

Alp Tuğan / 18.03.2024

Creative Coding

Three Steps Transformation to Computational Thinking →



Table of Contents

Table of Contents

Computational Thinking Framework

Let's make it more clear

DTFA Method

Subject / Problem

Imagine the abstracted forms

Feature Extraction*

Express the Panda Using Simple Shapes

Decomposition  (15 minutes)

Abstracted Shape

Flow  (15 minutes)

Translation    (15 minutes)

Algorithm   

BREAK

"First, solve the problem. Then, write the code."

Assignments

Computational Thinking Framework

The method for Problem-Solving

1. DECOMPOSITION
2. ABSTRACTION
3. PATTERN RECOGNITION
3. ALGORITHM DESIGN

Let's make it more clear

Start with your known register. Use verbal, visual, written representations.

1. Decomposition



Take notes or draw abstracted shapes using pen and paper. Do not code anything. Analyze the subject.

2. Flow



Identify the procedural flow to solve the problem. Switch between registers. Use diagrams and arrows or take notes again.

3. Translation



Open the code editor. Identify variables and functions from previous step. In this step start coding. Refer to p5Js reference page. Keep running your code whenever you add new lines to check if it is ok.

4. Algorithm



Implement the flow design to your code. Utilize conditional statements `(if/else)`, repetitions `for loops`, trigonometric functions `sin()`, and probability `random()` to reveal the creative potential of your application by experimenting with the code. If your code involves interactivity (mouse, keyboard, sound or camera), test your code for each potential use case.

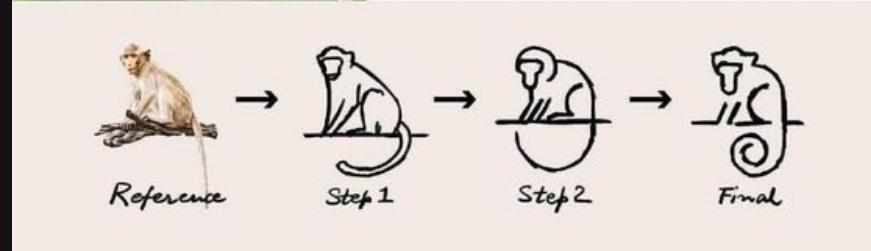
Subject / Problem

Note down everything you see on your interface design formally. Note down each element what you see. Every single color and shape matters.

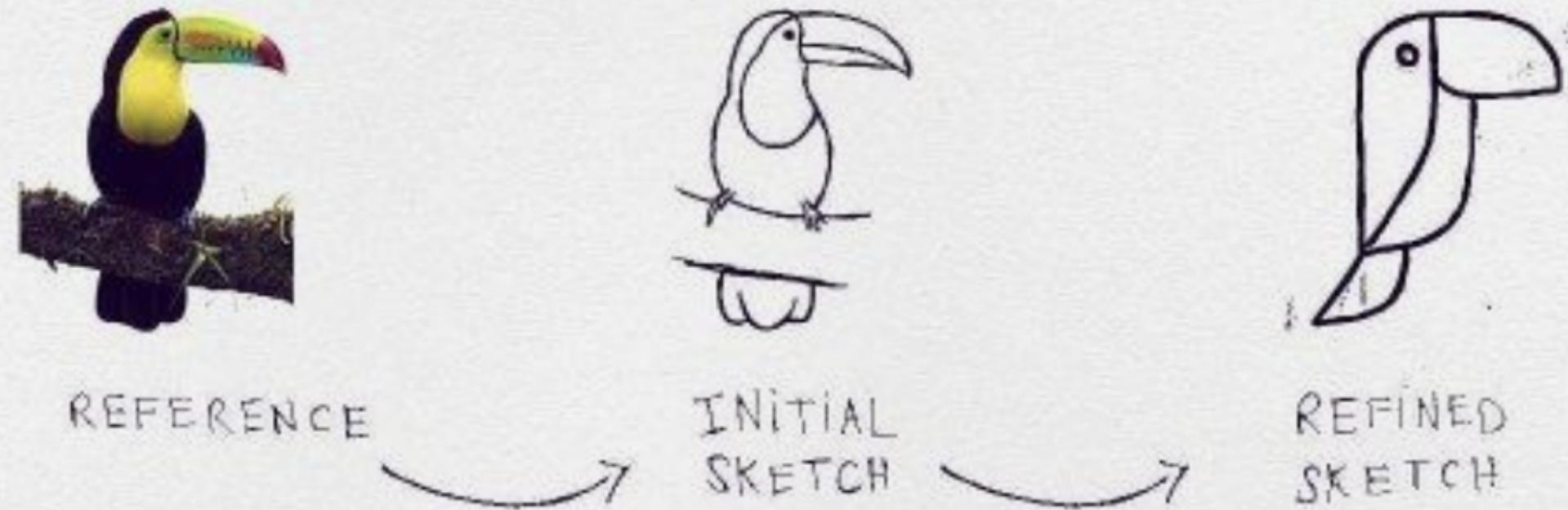




Imagine the abstracted forms



Feature Extraction*

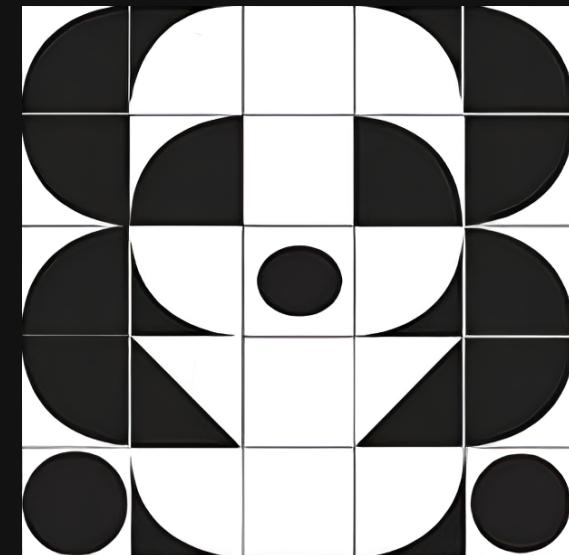


* The term Feature Extraction comes from Neurophysiology represents extracting information from imagery objects.

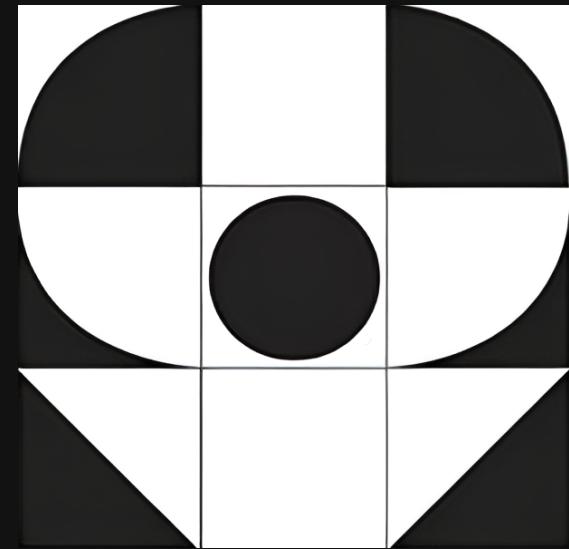
Express the Panda Using Simple Shapes



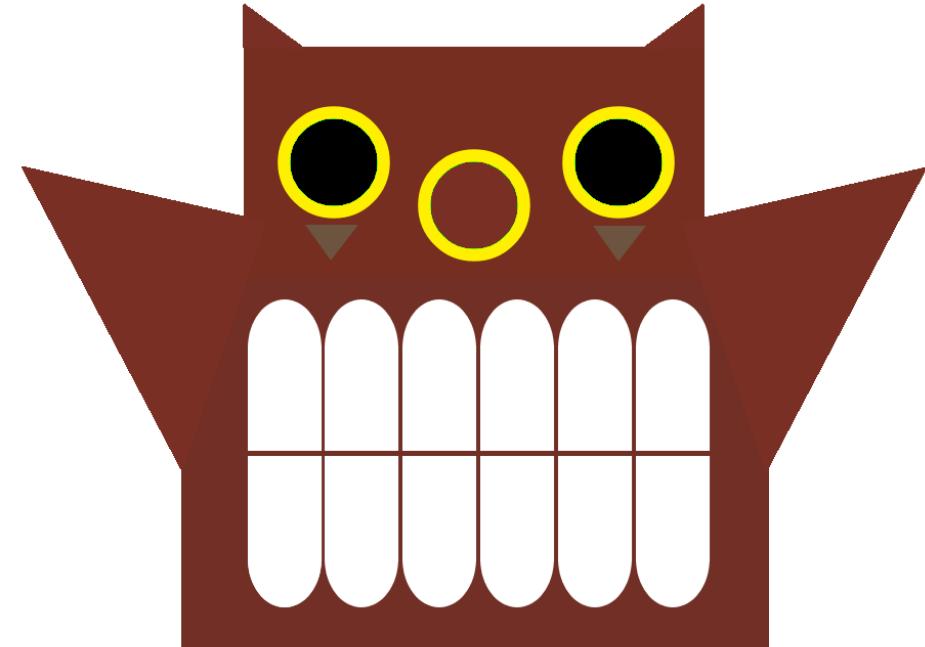
Original Image



Abstracted via Simple Shapes



Abstracted more via Simple Shapes

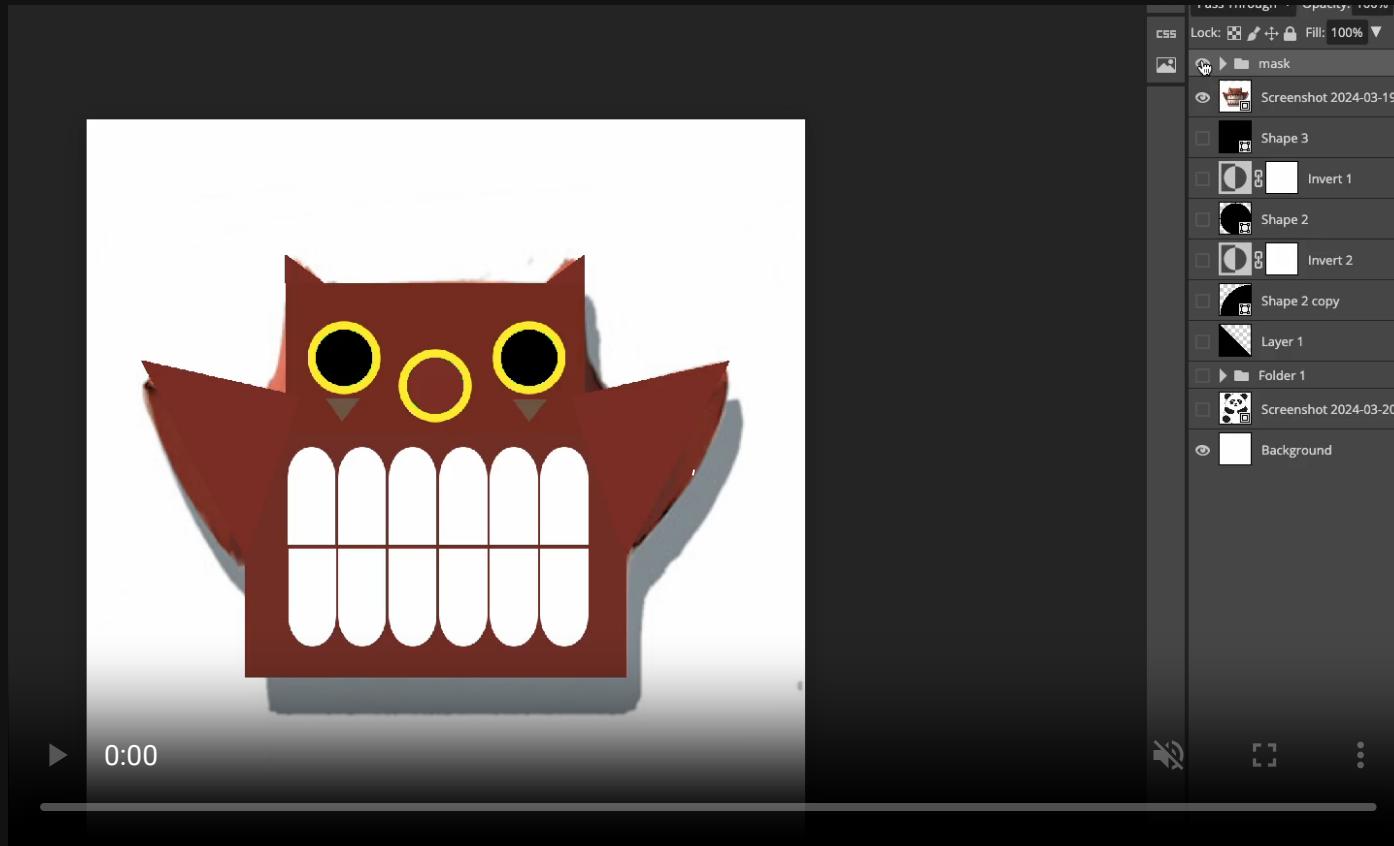


1. Decomposition (15 minutes)



- Determine your canvas size.
- Draw the abstracted version of the mask.
- Note down each shape that you see.
- Note down each shape properties
 - color: white, black, yellow, red
 - position: x and y positions.
 - stroke: Is there stroke around the shape?
 - size: width and height

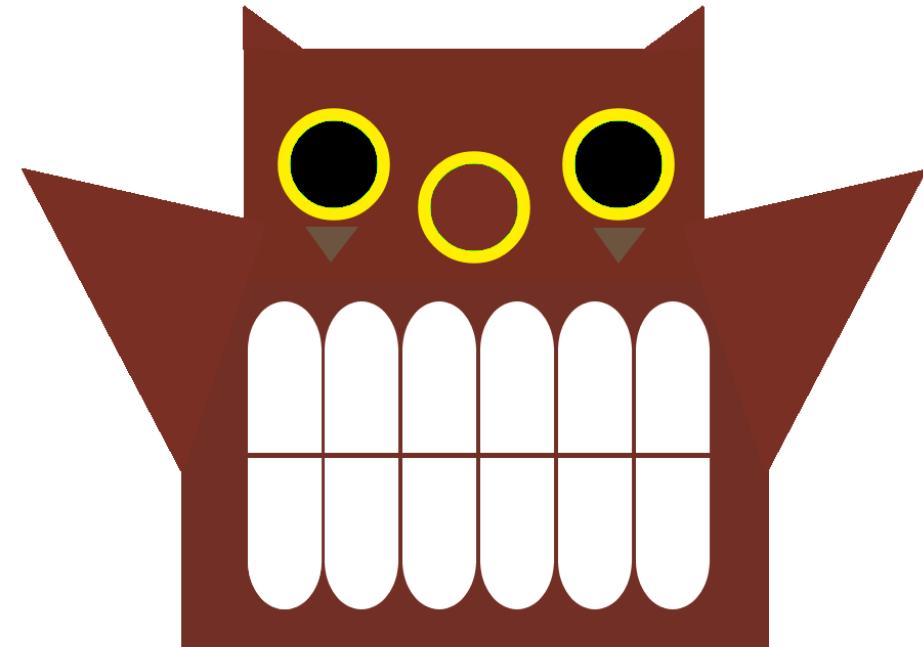
Abstracted Shape

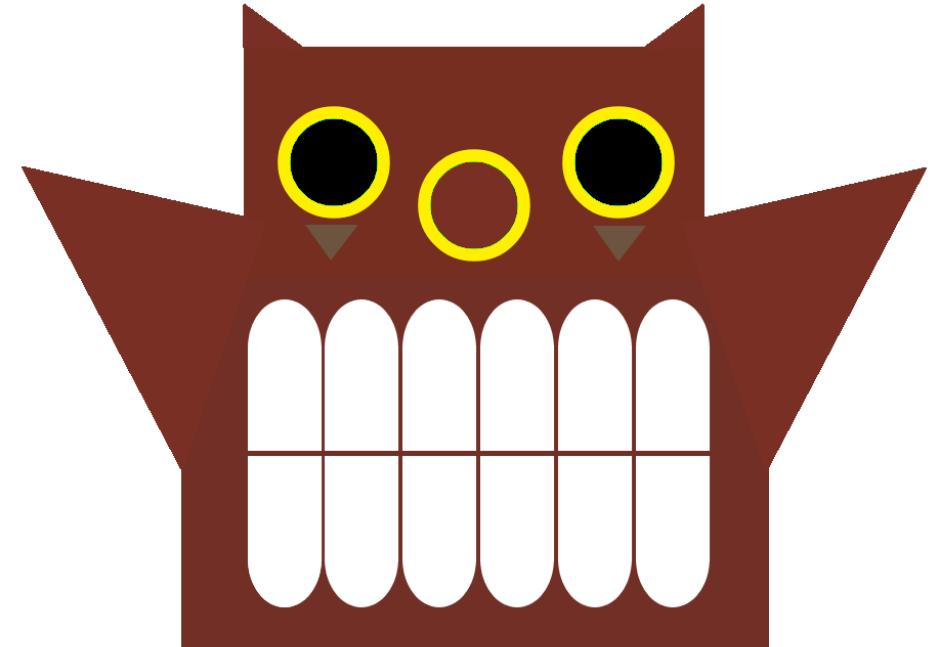


2. Flow (15 minutes)

In this step, we set the order of our notes. We determine what to draw first. Is there any hierarchy between the objects you identified in the previous step? What are the specific colors?

- Determine variables such as color values.
R,G,B codes
- Decide the drawing order of your shapes.
- Layer **1** → Rectangles for the mask base must be in lowest layer.
- Layer **2** → Triangles can sit onto rectangles.
- Layer **3** → Eyes, nose, makeup triangles, and teeth can sit on top most.
- Finally, determine the repetitions. What is repeated in this mask?





3. Translation (15 minutes)



Determine the equivalents of your notes in P5Js language. Create code blocks to reorder them according to the flow.

- Mask shape 1 on top → `rect(x,y,w,h)` $\mu \geq$
- Mask shape 2 at bottom → `rect(x,y,w,h)`
- Mask shape 1-2 color → `fill(r, g, b)`
- Mask shape 3 Horn Left → `triangle(x1, y1, x2, y2, x3, y3)`
- Mask shape 4 Horn Right → `triangle(x1, y1, x2, y2, x3, y3)`
- Continue to code...



4. Algorithm

↻ abc 12
34

- Open your editor.
- Declare all required variables such as color(R, G, B)
-



BREAK

10 mins.

 [Link to submission page](#)

p5.js Community Sketch

p5.js Community Sketch Open Call

We are accepting p5.js sketches from our community! Curated sketches will be featured on the new [p5.js](#) website launching this spring. Please ensure to include `describe()` in your submitted sketch code, so it is accessible for screen reader users.

Curated by [Zainab Aliyu](#), the inaugural collection is especially interested in sketches that explore access, experimental time-keeping, text as image, and algorithmic improvisation.

Submission Deadline: **March 31st**

 [Slideshow](#)



Curated by
[p5.js](#)

Submissions by 0

Followed by 43 - [unfollow](#)



"First, solve the problem. Then, write the code."

John Johnson

Assignments

Please READ! If you don't understand ask me, or your friend, or to an AI chatbot. No Late Submissions   

- Use Decomposition, Flow, Translation, and Algorithm Design steps for your mask generation. I want to see how you implement your ideas. Upload text, diagram or any kind of material that you include your programming activity. Create a pdf for your design process as I show you in this presentation. **(40 points)** [Tutorial](#)
- Write down the mask code. **(20 points)**
- Apply ML5 library to enable head tracking on camera. **(20 points)** [ML5 Face Tracking Example](#), [Face Tracking with Mask Example](#)
- Skim the attached document (Form+Code Book, Repetition pg. 42-65)
- Choose an image that includes repetition in the artwork from the follwing link.
-  Submit the openprocessing link.
-  Submit the sketch source code as zip file as well.