

Generative Art & Design using the Programming Language PROCESSING

Downtown Magnets High School (051576)

Submitted: Jun 23, 2017

Decision: Aug 3, 2017

APPROVED

Basic Course Information

Title:

Generative Art & Design using the Programming Language PROCESSING

Transcript abbreviations:

GENERATIVE ART & DES 1AB / 200143/4

Length of course:

Full Year

Subject area:

Visual & Performing Arts ("f") / Visual Arts

Integrated (Academics / CTE)?

Yes

Grade levels:

10th, 11th, 12th

UC honors designation?

No

Course learning environment:

Classroom Based

Course Description

Course overview:

Students will learn some fundamental topics – color, shape/form and light – typically taught in a 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 is the ALGORITHM, encoded in a computer program written in the programming language Processing, and applied to 2D- and 3D- virtual spaces.

The course will comprise several topics from Part 2 of the book *Generative Design: Visualize, Program, and Create with Processing*, Authors: Bohnacker et al. 2012. Other units and material material will come from other similarly-themed books and sources. There are 9 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/code> (<http://www.generative-gestaltung.de/code>)

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.

Prerequisites:

COMP SCI 1AB : Computer Science 1 (or an equivalent year-long programming course) (Required)

Co-requisites:

None

Course content:

Unit 1: Color

Unit 1: Color

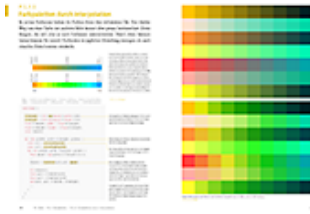
Students will review the RGB (Red,Green,Blue) additive model for the 16 million (2^{24}) possible colors one can generate programmatically. They will also explore color definitions based on the HSB model (Hue, Saturation, Brightness/Luminance).

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

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.

Video example of a color gradient: http://www.generative-gestaltung.de/P_1_2_1_01 (http://www.generative-gestaltung.de/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?
2. Can any patterns be discerned in the gradients?
3. Do the palettes "work", i.e. do they have a desired aesthetic effect, either harmonious or disharmonious?

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.

Unit 2: Form/Shape

Unit 2: Form/Shape

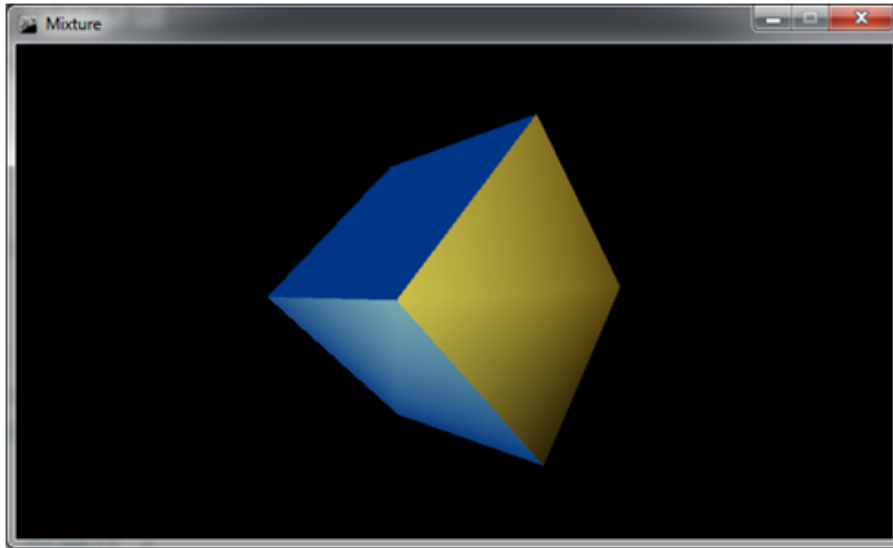
This rather large unit is an exploration of the visual effects of three distinct principles as applied to shapes or shape families: (1) Repetition using a grid; (2) Iteration using dumb and intelligent agents; and (3) Interaction via user drawing. Such compositions can be viewed on (at least) two levels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

One Key Assignment: Drawing with Animated Brushes (Principle 3, above): Drawing with Type (Letters). As the mouse moves, it leaves letters on the screen surface. The size, angle and color of the letters can be controlled by various mouse events (left/right presses) and keyboard inputs (e.g. arrow/directional keys, plus and minus keys). Students draw three compositions: (1) Representational; (2) Abstract; and (3) a Word Cloud, where the composition takes into account word meanings. Prior to drawing the compositions, students study the decorative characteristics of various Typefaces/Font-Families and Calligraphic styles via compositions in which they are used.

Example of a representational composition: http://www.generative-gestaltung.de/P_2_3_3_01_TABLET
(http://www.generative-gestaltung.de/P_2_3_3_01_TABLET)

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.



Unit 4: Randomness and noise

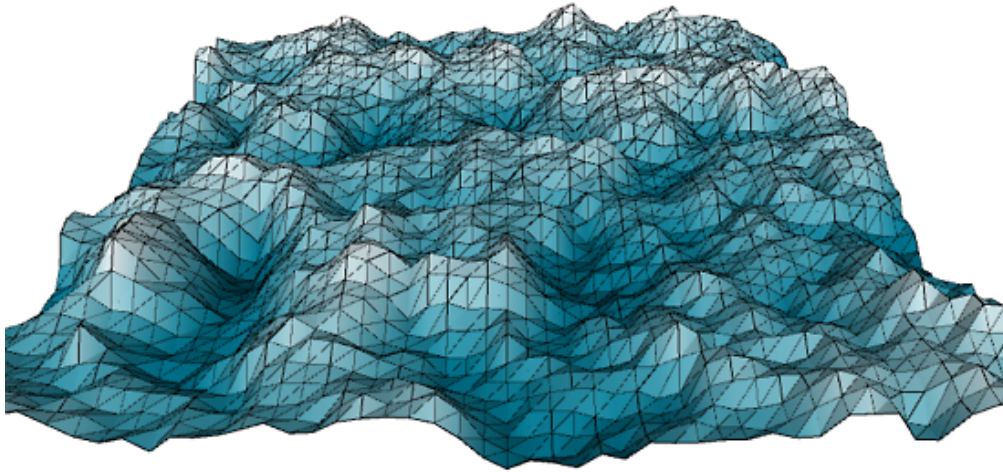
Unit 4: Randomness and noise

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.

One Key Assignment: As an engagement exercise, students examine the output from two programs for generating textures: (a) the random program produces a uniform gray texture (http://www.generative-gestaltung.de/M_1_3_02 (http://www.generative-gestaltung.de/M_1_3_02)), while (b) the noise program generates cloud-like patches with smooth transitions between regions (http://www.generative-gestaltung.de/M_1_3_03 (http://www.generative-gestaltung.de/M_1_3_03)).

Exercise http://www.generative-gestaltung.de/M_1_5_01 (http://www.generative-gestaltung.de/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: http://www.generative-gestaltung.de/M_1_4_01 (http://www.generative-gestaltung.de/M_1_4_01).



Unit 5: Oscillation figures

Unit 5: Oscillation figures

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).

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.generative-gestaltung.de/M_2_4_01 (http://www.generative-gestaltung.de/M_2_4_01) and http://www.generative-gestaltung.de/M_2_5_01 (http://www.generative-gestaltung.de/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> (<http://giphy.com/search/sine-wave>),

Kinetic Sculptures <https://www.youtube.com/watch?v=dkkm518EsNo> (<https://www.youtube.com/watch?v=dkkm518EsNo>)

<https://www.youtube.com/watch?v=ICixCazf6-k> (<https://www.youtube.com/watch?v=ICixCazf6-k>)

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

Unit 6: Formulated Bodies

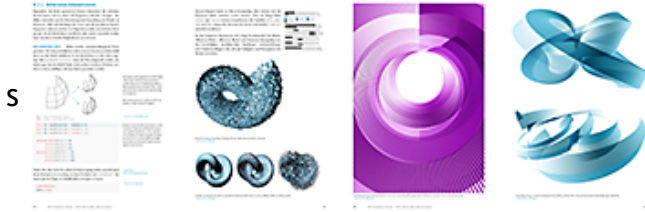
Unit 6: Formulated Bodies

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/M_3_4_01_TOOL (http://www.generative-gestaltung.de/M_3_4_01_TOOL)

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/M_3_5_01 (http://www.generative-gestaltung.de/M_3_5_01)



Unit 7: Attractors

Unit 7: Attractors

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/M_4_3_01_TOOL (http://www.generative-gestaltung.de/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.

One Key Assignment:

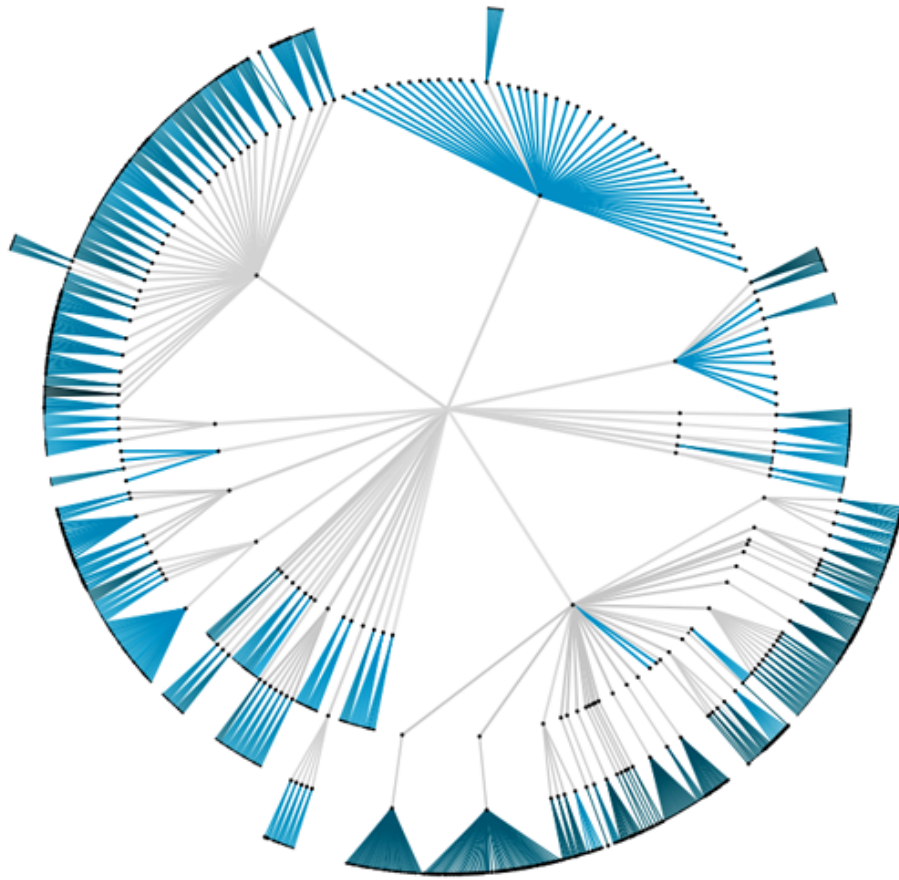
Students examine classical gravitation and other natural attractive/repulsive forces using various sketches in Processing's Examples that illustrate and isolate particular aspects of classical attraction/repulsion phenomena in physics, e.g. Spring, Springs, SoftBody, GravitationalAttraction3D, AccelerationWithVectors. 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 the 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 8: Data Visualization

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:

(1) 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.generative-gestaltung.de/M_5_5_01_TOOL (http://www.generative-gestaltung.de/M_5_5_01_TOOL)



(2) 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/M_6_4_01_TOOL (http://www.generative-gestaltung.de/M_6_4_01_TOOL)

(3) 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.

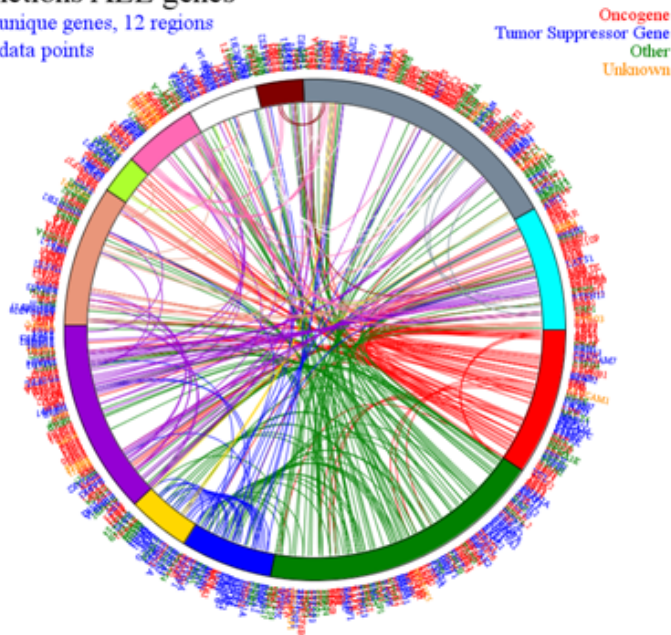
One Key Assignment:

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.

Functions ALL genes

339 unique genes, 12 regions

525 data points

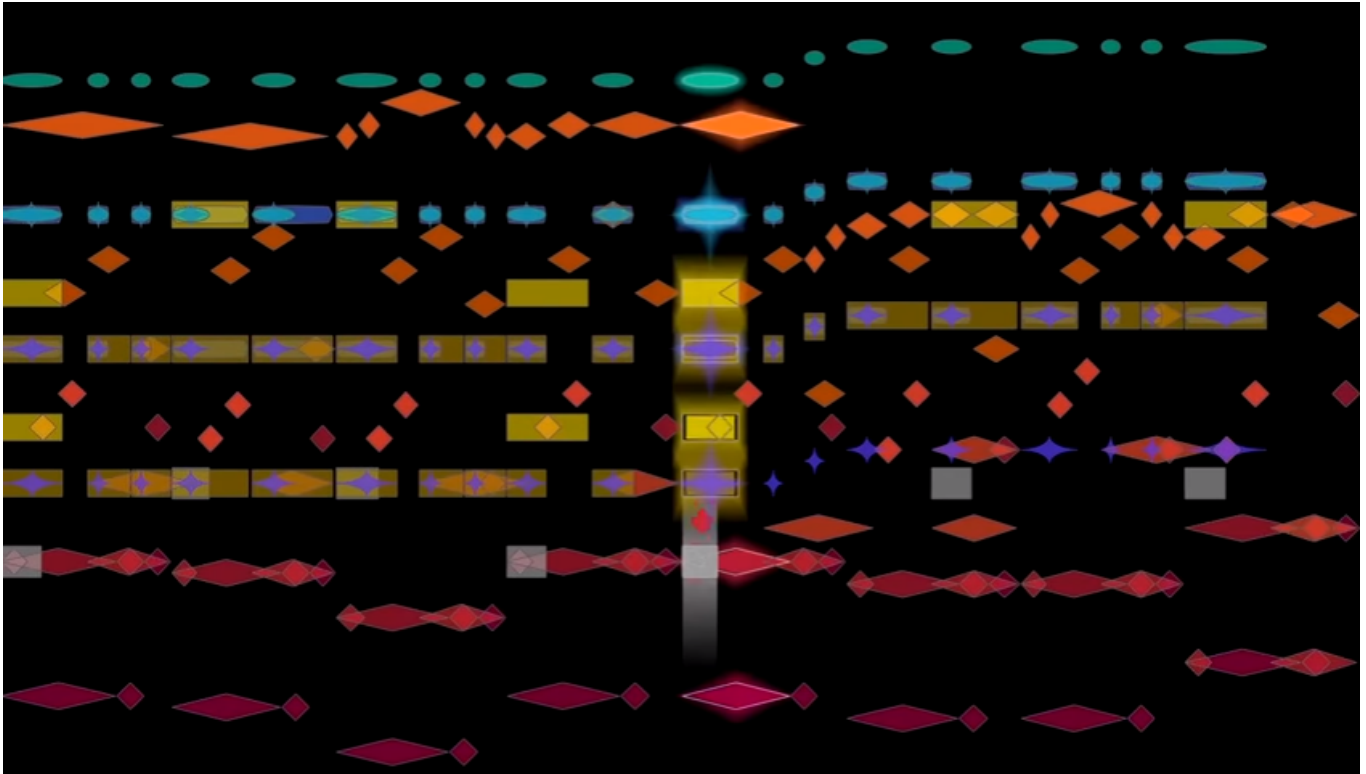


Unit 9: Music Visualization

Unit 9: Music Visualization

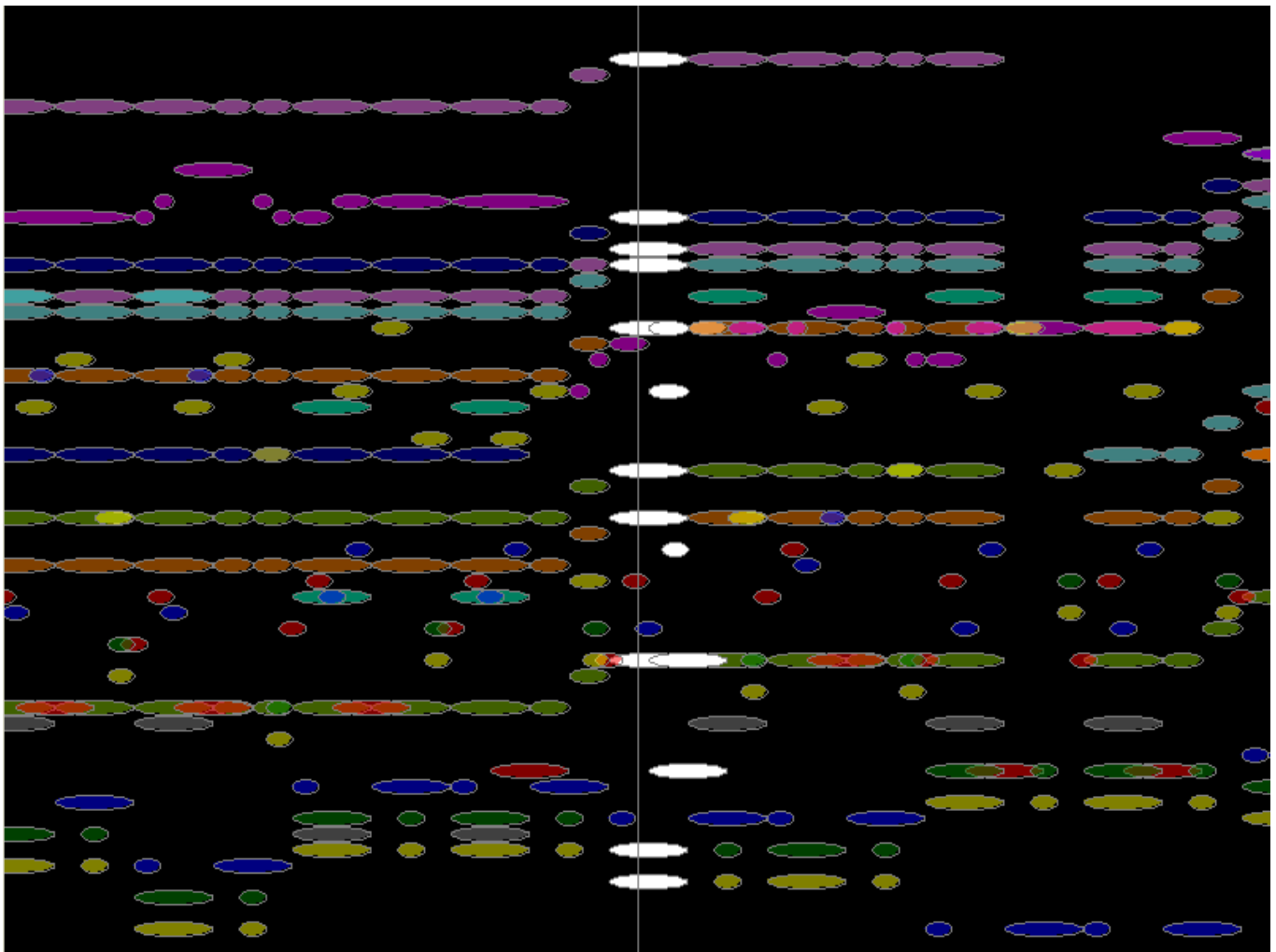
One very specific kind of data visualization problem, the visualization of music, has been implemented extremely creatively by Stephen Malinowski (moniker: smalin; YouTube Channel: Music Animation Machine). An example for Beethoven's Symphony 7, Allegretto can be seen at <https://www.youtube.com/watch?v=ffYKCNY6kUk> (<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.



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.



Course Materials

Textbooks

| Title | Author | Publisher | Edition | Website | Primary |
|--|---|---|----------------------|---|---------|
| 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/ | Yes |
| Form + Code In Design, Art and | Reas, Casey; McWilliams, Chandler | Princeton Architectural Press / New | First Edition / 2010 | http://formandcode.com/ | No |

| | | | | | |
|---|-----------------------|---|--------------------------|---|----|
| Algorithms for Visual Design Using the Processing Language | Terzidis, Kostas | Wiley Publishing, Inc. | First Edition / May 2009 | http://www.wiley.com/WileyCDA/WileyTitle/productCd-0470375485.html | No |
| 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 | No |
| 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 | No |
| Processing: Creative Coding and Computational Art | Greenberg, Ira | Apress / Springer-Verlag, New York | Second Edition / 2007 | http://www.apress.com/us/book/9781590596173 | No |

Websites

| Title | Author(s)/Editor(s)/Compiler(s) | Affiliated Institution or Organization | URL |
|-----------------------|---|---|---|
| Generative Gestaltung | Benedikt Gross, Hartmut Bohnacker, Julia Laub, Claudius Lazzeroni | Schmidt-Friderichs GmbH, Mainz, Germany | http://www.generative-gestaltung.de/ |

Additional Information

Course Author:

Scott Portnoff
Teacher
srp4379@lausd.net
3236271152 ext.

Comment to UC:

