Algorithmic & Computational Thinking In Design

By Nettrice R. Gaskins, PhD

Definition

An **algorithm** is a well-defined procedure that takes some value, or set of values, as input and produces some value, or set of values, as output. An algorithm is thus a sequence of steps that transform the input into the output.

Algorithmic thinking is a way of getting to a solution through the clear definition of the steps needed. Rather than coming up with a single answer to a problem, students can create algorithms for many solutions.

Computational thinking is a set of problem-solving methods that involve expressing problems and their solutions in ways that a computer could execute.

Introduction

Dutch artist **MC Escher** used an algorithm (script) to create this image of butterflies:

1. Draw a butterfly.
2. Color the butterfly's wings in such a way that mutually orthogonal circles are formed.
3. Arrange the butterflies in a group or pattern, starting with bigger objects that get smaller and smaller as they approach the center.
4. Rotate each row of butterflies to fill the space.

Image on the left: "Circle Limit with Butterflies" by MC Escher

Introduction

The steps to create Escher's art constitute an algorithm. They are not a computer program—they are just specific directions on how to achieve something.

In order to simulate this art on a computer you will need a programming language that generates patterns as data structures and each butterfly (shape) as an object.

Introduction

With Escher’s algorithmic thinking process in mind we can look at many different real life patterns (data structures) and identify algorithms that can be used by a computer to generate shapes and patterns.

Images counter-clockwise from top left: Mapuche textile print, Mardi Gras Indian patches, Native American star quilt, Kuba cloth.

General Terms

Program

Variables

Data & Math

Modify & Control

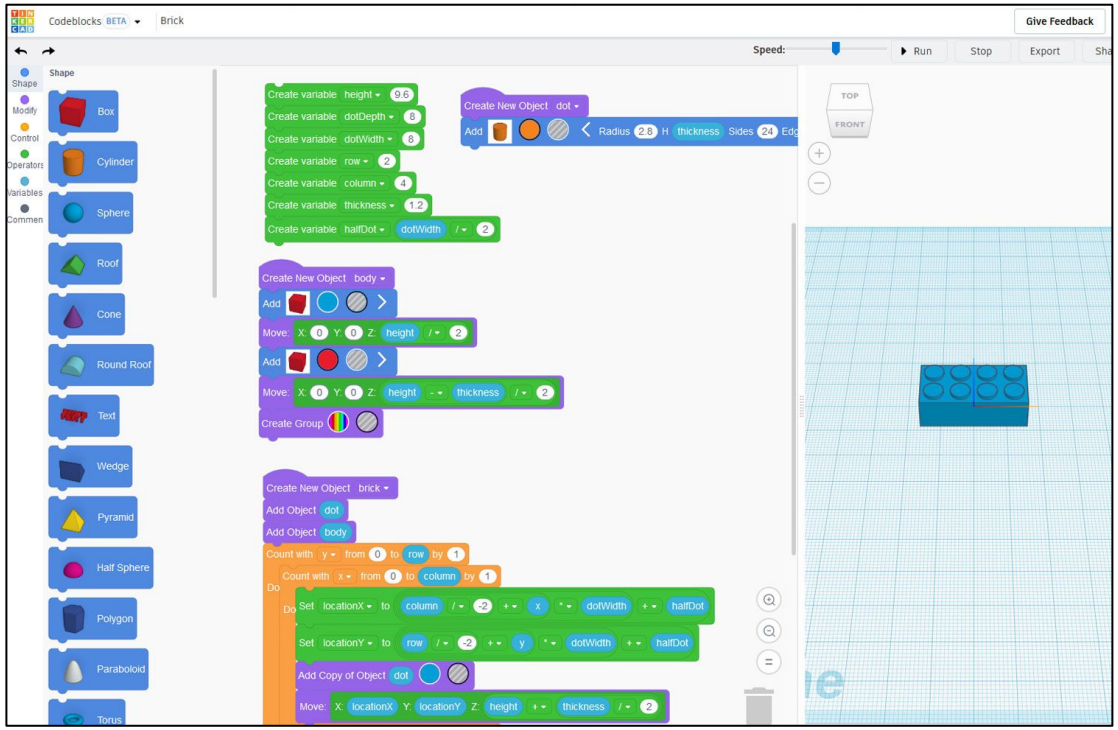
We can think of computational thinking as baking a cake. The finished cake is the program that contains different components (i.e. blocks).

The Program

In computing, a program is a specific set of ordered operations for a computer to perform or follow, to make something.

People create programs using **programming languages**.

The Program



Codeblocks uses a visual programming language to create 3D designs that can be 3D printed.

The program generates a 3D object.

Variables

Variables are components that perform calculations or execute instructions.

In visual programming, **variable blocks** are used to hold values and strings in variables… like bowls or cups are used to hold ingredients.

Variables

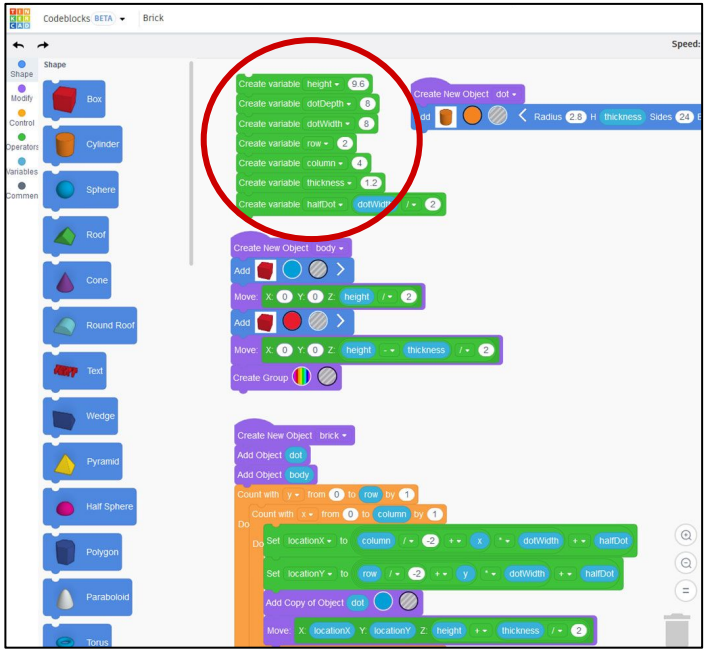
Variables can be created and used in other blocks.

This is like using the same measuring cup for different tasks.

> create variable “cup”

> indicate how many cups or what kind of cup

> use this variable (in script) when needed

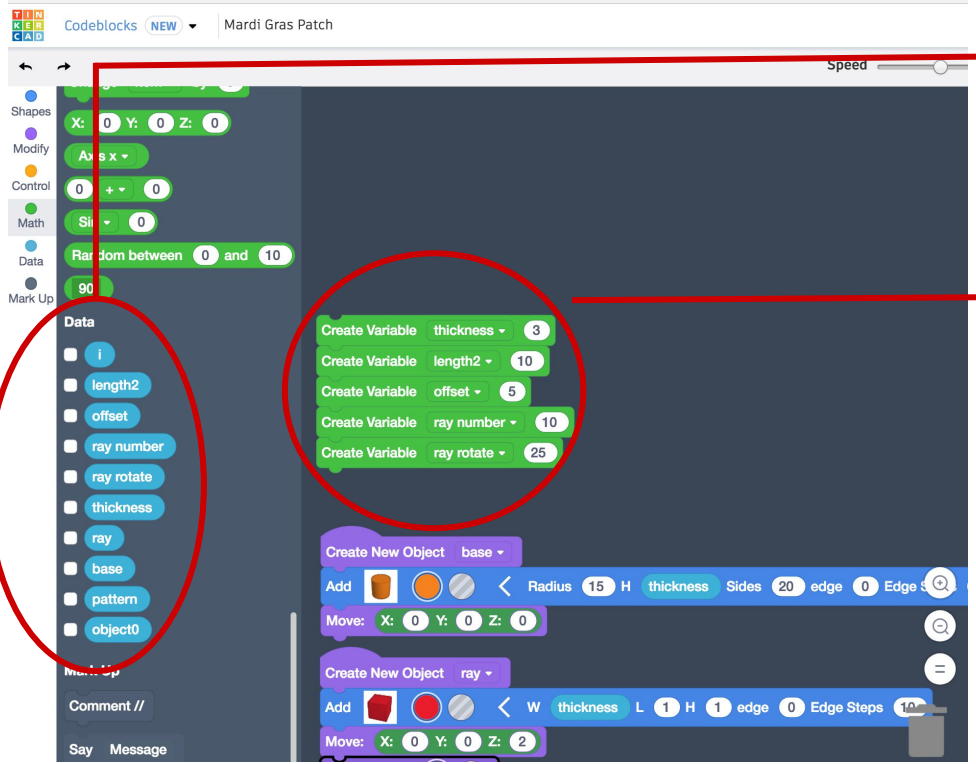


Data & Math

Data lists all the variables created in the program (like the ingredients list for a cake).

Math defines variables, changes their values, and generates random numbers.

Data & Math



Data (blue) lists all the variables created in the program.

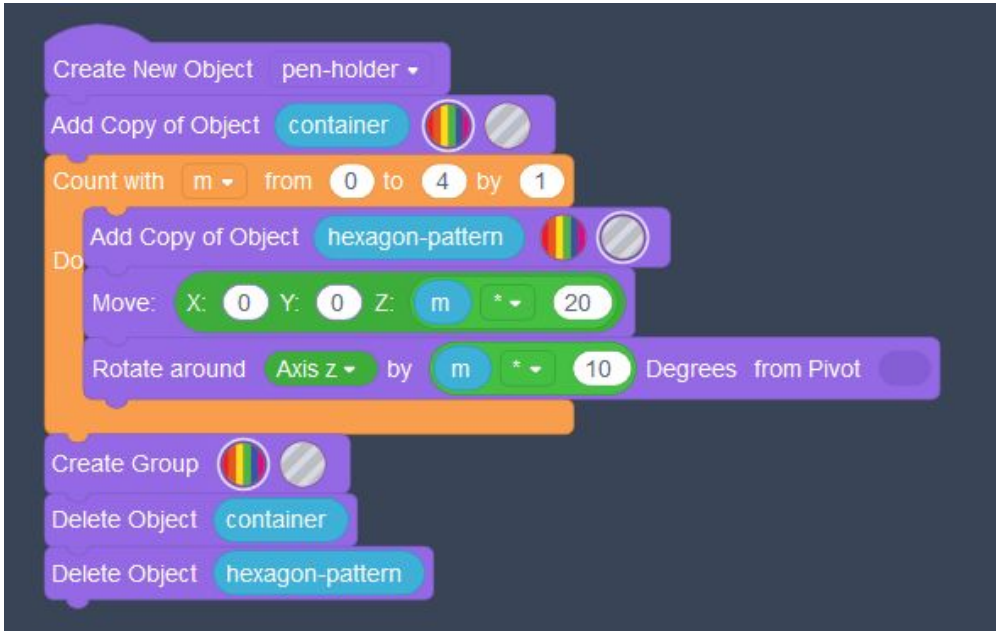
Math (green) defines variables, changes their values, and generates random numbers.

Modify & Control

Modify allows you to transform or change something in a recipe or script.

Controls help you create a more efficient recipe (script).

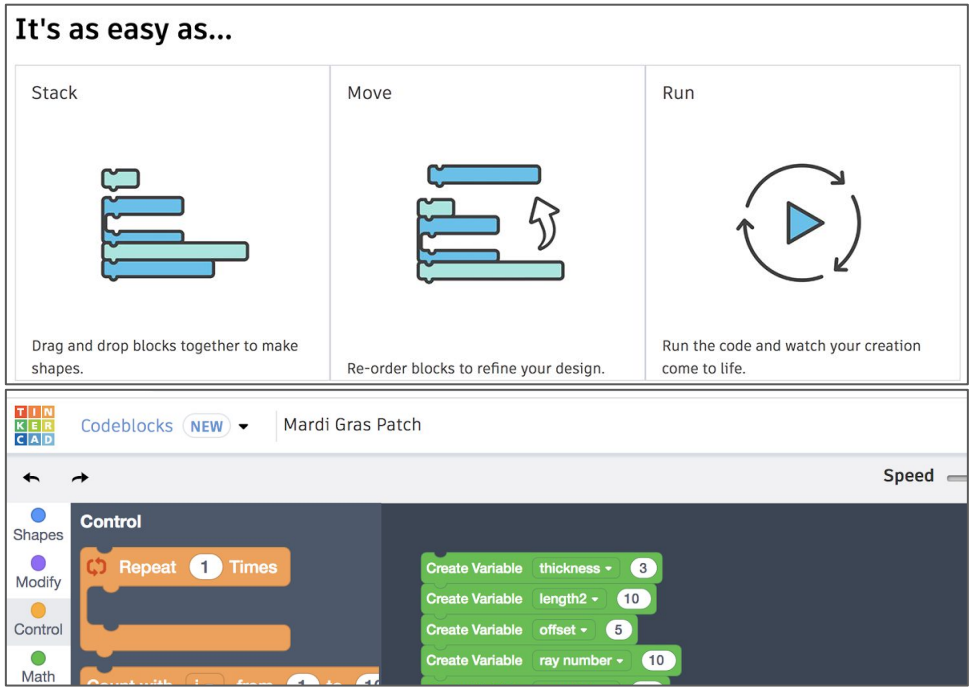
Modify & Control



Modify blocks (purple) allow you to change the attributes of existing shapes, like their size, color, position, and rotation.

Control blocks (orange) let you define loops to repeat actions.

Tinkercad Codeblocks

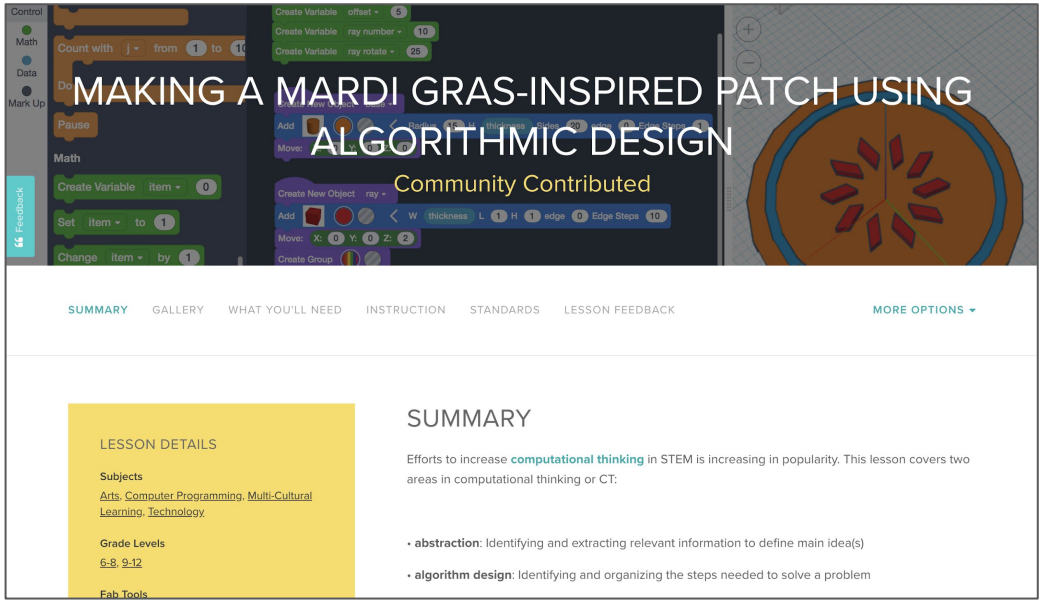


Tinkercad Codeblocks procedurally generates 3D shapes using a visual programming language or VPL.

The resulting model can be exported and 3D printed.

Tutorial: https://maker.pro/custom/tutorial/an-introduction-to-tinkercad-codeblocks

SCOPES-DF Lesson



This lesson covers two areas in computational thinking or CT:

- **abstraction**
- **algorithmic design**

Students use algorithmic and computational thinking, with Codeblocks to generate 3D designs.

Lesson Plan: https://www.scopesdf.org