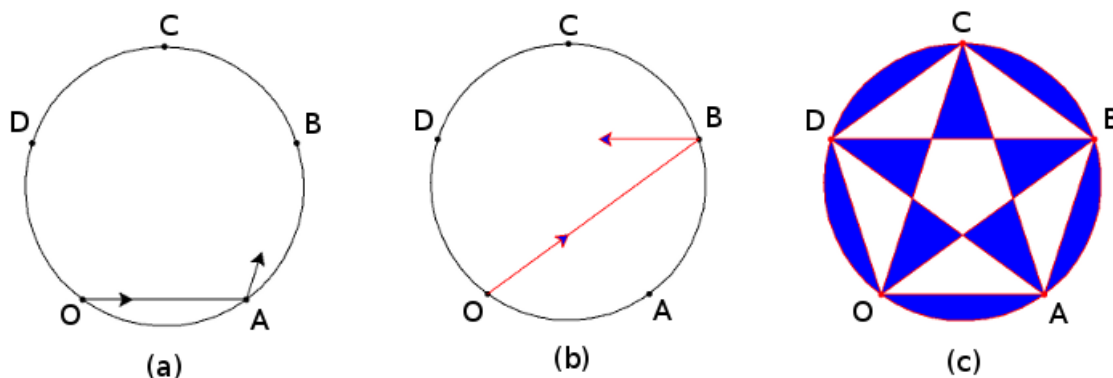


UCL/CAS Training for Teachers
Algorithms and Programming Module 1



WORKBOOK 9

ACTION GEOMETRY: JOINING THE DOTS

COMPUTATIONAL THINKING

Make and Make Sense; Hands On and Minds In. Seymour Papert

- ❖ Addressed to Teachers
- ❖ Action Geometry Unplugged: starting with the square, we explore the properties of simple regular polygons, circles, stars with a view to designing our own unique coloured patterns. We use an unplugged approach with the sprites and turtles of Scratch, Logo and Python and Coffeescript to investigate the static and dynamic geometry of shapes. We employ learning and teaching strategies which call on experimentation, deduction from symmetry, pattern recognition, induction, decomposition and generalisation.
- ❖ Activities are graded: easy to hard – 0 to 5*. You should attempt every activity marked without a star or marked with '*'.

UCL/CAS Training for Master Teachers
Algorithms and Programming Module 1

CONTENTS

Mission 1: ‘Unplugged’ Action Geometry Background to Geometric Shapes	3
Activity 1 Moonshine: Make and Make Sense	3
Worksheet 2 “Action Geometry”: Mimic the sprite/turtle	3
Mission 2: Join the Dots: Hands On	3
Mission 3: Join the Dots: Hands On and Minds In	5
Mission 4: Spinning Out the Results for Complete Stars.....	6

MISSION 1: 'UNPLUGGED' ACTION GEOMETRY BACKGROUND TO GEOMETRIC SHAPES

ACTIVITY 1 MOONSHINE: MAKE AND MAKE SENSE

- ❖ The moon does an exact spin of 360 degrees on its axis -- SPIN(360) -- as it orbits around the earth in 28 days. That's why, everywhere on planet earth, we only see one side of the moon. Can you construct a simple *dynamic model* involving two persons -- one representing the earth the other the moon to describe the moon's dual motion around the earth (**abstraction**) and so illustrate and explain this curious phenomenon? (Walking the talk)
And make a similar walking model for the motion of planet earth around the sun.

WORKSHEET 2 "ACTION GEOMETRY": MIMIC THE SPRITE/TURTLE

- ❖ In this section we use our pencil carrying pet/robot to mimic the tracing action of the sprite/turtle of Scratch and Python, and its awareness only of its:
 1. present position
 2. turning angle/direction.
- ❖ The emphasis in this workbook is on learning by doing, experimentation, deduction from symmetry, induction and **generalising**.
- ❖ What is the difference between the reasoning of deduction and induction?

MISSION 2: JOIN THE DOTS: HANDS ON

Each circle contains a number of equally spaced points around its circumference. In this activity we use coloured pencils and a straight edge to join dots.

POLYGONS

In circles (a), (d), (e), (f) start at O, join dots OA, AB, ...

When tracing each polygon your pet/robot turns at each vertex until it returns to O and then turns to face A again --- it's starting position. A process or a program which returns to its starting point and faces in the direction it first moved in, we call a RETURN process or program.

In addition, can you see, that in drawing a polygon and executing a RETURN process, your pet/robot has turned anti-clockwise through 360 degrees in total. We shall call this a (LEFT) SPIN(360).

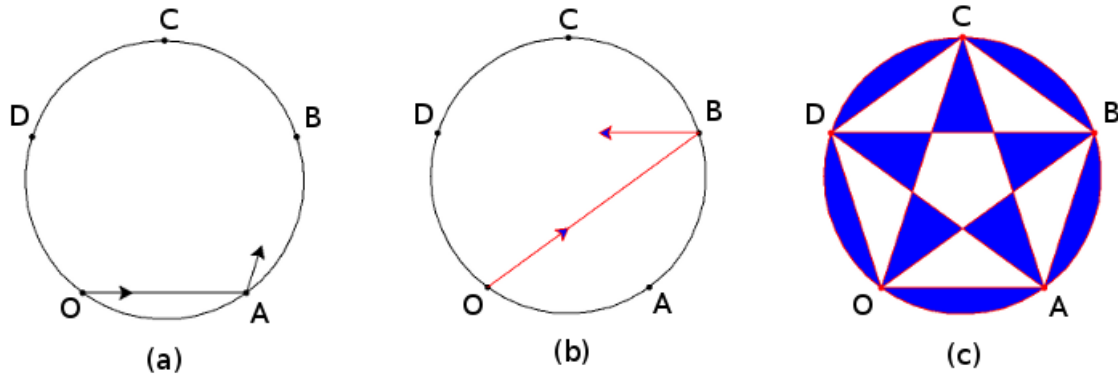


Figure 1. *action geometry*: for the pentagon and pentagram(5 point star)

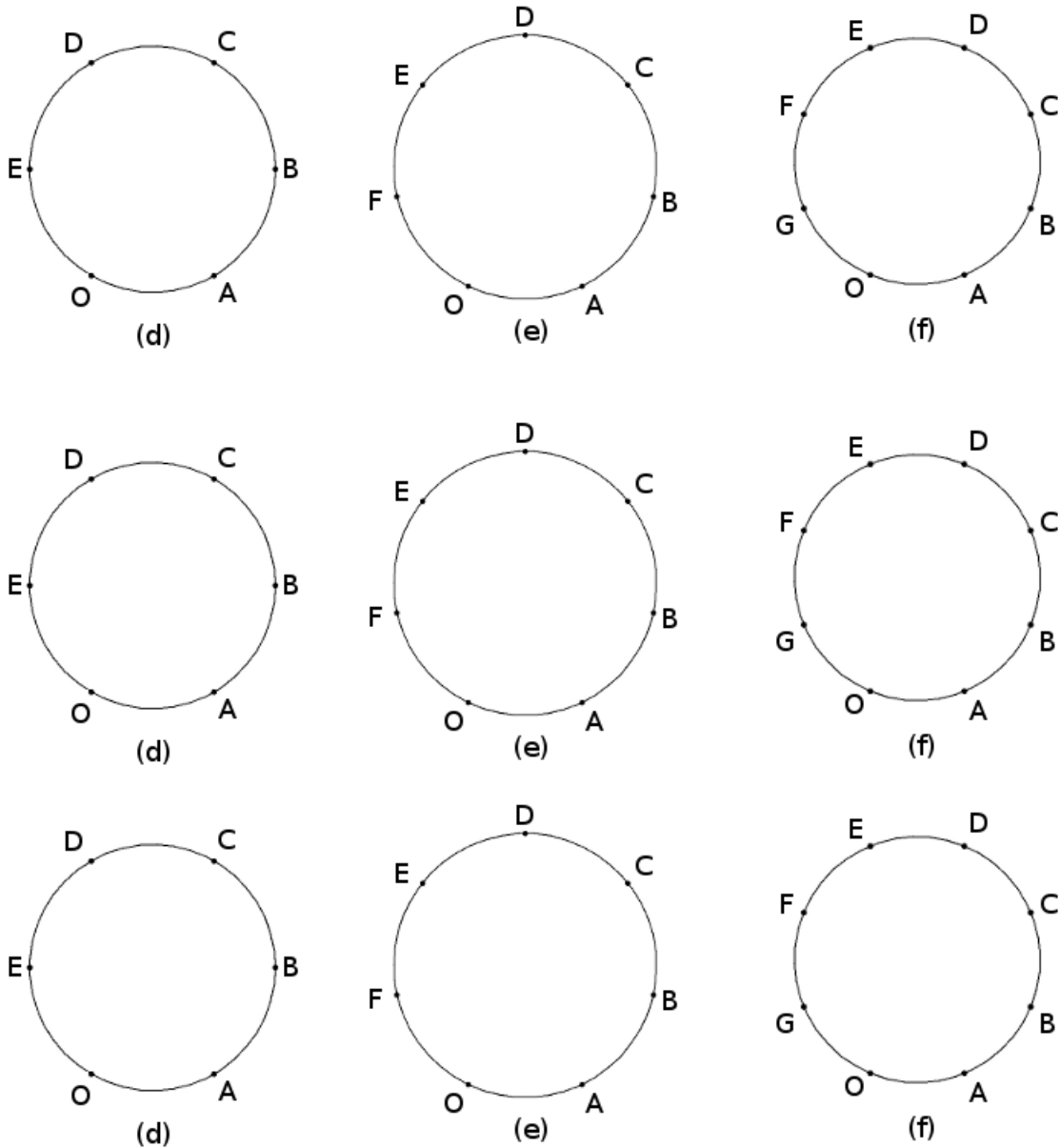


Figure 2. *action geometry*: for the hexagon(6 sides), heptagon(7 sides) and octagon(8 sides).

MISSION 3: JOIN THE DOTS: HANDS ON AND MINDS IN

We have several copies of circles (d), (e), (f), available in Figure 2 to allow for trial and error and to compare resulting stars in Activity 2. We use two equivalent ways of stating each instruction:

Start at O, skip 1 dot --- (join every 2nd dot) --- Join OB, BD

Start at O, skip 2 dots --- (join every 3rd dot) --- Join OC, C to next 3rd dot

Start at O, skip 3 dots ...

In all the processes below we attempt to make each drawing a RETURN process that is returning to O and facing along its original starting direction of motion.

STARS (POLYGRAMS)

Use different coloured pencils to distinguish stars drawn on the same circle.

1. In Figure2 (b), Start at O, skip 1 dot, join OB, BD ... to form a pentagram
2. In (e), Start at O, skip 1 dot, join OB, BD ...
3. In (e), Start at O, skip 2 dots, join OC, CF ... (**generalising**)
4. *In (b), What happens when you start at O, skip 2 dots, join OC, CA... ?
5. *In 2. and 3. you have drawn two 7-side(point) stars (heptagrams). Are they different?
6. *In (e), Start at O, skip 3 dots, join OD, DA... (**generalising**). Is this star different to 2. and 3.?
7. *In (e), Does skipping 4 dots give you a different star again? (**generalising**)
8. ** How many different stars with 7 sides are there? (**generalising**)

❖ All the stars we have drawn so far have been drawn by joining up all the points in the star in a continuous uninterrupted flow visiting each vertex once and returning to O. The stars resulting from this way of drawing we call COMPLETE stars. In these examples so far drawing is a RETURN process if the pet/robot returns to O and faces along its original starting direction.

9. *(d) Start at O, skip a dot, Join OB, BD ... What happens?
10. **(d) You can complete a hexagram(star of David) by moving to A and joining A to C C to E and so on. The hexagram is an example of an INCOMPLETE or COMPOSITE star made up of 2 or more distinct lower order polygons (in this case triangles).
11. ****(g) How many different 8-sided COMPLETE stars are there? (**generalising**). How many different COMPOSITE stars?
12. ***When does skipping one dot (joining every 2nd dot) not work to produce a star for a polygon? (**generalising**)
13. ***When does skipping two dots (joining every 3rd dot) not work to produce a star for a polygon? (**generalising**)
14. *****Can you make a general observation about joining dots **not working** to produce a COMPLETE star for a polygon? (**generalising**)
15. ****Draw up a table with the number of different COMPLETE stars, different COMPOSITE stars for stars with 5 points to stars with 12 points. (Data collection and interpretation)
16. *****If n is a prime how many different COMPLETE and cOMPOSITE stars (n-grams) are there? (Induction and **generalising**.)

MISSION 4: SPINNING OUT THE RESULTS FOR COMPLETE STARS

1. ***In (b), Remembering that your pet/robot executes a SPIN(360) in drawing a pentagon, can you work out how many degrees your /pet/robot turns through in total when drawing the pentagram by joining every 2nd dot, and returning to face in the original starting direction OB?
 2. *** and the heptagram in (e) by joining every 2nd dot and facing original starting direction OC?
 3. *** Can you work out how many degrees your pet/robot turns through in total when drawing the heptagram in (e) by joining every 3rd dot and facing the original starting direction OD? (induction), (**generalising**).
 4. ***** Is there a connection between the number of dots your pet/robot skips at each move and the number of degrees your pet/robot/ turns through in total when it draws a star and returns to O and faces in its original starting direction (induction)?
 5. *****The pentagram is drawn with 5 sides (and 5 vertices/points). If the turning angle at each vertex by your pet/robot is the same (by symmetry) how do you work out the value of this turning angle?
- ❖ There will be a follow up discussion with more exploration in Workbook 7 at ispython.com/repository-1/ and more hints and solutions on ispython.com/discussion --- to be completed.