Contextualizing Programming with Algorithmic Art Practices Using Computational Thinking Principles for Undergraduate Design Students

PHD Dissertation Alpha Version →

Overview

- Chapter 1: Introduction
- Chapter 2: Literature Review
- Chapter 3: Theoretical Background
- Chapter 4: Methodology
  - Phase A: ALAP Research
  - Phase B: De-scription / In-scription Method
- Chapter 5: Conclusions

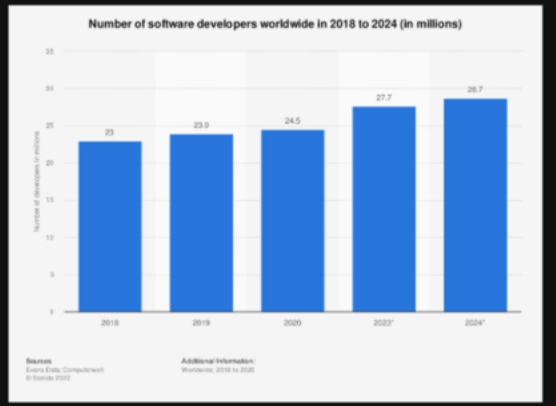
Introduction & Background

Programming knowledge becomes a required skill more than before.

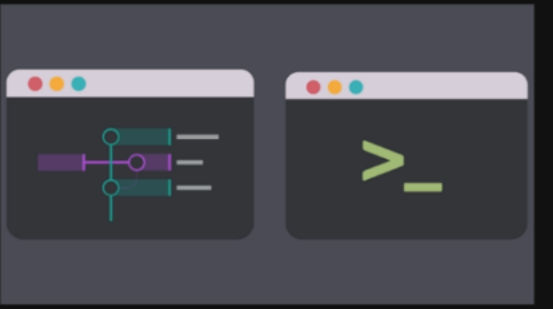
- Paradigmatic Shift in Technology Usage... But Why?
- Increasing Number of Programmers (1.1. The 4th R)
- Changing Grammars in Technology (GUI → CBI) (1.2. Emerging Tendencies)
- Fresh Problem-Solving Paradigms (1.3. Computational Creativity)

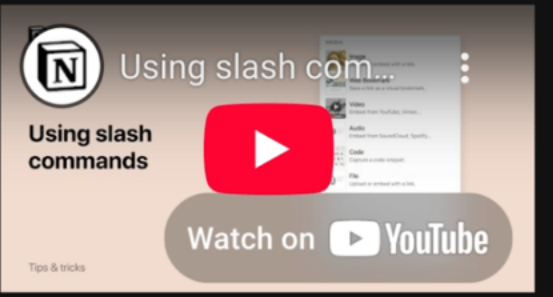
Cases of The Paradigmatic Shift

Increasing Number of Programmers (Statista, 2020)



GUI → CBI (Command-based Interface)



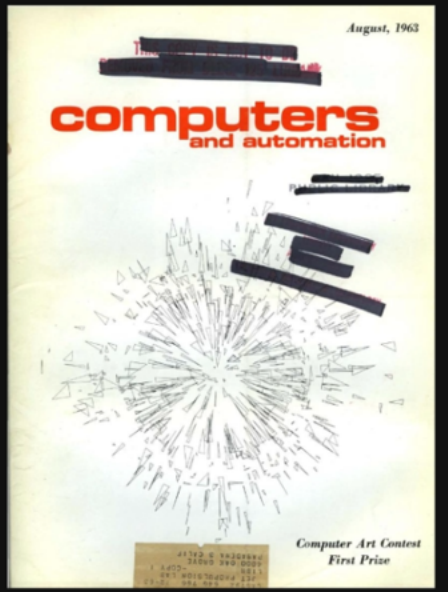


This Presentation is another proof-of-concept

New Paradigms For Computational Creativity

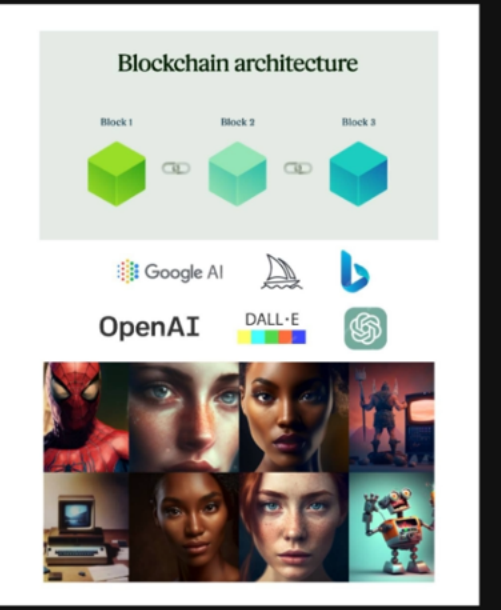
Evolution of computational environments as instruments

Conceptual Art, Algorithmic Art and its sub-branches - 60s



1980-2000 Personal Computers, Discipline-specific software tools

2000-Today Internet, Blockchain, Digital Art, Artificial Intelligence



CBI AI Tools

Eg. Mid Journey, DALL-E

"Prompt: *A Panda fixing the rooftop.*"

Scripting

Eg. Adobe Illustrator


```
var aDoc = app.activeDocument;
if (aDoc.selection.length > 0) {
if (aDoc.selection.length < 2 && aDoc.selection[0].typename == "TextFrame") {
var aTFrame = aDoc.selection[0];
var theChars = aTFrame.characters;
var charLength = theChars.length;
var startSize = prompt("size of first character", 25,"start size");
var endSize = prompt("size of last character", 5,"end size");
var step = (startSize-endSize)/(charLength-1);
for (i = 0; i < charLength; i++) {
    theChars[i].size = (startSize - i*step).toFixed (2);
    redraw();
    }
} else {alert("Please select only one text frame")}
} else {alert("No selection")}
```

More Than Technological Literacy

The act of being active actors rather than passive consumers.

Encourage individuals to open the Black-Box. How?

- ANT
- Constructionist Learning Theories
- Computational Thinking
- Context-based Learning

Statement

Research indicates that contextualized programming education positively impacts students' learning and enthusiasm for the subject. However, there is a requirement for resources and explicit instructional methods specifically designed for students in design-based undergraduate programs, where visual learners are prominent. To bridge this gap, this study introduces materials (ALAP) and a method (D/I) that can be implemented in programming education within the context of algorithmic art.

Research Gap

- Contextualizing the knowledge works (Hansen, 2019; Guzdial, 2009).
- There is a need for elaborate approaches on programming educaiton (Brown & Wilson, 2018).
- Most studies focus on assesment of Computational Thinkig (CT).
- The necessity of CT tools for specific contexts.
- Most studies claim that contextualizing programming practices are beneficial.
- But how?

Research Question

How can we contextualize programming fundamentals through algorithmic art practices to improve students' computational thinking skills and engagement in higher education?

SRQ1: *"What are the common practices used in algorithmic works of art related to programming fundamentals, especially in creative coding?"*

SRQ2: *"How can we relate common practices used in algorithmic art with computational thinking to provide hands on tools that can be used as teaching and learning material in a visual context?"*

Research Contributions

1. De-scription/In-scription Method
2. Tools for Computational Thinking
3. Taxonomy for Algorithmic Art practices
4. Online Algorithmic Art Database
5. Case study of a 4-weeks long practical method
6. Methods and data for researchers

Litereture Review Summary

- Programming is a tedious process for students
- The prejudice against coding among students causes declining attendance to computing classes (Allwood, 1986; Winslow, 1996; Robins et al., 2003; Ring et al., 2008; Yardi & Bruckman, 2007).
- The lack of inadequate computer literacy education at earlier ages (Guzdial, 2009; Yardi & Bruckman, 2007)
- The wrong choice of programming language and out-of-date course materials (Brown & Wilson, 2018; Robins et al., 2003; Guzdial, 2009; Hansen, 2019).
- Contextualizing Programming Fundamentals with Art Increases Student Engagement (Liao & Pope, 2008; Guzdial, 2009).

Theoretical Foundations

Actor-Network Theory: A birds-eye View of The Research

ANT provides an overall umbrella for the purpose of this research to analyze and develop methods and tools for the study.

- ANT ontologically aligns well with Constructionist and Pragmatist paradigms, pointing out the existence of multiple realities while analyzing the social world.
- The research problem (engagement of design students in programming classes) represents all the subjects above as macro-actors.
- "Sensual Objects" from Harman's OOO (prejudices, translation).

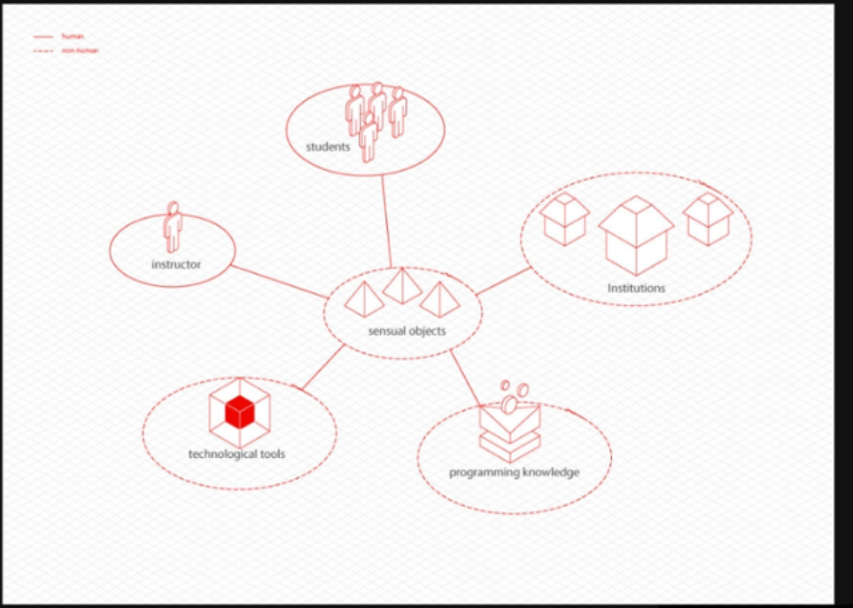
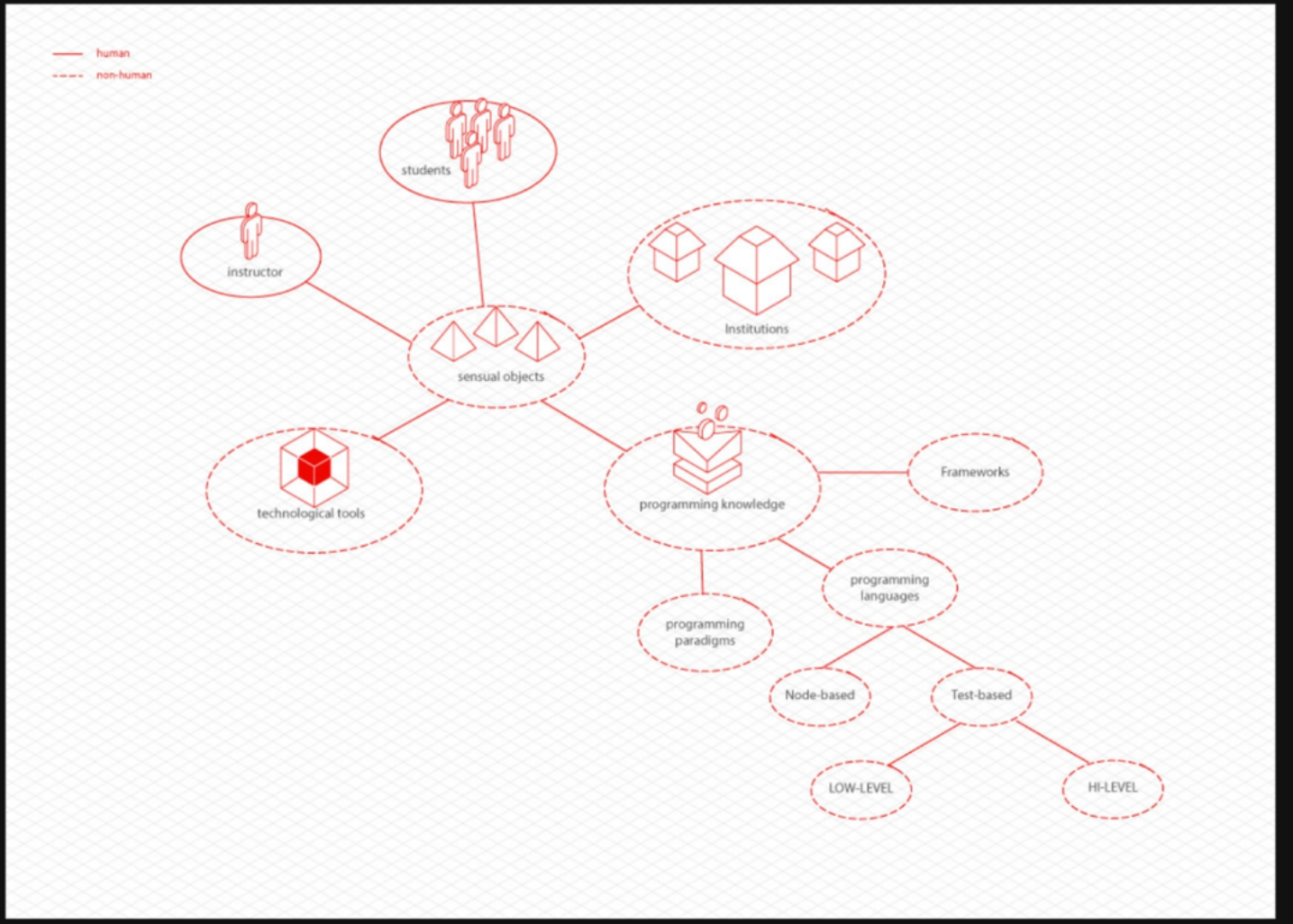


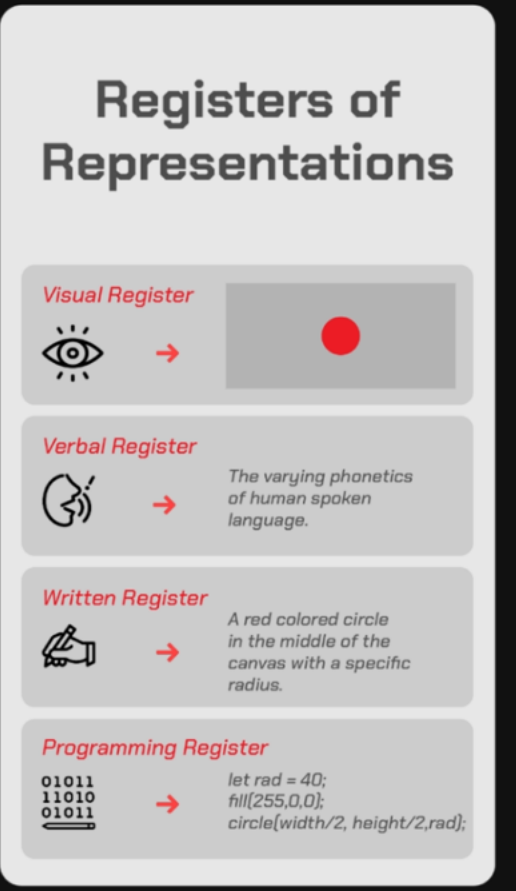
Figure: Research Network Actors


human
non-human
students
instructor
Institutions
sensual objects
technological tools
programming knowledge
Frameworks
programming languages
programming paradigms
Node-based
Test-based
LOW-LEVEL
HI-LEVEL

What is register?

A register of representation refers to a specific semiotic system used to express mathematical concepts, such as natural language, symbolic notation, graphical representations, or visual displays. (Duval, 2006)

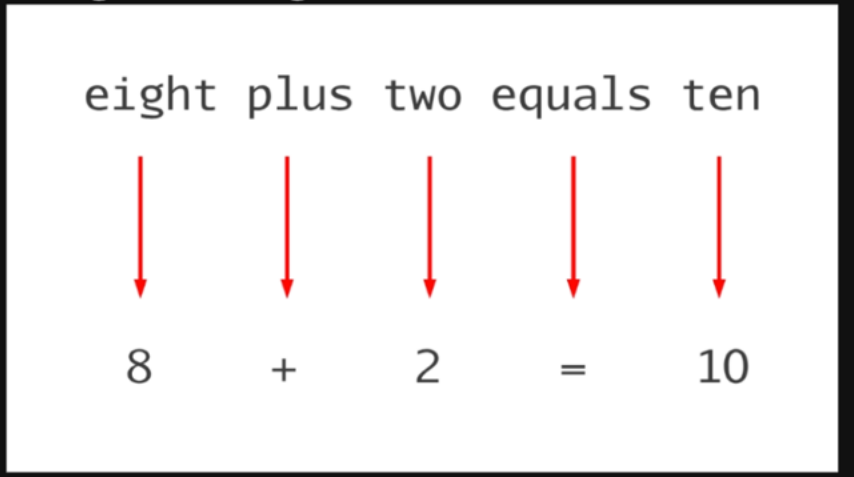
1. Visual Register
2. Verbal Register
3. Written Register
4. Programming register



Register Conversion

Congruent and Non-congruent registers

Congruent Register Conversion

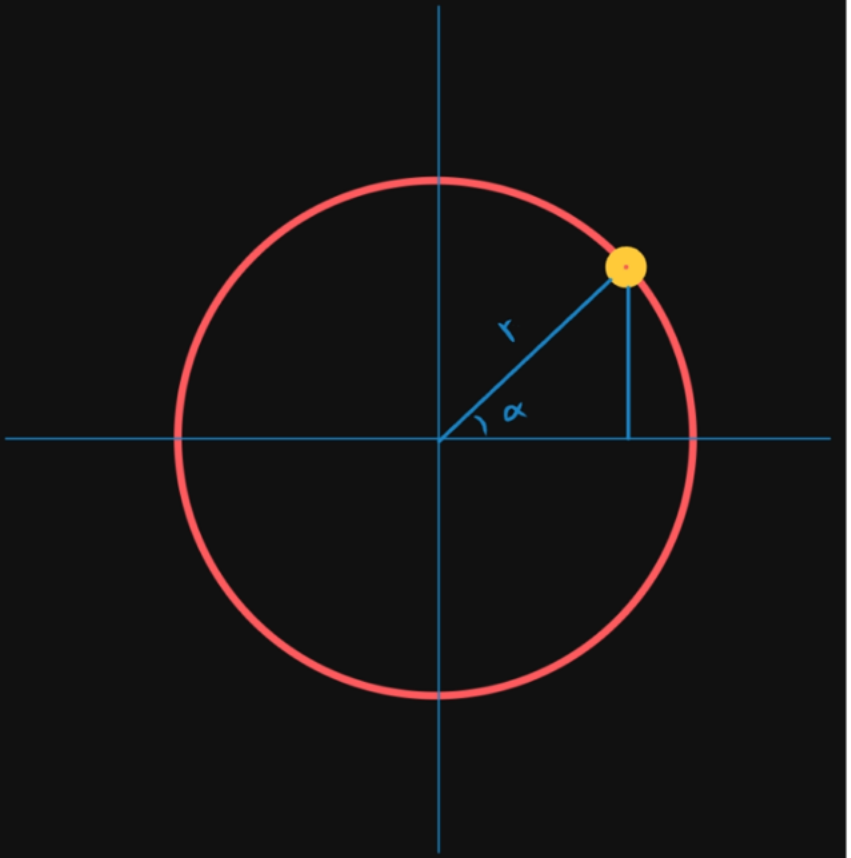


Non-congruent Register Conversion

`Adding eight and half of four gives the sum of ten`, the word "adding" comes before the numbers, two is derived from the half of four and the conversion becomes non-congruent.

How does Duval's theory relate to the programming?

In the context of programming education, non-congruent conversions can arise when students need to translate between natural language descriptions of problems and their corresponding programming code representation - (Bråting & Kilhamn, 2021).



Point on a circular path (Verbal)

Point on a circular path (Trigonometric)

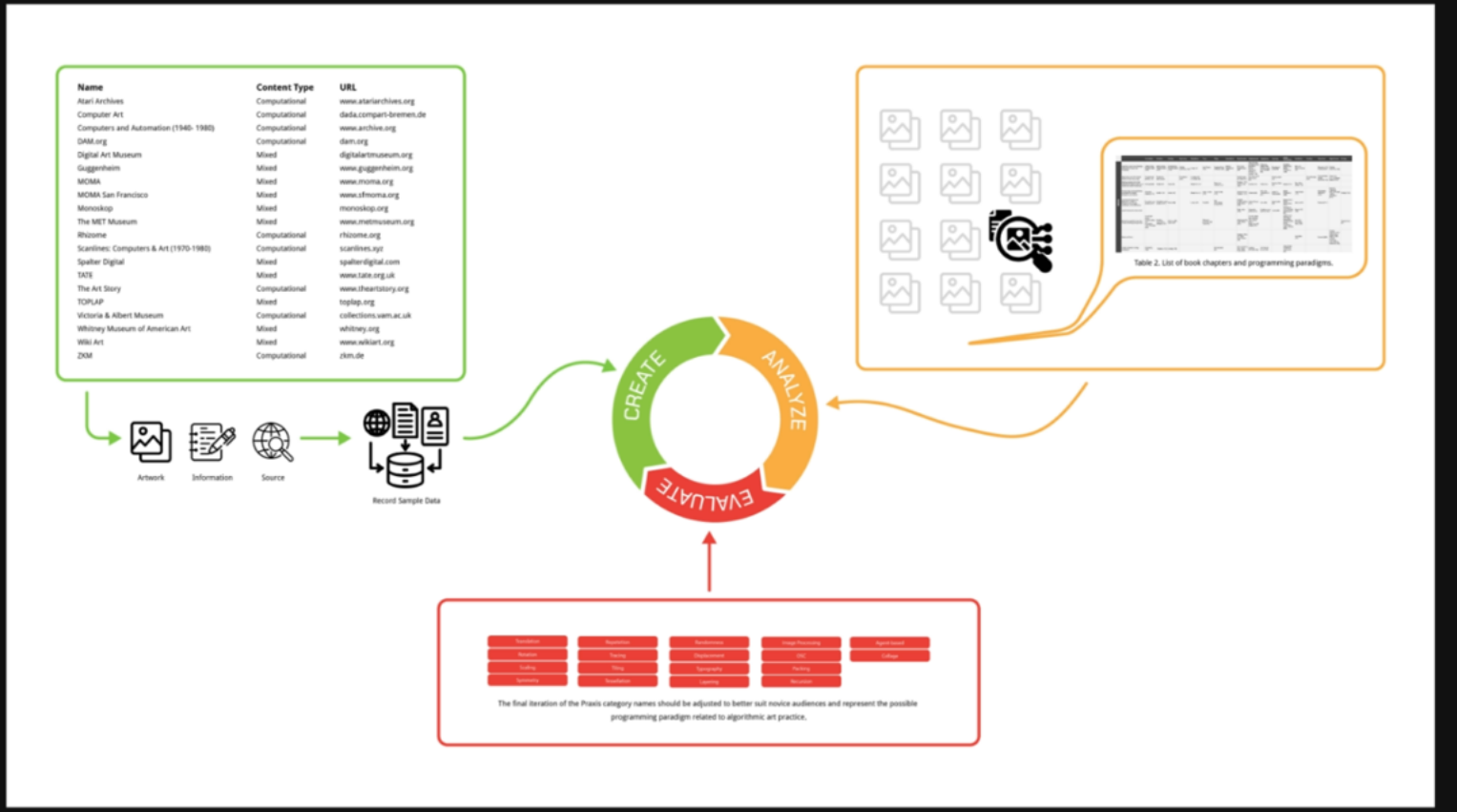
$x = \cos(\alpha) * r$

$y = \sin(\alpha) * r$

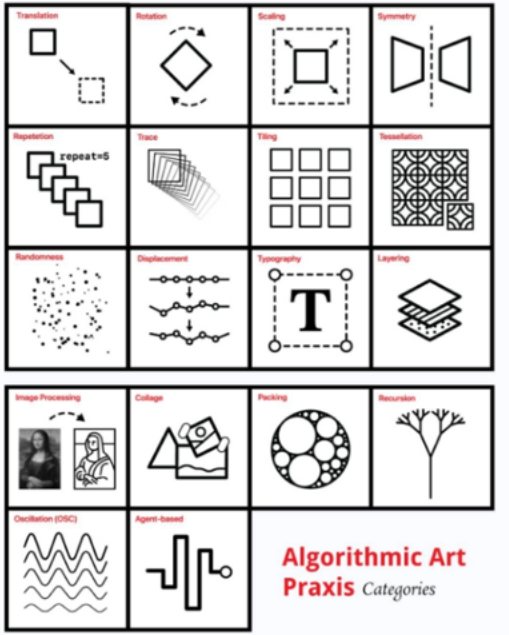
Methodology

- Phase 1: ALAP Database, Categories and Cheat Sheet
- Bridge: Theory of Semiotic Registers
- Phase 2: De-scription/In-scription Method

Categorization Rationale



Outcomes: ALAP Categories



Algorithmic Art Praxis

Cheat Sheet

ALAP Category	Icon	Description	Sample Artworks
Translate		Represents positional changes of the elements relative to each other or the canvas.	
Rotate		Represents orientational changes of the elements relative to each other or the canvas.	
Scale		Represents dimensional changes of the elements relative to each other or the canvas.	
Symmetry		Represents mirrored elements in vertical, horizontal, or custom axis relative to each other or to a point on the canvas.	
Repetition		Represents occurances of a single or group of elements with or without formalistic modifications on the canvas.	
Traces		Represents occurances of graphical elements (lines, curves) along with or without a continous path. The opacity of the repeating pattern may vary on the canvas.	
Tiles		Represents a grid-based distribution of elements on the canvas. Individual graphical objects in the grid do not have to be continuous, mixed, or same with each other.	
Tessellation		Represents a continous distribution of the elements on the canvas. Each tile must have a relational and formal connection in order to create unique patterns. In short, every Tessellation involves Tiling, but not every Tiling can be considered a Tessellation.	
Randomness		Represents the graphical elements as if they were randomly positioned, scaled, or colored on the canvas. The random behavior may be observed as displacement of points on lines or curves.	
Displacement		Represents the positional change of the contour points of graphical elements on the canvas. For example, a straight line consisting of four points can be transformed into a zig-zag shape by moving the points in different directions.	
Typography		Represents the use of typographic elements on the canvas.	
Layers		Represents stacked or redrawn graphical elements on to each other using different colors.	
Image Processing		Represents the recreation of a preloaded image in different styles on the canvas.	
Oscillation (OSC)		Represents occurrence of sinusoidal abstract forms and wave-like shapes on the canvas.	
Packing		Represents fitting the objects into a limited space (a.k.a space-filling or packing algorithm). The rule is that objects must not interfere with each other.	
Recursion		Represents a process that repeats itself to create patterns and shapes, which can continue indefinitely or become highly intricate. For example, a tree starts with a trunk, then splits into two main branches. Each branch further splits into two smaller branches, and so on...	
Agent-based		Represents the creation of a graphical composition showcases continous formalistic features. For example, drawing a sketch without holding up the pencil.	
Collage		Represents the traditional style collage in art. Images can be cropped manually and then transferred to the computer, and using programming practices, they can be positioned on the canvas.	

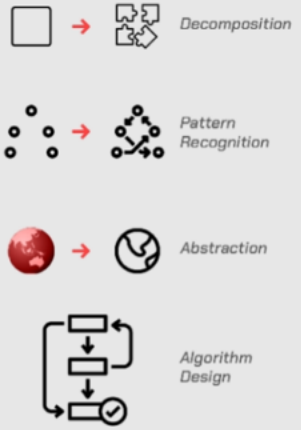
Bridging Phase1 and Phase 2

Semiotic Registers

The Methodological Framework

Computational Thinking

Computational Thinking



ALAP Categories

Algorithmic Art Praxis

Categories identified in the previous research;

Symmetry
Rotation
Scaling
Trace
Layering
Tiling
Tessellation
Image Processing
Collage
Typography
Translation
Displacement
Repetetion
Recursion
Packing
Randomness
Agent-based
Oscillation (OSC)

Multiple Registers

Semiotic Representations

Register Conversion

Registers of Representations

Visual Register

Verbal Register

The varying phonetics of human spoken language.

Written Register

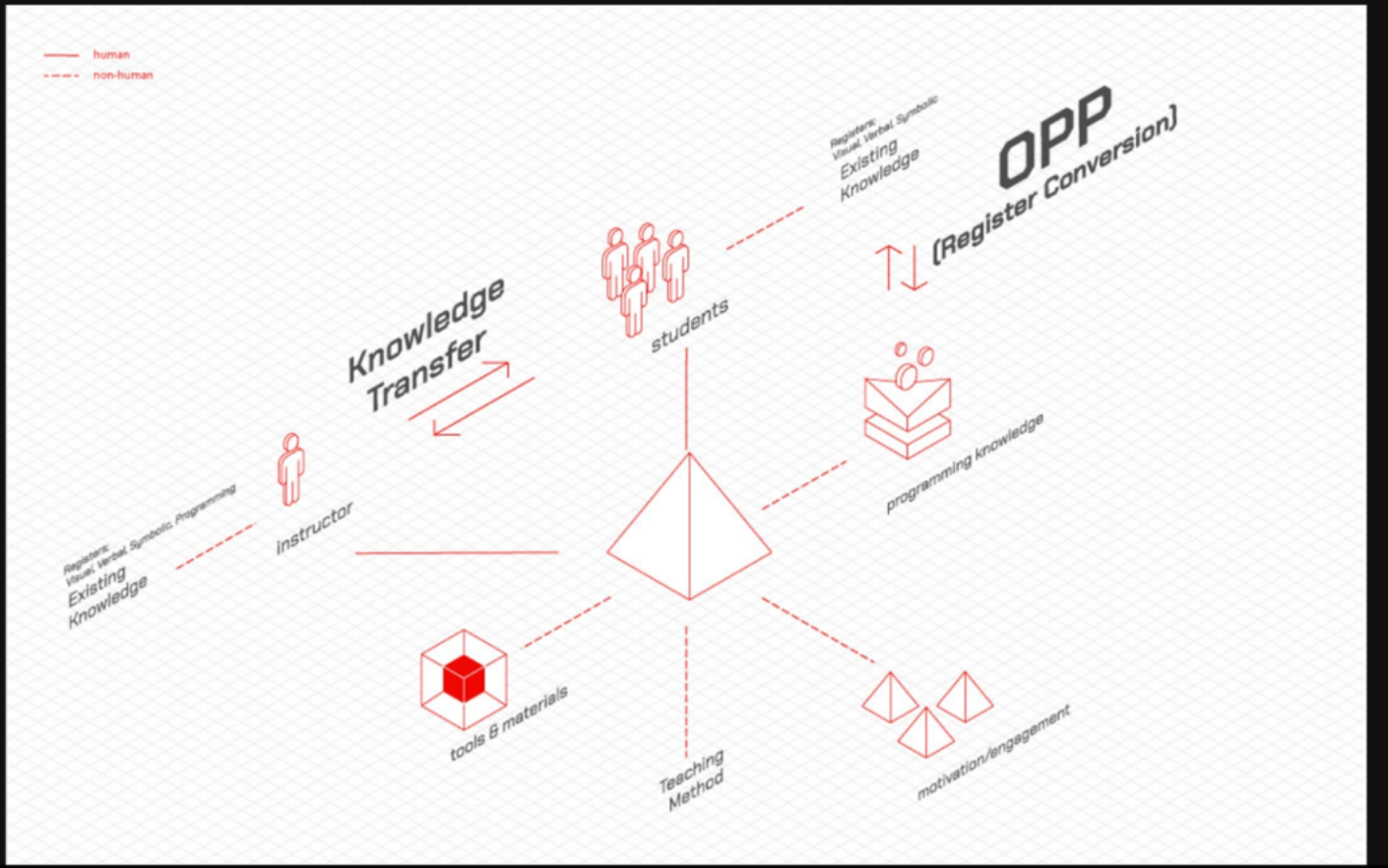
A red colored circle in the middle of the canvas with a specific radius.

Programming Register

```
01011
11010
01011
```

```
let rad = 40;
fill(255,0,0);
circle(width/2, height/2,rad);
```

OPP


human
non-human
Registers:
Visual, Verbal, Symbolic, Programming
Existing
Knowledge
Knowledge
Transfer
Instructor
students
Registers:
Visual, Verbal, Symbolic
Existing
Knowledge
OPP
(Register Conversion)
programming knowledge
tools & materials
Teaching
Method
motivation/engagement

Methodology

De-scription

Step	Name	Registers
1	CHOOSE	Visual
2	ANALYSIS	Visual Verbal Written
3	PROCEDURAL FLOW	Visual Verbal Written

OPP
(Register Conversion)

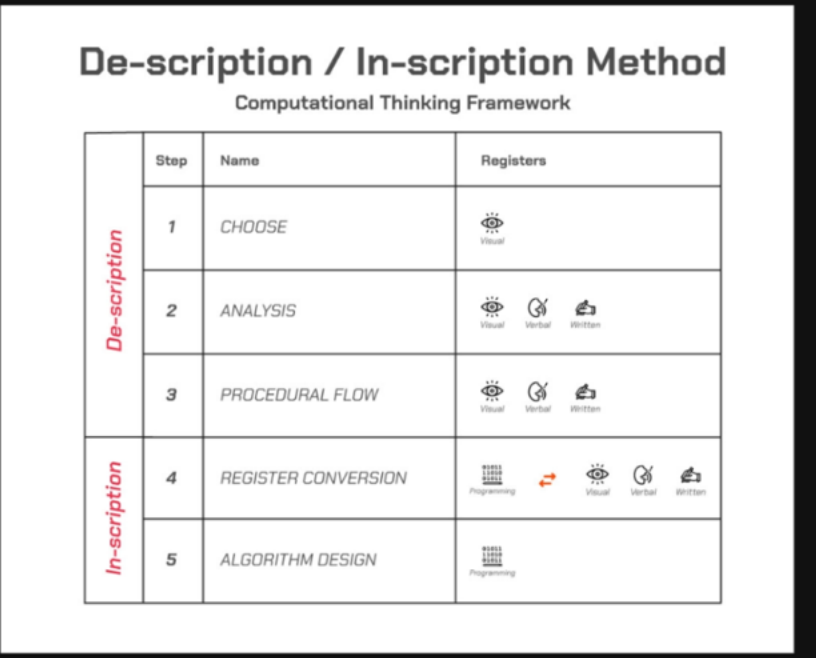
p5.Utils

In-scription

Step	Name	Registers
4	REGISTER CONVERSION	Programming ⇄ Visual Verbal Written
5	ALGORITHM DESIGN	Programming

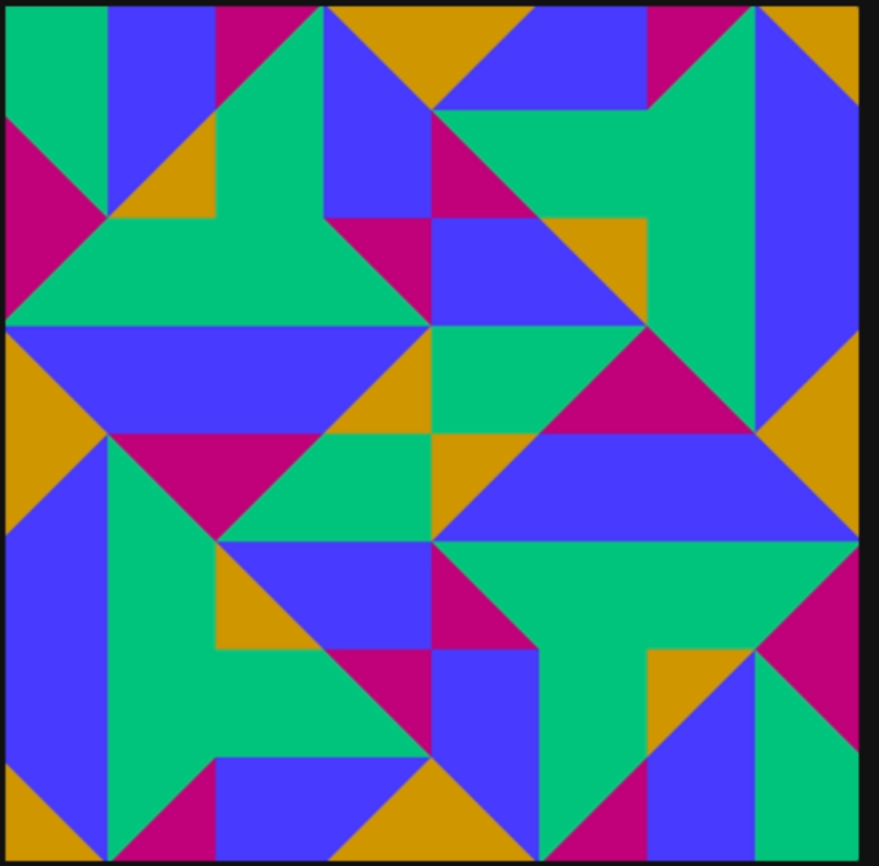
Phase 2: De-scription / In-scription Method

1. Choose an image from the database
2. Analyze it using pen and pencil or any other tool like drawing tablets familiar to the learner.
3. Determine the instruction order.
4. Register Conversion Stage.
   - Use the cheat-sheet.
   - Research using the cheat-sheets (web-sites, previous assignments, ALAP codes).
5. Algorithm Design



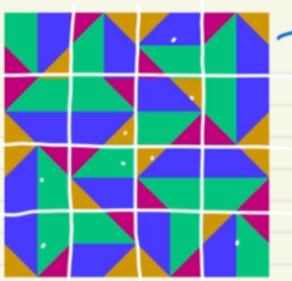
Method Applied

1. Choose: Student selected the image below

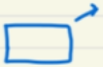


Paul Brown, "4^16-2030311203312120", 2005

2. Analysis



→ the image can be decomposed in the grid

↓  → 8 different squares.

↓ worked grid by grid, each rect carries, triangles

there are 8 squares and are all the same

- 2 rect inside the grid (square)
- Create two triangles on top

36, 222, 143

35, 68, 120

235, 195, 48

239, 23, 121

3. Procedural Flow

Flow

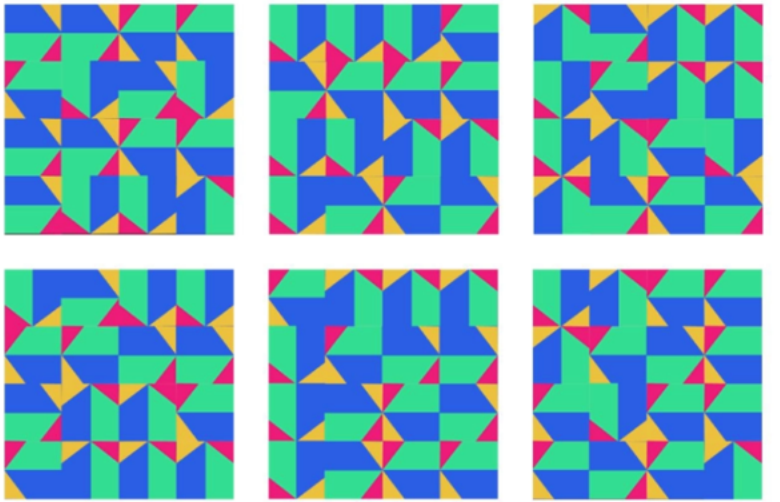
1) draw the grid
2) draw the shape inside one square
3) rotate it

① draw two rect
② draw triangles on top of each grid as it is in the image
③ Fill the color
④ use rotate, translate

4. Register Conversion

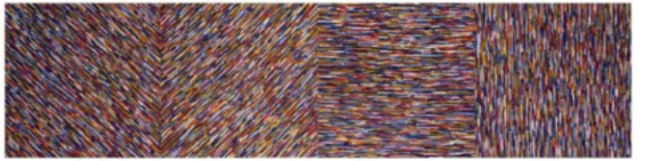
Written Register	Programming Language Register
Green color →	`let green; green = color(36, 343, 143);`
Triangle →	`triangle(x1,y1,x2,y2,x3,y3);`
Square →	`rect(xr1, yr1, sr1, sr1);`
Create canvas 800 by 800 px →	`createCanvas(800, 800)`

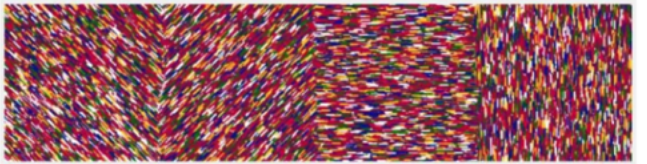
5. Algorithm Design



Other Students

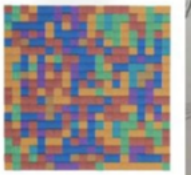
1 Student A

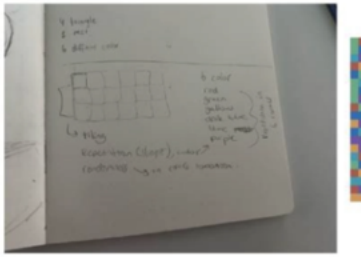


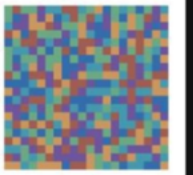


2 Student B

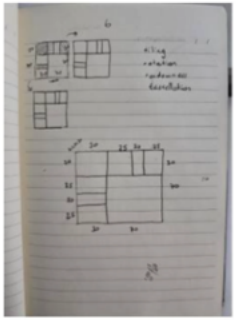
4 Student C

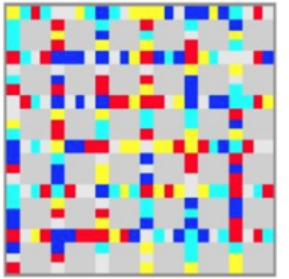




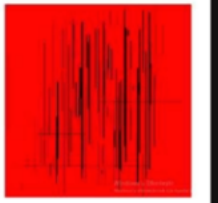


3 Student D

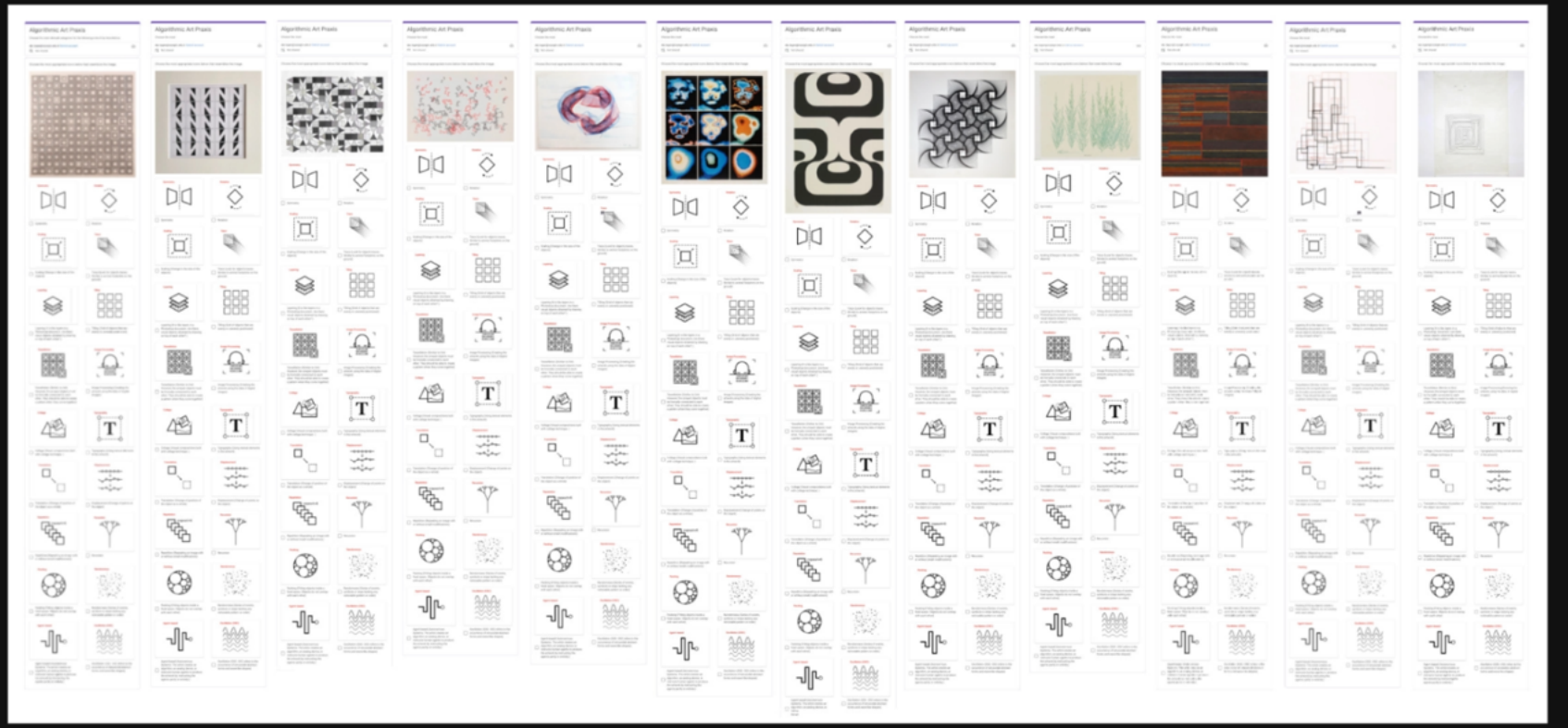




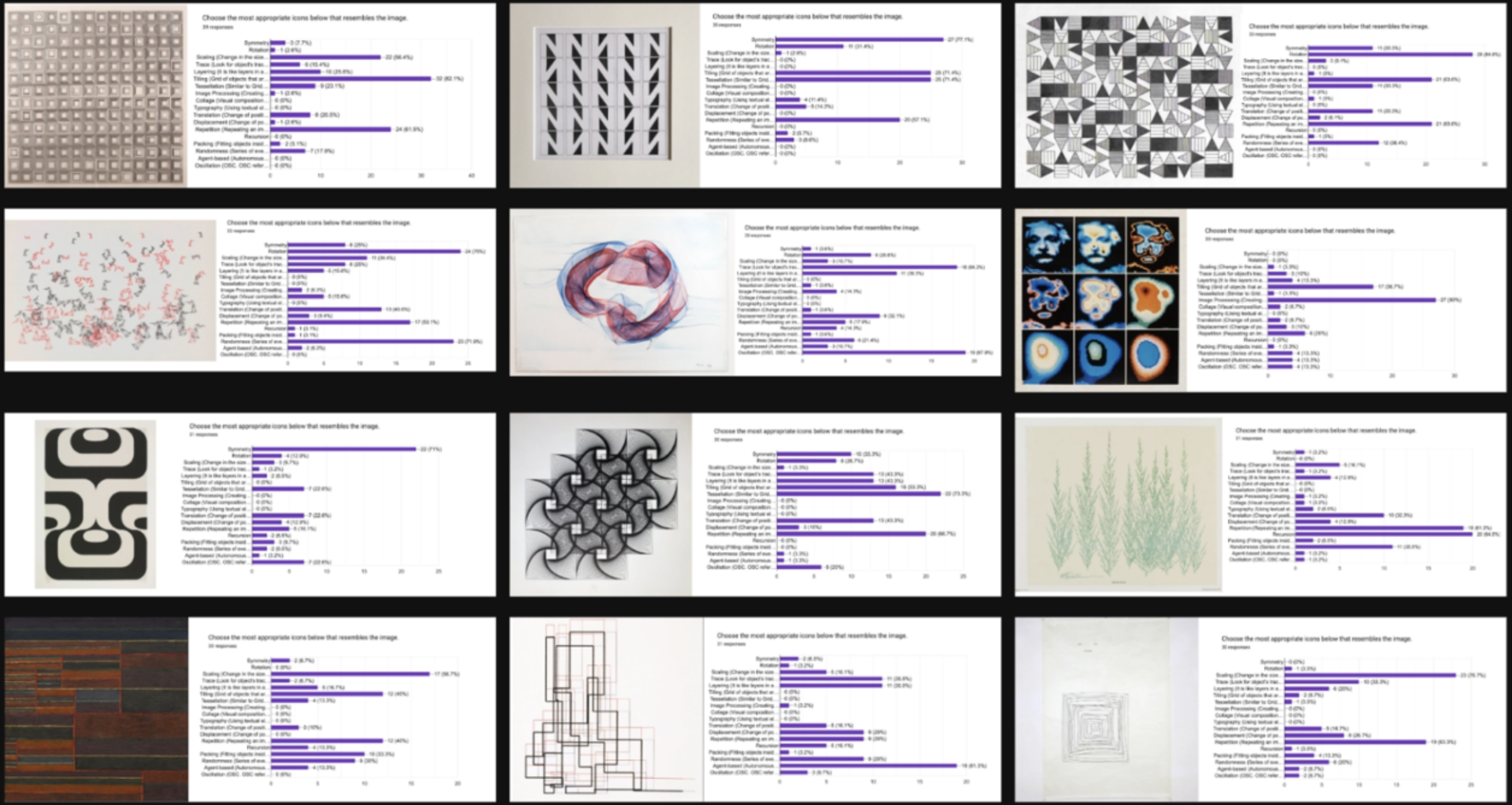
5 Student E



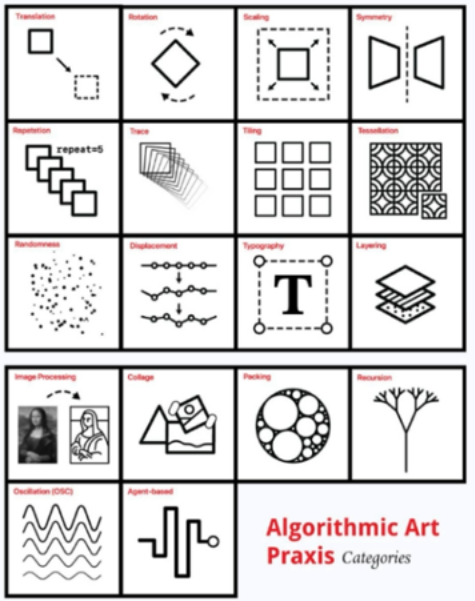
Survey 1 - ALAP Categories



Results (Survey 1)



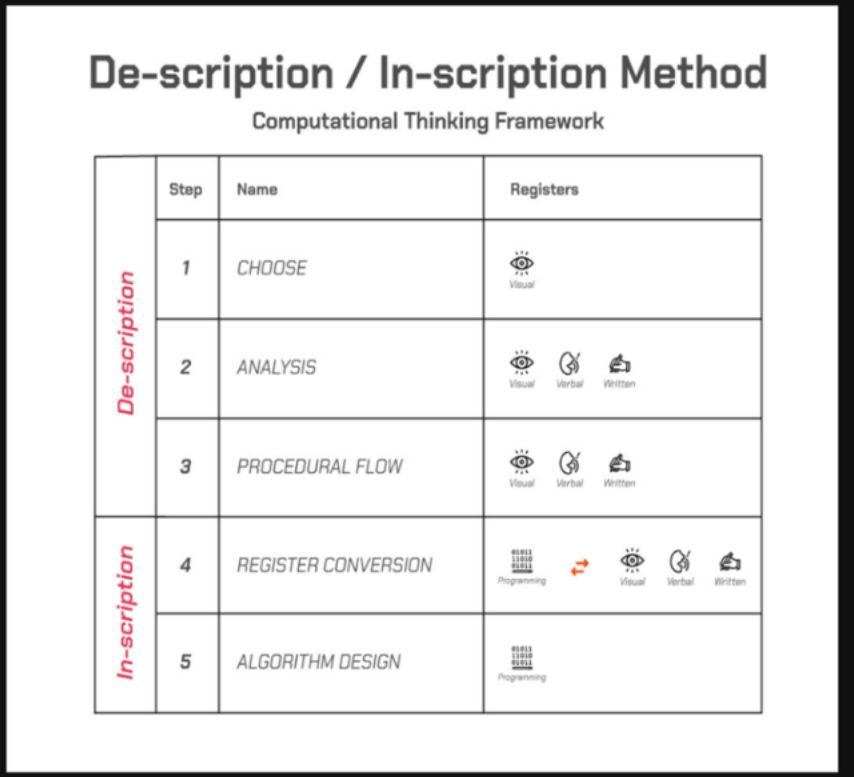
Conclusion: ALAP Survey



Results show that;

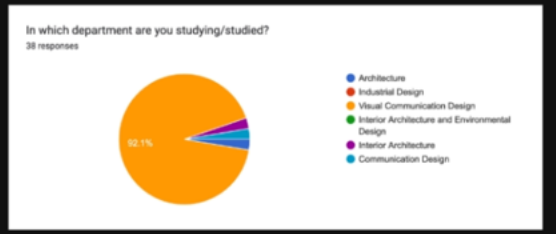
- Participants comprehend categories.
- Some of the selected artworks take time to identify. The artworks, including abstract and natural forms (e.g., Desmond Paul Henry's), become more challenging than those generated with simple shapes, such as Vera Molnar's geometric works.
- Increased engagement.
- Peer assessment.
- In-class discussions increased.
- Participants use cheat-sheets while asking questions.

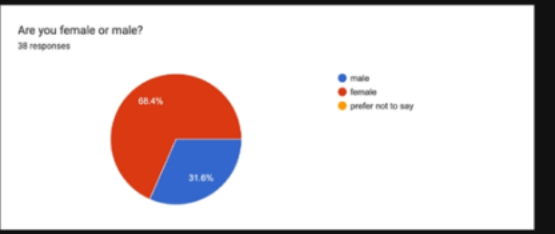
Results (Survey De-scription/In-scription)

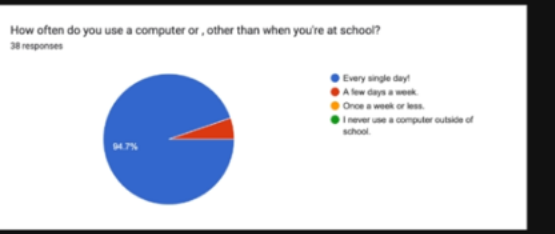


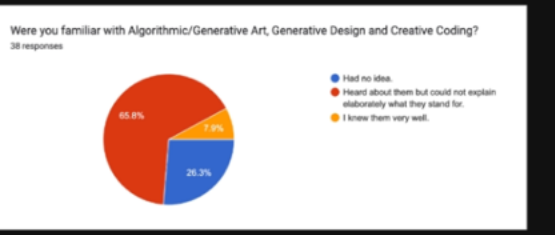
- The survey is divided into two main sections.
- The first section, related to participants' backgrounds, providing essential demographic information.
- The second section, focuses on the efficiency of the De-scription/In-scription method.

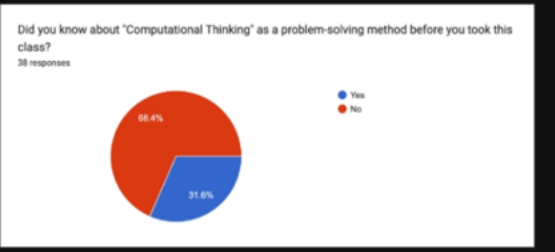
Participants' Background



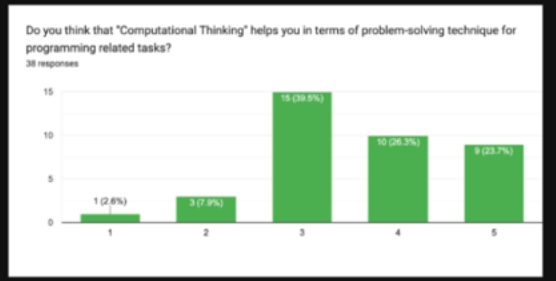


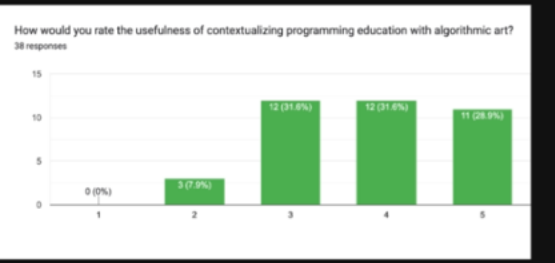


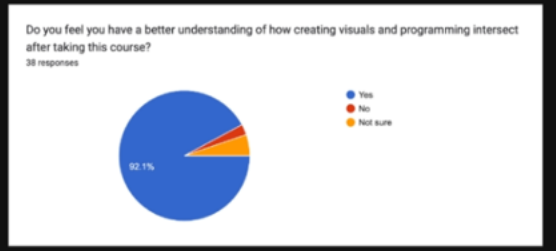


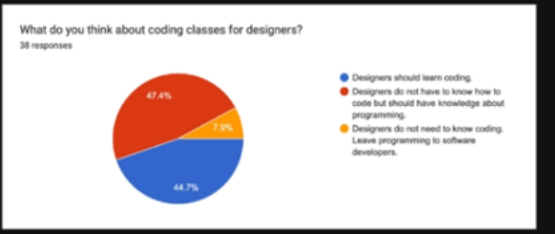


Is The Method Effective?









Conclusion

Equipping students with opportunities to express themselves through visual aids like the ALAP categories can significantly enhance the learning experience, comprehending cognitive processes, and fostering self-assurance and willingness to articulate views.

- ALAP Database provide source material for contextualization.
- ALAP categories eases the process of register conversion.
- ALAP cheatsheet improves communication between the instructor and the learner (CT Tools).
- D/I Method provides an explicit, step-by-step, problem-solving approach.
- Improvement of student ↔ instructor communication.
- Increased self-confidence results in higher engagement.

Thank you.

The End