The journey around the circle takes a longer time than the journey across the chords and the radii.

Abstraction for the Simulation

We want to develop an algorithm that will simulate the two journeys and tell us which journey takes the longer time. (We will use turtle graphics initially).

Some of the key components of the problem are :

- The 2 turtles must start at the same point and travel at the same speed.
- The time or distance must be recorded for each turtle.
- If they journey together, the circle turtle should always finish second. If they journey separately, the time for the circle journey should always be greater.
- The solution does not depend on any particular value of the radius.
- Four key coordinates are needed to represent O, A, B and C.

Writing your Thinking

Take 2 minutes to think about how you tackled this problem.

- **Did you use pen and paper to help visualise possible solutions?**
- Did you draw on the diagram to represent the movements of the turtles?
- **Did you imagine scenarios : what would happen if the chord was a diameter or what is the longest distance along the straight lines?**
- Did you choose a value for the radius and make some calculations to compare the journeys?

Using Think-Pair-Share-Square (TPSS), go through how you and your partner were thinking about how to solve the problem.

Write an algorithm to solve this problem in a computational way.

Try to write a version of your algorithm in pseudo-code.

Pseudo-Code

#Initialise the simulation environment

Import the turtle modules and Create a canvas

Define 2 instances of turtles and give them distinguishing properties

Set each turtle position to A

Set radius to a value

Create a list of 4 coordinates [B, C, O, A]

#The circle journey and the chord-radii journey

Start the clock

    Journey around the circle

Stop the clock and record time on canvas

Start the clock

    Travel from A – B – C – O - A

Stop the Clock and record time on canvas

Key Notes

The speed of the turtles can be set by the attribute speed. This is the speed at which the shapes are being drawn.

Can you see how this might create difficulties for how our simulation may be accurate or realistic?

The turtle graphics are quite limited compared to say pygame graphics. Only one turtle can move at any one time. To program the turtles to look like they are racing is challenging. A solution like this is an example of a **HEURISTIC**. The problem is not fully solved but solved for practical purposes, given the context we are working in.

This video shows how to use heuristics to approach problem solving.