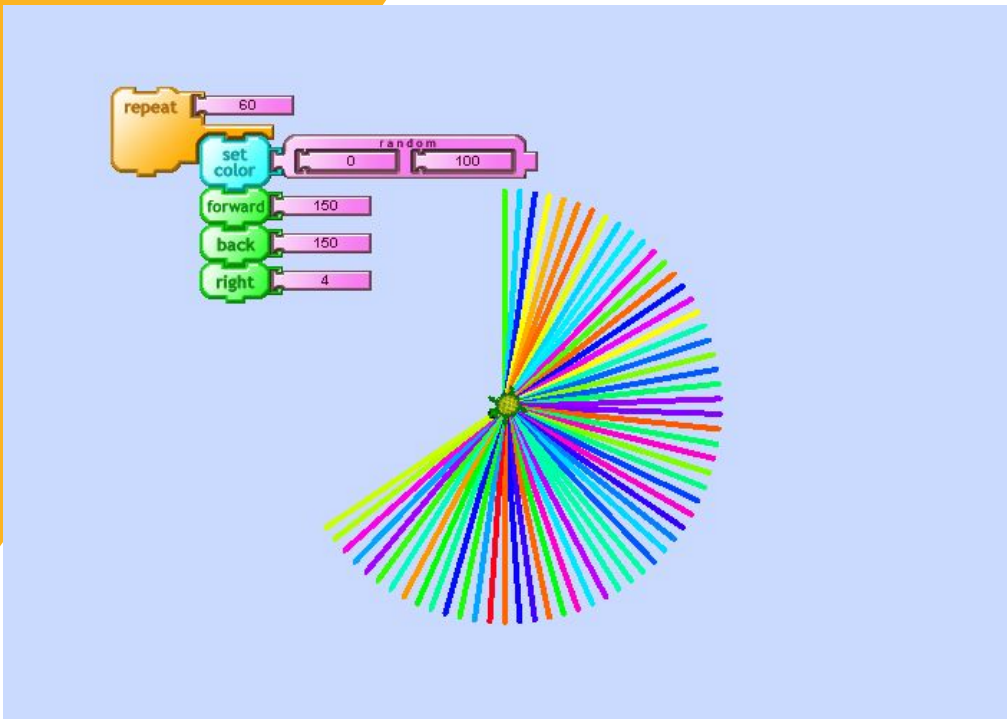
Lighthouse Community Public Schools
444 Hegenberger Road
Oakland, CA 94621

creativitylab@lighthousecharter.org
LighthouseCreativityLab.org

TURTLE ART

A Guide from the Creativity Lab



Authors
Aaron Vanderwerff
Laura Kretschmar
Miriam Leshin

Contributor
Anna Milada Grossi

About This Project

Turtle Art is free software that lets users design and draw using simple programming commands. Both an exercise in coding and an outlet for creative expression, Turtle Art challenges students to think critically, while giving them room for play.

By itself, Turtle Art serves as a great introduction to programming, which students can later build upon with more complicated programming languages.

You needn't have a mastery of Turtle Art in order to begin this unit. Once you have a basic understanding, feel free to have your students explore on their own, and discover areas within the program that even you may not be familiar with. This helps students to develop agency.

Our Story

At Lighthouse, we also use Turtle Art to help students think critically about concepts like area, perimeter, angles, and the Cartesian coordinate system. The lessons that follow are intended to span several weeks.

This project is designed to give students ownership of their learning by offering minimal instruction. Students learn through the iterative process of design, test, and redesign. This is a crucial element of this project, and should not be viewed as supplemental. We encourage you to customize this project to the needs of your class, but maintain the ideal of students learning through exploring.

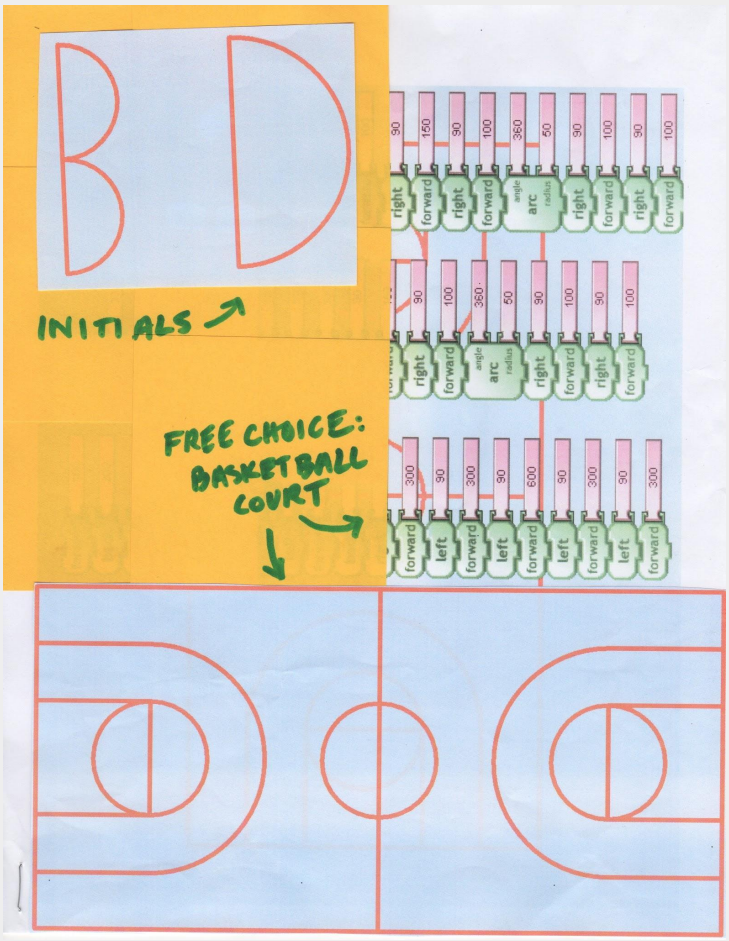
Materials & Tools

TOOLS

- Macs/PCs
- Turtle Art1
- Projector (Optional)

Learning Targets

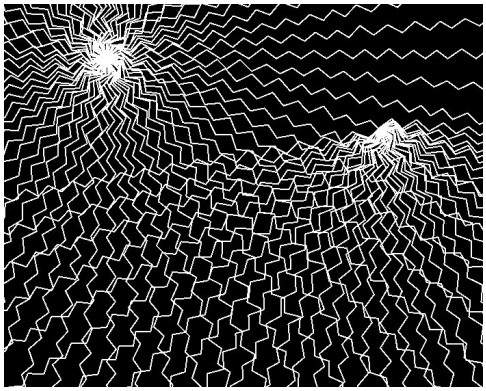
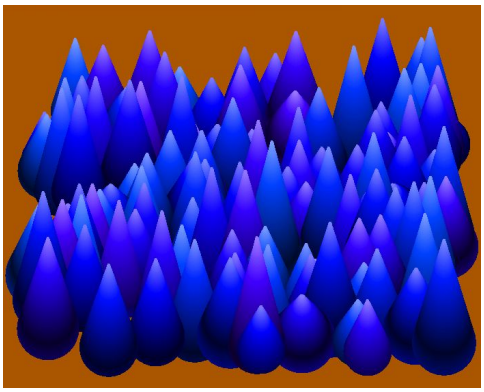
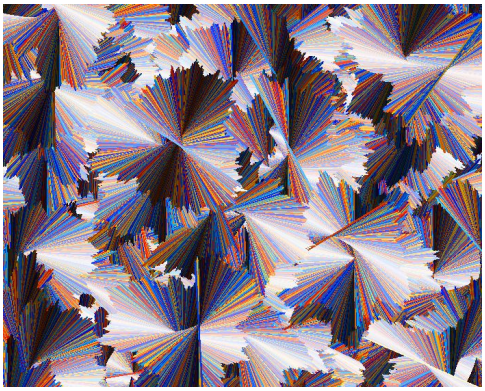
- I can explore complexity.
- I can imagine next steps in making a piece.
- I can reach beyond my current capacities through playful exploration.



Context: Before we make...

Show students the complex designs created with Turtle Art on the Turtle Art image gallery (located at turtleart.org) Reinforce to your students that the images created were done so only using Turtle Art. Have students explore the remaining blocks on their own, in hopes of achieving complexity in their designs, with features like changing pen colors, and line widths. As students make discoveries, write them on the board for everyone to see.

Gallery TurtleArt.org
Getting Started With Turtle Art
Turtle Geometry
Coding For Art
Art Made With Coding
From Art to Science

Material Management

- Students can work in pairs. Use this as a chance to develop collaboration skills. Students should take turns being at the computer and recording data in a notebook.
- Students should turn to each other for help, before turning to their teacher. Establish this early on as a norm.
- Show students the Question Mark tool, for help with navigating certain aspects and tools of Turtle Art.

Standards Assessed in Turtle Art

- I can use Turtle Art to draw basic shapes.
- I can incorporate subroutines into a program.
- I can draw shapes of specific area and perimeter.
- I can draw shapes at specific points.
- I can add complexity to my programming, through the use of variables.

Warm Up

Warm students up with a series of area and perimeter problems, in their notebooks or using worksheets. Then, give them a series of challenges using Turtle Art:

- Draw a polygon with an area of 100.
- Draw a polygon with an area of 200.
- Draw two polygons with the same area, but different perimeters.
- Draw two polygons with the same perimeter, but different areas.
- Draw both the biggest and smallest shapes you can.

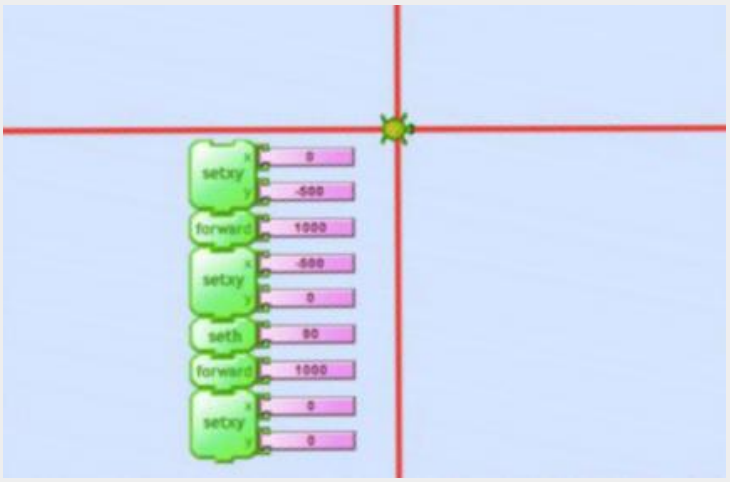
Cartesian Coordinates

Ask students how they can repeats shapes at different parts of the screen. They will frequently say you can click and drag the turtle, to move it. Demonstrate this method, then ask students to find alternative ways, using only blocks. They will usually identify the *pu/pd* block, *seth*, and *setxy*. After initial exploration and whole class sharing, focus your class on *setxy*.

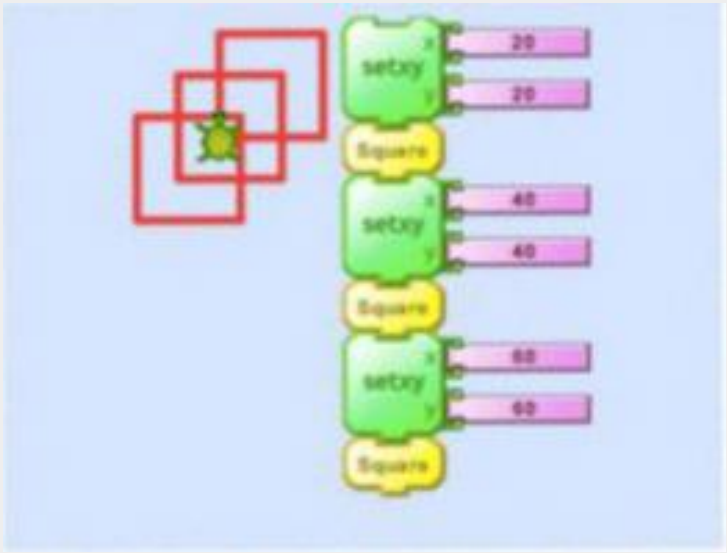
Using *setxy*, challenge students to draw shapes in each corner of the screen; in a row; and in a column. What patterns do students notice when they change the turtle's x and y values? They should notice that changing these values changes the turtle's horizontal and vertical positions, respectively. Draw a series of shapes on a whiteboard, then project Turtle Art over them. As a class, have students work together to set the turtle to the correct coordinates, and correctly (and efficiently) create the codes to perfectly trace the shapes.

The Origin

Project Turtle Art onto a whiteboard. Challenge students to write a program that will draw x and y axes. Give students ten minutes to determine what x and y values must be input to bring the turtle back to the center of the screen, using their own computers. Once they discover the answer is 0,0, define the center as the origin.
.



The Origin



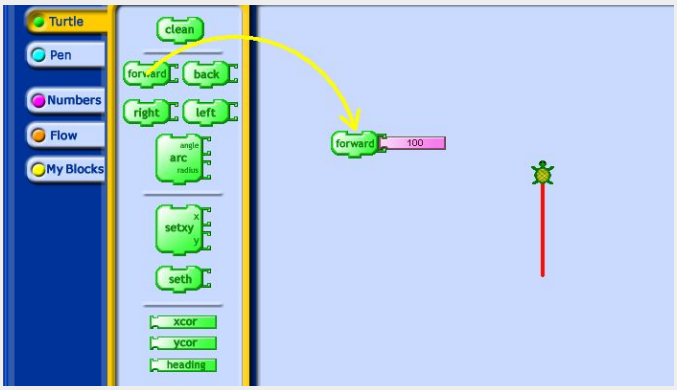
Cartesian Coordinates

Step-By-Step Guide

To draw a basic shape:

1. Select a Category
2. Click and drag a command block from the sidebar to the center of the screen.
3. Double-click a command block, to make the turtle perform that action.
4. Certain blocks, like *forward* and *right*, let you input a numerical value, to control what distance the turtle will move, or how many degrees it will rotate.
5. Link several blocks together to create a stack. If the first block in a stack is double-clicked, the turtle will perform all of the commands, in sequence.

Select a Category

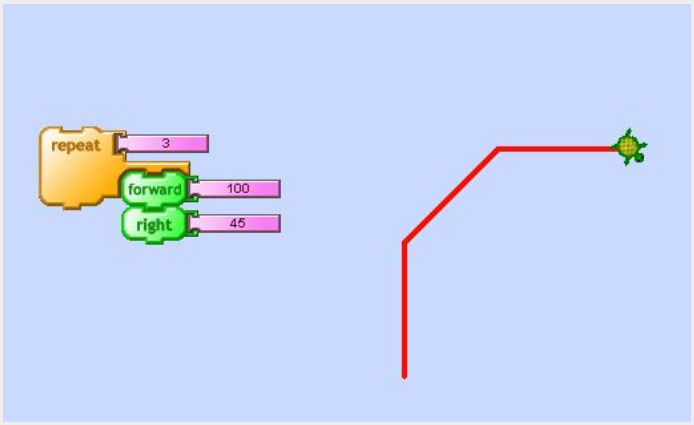


Click and drag a command

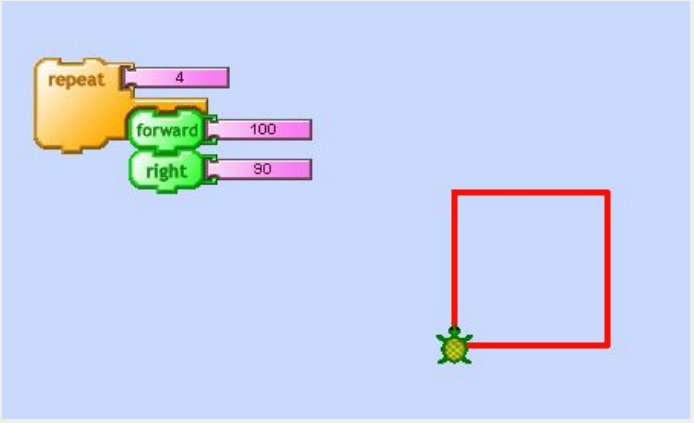
Notes

Show students how they can click and drag blocks to the center of the screen, and how double-clicking a command causes the turtle to perform the action. With only this basic demonstration, challenge students to make their turtle draw a square, using only the *clean*, *forward*, *back*, *left*, and *right* blocks. Give them roughly twenty minutes to work. If they finish early, challenge them to draw squares of different sizes or explore other blocks.

Have students begin experimenting with the *repeat* block. How do they think they can use it? Challenge them to draw squares using only three commands in total, incorporating *repeat*. As students show comprehension, introduce my blocks, the Turtle Art name for subroutines. Demonstrate how an entire stack of commands can be turned into a single block, to be incorporated into their larger program as needed. In the example to the right, we've created a square using the *repeat* command. You can top the stack with a my block and call it "*square*", creating a single block.



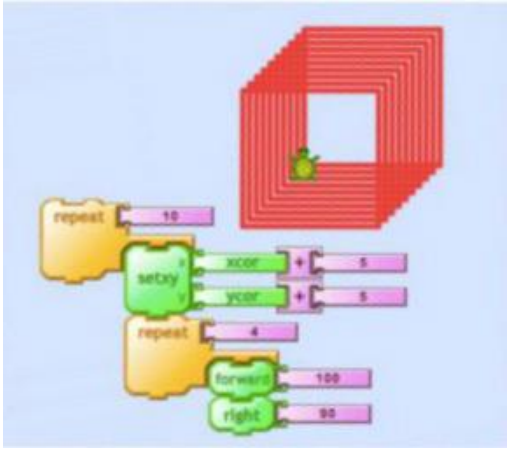
Repeat Block



Creating a square with repeat blocks

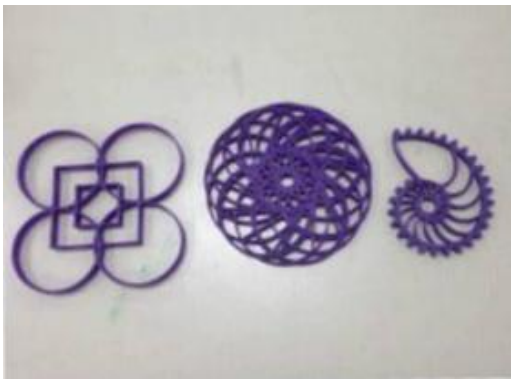
Turtle Art with Older Students

Introduce students to the + (plus) and – (minus) blocks. Are students able to figure out how the work? These commands add a level of complexity. Demonstrate a program that draws a shape in various locations, using these commands. Have students explain the commands the turtle is following. Blocks like *xcor* and *ycor* are Turtle Art variables.



To demonstrate their comprehension, students should plan a project that incorporates as many elements of Turtle Art as possible. They should strive for complexity, efficiency, and precision in their work. Possibilities for a project might include designing:

- Your ideal house, garden, or other space
- A sports team logo
- Your favorite animal



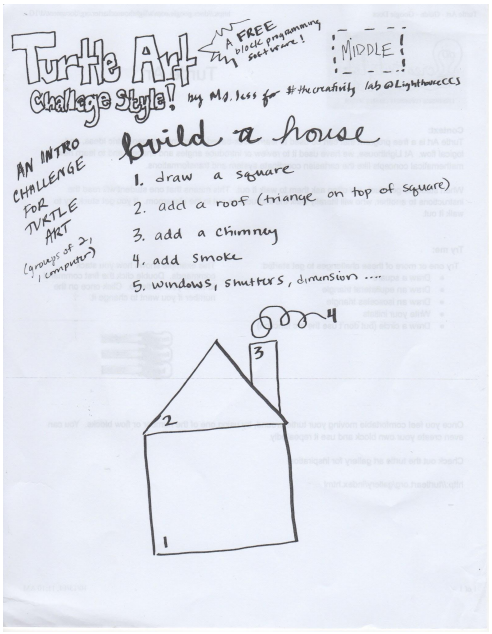
Students should draw/describe their proposed project with a broad idea of the programming involved, to be turned in to the teacher for approval before beginning. Once approved, give students several class periods to work, and encourage them to tinker and play. Allow them to change their design over time.

Students can also, when the equipment is available, Import Turtle Art designs to 3D printing software. Print them out and use them as stamps, or to make imprints in clay.

Looking Closely

Come back together as a group. Did any students manage to draw a square? How did they do that? Ask students to clarify what each command does:

Forward—The turtle moves towards its head. (Not "Up.")
Back—The turtle moves towards its tail. (Not "Down.")
Right—The turtle turns right. (It does not move to the right.)
Left—The turtle turns left. (It does not move to the left.)
Clean—All previous drawings are erased.



Ask what happens when the numerical values are changed. Have students experiment. With *forward* and *back*, the numbers represent distance travelled. With *right* and *left*, the numbers represent degrees rotated. What number must be entered to make the turtle turn completely "sideways?" (horizontal.) Guide students to discovering that the answer is 90. With these clarifications, challenge students to continue working. If they have already tackled drawing a square, challenge them to draw squares of different sizes, as well as other shapes.