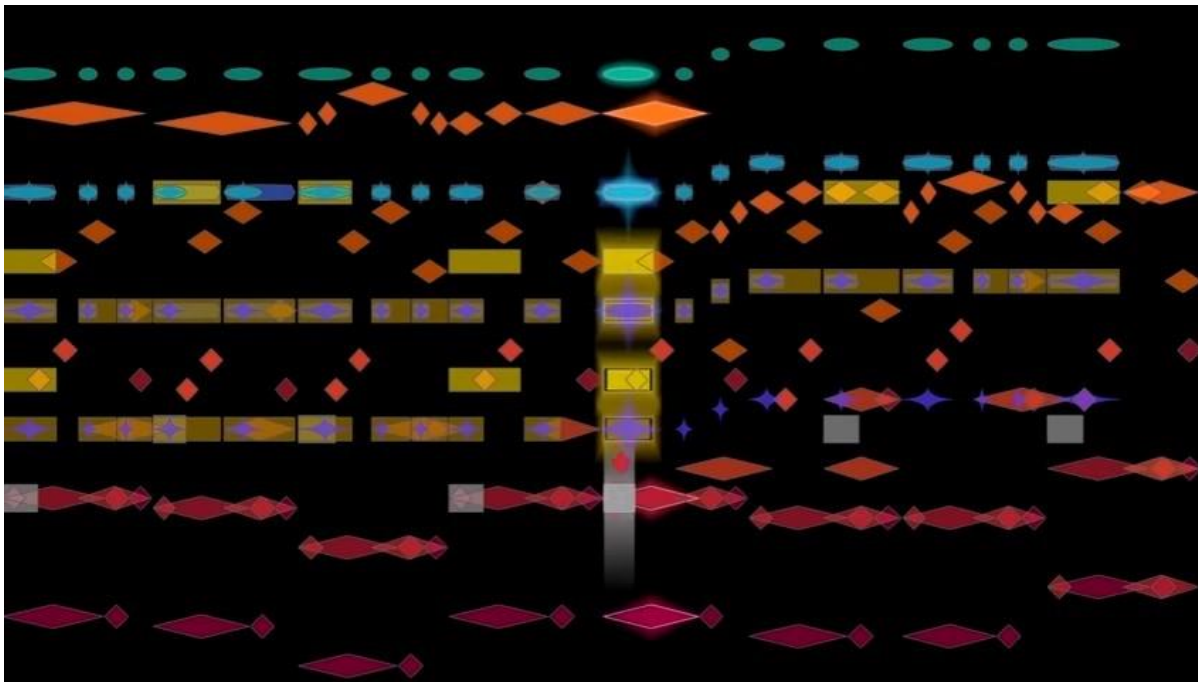
Unit 9: Music Visualization

Unit Summary

One very specific kind of data visualization problem, the visualization of music, has been implemented extremely creatively by Stephen Malinowski (moniker: smalin; YouTube Channel: Music Animation Machine). An example for

Beethoven's Symphony 7, Allegretto can be seen at <https://www.youtube.com/watch?v=ffYKCNY6kUk> (image below).

Malinowski represents each instrument's notes visually (diamonds and rectangles, in this example) and displays them along a leftward-scrolling timeline. An instrument's notes appear in a narrow single-colored band, and the bands are distributed from top to bottom according to the instrument's pitch, e.g. flute at the top, bass at the bottom. The animation's movement across the screen is synchronized with a recorded performance, and the visual shapes representing notes currently playing cross a middle vertical bar where they are highlighted. Each note's relative pitch is easily discerned and a note's duration is directly proportional to the horizontal length of its visual counterpart. The result: the viewer can see in real time the durations and relative pitches of each orchestra section's notes as the music is heard.

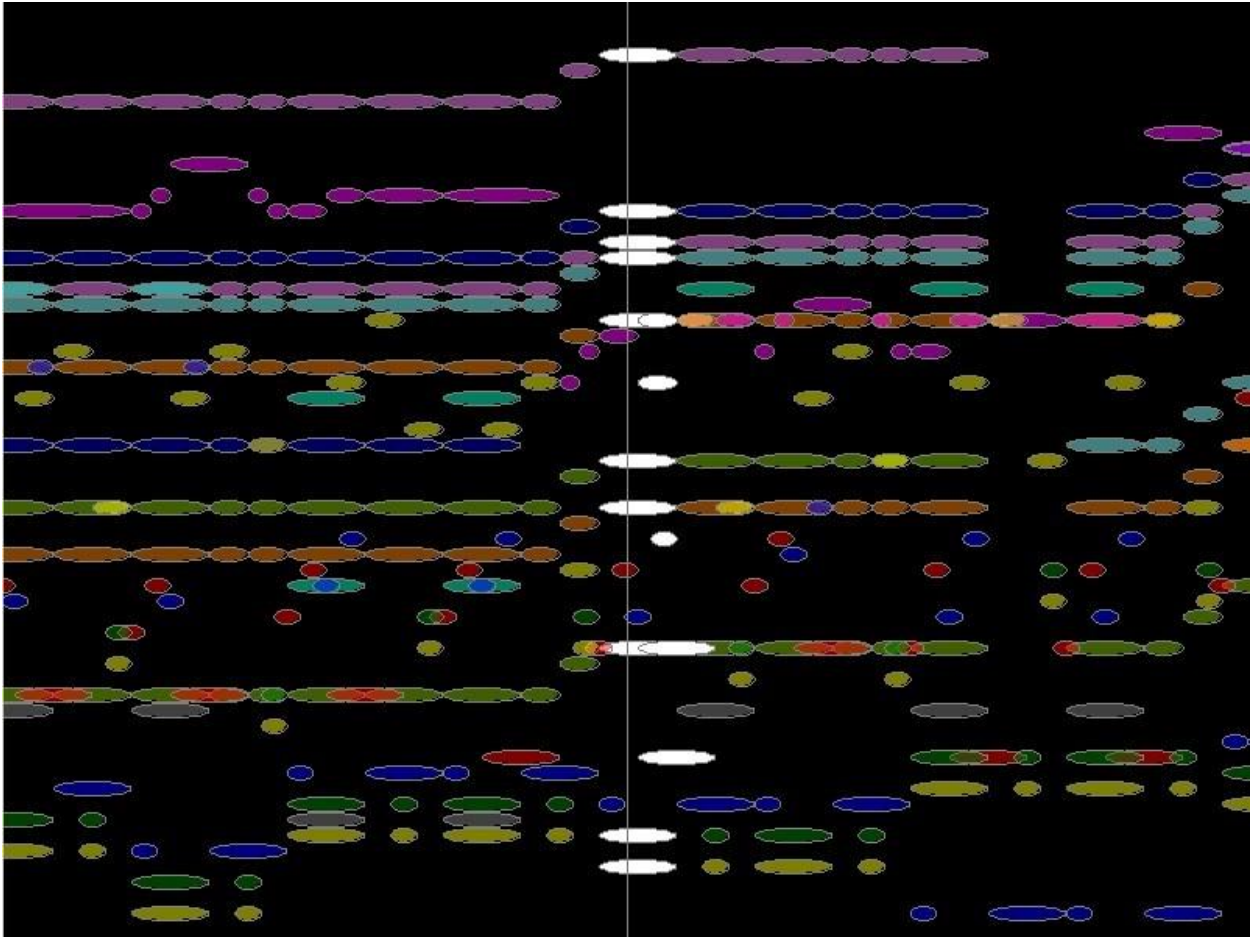


Unit Assignment(s)

One Key Assignment:

Students will build an extended and more complicated version of a music animation that is more akin to a player piano.

Rather than making a single animation for a "canned" performance, the program will be able to play any piece of music "fed" to it. Students will first study staff notation. They will then decide upon an encoding scheme for representing the notes (pitch, volume, duration) of a musical score so that they can be systematically written into a text file. The program will read in and parse this data file and build an internal representation of the notes played by each instrument. A graphic timeline of the composition will then be generated and begin to scroll across the screen. The notes will be sent to the computer's internal music synthesizer and soundbank. The program will synchronize the speed of the animation so that when it detects notifications from the hardware synthesizer telling it which notes are currently playing, these notes will have reached the window's middle vertical bar and will be highlighted white. (MA:Cr1-1, MA:Cr1-2, MA:Cr1-3, MA:Cr1-4, MA:Cr1-5, MA:Cr1-6, MA:Cr1-7, MA-CR1-8, MA:Cr1-9)



This synthesized version Beethoven's Symphony 7, Allegretto (above) can be seen at the link below.
<https://d3id26kdqbehod.cloudfront.net/DOWMHS/2020/10/05/Gi3ppr0lms0QZfuNaoJUQ15G1SIUnySOFUkoEZG774GPbp2EAAjkAfJOCBvy/480p.mp4>

Section 2: Introduction to Unity 3D and Virtual Reality

Unit Summary

Students will learn the basics of Design and C# Programming needed to create Virtual Reality (VR) Games.

Lessons are through the Unity/Learn website (learn.unity.com).

There is also a culminating project: **Rubik's Cube**, which students will build in both Processing and Unity.

Unit Assignment(s)

learn.unity.com Modules / Lessons

Unity Game Dev Course

Beginner Art: Unity Game Dev Course

Beginner Fundamentals: Unity Game Dev Course

Intermediate Art: Unity Game Dev Course

Intermediate Fundamentals: Unity Game Dev Course

Advanced Art: Unity Game Dev Course

Advanced Design: Unity Game Dev Course

Advanced Fundamentals: Unity Game Dev Course

Beginner (supplemental modules)

Beginning 2D Game Development

Introduction to 3D Animation Systems

Real-time Animated Storytelling

Unity's Game Prototyping Series

Create with Code Live: Summer 2020

Intermediate (supplemental modules)

Exploring Unity 2018 LTS

Buried Memories: High Fidelity Game Visuals

Zoe - VR for Education

Intermediate 3D Game Development

Culminating Project

Processing version: Rubik's Cube (<https://sites.google.com/site/krishnabharadwaj85/rubikcube>)

Unity version: Rubik's Cube (<https://www.megalomobile.com/lets-make-and-solve-a-rubiks-cube-in-unity/>)

([MA:Cr1-1](#), [MA:Cr1-2](#), [MA:Cr1-3](#), [MA:Cr1-4](#), [MA:Cr1-5](#))

Course Materials

Title	Author	Publisher	Edition	Website
Generative Design: Visualize, Program and Create with Processing	Bohnacker, Hartmut; Gross, Benedikt; Laub, Julia; Lazzeroni, Claudius	Princeton Architectural Press, New York	First Edition / 2012	http://www.generative-gestaltung.de/1/
Form + Code in Design, Art and Architecture: A Guide to Computational Aesthetics	Reas, Casey; McWilliams, Chandler	Princeton Architectural Press, New York	First Edition / 2010	http://formandcode.com/
Algorithms for Visual Design Using the Processing Language	Terzidis, Kostas	Wiley Publishing, Inc.	First Edition / May 2009	https://www.wiley.com/en-us/Algorithms+for+Visual+Design+Using+the+Processing+Language-p-9780470503188
Generative Art: A Practical Guide Using Processing	Pearson, Matt	Manning Publications, Inc. / Shelter Island, NY	First Edition / 2011	https://www.manning.com/books/generative-art
Processing: A Programming Handbook for Visual Designers and Artists	Reas, Casey; Fry, Ben	MIT Press, Cambridge	First Edition / 2007	https://mitpress.mit.edu/books/processing
Processing: Creative Coding and Computational Art	Greenberg, Ira	Apress / SpringerVerlag, New York	Second Edition / 2007	http://www.apress.com/us/book/9781590596173