Creative Coding Major Project

IDEA9103 Creative Coding
School of Architectural Science
The University of Sydney

Group Submission Authors:
Sona Sun, SID: 540977772
Lachlan Wray, SID:460411204

Table of Contents

1 Introduction

This project challenges us to apply our coding skills to digitally represent a chosen artwork using p5.js. The assignment aims to test our ability to translate visual art into an interactive, modular code structure, capturing the essence and meaning of the original piece. Through this work, we explore how programming can convey artistic expression, using techniques learned throughout the semester to bring the artwork to life on a digital canvas.

In response to the current global uncertainty and conflict, we chose to recreate Pablo Picasso's *Dove of peace* using p5.js, due not only for its aesthetic simplicity but for its universal representation of peace and hope. Through this artwork, we want to express our desire for a more harmonious and peaceful world.

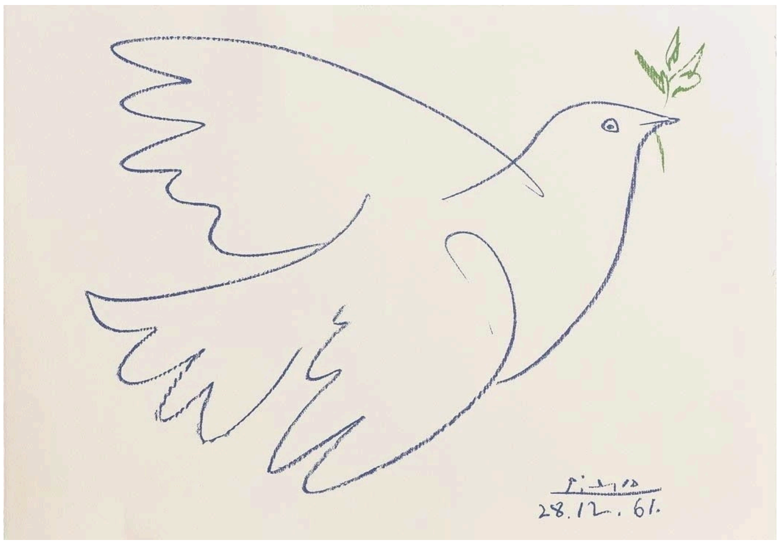


Figure 1 Pablo Picasso - Dove of Peace

2 Research and inspiration

2.1 Background of the original artwork

Pablo Picasso's *Dove of Peace* holds significant historical and symbolic value in both his life and the 20th-century peace movement. Created in the wake of World War II amidst Cold War tensions, Picasso's dove symbolized a world yearning for peace (Waga, 2023). A lifelong pacifist who had witnessed the devastation of war, Picasso used the dove to represent hope, unity, and a vision of a conflict-free world (Cohen, 2014). It became one of the most recognized symbols of peace, especially after featuring prominently at the 1949 World Peace Congress (Waga, 2023).

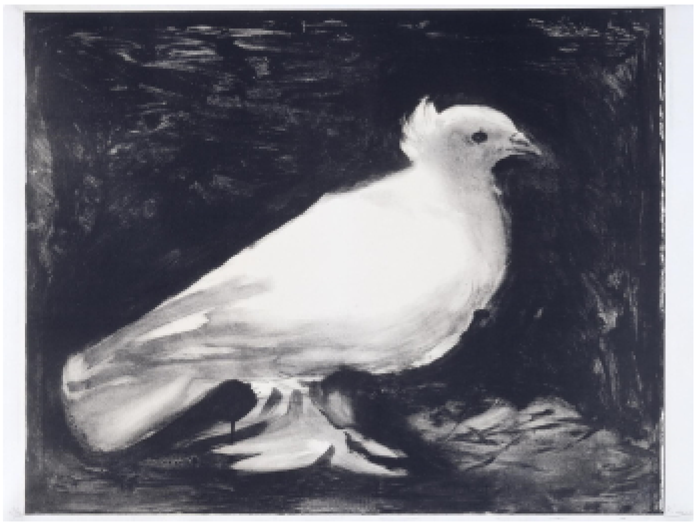


Figure 2 Realistic depiction of *the Dove of Peace* Picasso initially made in for the World Peace Congress, 1949

The dove's minimalist, fluid lines by Picasso (Figure 1) conveyed a sense of purity and elegance, reinforcing the theme of peace. This abstracted form ensured the dove's accessibility and recognizability across cultures, making it a powerful symbol of peace that transcended language barriers (Art in Context, 2024).

2.2 Recreation Inspirations:

Inspired by Picasso's vision of simplicity and universal appeal, our recreation of the Peace Dove is crafted using filled shapes to create a fluid, organic quality that enhances its elegance. The use of white preserves the purity of the original artwork.

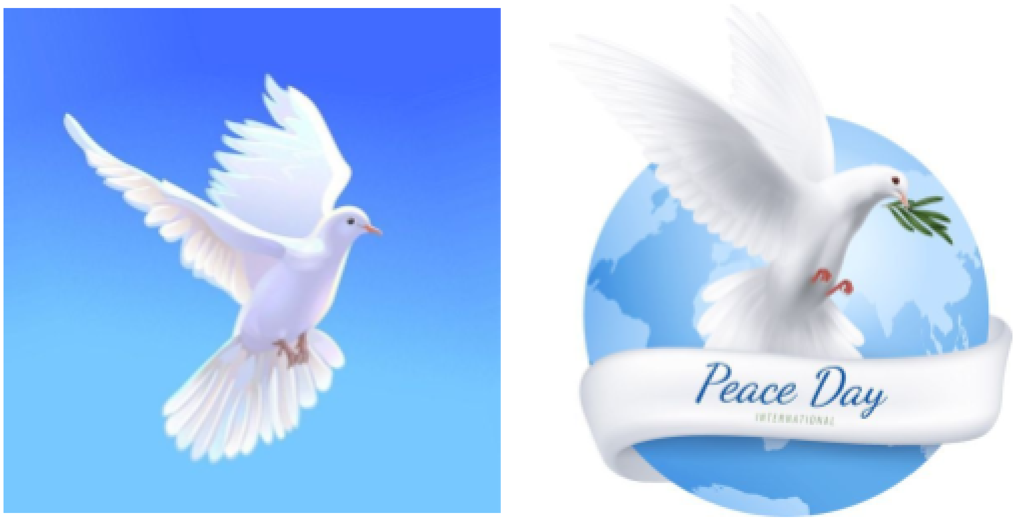


Figure 3 Inspiration for use of colour

Source: International Peace Day

Source: Images By Narendra Sagar On Pigeons E85

The soft gradient blue background in Picasso's *Dove of Peace* not only symbolizes essential elements like the ocean, sky, and earth, but also creates a calm and peaceful feeling, adding depth to the universal themes of peace and harmony. While Picasso's dove symbolized a call for peace, our 3D animation aims to bring it to life in a way that reflects today's world, reminding us to cherish the peace we have. The slow, calm motion of the dove's flight enhances its presence and deepens its symbolic impact.

3 Technical planning

Our group aims to represent the peace dove in p5.js using a structured and modular approach with clean, reusable code while ensuring the dove fits well within the canvas.

Generating point data

The first challenge for generating intricate curves to replicate the outer profile of a dove meant an approach was to be developed to create a set of points for each curve. It was found this could be achieved in OnShape, a 3d CAD (computer aided design) software, as per the FeatureScript (OnShape native language) (Brown, G, 2024).
Each significant part of the dove (ie the body, eye, left and right wing etc.), can then be represented in a separate .csv file of x,y coordinates.

3.1 Classes for Modularity

To maintain a clear and organized structure, we decompose the artwork into individual components, each of which can then be drawn with a class.

Sketch Class

This class will load point data from CSV files and store the points in an array. The class has methods for:

- Loading Points: reads coordinates from CSV files and stores them as Point objects.
- Connecting Points: draws lines between points to form shapes, note the higher the density of points on a given curve, the smoother the curve looks

Scene Class

This class manages the background drawing, adding visual depth with gradient colour.

- **drawGradient()**: a method will design the gradient with colours and map it over the canvas dimensions (a smooth transition from white (c1) to light blue (c2) vertically).

3.2 Functions for Drawing and Animation

Additionally, separate functions will be used in the main sketch.js file to:

- Adaptive Screen Size and Scaling

- Translation and Positioning: translate() is used to position the dove at the center of the screen. Thus, any rotation/animation is centered.

4 Implementation

This section showcases the iterations and progress made throughout the process, highlighting the stages of our work and the outcomes achieved. It includes screenshots of various outputs and demonstrates how our design evolved.

1. Firstly, to extract a series of points from a 2d sketch, which was found to be possible using OnShape. The Featurescript tool was initially tested with basic shapes (Brown, G., 2024) and then applied to create a 2D dove sketch in Onshape. The arc tool was used to form the various sections of the dove, allowing for smooth yet intricate curves.

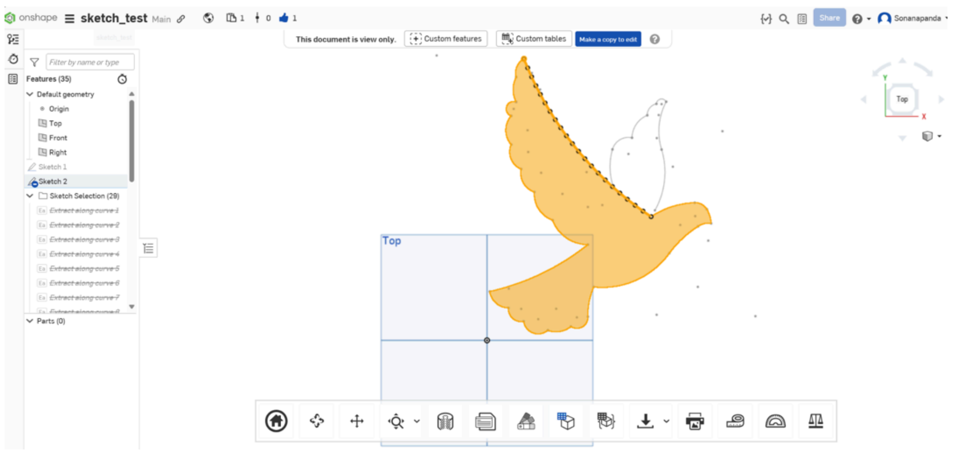


Figure 4 Draw a 2D dove sketch in Onshape

1. Microsoft Excel was used as an intermediate tool to export the table from OnShape, into a CSV format, so that each value was separated by a comma. All the arcs drawn were concatenated in OnShape, so there was no clear start/finish for each arc, and ID numbers increased across the whole drawing.
2. Instead, the Featurescipt was applied to each arc, and a separate table was exported to Excel for each arc. The main issue with this however was each table had an ID number starting at 1, except with inconsistent direction (decreasing vectors for some, increasing for others). We used this ID number as a guide for each arc, with the code logic to start a new arc if the ID decreased. Initially we tried drawing each arc with the p5.js vertex() function, however this caused issues with repeating values, and lines which would cut across from the start to end of some arcs. See Appendix A.
3. A new p5.js Project was created and used the loadTable() as a function to import the CSV file with the x, y coordinates representing the dove's outline.
4. Define Classes and Functions for Drawing:

- Create a Point Class to represent each x, y coordinate as a point. Define a Sketch Class: This class will handle the dove drawing, using an array of Point instances.

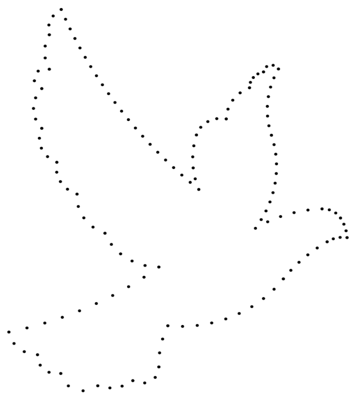


Figure 5 Using the points to draw the dove.

- In the Sketch class, a connectPoints() function was defined to connect the sequence of points to render the outline of the dove on the canvas. A jitter function (repeated sketches with random variations to the points) was applied such that it gave the sketch a natural drawing look.

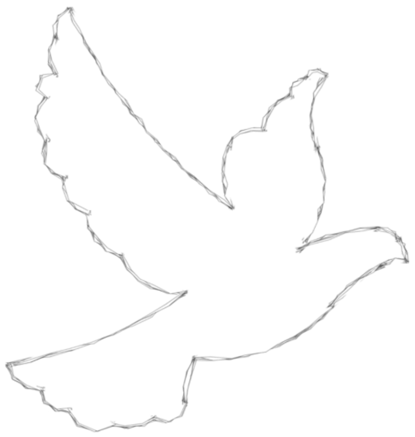


Figure 6: draw lines between points to create the dove

Inspirations for creating an animated flapping of the wings (from Purple Pie Studios, 2021) were of interest and methods to creating 3D transformations were investigated. It was found In p5.js projects, WebGL can be used to introduce 3D transformations, allowing for the rotation, scaling, and translation of shapes in three-dimensional space. This enhances interactivity and depth in digital art projects, providing a dynamic way to engage with visuals on the canvas.

- First, the project canvas was to be defined using canvas (width, height, **WEBGL**).

- A function was created for animating the rotation of the wings. Issues were evident that for a defined background, the wing that rotated in the negative z direction disappeared behind the background layer (which was at z=0).
- The whole bird drawing was then to be translated forwards in the positive z direction until the wings rotation was visible across the whole range of motion.
- Shader files were experimented with, however layers were difficult to allow for other background effects, so the project was reverted to using standard p5 functions to create a background layer.

The background class required a gradient color with the feeling of height generated by the fading of the blue to white. Testing and Refinements were completed with these concepts being considered:

- Adjust colors, opacities, and layering to finalize the visual appeal.
- Test the final drawing and animations on different screen sizes and adjust any parameters as necessary.
- Delays were evident in the rendering process, so a buffering procedure was implemented to render the graphics outside of the canvas before being displayed (Moussa, A, 2022). This concept can be implemented using the p5 function createGraphics() and texture().
- Fine-tune the positioning, scaling, and animation timing to achieve a smooth and cohesive look.
- Adjust Positioning and Scaling, a limiting function was implemented because the image wouldn't animate correctly for scaling over a certain range, the rotation function would need to be revised further.

The final implementation provided a flexible, modular codebase that brings Picasso's dove to life in an animated, modernized form embodying the theme of peace.

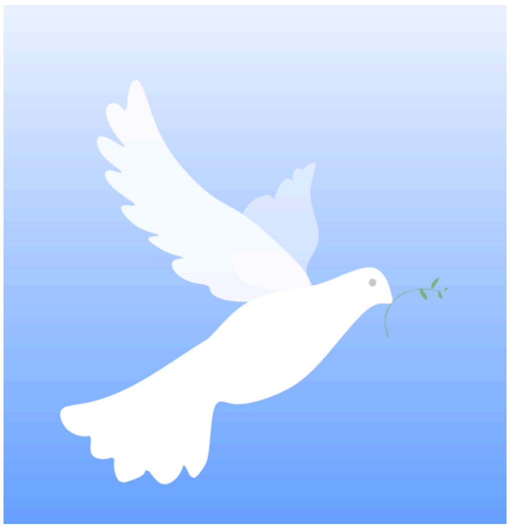


Figure 7 Final outcome for group submission

5 Technical overview

The following section provides a detailed overview of our final code, including key snippets and explanations, along with a visual map illustrating the structure and relationships between major code components.

5.1 Project Structure and Flow

The program is organized to load data points from CSV files, use classes to handle data and drawing, and apply animations for a fluid dove illustration. Each section performs specific tasks to ensure the smooth functioning of the dove illustration. Below is a flowchart to represent the structure:

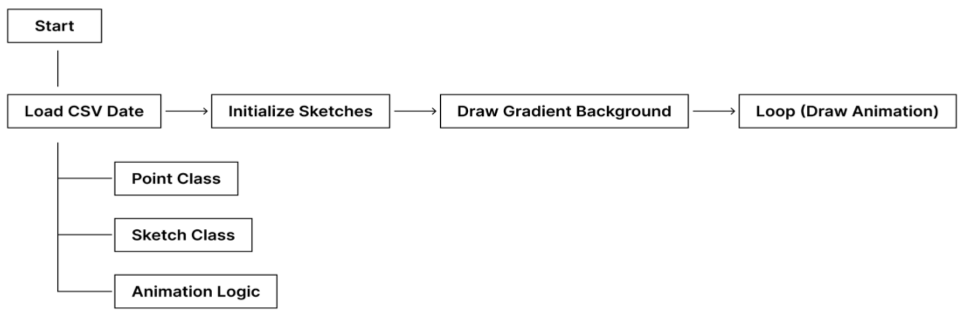


Figure 8 Project flowchart

5.2 Major Code Components

- Data Loading and Initialization

The preload() function loads data files containing x, y coordinates for each part of the dove and stores them in the sketches array. Each file is loaded as a Sketch instance, with a specific color and opacity, and is stored in the sketches array.

- Scene and Sketch Classes

The Sketch classes represent the dove's structure. Each coordinate is stored as a Point, and each component (e.g., wings, body) is managed as a Sketch. The final implementation used the Spline Feature in OnShape, and enable easy exporting of all the points into a single table for each section of the dove.

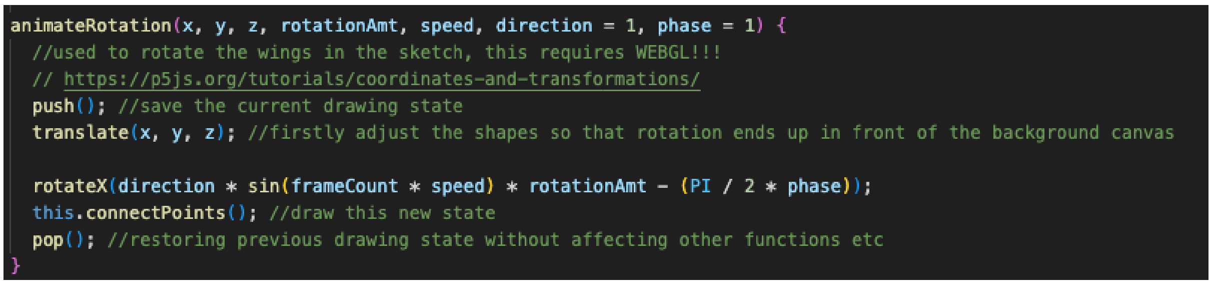
5.3 Sketch Class

This class handles loading points, translating and scaling the points, and then connecting the points into a shape with a specified colour.

Drawing and Animation Logic

A method within the Sketch class was defined for both translation oscillation, and for rotation oscillation. The wings were drawn in the canvas using the animateRotation method, and the rest were drawing using the animateOscillation method. The draw() function handles the main drawing process, calling each sketch to render the dove components with animation effects.

For example, the animateRotation method was constructed like this:

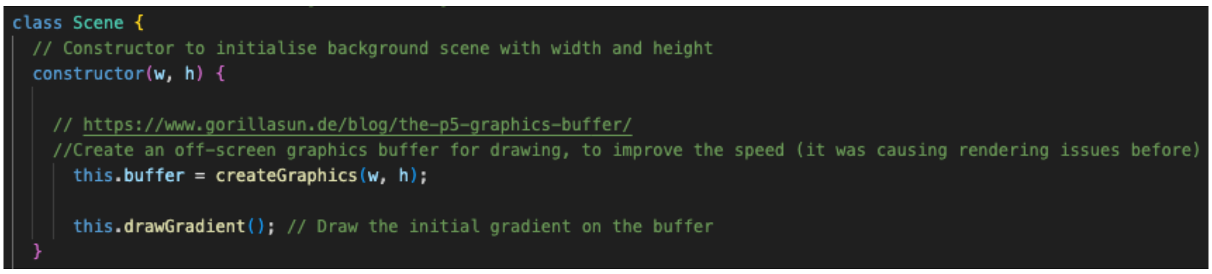


```
animateRotation(x, y, z, rotationAmt, speed, direction = 1, phase = 1) {
  //used to rotate the wings in the sketch, this requires WEBGL!!!
  // https://p5js.org/tutorials/coordinates-and-transformations/
  push(); //save the current drawing state
  translate(x, y, z); //firstly adjust the shapes so that rotation ends up in front of the background canvas

  rotateX(direction * sin(frameCount * speed) * rotationAmt - (PI / 2 * phase));
  this.connectPoints(); //draw this new state
  pop(); //restoring previous drawing state without affecting other functions etc
}
```

5.4 Scene Class

A Graphics buffer was setup with the constructor for the class, as per this standard implementation of a buffer:

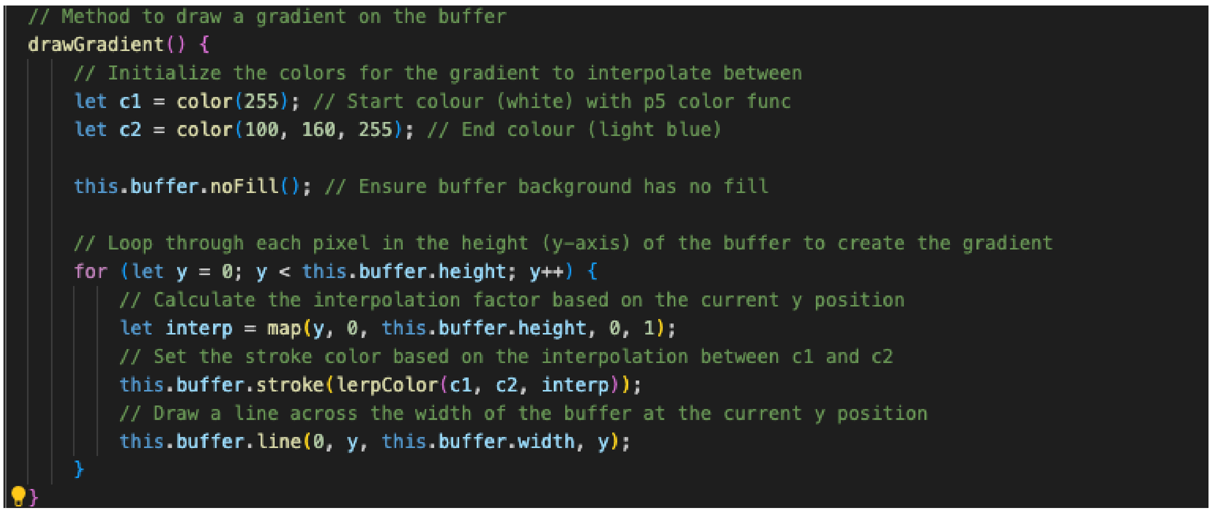


```
class Scene {
  // Constructor to initialise background scene with width and height
  constructor(w, h) {

    // https://www.gorillasun.de/blog/the-p5-graphics-buffer/
    //Create an off-screen graphics buffer for drawing, to improve the speed (it was causing rendering issues before)
      this.buffer = createGraphics(w, h);

      this.drawGradient(); // Draw the initial gradient on the buffer
  }
```

A method was defined with the Scene class; drawGradient(), whichsets up a soft blue gradient to enhance the dove's peaceful theme. The lerpColour() Function was used to create the interpolated values of colour across the dimensions canvas, between colours bounds c1 and c2.



```
  // Method to draw a gradient on the buffer
  drawGradient() {
      // Initialize the colors for the gradient to interpolate between
      let c1 = color(255); // Start colour (white) with p5 color func
      let c2 = color(100, 160, 255); // End colour (light blue)

      this.buffer.noFill(); // Ensure buffer background has no fill

      // Loop through each pixel in the height (y-axis) of the buffer to create the gradient
      for (let y = 0; y < this.buffer.height; y++) {
          // Calculate the interpolation factor based on the current y position
          let interp = map(y, 0, this.buffer.height, 0, 1);
          // Set the stroke color based on the interpolation between c1 and c2
          this.buffer.stroke(lerpColor(c1, c2, interp));
          // Draw a line across the width of the buffer at the current y position
          this.buffer.line(0, y, this.buffer.width, y);
      }
}
```

GitHub Links

The individual components for this submission can be found here:

Sona Sun: https://github.com/sonanapanda/ysun0714_9103_GroupF.git

Lachlan Wray: https://github.com/lwra4681/IDEA9103-Major-Coding-Assignment.git

6 References

Barash, D. P., & Webel, C. P. (2021). *Peace and conflict studies* (5th ed.). Sage Publications.

Brown, G. (2024). *Onshape forum examples.* Retrieved from https://forum.onshape.com/discussion/23901/export-or-collect-the-coordinates-of-points-on-a-sketch

Cohen, R. (2014). Picasso's dove: Art, politics, and peace. *The Peace Review, 26*(4), 564–572.

Galtung, J., & Fischer, D. (2013). *Johan Galtung: Pioneer of peace research* (Vol. 5). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-32481-9

Moussa, A. (2022). *Graphics buffer in p5.* Retrieved from https://www.gorillasun.de/blog/the-p5-graphics-buffer/

Purple Pie Studios. (2021, January 12). *Bird animation tutorial in After Effects* [Video]. YouTube. https://www.youtube.com/watch?v=z9zpV8E2jWE

WEBGL coordinates and transformations. (2024). Retrieved from https://p5js.org/tutorials/coordinates-and-transformations/

Waga, N.-O. (n.d.). How Picasso's 'dove of peace' became a worldwide symbol of hope and unity. *Forbes.* Retrieved October 18, 2024, from https://www.forbes.com/sites/neloliviawaga/2023/10/15/how-picassos-dove-of-peace-became-a-worldwide-symbol-of-hope-and-unity/

7 Appendix A - Previous Iterations: