Algorithmic Thinking : Programs that Solve Well-defined Visual Problem

Elodie Fourquet
Colgate University

In early 2014 I created and taught a CS0 course for women students at Mount Holyoke College. I chose to emphasize algorithmic thinking as a way of solving problems on and off the computer, so I needed concrete and appealing problems for the students to solve. Having the students create their own problems was my way of doing so.

Based on a one week introduction to visual design, each student designed the image or animation that they would later program. As they did so outside class, the students and I collaborated in class solving the problems of implementing a design I had chosen: solving design problems of their own overlapped with algorithmic solution of implementation problems in class. Subsequently, they applied the classroom lessons to their own designs. The result was a collection of programmed images and animations the students were eager to show.

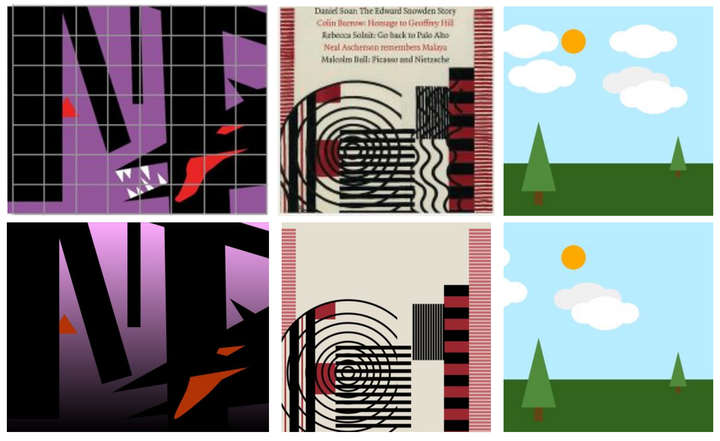
The dual nature of their practice appealed to the students: they became highly motivated; they took ownership of their work and of the skills they acquired; they formed a cohesive community working together in the lab. Modelling on the classroom work in which they participated was effective in building programming skills.

In Class

Processing—A language designed by artists for artists. Students aspire to artistic goals and endeavours and therefore are engaged and keen to practice problem solving in a visual art framework. However, students are not artists, not yet, they need training, models and structure.

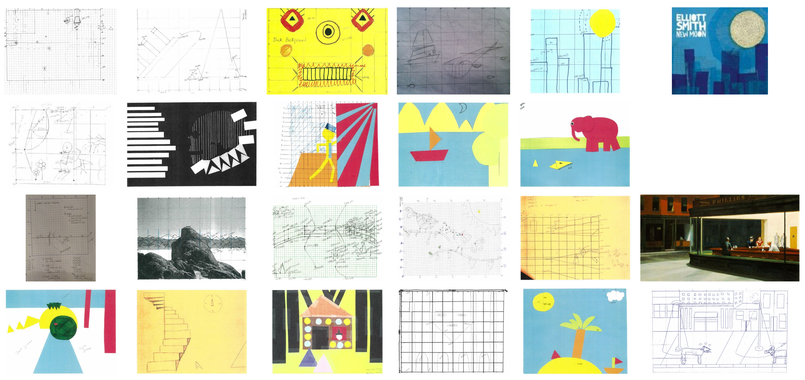
Design Principles—I presented the design process described in *Picture This: How Pictures Work* by Molly Bang. As shown on the left a sequence of pictures creates a representation of the Little Red Ridding Hood: contrasting alternatives, working incrementally. I believe this book is key to the students achievement in creating simple and effective pictures that tell stories.

In class version of each assignment. For the two left most column, the first row presents the designs (aka blueprints) that guided the programs we wrote in class while the second row contains the output of these programs. The right most column shows two frames of the simple animation created in class.



Assignment 1: Composition

For Assignment 1 students first design their image composition using paper cuts and/or sketches. Abstraction, simple shapes and few colors are used to tell a story.



Students then translate their physical plans to a Processing program composed of a sequence of calls to graphics functions. A coordinate system and a grid is required to map the picture design to the function call statements with appropriate parameters. In addition mastery of the color model, order of instructions and the painter's algorithm is required.

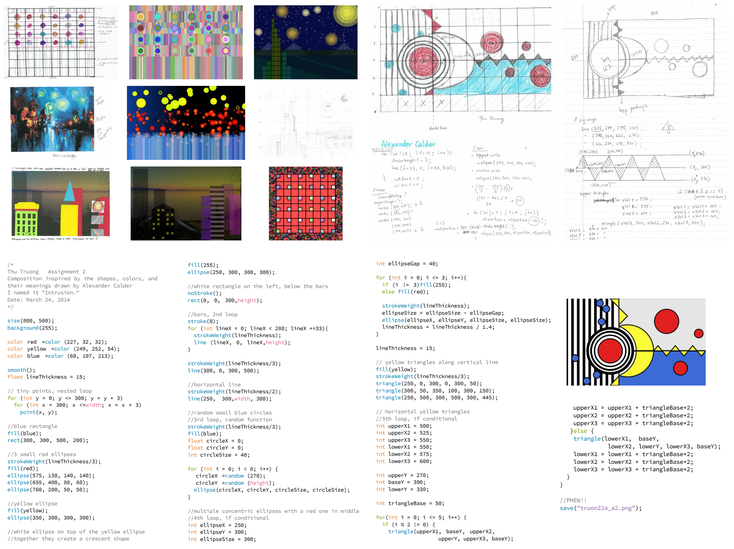
Match their output to the paper plans above!

Example Code

```
/*Assignment 1   02/07/14
Based on Edward Hopper's Nighthawks (1942)*/

size(1000,800);
background(255);

smooth();
noStroke();

//the lowest part of the bar
fill(11,54,36);
quad(400,520,1000,650,1000,880,400,680);

//upper orange part of the background store
fill(139,52,17);
rect(0,0,700,200);

//black top right corner
fill(0);
rect(700,0,200,200);

//5th small window
stroke(15,95,43);
strokeWeight(70);
strokeCap(SQUARE);
line(530,50,530,180);

noStroke();
//the bar sign
fill(75,23,4);
quad(455,80,1000,-20,1000,50,455,150);

//triangle part where the people sit
fill(250,243,103);
triangle(400,520,1000,400,1000,650);

//the big wall window
fill(51,82,81);
quad(400,200,900,230,900,420,400,520);

//little quad on the right
fill(247,168,47);
quad(900,230,1000,150,1000,440,900,420);

//the ceiling
fill(250,247,192);
triangle(400,200,1000,150,900,230);

//the 4th small window
stroke(15,95,43);
strokeWeight(70);
strokeCap(SQUARE);
line(410,50,410,180);

noStroke();

// the quad right below the bar sign
fill(7,36,24);
quad(400,150,400,200,1000,150,1000,50);

//the woman in red
fill(255,0,38);
quad(800,480,830,435,800,500,780,444);

//the man in black
fill(0);
quad(650,500,680,490,700,560,630,560);

//the man's hat
fill(0);
ellipse(680,490,40,25);

stroke(15,95,43);
strokeWeight(70);
strokeCap(SQUARE);

//the other small windows
line(50,50,50,180);
line(170,50,170,180);
line(290,50,290,180);

stroke(0);
line(50,140,50,180);
line(170,90,170,180);
line(290,120,290,180);

noStroke();

//the front of the background store
fill(11,54,36);
rect(0,200,400,320);

//the two black thin pillars on the front of the

stroke(0);
strokeWeight(10);
line(170,200,170,520);
line(230,200,230,520);

//the two black windows on the front of the store
strokeWeight(100);
line(80,230,80,480);
line(320,230,320,480);

noStroke();

//ground 1
fill(130,137,134);
rect(0,520,400,160);

//ground 2
fill(53,96,106);
quad(0,680,400,680,800,800,0,800);

//the waiter in white
fill(255);
beginShape();
curveVertex(880,500);
curveVertex(880,500);
curveVertex(940,510);
curveVertex(965,540);
curveVertex(970,580);
curveVertex(970,580);
endShape();
```

Assignment 2: Patterns

Assignment 2 follows the same process but the design can be abstract and needs to contain patterns. The implementation therefore requires loops.



```
/*
Thu Truong   Assignment 2
Composition inspired by the shapes, colors, and
their meanings drawn by Alexander Calder
I named it "Intrusion."
Date: March 24, 2014
*/

size(800, 500);
background(255);

color red =color (227, 32, 32);
color yellow =color (249, 252, 54);
color blue =color (60, 187, 213);

smooth();
float lineThickness = 15;

// tiny points, nested loop
for (int y = 0; y <= 300; y = y + 3)
  for (int x = 300; x <=width; x = x + 3)
    point(x, y);

//blue rectangle
fill(blue);
rect(300, 300, 500, 200);

//3 small red ellipses
strokeWeight(lineThickness/3);
fill(red);
ellipse(575, 130, 140, 140);
ellipse(655, 400, 80, 80);
ellipse(760, 200, 50, 50);

fill(255);
ellipse(250, 300, 300, 300);

//white rectangle on the left, below the bars
noStroke();
rect(0, 0, 300,height);

//bars, 2nd loop
stroke(0);
for (int lineX = 0; lineX < 200; lineX +=33){
  strokeWeight(lineThickness);
  line (lineX, 0, lineX,height);
}

strokeWeight(lineThickness/3);
line(300, 0, 300, 500);

//horizontal line
strokeWeight(lineThickness/2);
line(250, 300,width, 300);

//random small blue circles
//3rd loop, random function
strokeWeight(lineThickness/3);
fill(blue);
float circleX = 0;
float circleSize = 40;

for (int i = 0; i < 8; i++) {
  circleX =random (270);
  circleY =random (height);
  ellipse(circleX, circleY, circleSize, circleSize);
}

//multiple concentric ellipses with a red one in middle
//4th loop, if conditional
int ellipseX = 250;
int ellipseY = 300;
int ellipseSize = 300;

int ellipseGap = 40;

for (int i = 0; i <= 3; i++){
  if (i != 3)fill(255);
  else fill(red);

  strokeWeight(lineThickness);
  ellipseSize = ellipseSize - ellipseGap;
  ellipse(ellipseX, ellipseY, ellipseSize, ellipseSize);
  lineThickness = lineThickness / 1.4;
}

lineThickness = 15;

// yellow triangles along vertical line
fill(yellow);
strokeWeight(lineThickness/3);
triangle(250, 0, 300, 0, 300, 50);
triangle(300, 50, 350, 100, 300, 150);
triangle(250, 500, 300, 500, 300, 445);

// horizontal yellow triangles
//5th loop, if conditional
int upperX1 = 500;
int upperX2 = 525;
int upperX3 = 550;
int lowerX1 = 550;
int lowerX2 = 575;
int lowerX3 = 600;

int upperY = 270;
int baseY = 300;
int lowerY = 330;

int triangleBase = 50;

for(int i = 0; i <= 5; i++) {
  if (i % 2 != 0) {
    triangle(upperX1, baseY, upperX2,
             upperY, upperX3, baseY);

//PHEW!!
save("truon22a_a2.png");
```

Assignment 3: Animations

Assignment 3 includes function definitions to draw similar objects (varying their size and/or color), animation and mouse/keyboard interactions.

Discussion

During the term student's enthusiasm guided the course development: they were keen to learn new concepts to make complex designs; they frequently asked about graphics research. They requested to learn about gradient for Assignment 1! They enjoyed discussing the contrast between their work and the 3D equivalent.

Students were surprised by the approach and methodology of the course. They didn't expected programming to be so fun and computer science to involve creativity. They wholeheartedly embraced the course, which changed their view of computer science.

Most important several students expressed unprompted interest in taking a follow-on course. Two students were part of a team of five who went to the 36 hours hacking at Yale University the following November.

- What is their view of computer science one year later?
- Was the women classroom a contributor to this special experience?
- How does this curriculum compare to other CS0 which use a visual approach?