DT Challenge Year 7/8 Blockly – Turtle

https://groklearning.com/course/aca-dt-78-py-turtle/

About this activity

Turtle graphics is a term in computer vector graphics, using a relative cursor (the "turtle") upon a Cartesian plane. Students learn the first commands, such as move and turn to change the position of the turtle on the screen. By combining these simple commands, students can draw simple geometric shapes, such as a square or a line.

Age

This course targets students in year 7 and 8, though it can also be used as an advanced course for students in earlier years, especially those who are ready to move beyond block based visual coding languages.

Language

Python — a general purpose text based programming language

Time

The course is designed to be completed in 20 hours of class time.

Key Concepts

Key Concept	Coverage
Abstraction	
Data: collection, representation, interpretation	Data representation as integer and string
Specification, algorithms, implementation	Simple Algorithms, loops
Digital Systems	
Interaction	Interaction (command line input/output)
Impact	

Objectives (Content Descriptions)

ACTDIK015	Examine how whole numbers are used to represent all data in digital systems
ACTDIP019	Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition)
ACTDIP020	Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input.

Funded by the Australian Government Department of Education and Training.

What are we learning? (Abstract)

At the conclusion of this lesson students will be able to:

- Write programs using a general purpose programming language
- Recognise that breaking down a problem into smaller steps (deposition) makes it easier to solve problems
- Debug algorithms
- Recognise that steps in algorithms need to accurate and precise.
- Recognise that problems can have multiple solutions
- Utilise for loops in programs
- Utilise user input
- Add colours to their graphics, using lines, fill and background
- Produce a range of different shapes and manipulate existing ones
- Use angles
- Define the term *algorithm*
- Define the term *decomposing*
- Define the term *branching*
- Define the term *iteration*

Module outline

The course consists of seven modules:

1. Introducing the Turtle
   This module introduces programming with the Turtle, a Logo style drawing program that students control in Blockly, a visual, block based coding environment.
2. Angles with the Turtle
   This module introduces students to drawing angles using the Turtle. The course mostly uses 90 degree and 60 degree angles, though some others are also introduced.
3. Looping with the Turtle
   This module introduces the concept of iteration: using loops to instruct the Turtle to do something a number of times. The ‘repeat’ block simplifies drawing patterns and shapes.
4. Add a Dash of Colour
   This module adds colours - filling shapes with colour, and also changing the line colour that the Turtle draws with.
5. Asking Questions
   This module introduces students to the concept of *user input* - allowing programs to react to information received from the user.
6. Making Decisions
   This module introduces decisions, and allows students to write programs that change their behaviour on the basis of information from the user.
7. Putting it all together
   This module is an extension module and allows students a chance to put everything they’ve learnt together.
8. Playground!
   The module includes a Turtle Playground where students can create and save their own drawings. The Turtle Playground is a good place for teachers to additionally set their own challenges for students.

Funded by the Australian Government Department of Education and Training.

Lesson Preparation

Requirements
In order to complete this lesson you need an internet connection and some pencils and graph paper for activity 1, and pencils and paper for activity 2.

Introduction— what is programming?

What does it mean to program a computer?
Programming a computer means giving it instructions - a sequence of steps - to follow. Another word for this is an algorithm.

New Vocabulary

Algorithm: A set of rules or step by step instructions to solve a problem or achieve an objective.
Decomposing: Breaking down a problem into parts to then deal with individually. Imagine making a cake but you only had the final picture. The recipe is the instructions that break the problem up so it is easier to follow.
Branching: A programming instruction that directs the computer to another part of the program based on the results of a decision or comparison
Iteration: Executing a section of code a number of times as part of the program

Discussion Questions

- How do we make a computer do exactly what we want?
- How can we input and output information with a computer?
- Where have we seen examples of computer graphics?

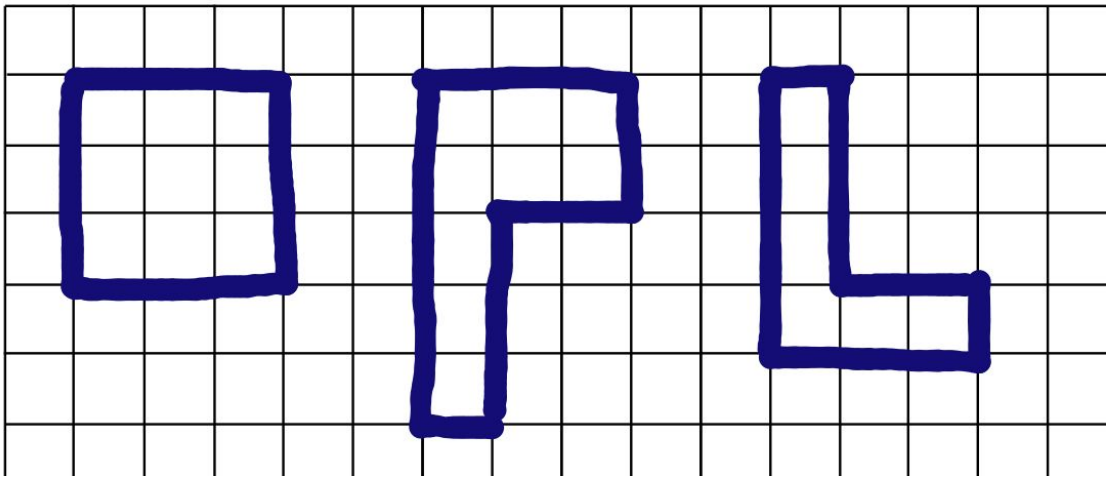
Funded by the Australian Government Department of Education and Training.

Activity 1 - Transfer Shapes

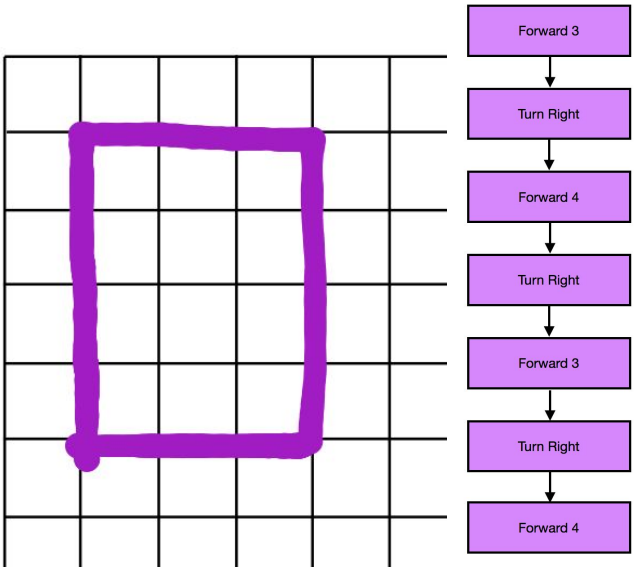
Activity Overview

- Ask the students to come up with a simple image on graph paper.
- The students then develop an *algorithm*
- They should then give their partner or someone in their group their algorithm.
- Then they compare with their partner to see how close to the original their new image is.

Ask the students to come up with a simple image using only right angles and lines on some graph paper. Advise the students to choose a simple shape such as a letter, or symbols like plus signs or squares. Also remind them that it must be possible to do without lifting their pencil up. The shapes should all start at the same spot, in the top left corner, and going horizontally unless their algorithm tells them to turn. Some examples:



The students should then develop an *algorithm* that will draw the image using directions such as “Draw the line forwards 5 squares” and “turn right”. At this point they will realise just how many steps is involved for one shape, and may want to make it more simple. One example of a shape and its algorithm;



The student should then give their algorithm to a partner to follow, and to attempt to follow their directions exactly and draw the shape, without the student showing them the original shape.

Once their partner has completed their instructions, they can compare with the original image to see how close their new image is.

Discussion questions:

- Were there more steps involved to make a simple shape than you first thought?
- Was it difficult to follow directions given to you?

Extension ideas:

- The shapes could contain multiple colours or even fill.
- Give the student an option of ‘pencil up’ and ‘pencil down’ so the shape doesn't have to be all connected.
- Let the students make a shape with 45° angles.

Activity 2 - Blind Pictures

Resources:

Participants will need two pieces of paper and a pencil or pen to draw an object.

Activity objectives

- Follow a simple sequence of steps algorithm carefully to solve a problem
- Experience how difficult it is to describe an algorithm unambiguously and succinctly
- Practice using lengths, angles, proportions and spatial language to describe a design
- Practice communicating and collaborating to solve a problem

Instructions

Discuss: In this activity you will use instructions to communicate an image. This is how digital technologies represent scalable images (such as SVGs or EPS) and how printers produce high-quality output.

Instructions:

1. Students should work in pairs: a drawer and an instructor
    a. the drawer from each pair needs close their eyes
    b. the instructor needs to look to the screen for what they must draw
2. Jump to the next slide so only instructors can see the house
3. Instructors should draw their version of the house and hide their own version
4. Cover up the slide again and everyone should open their eyes
5. Instructors give their drawer step-by-step instructions to replicate their image
6. After finishing their drawings, compare the images

Optional: Have partners swap roles, and draw a car, but this time the instruction givers cannot see what is being drawn, and therefore can’t give adaptive feedback

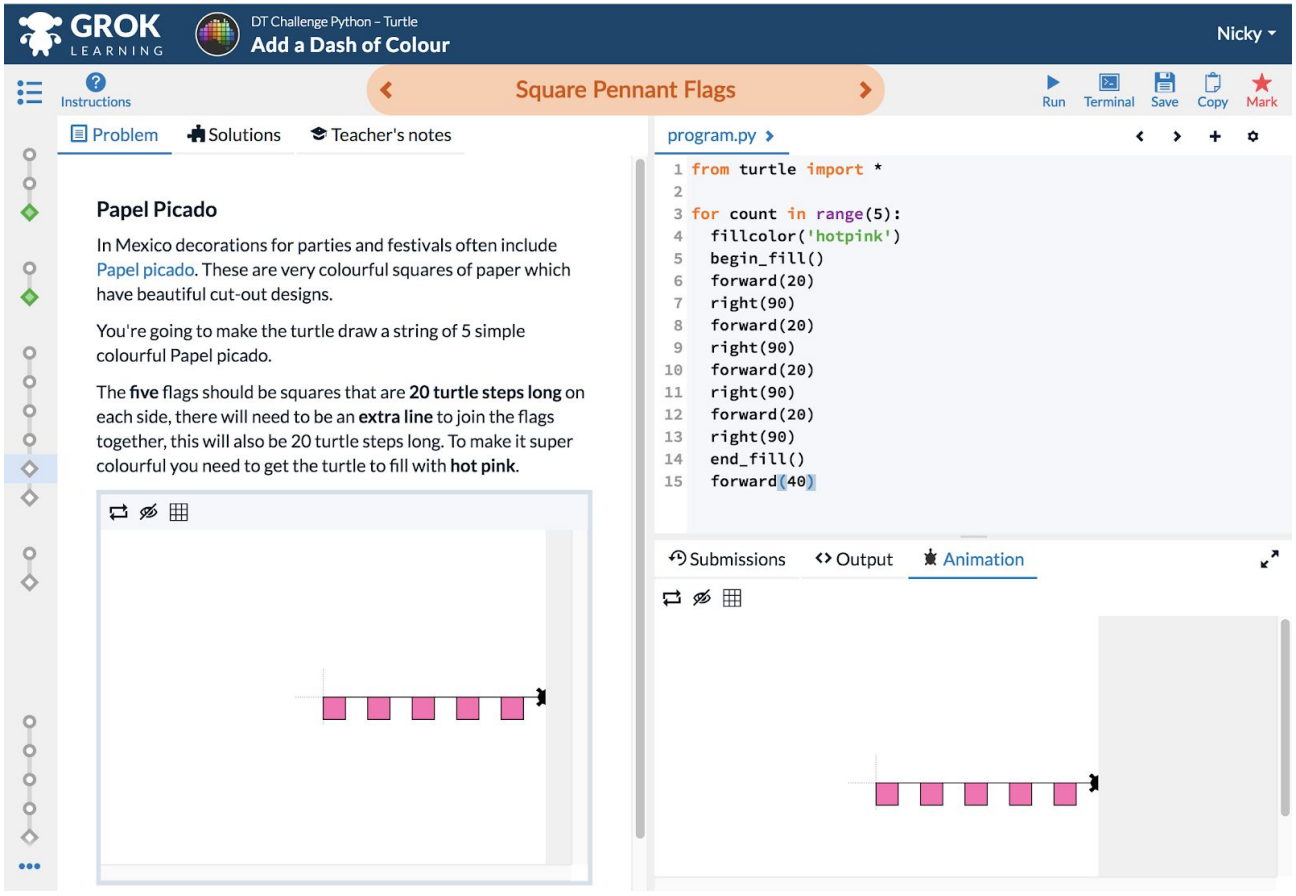
Note: Hiding the target image on the screen discourages specific vocabulary, e.g. house, window, door --- after the activity have a discussion about whether using this vocabulary would have helped.

Funded by the Australian Government Department of Education and Training.

Activity 3 - Turtle

Activity overview

- Students should read interactive notes and complete the coding challenges.
- The students should write their code in Python, referring to the notes as required.
- Click the **Run** button to check your code by running it. Then press the **Mark** button to see if your code is correct and to see feedback for your solution.



Review:

Reflection

Computers are good at following instructions, but those instructions need to be very accurate! What happens if the instructions aren't quite right?

Loops and decisions are useful to automate tasks. Can you think of a tasks at home or at school that you would want to automate? How would you approach this automation?

Funded by the Australian Government Department of Education and Training.