Pencil Code Program: Chaos game

*Please refer to the online lesson plan on the DT Hub to access all website links and additional resources.*

# DT + Mathematics

# Years 7-8

The ‘chaos game’ is a way to see how patterns can result from certain random events.

Use this program to run the ‘chaos game’, randomly moving the turtle to create a pattern (for more information, search 'chaos game’).

Have students analyse or fill in or change parts of the pencil code program. This program could be used to further your understanding of how you could use Pencil Code in the classroom, as a demonstration or discussion with your students, or as a way to introduce various CT concepts (see online guide)**,** such as pattern recognition or abstraction, to your students by inviting them to extend the existing functionality of the program.

## Learning hook

Roll out a bag of marbles onto the floor. Look at the random pattern made. If the bag of marbles was released would the same pattern be made? Ask students to explain.

Discuss random other events and randomisation with relevant examples.

Ask students, ‘Would you expect to see patterns with random events?’ Why or why not?

Introduce the ‘chaos game’ as a way to see how patterns can result from certain random events.

Use this program to run the ‘chaos game’, randomly moving the turtle to create a pattern (for more information, search 'chaos game’).

## Learning demonstration

Provide students with a link to pencilcode.net and the pencil code program. Run the program and ask students to describe their observations. Ask the following questions:

* What are the parts of the program? What results onscreen?
* What is the program doing?
* At what point do you see a pattern emerging?

## Learning construction

Have students analyze or fill in or change parts of the pencil code program. This program could be used to further your understanding of how you could use Pencil Code in the classroom, as a demonstration or discussion with your students, or as a way to introduce various CT concepts, such as pattern recognition or abstraction, to your students by inviting them to extend the existing functionality of the program.

### ***Ways to remix the program***

1. Change fd value from 200 to another number for each sprite. What happens when you don’t use the same value for each of the three dots (sprites)?
2. Change the speed from 1000 to another value
3. Change the range from 1..2000 to another value eg. 1..50000
4. Change the value of the factor multiplying the fd distance(p), from 0.5 to another value eg try 0.75 for starters

## Computational Thinking Concepts\*

|  |  |
| --- | --- |
| **Concept** | **Definition** |
| **Abstraction** | Identifying and extracting relevant information to define main idea(s) |
| **Pattern Recognition** | Observing patterns, trends, and regularities in data |

*\* Explore the Computational Thinking Concepts Guide on the DT Hub for a list of the CT concepts noted on ECT, including tips for implementing each concept in your classroom*

# Pencil Code Program

Copy/Paste the following program into a ‘Blank Editor’ on the Pencil Code website (new.pencilcode.net)

|  |
| --- |
| # Copyright 2015 Google Inc. All Rights Reserved.    # Licensed under the Apache License, Version 2.0 (the "License");  # you may not use this file except in compliance with the License.  # You may obtain a copy of the License at    # http://www.apache.org/licenses/LICENSE-2.0    # Unless required by applicable law or agreed to in writing, software  # distributed under the License is distributed on an "AS IS" BASIS,  # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  # See the License for the specific language governing permissions and  # limitations under the License.  v = [  new Sprite('red dot').fd(200)  new Sprite('blue dot').rt(120).fd(200)  new Sprite('green dot').lt(120).fd(200)  ]  speed 1000  for [1..2000]  p = random v  turnto p  fd distance(p) \* 0.5  dot black, 2  await done defer() |

# 

# Sample Output

# Screen Shot 2015-06-09 at 12.01.58 PM.png

**Additional Resource Links**

* Visit http://pencilcode.net/ to explore the Pencil Code development environment
* See Pencil Code: A Programming Primer for more than 100 example programs written in CoffeeScript

| **Links with Digital Technologies Curriculum Area** | | |
| --- | --- | --- |
| **Strand** | **Year** | **Content Description** |
| **Processes and Production Skills** | **7-8** | Design algorithms involving nested control structures and represent them using flowcharts and pseudocode (AC9TDI8P05)  Trace algorithms to predict output for a given input and to identify errors (AC9TDI8P06)  Implement, modify and debug programs involving control structures and functions in a general-purpose programming language (AC9TDI8P09) |

Keywords; Computational thinking, Programming, Logo, Turtle, Random, Patterns, Pattern recognition, Text programming, Algorithms, Geometry

## Administrative Details

|  |  |
| --- | --- |
| **Contact info** | For more info about Exploring Computational Thinking (ECT), visit the ECT website (g.co/exploringCT) |
| **Credits** | Developed by the Exploring Computational Thinking team at Google and reviewed by K-12 educators from around the world. |
| **Last updated on** | 28/01/2025 |
| **Copyright info** | Except as otherwisenoted, the content of this document is licensed under theCreative Commons Attribution 4.0 International License, and code samples are licensed under theApache 2.0 License. |