THE EDUCATION FUND

FOR EXCELLENCE IN MIAMI-DADE PUBLIC SCHOOLS

2022-2023

Ideas with IMPACT

Idea Packet Sponsored by:

FPL

Coding the Future with Art

Coding the Future With Art

STEAM Lesson

Anna Weiss

a_weiss@dadeschools.net

Highland Oaks Middle School

School code: 6241

(305)932-3810

(305)932-0676 FAX

For information concerning Ideas with Impact opportunities including Adapter and Disseminator grants, please contact:

Audrey Onyeike,
Program Director
Ideas with IMPACT
The Education Fund
305-558-4544, Ext. 113
Email: audrey@educationfund.org
www.educationfund.org

Table of Contents

- Goals and Objectives
- Course Outline and Overview
- Lesson plan
- Resource List
- Student Work Samples
- Adapter Application

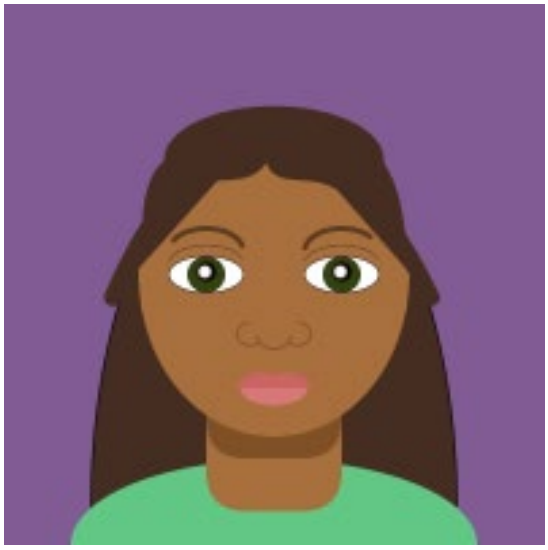
Goals and Objectives

The goal of the **Coding the Future with Art** lesson is to help teachers to provide their students with an exciting way to understand relationship between math, computer coding, and art; use coding in a creative way; foster self-esteem; introduce possible careers in coding as it relates to the arts; and to provide a new venue for self-expression.

The objective for this lesson is to help teachers set up Khan Academy account, select a subject, create a new class, add students, and assign them lessons from Computer Programming. Then teachers will explore all parts of the lesson from preparation to submission of work to a Code/Art Contest. Finally, teachers will be able to create their own practice portrait by following a video created by Code/Art.

Course Outline and Overview

- Introduce Code/Art
- Explain lesson objectives
- Talk about preparation for the lesson and required documents:
    1. Sample email
    2. Student lesson handout
    3. Tips for success
    4. Key vocabulary
    5. Self-portrait coordinate sheet
    6. Code/Art Miami Project Submission instructions
- Create teacher's account
- Create students' accounts
- Using a how-to video, guide teachers in creating a practice portrait



Code/Art Coordinate Sheet

Name:_____ **Date:**_____

Name of the Project

_____

(0,0) (200,0) (400,0)

(0,200) (400, 200)

(0,400) (200,400) (400,400)

Class 1 (85 minutes) – Introduction and creating accounts

Students will be introduced to the concept of Code/Art, its creation and future application. Students will be introduced to JavaScript by watching a presentation on a Promethean board. Students will see the work of other students over the past few years. Then they will get their account usernames and passwords. Finally, they will take brief notes on how to start a program from how-to video.

Classes 2-3 (3 hours) – How-to video

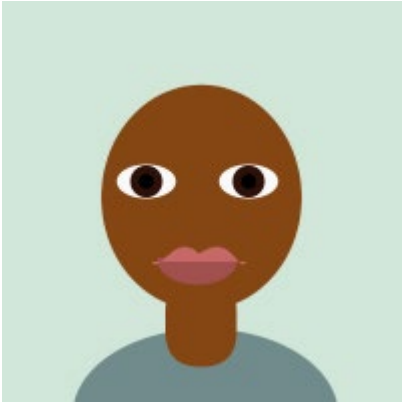
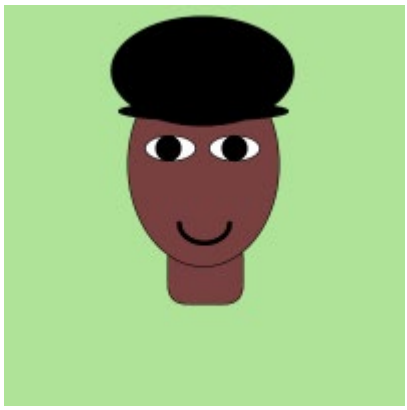
Students will watch a how-to video on making a practice portrait slowly with pauses so that they can code their practice portraits. The video will include explanation of the grid set up, basic functions, and how to explore each function individually and separately from the portrait they are coding. There will be new vocabulary and concepts. Teachers should make a parallel between the grid in JavaScript and the grid used in math classes. Students are encouraged to find similarities and differences that will help them understand both grids better.

Class 4 (85 minutes) – Sketch for self-portrait on Coordinate sheet

Students will draw and color their self-portraits while looking at the mirrors on a coordinate sheet. It is a grid similar to that of canvas in JavaScript. The edges are labeled with appropriate coordinates (0,0), (400,0), (0,400), and (400,400). That way when students start coding, they will be able to read the numbers on the sketch and be more precise with coordinates when coding.

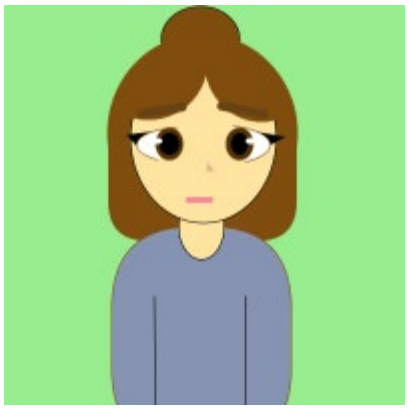
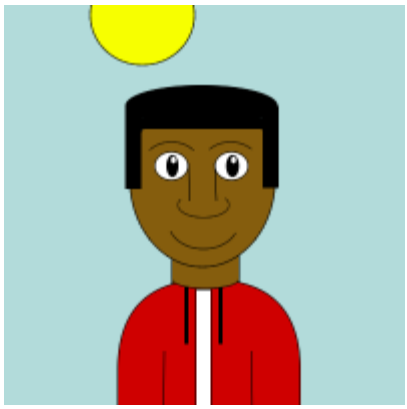
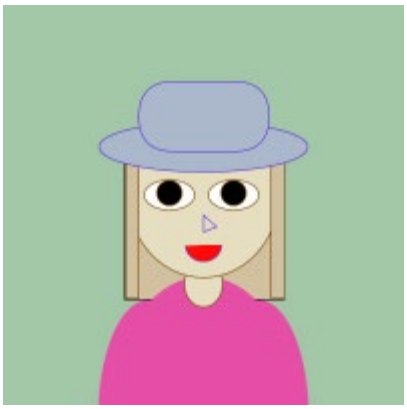
Class 5 (85 minutes) – Artwork in progress

Students will code a background color and their face or “sketch” the composition with coding.

Classes 6-7 (3 hours) – Artwork in progress

Students will add clothing and hair.

Classes 8-9 (3 hours) – Complete Artwork

Students will add background and any necessary details to complete their projects.

Class 10 (85 minutes) – Critique and reflection

Students will participate in class critique and write self-evaluations.

Visual Arts Lesson Plan

Grade: 6-8 Time: 8 hours

<u>Lesson</u>: Coding the Future with Art

<u>Standards:</u>

V.A.68.F.3: The 21st- century skills necessary for success as citizens, workers, and leaders in a global economy are embedded in the study of the arts.

V.A.68.F.1: Creating, interpreting, and responding in the arts stimulates the imagination and encourages innovation and creative risk-taking.

- V.A.68.F.1.4: Use technology skills to create an imaginative and unique work of art.

CTE-IT.68.PROG.02.09: Write a simple program

CTE-IT.68.PROG.02.10: Edit, compile, and run a program

CTE-TECED.68.ENTECH.05.02: Explain how knowledge gained from other fields of study interact with technology

CTE-AATC.68.VISUAL.01: Demonstrate an understanding of the visual arts career pathways

MAFS.7.G.1: Draw, construct, and describe geometric figures and describe relationship between them.

<u>Materials</u>:

Laptops, Student lesson handout, Tips for Success, Self-Portrait Coordinate sheet, pencils, erasers, color pencils.

<u>Activities</u>:

Students are introduced to JavaScript program in Khan Academy. Then they will watch a how-to video with frequent pauses and explanations, to code a practice portrait. After that students will draw and color their portraits on a grid. Then students will code a self-portrait using a drawing sketch to help with the placement of coordinates and choosing the correct shapes and functions. When

the project is completed, students will be able to see everyone's portrait and code on Promethean board and comment.

Vocabulary:

Coding, computer program, comments, coordinates, bug, debugging code, function, JavaScript, Khan Academy, loop, modular, parameters, processing.JS, syntax, variable

ESOL Strategies:

VC –Vocabulary in context

WGIR – Whole group to individual response

VAKT – Visual, Auditory, Kinesthetic, tactual

Performance Assessment/Evaluation

- Observation of process

Students get separate grades for notes, self-portrait sketch on a grid, work in progress, completed self-portrait, and written self-evaluation

- Group critique
- Observation of final product
- Self-assessment by student
- Homework:

Students will complete lessons in Introduction to J.S.: Drawing and Animation. The lessons include: Intro to Programming, Drawing Basics, Coloring, Variables, and Bonus: Resizing with Variables.

Integrated Curriculum

STEAM: Science, Technology, Engineering, Art, Math; vocabulary

Concepts/Skills

Art and Technology: using computers as a medium for self-expression, Math: finding coordinates on a grid, understanding basic geometrid shapes Science: science of technology is used to create artwork, Engineering: code can be modified to better the artwork.

Last Name: _____ First Name: _____Grade:__Period: _

Self-Assessment

Project: Code Art Self-Portrait with Superpower

How involved was I with planning and preparation of my project?
Circle the most fitting number of possible points per each questions.

QUESTIONS:

1.	0	1	2			Matched the shape of my face and neck
2.	0	1	2	3		Created realistic skin tone, shadows, and highlights
3.	0	1	2	3	4	Eyes have correct shape, iris, pupil, and highlight
4.	0	1	2	3	4	Created successful eyelashes and eyelids
5.	0	1	2			Created a successful nose that goes well with my face
6.	0	1	2	3		Created a successful mouth that looks like mine
7.	0	1	2	3	4	Hair is creative, and matches my color, texture, an style
8.	0	1	2			Eyebrows are successful, and match the hair color
9.	0	1	2			Ears are made successfully
10.	0	1	2	3		Background is creative, and matches the theme
11.	0	1	2	3		Clothing and accessories are creative and match the theme
12.	0	1	2	3	4	Completed artwork in a timely manner

Reflection Questions:

Please answer the following questions using complete sentences:

1. What super power did you choose? **Why**? _____
_____
_____
2. What is the most successful part about your project? **Why?** _____
_____
_____
3. What **3 Coding concepts** did you learn about during this project? _____
_____
_____
4. What was the ***most challenging*** part about this artwork? **How did you overcome it**?
_____
_____
5. What was the ***least challenging*** part about this artwork? **Why**? _____
_____
_____
6. What would you change about your artwork? **Why/how**_____
_____
_____

Resource List

Websites:

www.khanacademy.org

https://code-art.com

https://code-art.com/competition

https://hub.code-art.com/teacher/

https://view.publitas.com/code-art/codeyourself2021-book-6-22-2021

https://www.khanacademy.org/computer-programming/top-self-portraits

Videos:

https://hub.code-art.com/courses/intro-lessons/lessons/video-self-portrait/

Important Info for JavaScript Coding

- www.khanacademy.org

Canvas is 400 pixles wide (X axis) by 400 pixles tall (Y axis)

- **The grid starts on the top left corner of the canvas with point (0,0), and ends at the bottom right with the point (400,400).**

- **Function is the short cut for many lines of code.**

- **Every command has to have ; at the end for computer to read it.**

- **Click ENTER to type the next line of code**

- **In any command that has more than one word, the following words are written without space, but the first letter of every word is capitalized.**

- **To write a note to yourself about the code, start with // , that way the computer will not read it.** The note will appear in green.

- **background(R,G,B); is the color on the back of the artwork, there R is red, G is green, and B is blue. The numbers are from 0 to 255, with (0,0,0) being black, and (255,255,255) being white.**

- **fill(R,G,B);** is used to change the color. The default color is **red**, and appears as (255,0,0). This command colors all the shapes following it. To change the color, type the fill command again and choose a different color. The first shape created is appears white with black outline. The fill command must be on top of the shape that needs to be colored in.

- Adding a **4th** number changes the transparency of the color.

- **rectMode(CENTER);**

This command lets the starting point of the rectangles and squares start in the middle of that shape. If you do not write this command, the rectangles and squares will start from the top left corner.
It needs to be written only once as the first line of the code followed by the background.

- **arc(x,y,w,h,start,stop);**

This command is used to make ant part of the circle. The start and the stop are degrees of the circle. The measurement starts at 0 degrees and moves clockwise.

- **noFill();**

Use this command to create any shape with an outline without being filled with color.

- **noStroke();**

This command is used to draw a shape with color inside but without the outline.

- **stroke(number);**

This command is used to make the lines thicker. The higher the number, the thicker the outline.

- **ellipse(x,y,w,h);**

This command creates ovals and circles. The X and Y values represent the middle of the circle on the grid.

- **Scrubbing tool** on top of the numbers lets you change them easily. Pressing on the arrows moves the number one at a time. To move the numbers quickly, move the middle arrow right to increase the number, and left to decrease it.

- **rect(x,y,w,h);** command creates squares and rectangles.

- Adding a **5th** number after h, will affect the shape of the corners. Larger number will curve the corners more, smaller – less. The numbers range from 0 to 100. At the number 100, the rectangle turns into a circle.

- To find more information about each individual Function, click on **Documentation** at the bottom center of the screen. From there, click on any Function to practice using it on a separate screen. The practice can be saved or discarded.

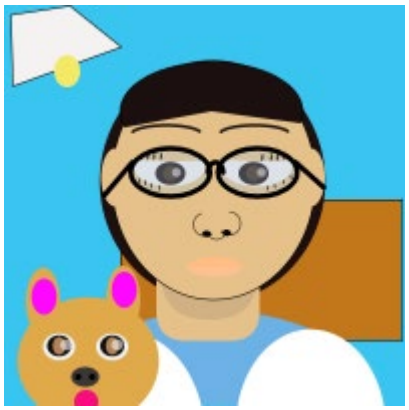
- The lines of code work like stickers on the paper. The last commands that are lower on the code page cover the other ones. The order of the code lines may be changed any time.

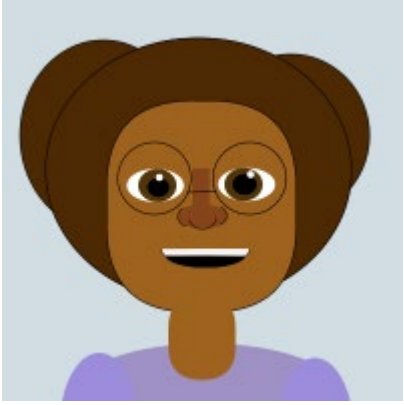
- Most important – **ALWAYS SAVE YOUR WORK!**

Very small sample of Code:

```
rectMode(CENTER);

background(0, 171, 201);

noStroke();

fill(255, 160, 43);  //orange

ellipse(41,50,82,88); //sun

fill(255, 255, 255);  //white

ellipse(87,77,180,52);  //cloud…

… fill(5, 0, 71);  //navy blue

arc(202,400,354,231,180,360);  //shirt

fill(133, 55, 0);  //skintone

rect(198,301,98,110,40);  //neck

fill(102, 50, 20);  //skintone shadow

rect(200,274,97,72,60);  //neck shadow

fill(133, 55, 0);  //skintone …
```

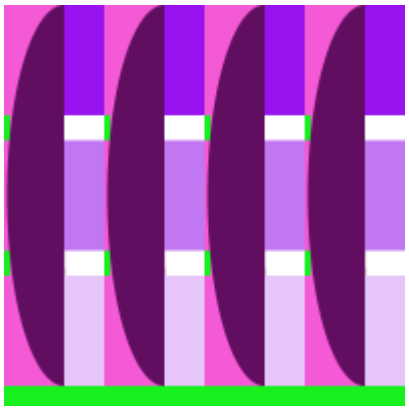
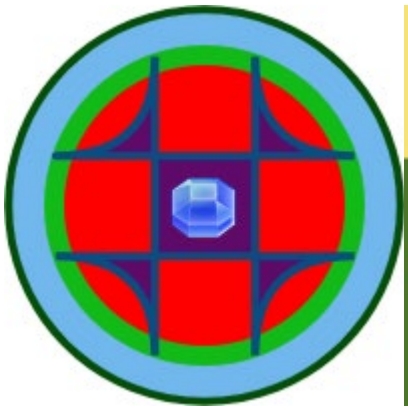
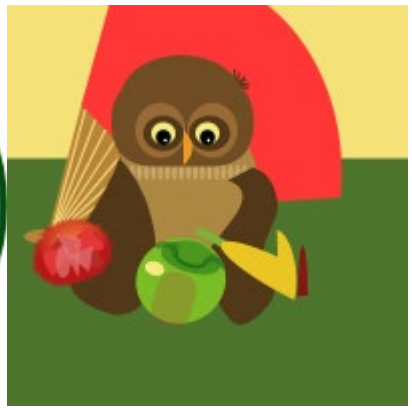
Student Work Samples

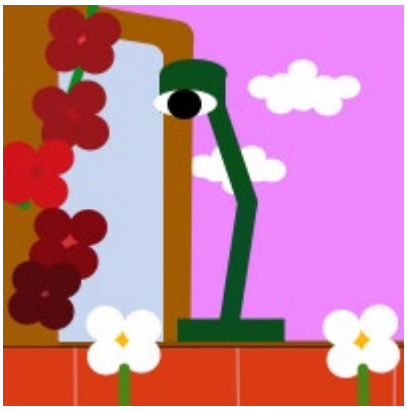




ADAPTER APPLICATION

Students can create other art with code, such as abstract, landscape, still-life, etc. Older students may also experiment more with variables and Bezier curves.



Older students may also experiment more with variables and Bezier curves. They can code a portrait in profile, create shadows, moving images, etc.
Younger students will use simpler functions and not that many lines of code.