2Logo User Guide

Contents

1 Introduction

2Logo gives children the opportunity to control an on screen Turtle using a range of commands.

Logo is a text-based coding language. 2Logo will give pupils guidance when their code does not match the correct structure of the language and will prepare pupils for the demands of more complex text-based coding.

2Logo allows children to practise basic single line Logo programming, then move on to multi-line programs which lead them to creating complex procedures and combining these procedures.

There is an in-built bank of turtle images and matching background to allow Logo to be adapted to a variety of purposes from exploration of mathematical concepts such as angles and shapes, to coordinate mapping using maps to History and Geography topic themed exercises.

Teachers can design 2Logo files and then set them as 2Dos for their pupils. Pupils can also create their own resources.

2Logo also includes Logo challenges - these take the form of an adventure game in which children must use their Logo skills to complete the game. The challenges are progressive with accompanying videos and hints that introduce and practice Logo skills. See the Logo Challenge section for full details.

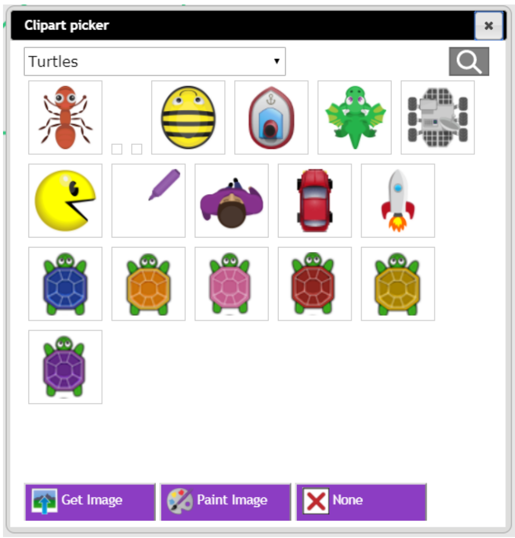
For more information about 2Dos see the 2Dos User Guide.

2 2Logo Screen Layout

Toolbar buttons

Most of the toolbar buttons are the same as all other tools in Purple Mash. The following buttons relate specifically to 2Logo.

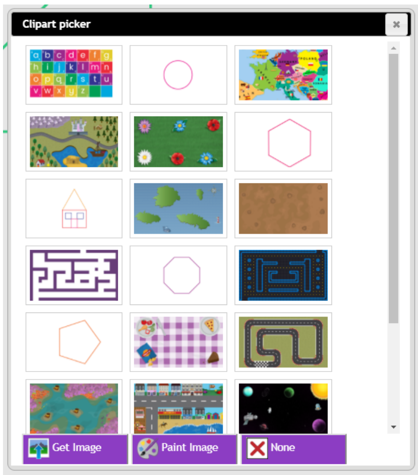
Use this button to select a different turtle design. There are several animated in-built designs or use Clipart from the Clipart picker or create designs by painting or uploading images.



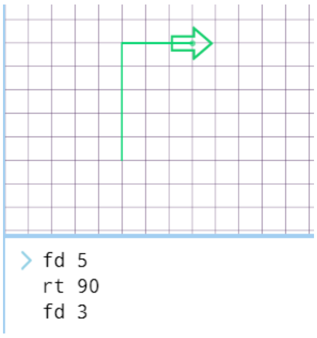
The turtle's starting position can be changed by dragging it to a different location.

Use this button to select a themed background design. There are several in-built designs or pupils can use Clipart from the Clipart picker or draw their own design. In-built designs include:

- A variety of polygons to create Logo instructions to re-create.
- An alphabet grid; can the turtle spell words?
- Maps; map a turtle journey using Logo, map the pirate's route to the treasure?
- A maze
- A computer game background with matching munching turtle.
- Racetrack and street map designs
- Outer space



Use this button to display a grid overlay that can be used to count steps forward and backwards. This is also useful for maths work connected to area and perimeter.

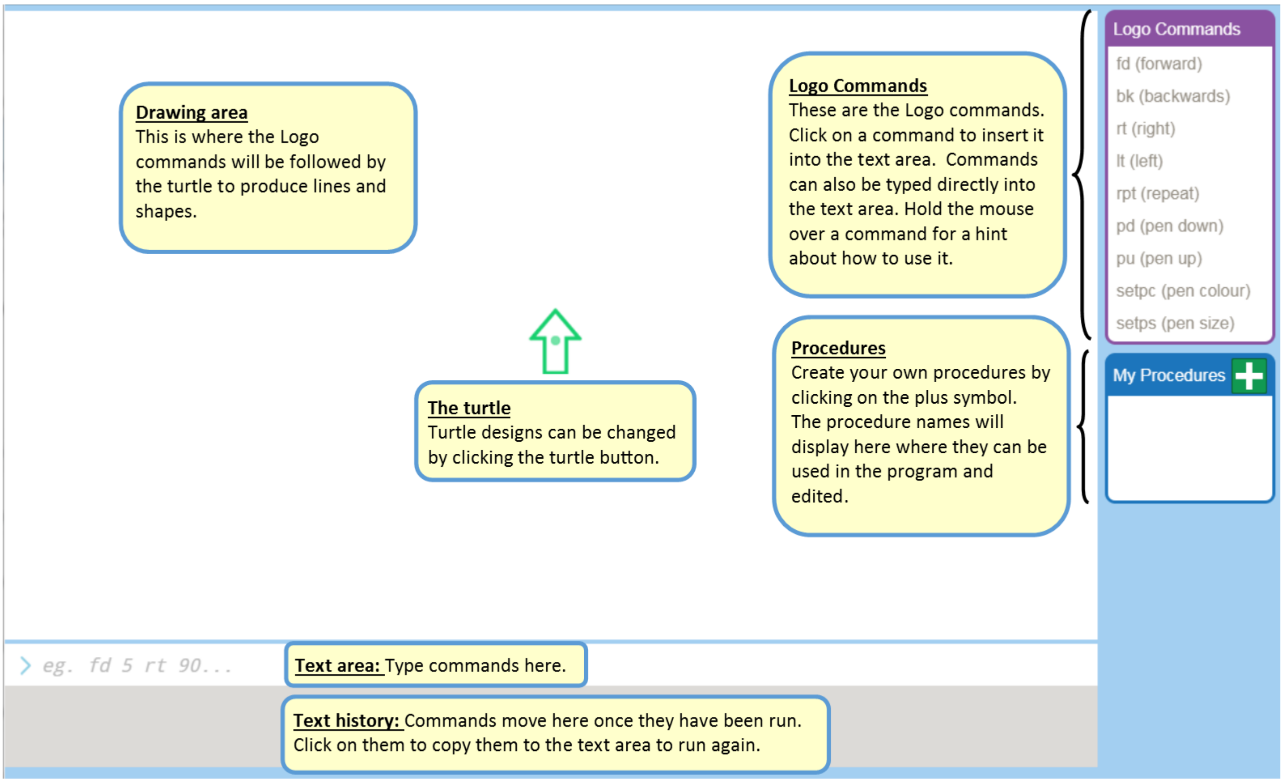


```
fd 5
rt 90
fd 3
```

This button resets the drawing area. Any drawing will be cleared, the turtle will return to its starting position and the pen size and pen colour will be reset to their original values.

This slider controls the speed that the turtle travels, select a slower speed to see the turtle follow each step more clearly.

Screen Layout



Drawing area This is where the Logo commands will be followed by the turtle to produce lines and shapes.

Logo Commands These are the Logo commands. Click on a command to insert it into the text area. Commands can also be typed directly into the text area. Hold the mouse over a command for a hint about how to use it.

Procedures Create your own procedures by clicking on the plus symbol. The procedure names will display here where they can be used in the program and edited.

The turtle Turtle designs can be changed by clicking the turtle button.

Text area: Type commands here.

Text history: Commands move here once they have been run. Click on them to copy them to the text area to run again.

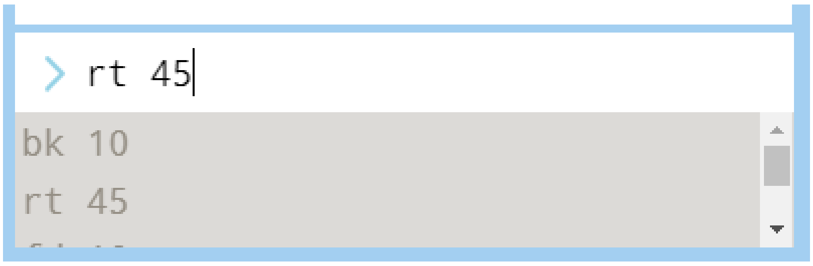
3 2Logo Modes

2Logo can be used in single-line mode or multi-line mode. Procedures can also be created.

Use the toggle button at the top to switch modes

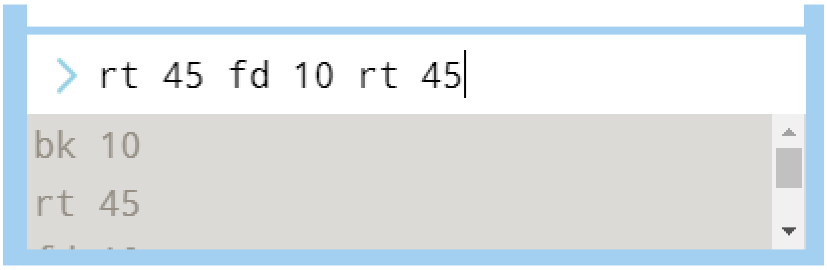
Single-line mode

This is the best mode for pupils who are just beginning to learn Logo. Lines of code run one line at a time. Type the code into the text area and press enter to run.



```
> rt 45
bk 10
rt 45
```

You can input more than one command in a line.

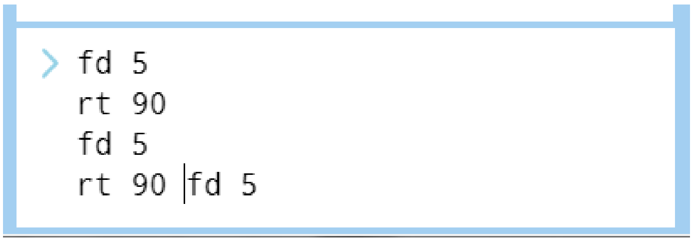


```
> rt 45 fd 10 rt 45
bk 10
rt 45
```

The command history appears in the greyed area below the text area. Click on a command here to copy it to the text area to re-run it.

Multi-line mode

In multi-line mode, type several lines of commands in the text area. Then click on the ▶ button to run the code.



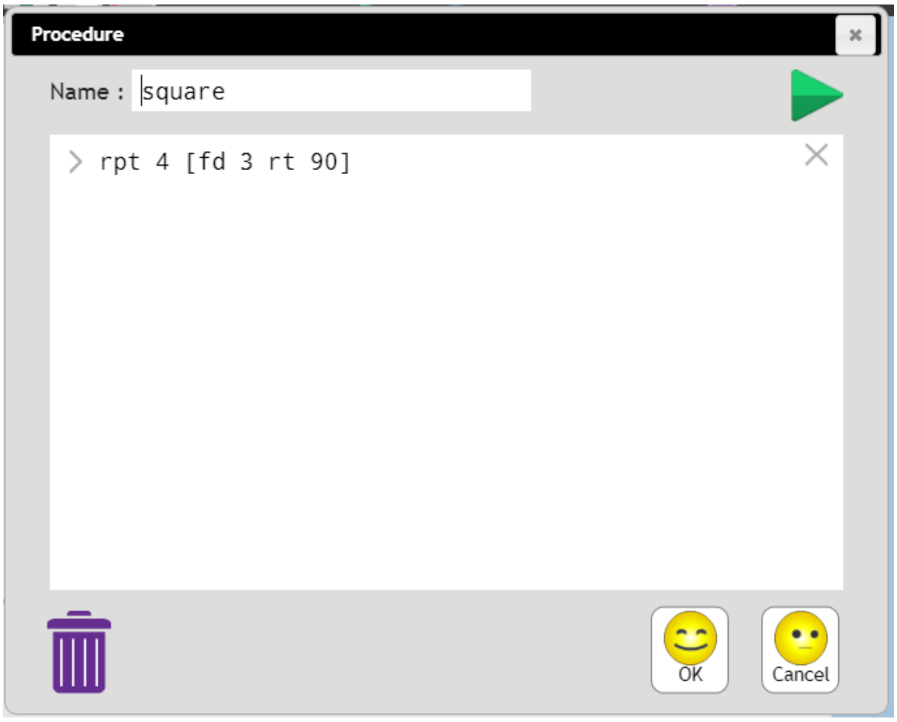
```
> fd 5
  rt 90
  fd 5
  rt 90 fd 5
```

The commands can then be edited and run again.

Procedures

Procedures are chunks of Logo with a procedure name that can be run by calling them by name. So instead of repeatedly writing the code to draw a cube when designing a pattern, you can call the 'cube' procedure and the turtle will draw a cube.

To add a procedure click on the ➕ button at the top right of the procedure box. If you are in multi-line mode, any text that is in the text area will get automatically copied to the procedure for editing. Give the procedure a name and then edit or add the Logo instructions.



Procedure

Name : square

```
> rpt 4 [fd 3 rt 90]
```

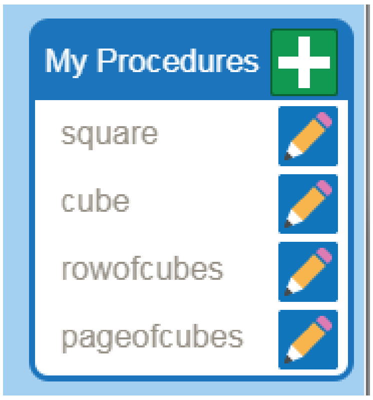
To test the procedure before using it in your program, click on the ▶ button at the top-right of the procedure window to see what it will create.

Click the OK button to save the procedure. The saved procedures can then be edited by clicking on the pencil icon next to their names.

To call the procedure type the procedure name or click on the name in the procedure window and it will appear in the text area. You can also use a procedure within another procedure for example write a procedure to draw a single square then write a procedure to draw a row of squares that calls and repeats square then a procedure that draws several rows of squares.

For example, the following file has 4 procedures:

- square draws a square
- cube draws a cube using the square procedure and some extra Logo code
- rowofcubes draws a row of cubes using the cube procedure and some extra Logo code
- pageofcubes draws a whole page of cubes



4 The Logo language

To make the screen turtle move, the user needs to input a command.

The format is the command followed by a space and then normally a number.

To make the screen turtle move forward type fd 1. Instructions can be typed in lower case, upper case or a mixture of the two.

There are a range of different instructions in Logo.

Instruction	Description	Example
BK	Move backwards a distance of units	BK 50 – Move the turtle back 50 units
FD	Move forward a distance of units	FD 50 – Move the turtle forward 50 units
LT	Turn left a given number of degrees	LT 90 – Turn the turtle 90° left
RT	Turn right a given number of degrees	RT 45 – Turn the turtle 45° right
RPT	Repeat a set of instructions a number of times	RPT 4[FD 10 RT 90] – This will draw a square. The instructions to be repeated must be inside square brackets.
SETPC	Set pen colour to a value or colour word	SETPC 1 – Pen colour is RED SETPC BLUE - Pen colour will be blue The colours that are included by name in 2Logo are; black, white, red, blue, green, yellow, orange, purple, pink, grey, brown, lime, cyan, magenta Advanced users can also create their own colours using HTML color codes e.g. setpc "#990000" - dark red (you need the # and the ")
SETPS	Sets the thickness of the pen	SETPS 1 (line will be thin) SETPS 20 (line will be thick) The default value is 4.
PU	Lifts the pen off the screen	
PD	Places the pen to begin drawing	

Comments can be added to the code. These are lines of text within the program that explains what is happening and makes it easier for others to interpret. To do this put a semi-colon before the line of text e.g.

; move forward 1

fd 1

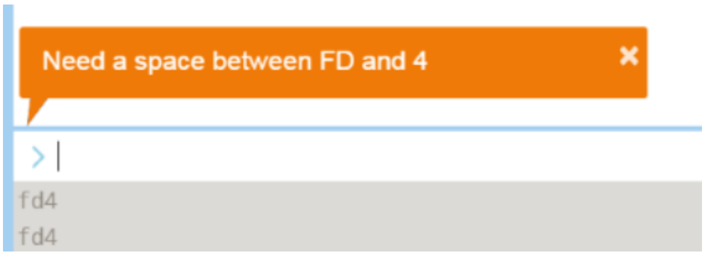
; next turn 90 right

rt 90

5 Error Reporting

If invalid commands are input into 2Logo. An error report will help the user to work out the problem.

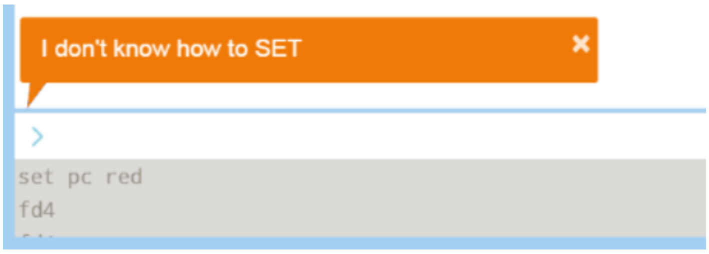
Missed space



Need a space between FD and 4

```
fd4
fd4
```

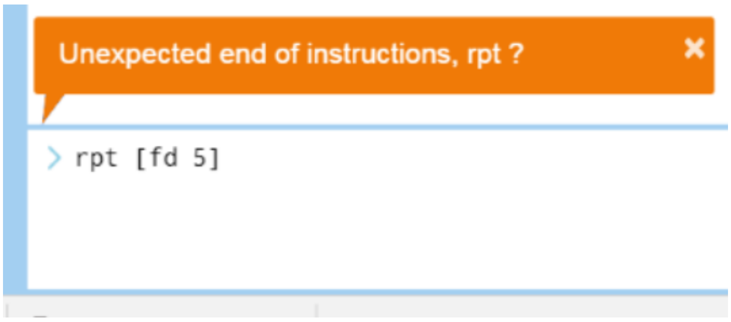
Extra space



I don't know how to SET

```
set pc red
fd4
```

Missed repeat number



Unexpected end of instructions, rpt ?

```
rpt [fd 5]
```

6 Saving and Sharing Logo

Logo that you have made can be saved and shared as 2Dos like any other resource. They can also be shared by pupils to blogs, displayboards and other links

Save the game and then click on the 🌐 for sharing options.

For further details about Share Links see the guide at Share Link User Guide.

For further details about 2Dos, see the manual at 2Dos User Guide.

7 Logo Challenges

There are 12 Logo challenges that take the form of an adventure game. They form a storyline in which a young time cadet jumps through time to fix errors in the time-line caused by a robot called Logonator. In each location, they have to solve a Logo puzzle in order to progress.

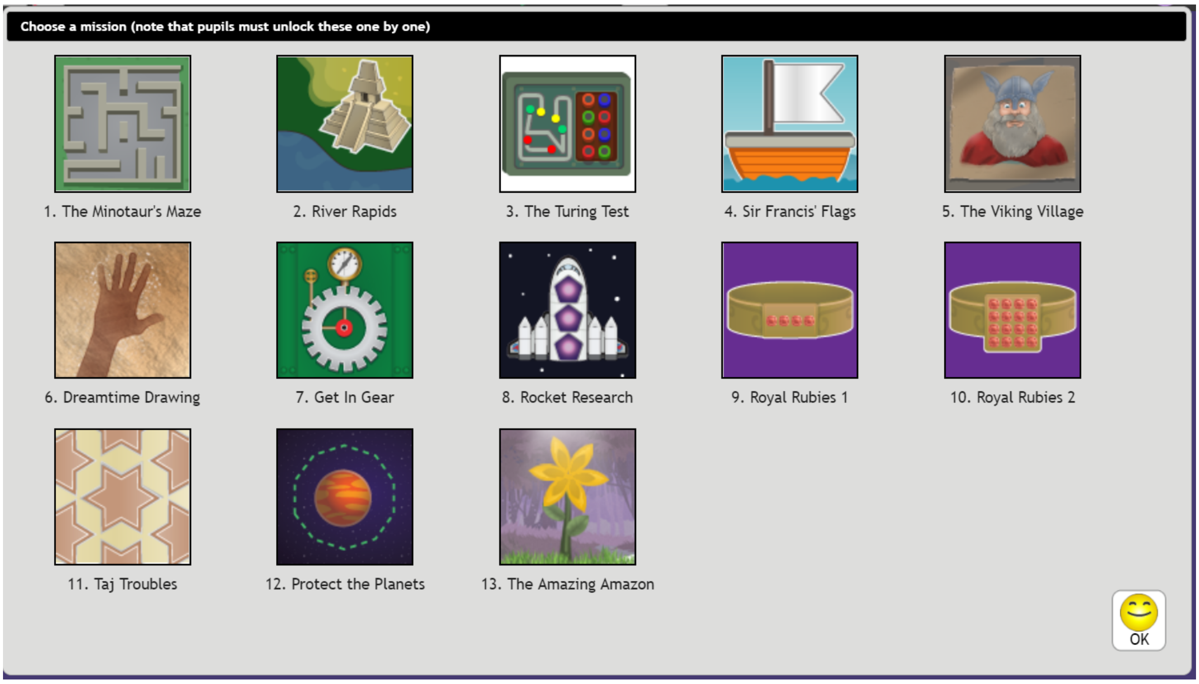
Each challenge has a video which takes the form of an update from TimeHQ telling them what they must do. The videos also include important teaching points about how to complete the challenge.

The challenges get progressively harder see the following sections for details of the Logo in each challenge.

Each challenge has a list of success criteria which players must fulfill.

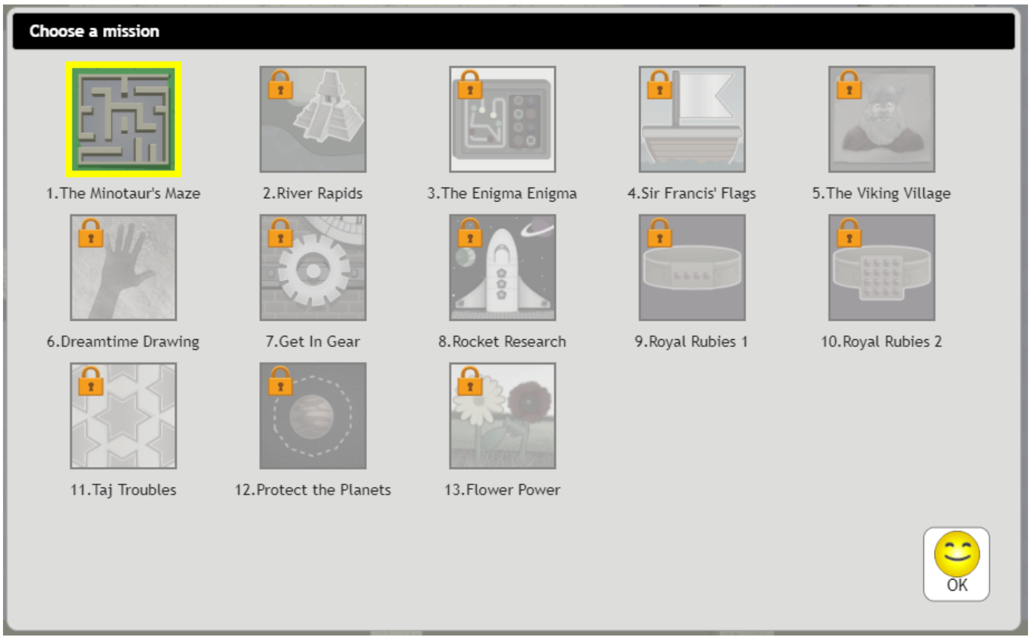
When a teacher opens the Logo Challenges they can set any challenge as a 2Do for their class(es) and try out all the challenges

When a pupil opens the challenges they only have the choice of the challenges that they have successfully completed or are in the process of working on, the rest are locked



Choose a mission (note that pupils must unlock these one by one)

Teacher view

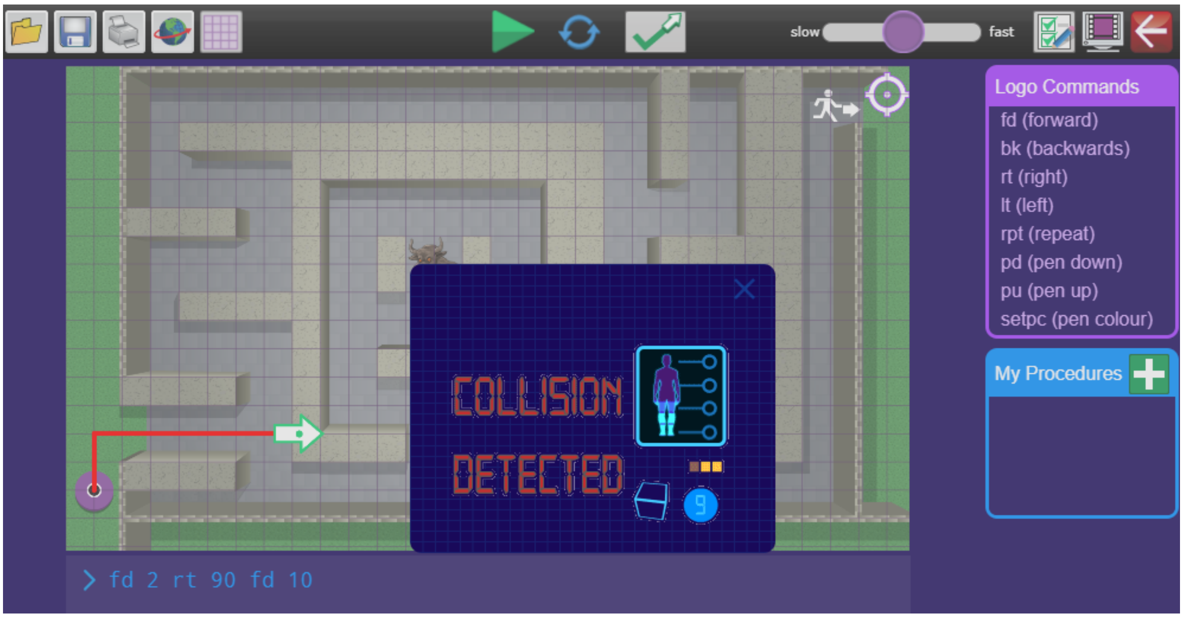

Choose a mission
1.The Minotaur's Maze
2.River Rapids
3.The Enigma Enigma
4.Sir Francis' Flags
5.The Viking Village
6.Dreamtime Drawing
7.Get In Gear
8.Rocket Research
9.Royal Rubies 1
10.Royal Rubies 2
11.Taj Troubles
12.Protect the Planets
13.Flower Power
OK

Pupil view

How to play the Challenges

Children should write their Logo in the text area at the bottom. It is essential that they build up their Logo code progressively so that their solution displays the full answer rather than typing one step, deleting it and then typing the next. This means that they will not have to start from the beginning every time they make a mistake. The following is an example using challenge 1

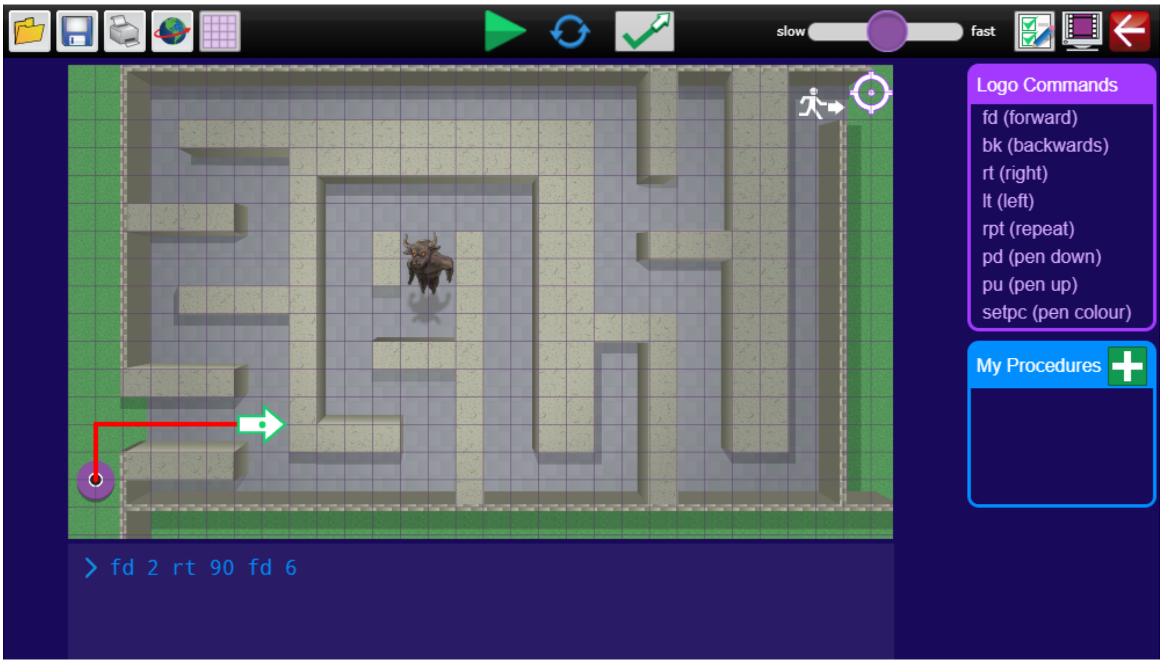
1) Write the first few instructions and press the ▶ button to try them out. Normally this involves some trial and error but children should be encouraged to try inputting a few steps at a time if they can.



COLLISION DETECTED

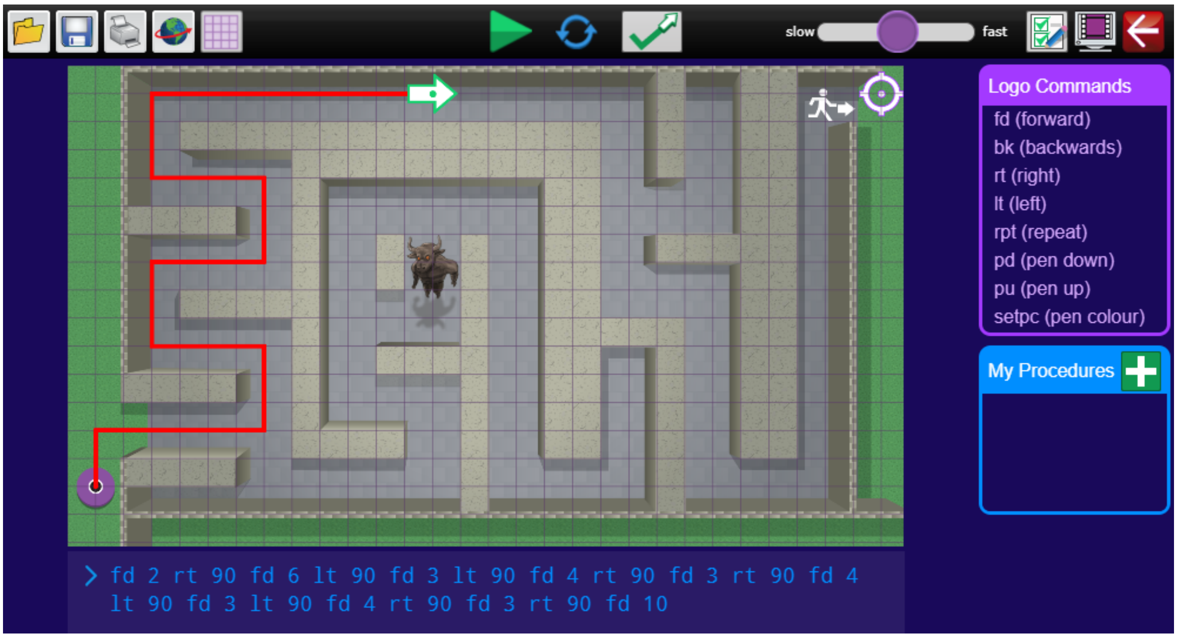
```
> fd 2 rt 90 fd 10
```

2) Correct any errors in the code, press the ⟳ button to reset the character and line and then re-run by pressing ▶. Repeat this process until the code is correct so far.



```
> fd 2 rt 90 fd 6
```

3) Add some more instructions and repeat steps 2 and 3 .



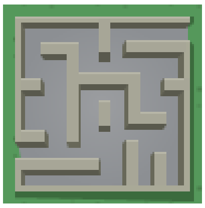
```
> fd 2 rt 90 fd 6 lt 90 fd 3 lt 90 fd 4 rt 90 fd 3 rt 90 fd 4
  lt 90 fd 3 lt 90 fd 4 rt 90 fd 3 rt 90 fd 10
```

4) Continue to build up the code until you think you have the correct solution. Pupils can make use of the play speed slider to run their code at different speeds slow fast. When the solution is reached, click on the check solution button. This gives pupils some idea of what is correct and what is missing or incorrect.

Use the slider to view their result and the correct result. They can then correct their challenge.

○ Your answer | Correct answer ○

7.1 Challenge 1: The Minotaur's Maze



Location and Time: Ancient Greece, 1705 BC

Story:

Logonator has travelled back in time to Ancient Greece to interrupt the time continuum. He rules over the city of Athens and each year he sends a young child to the island of Crete to be eaten by the minotaur - the awful monster that King Minos keeps as a pet in his labyrinth. But this child will grow up to be an important inventor. He must be saved!

Your mission is to lead the child through the Minotaur's maze to safety to protect the future.

Teaching points:

Use FD, RT and LT commands to navigate through the maze.

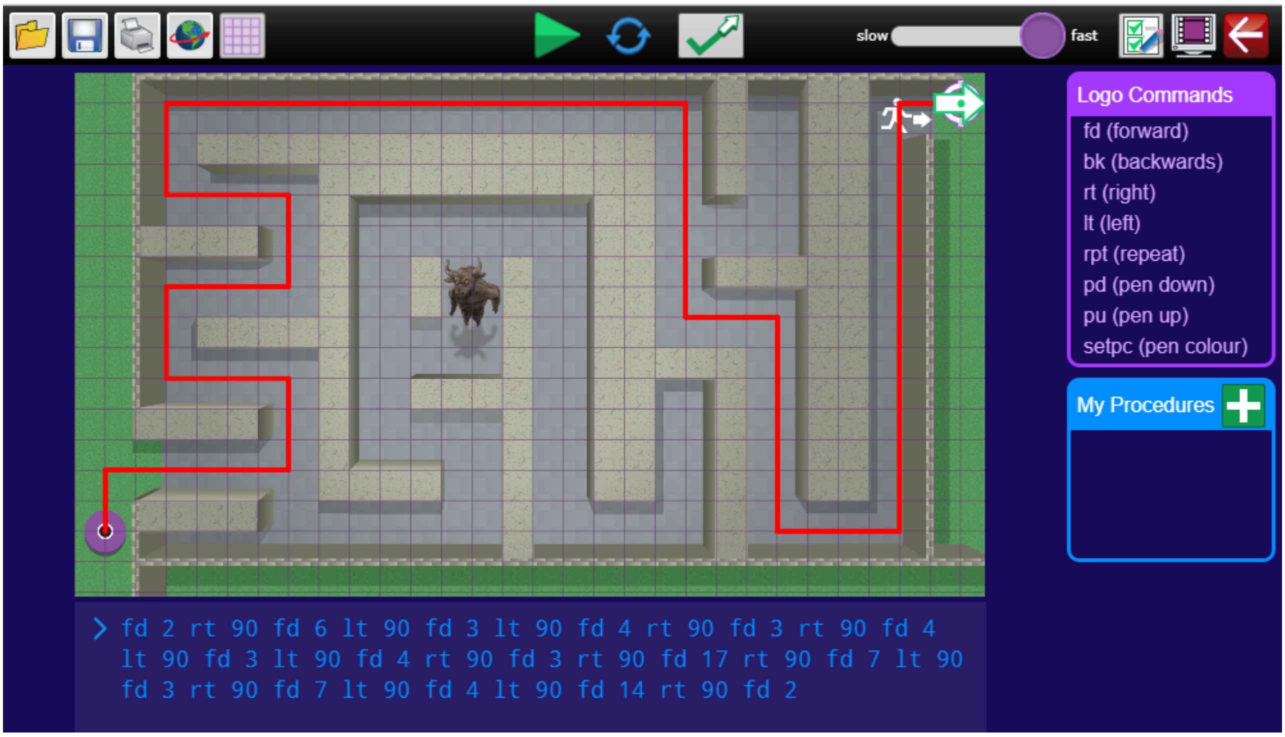
Use of degrees to turn - only 90° in this challenge.

Building up the code (see previous section).

Use of the grid to help with distances ▦ .

Solution:

fd 2 rt 90 fd 6 lt 90 fd 3 lt 90 fd 4 rt 90 fd 3 rt 90 fd 4 lt 90 fd 3 lt 90 fd 4 rt 90 fd 3 rt 90 fd 17 rt 90 fd 7 lt 90 fd 3 rt 90 fd 7 lt 90 fd 4 lt 90 fd 14 rt 90 fd 2



```
> fd 2 rt 90 fd 6 lt 90 fd 3 lt 90 fd 4 rt 90 fd 3 rt 90 fd 4
  lt 90 fd 3 lt 90 fd 4 rt 90 fd 3 rt 90 fd 17 rt 90 fd 7 lt 90
  fd 3 rt 90 fd 7 lt 90 fd 4 lt 90 fd 14 rt 90 fd 2
```

7.2 Challenge 2: River Rapids

Location and Time: Mexico, 1392 AD

Story:

Logonator has stolen the sacred Aztec golden statue from the great Pyramid at Teotihuacán.

The Aztec priests believe that the rain god is angry and that the harvest will fail this year if it is not returned. You must sail up the river and return the statue to its rightful resting place.

Teaching points:

You can use any route along the river you want, as long as you end up at the Pyramid.

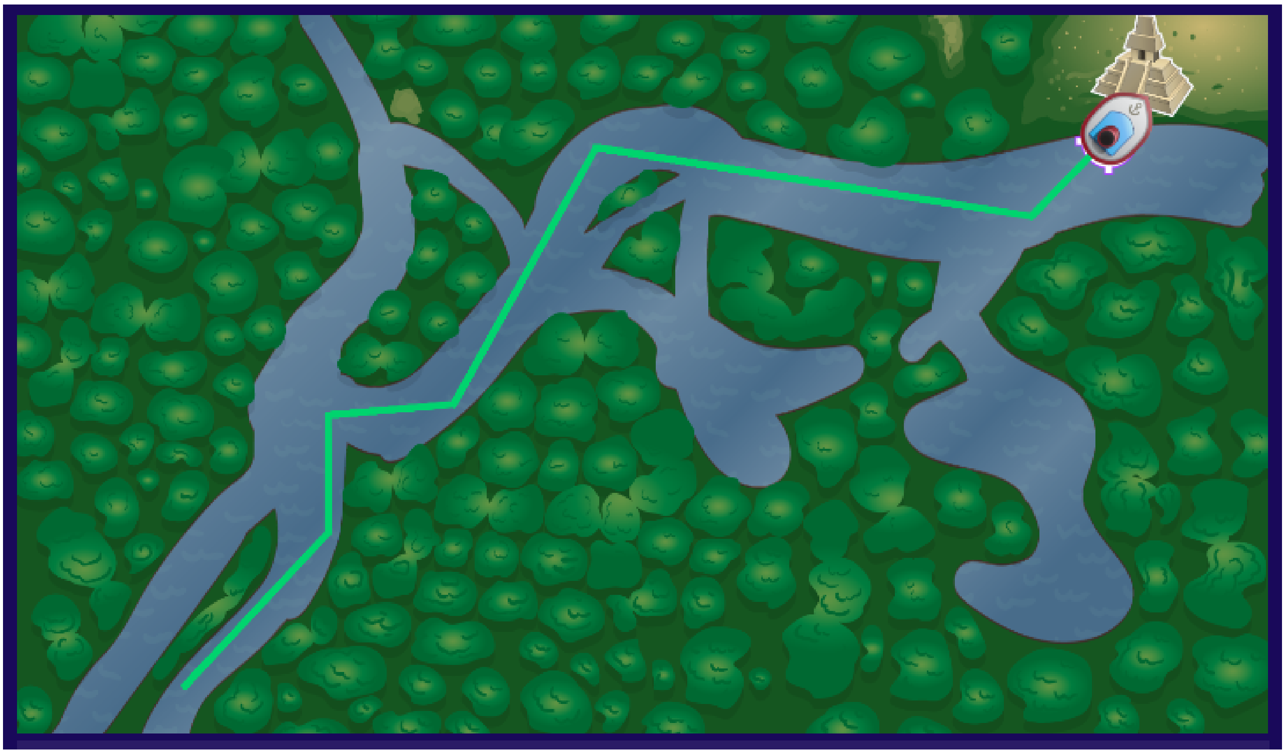
Use FD, LT and RT to plot your course.

Use of the protractor to determine the degrees to turn ◎.

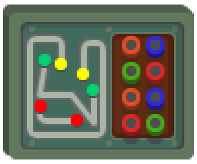
Building up the code.

Possible Solution:

rt 43 fd 5 lt 43 fd 2.8 rt 85 fd 3 lt 56 fd 7 rt 70 fd 10.5 lt 55 fd 3



7.3 Challenge 3: The Turing Test



Location and Time: Bletchley Park, England, 1941

Story:

Logonator has broken into Bletchley Park and has sabotaged the Enigma code-breaking machine, used to break the code that transmitted secret enemy messages during World War 2.

You must repair the machine and make sure that the Allies can intercept enemy communications, only then can they win the war.

Wire up the machine, taking care to use the right colour wires. The yellow terminals must be connected with yellow wire, and similarly for the green and red wires.

Teaching points:

Use of SETPC <colour> e.g. SETPC RED to change the color of the wire drawn to red.

Use the ready-made "goto" procedures to jump to the terminals e.g. gotob will jump to the beginning of the red wire with the arrow pointing towards 0°.

Use of the protractor to determine the degrees to turn ◎.

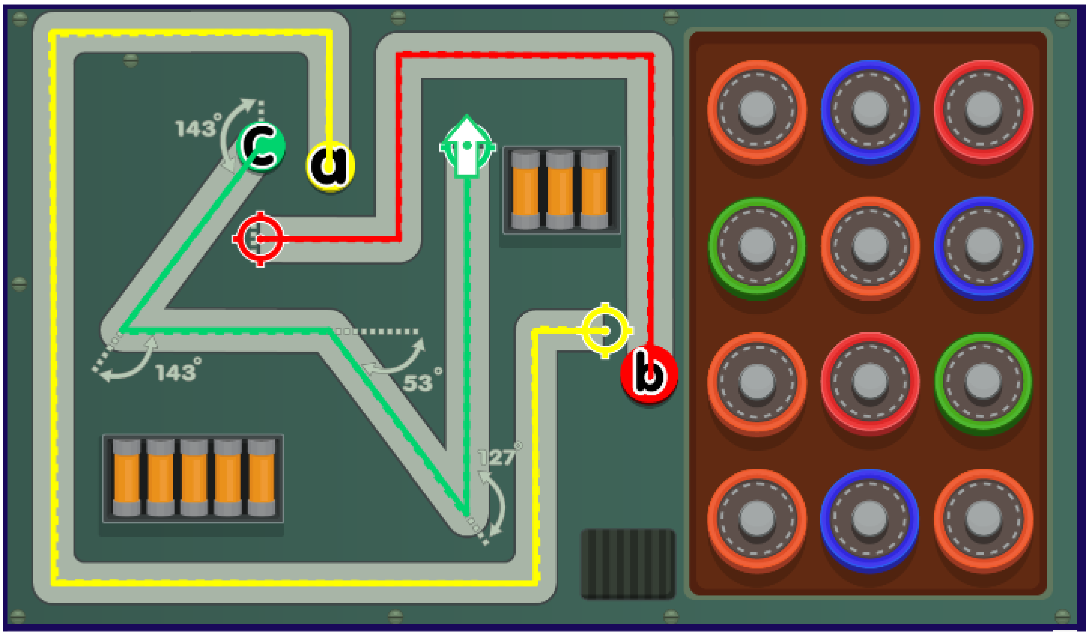
Encourage children to layout their Logo in a readable way using line breaks and comments (to comment put a ';' before the comment so the program doesn't see it as code (see solution)

Possible Solution:

```
;yellow wire
gotoa
setpc yellow fd 6 lt 90 fd 12 lt
90 fd 24 lt 90 fd 21 lt 90 fd 11 rt
90 fd 3 lt 90
```

```
;red wire
gotob
setpc red pd fd 14 lt 90 fd 11 lt
90 fd 8 rt 90 fd 6 rt 90
```

```
;green wire
gotoc
setpc green pd rt 217 fd 10 lt
127 fd 9 lt 37 rt 90 fd 10 lt 143
fd 16
```



7.4 Challenge 4: Sir Francis' Flags

Location and Time: The English Channel 1588 AD

Story:

Sir Francis Drake and his fleet are about to be attacked by the Spanish Armada, led by Logonator disguised as a Spanish Admiral. You must warn them!

There are no telephones in 1680 so you must display the flags on the message boat. When Francis Drake sees the correct flags he will know that the Armada is coming and be ready for it!

Teaching points:

Draw the four different shapes.

Use the ready-made "goto" procedures to jump to the lettered points.

Use of the protractor or the marked angles to determine the degrees to turn .

Encourage children to layout their Logo in a readable way using line breaks and comments (to comment put a ';' before the comment so the program doesn't see it as code (see solution)

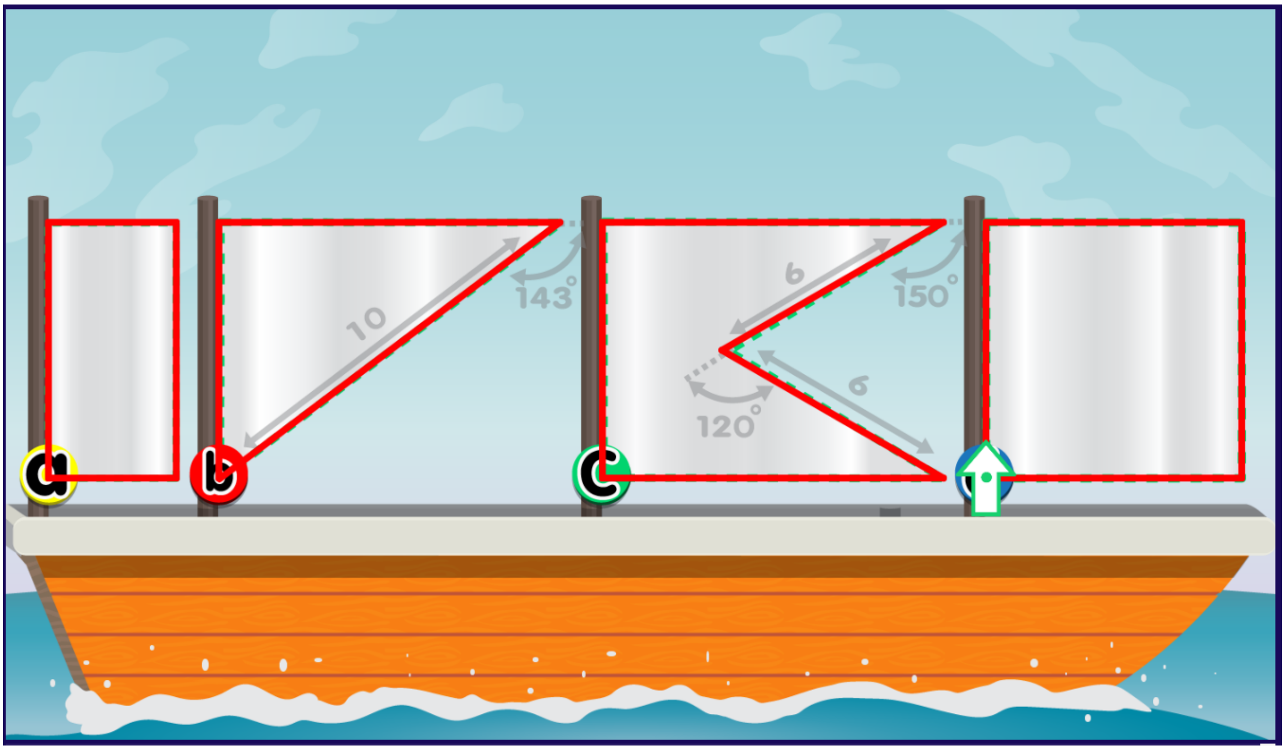
Possible Solution:

```
;A
gotoa
fd 6 rt 90 fd 3 rt 90 fd
6 rt 90 fd 3 rt 90
```

```
;B
gotob
fd 6 rt 90 fd 8 rt 143
fd 10 rt 127
```

```
;C
gotoc
fd 6 rt 90 fd 8 rt 150
fd 6 lt 120 fd 6 rt 150
fd 8 rt 90
```

```
;D
gotod
fd 6 rt 90 fd65 rt 90
fd 6 rt 90 fd 6 rt 90
```



7.5 Challenge 5: Viking Village

Location and Time: Iceland, 981 AD

Story:

Logonator has put a curse on the famous Viking, Erik the Red, giving him a terrible fear of water. This will prevent him from discovering Greenland and setting up the first Norse settlements. To lift the curse, you must work out the answer to Logonator's question and write the answer with rune shapes.

Teaching points:

Draw the four different runes to answer the question.

You will need to use the BK command to complete these.

Use the ready-made "goto" procedures to jump to the lettered points a, b, c and d.

Use of the protractor or the marked angles to determine the degrees to turn .

Encourage children to layout their Logo in a readable way using line breaks and comments (to comment put a ';' before the comment so the program doesn't see it as code (see solution)

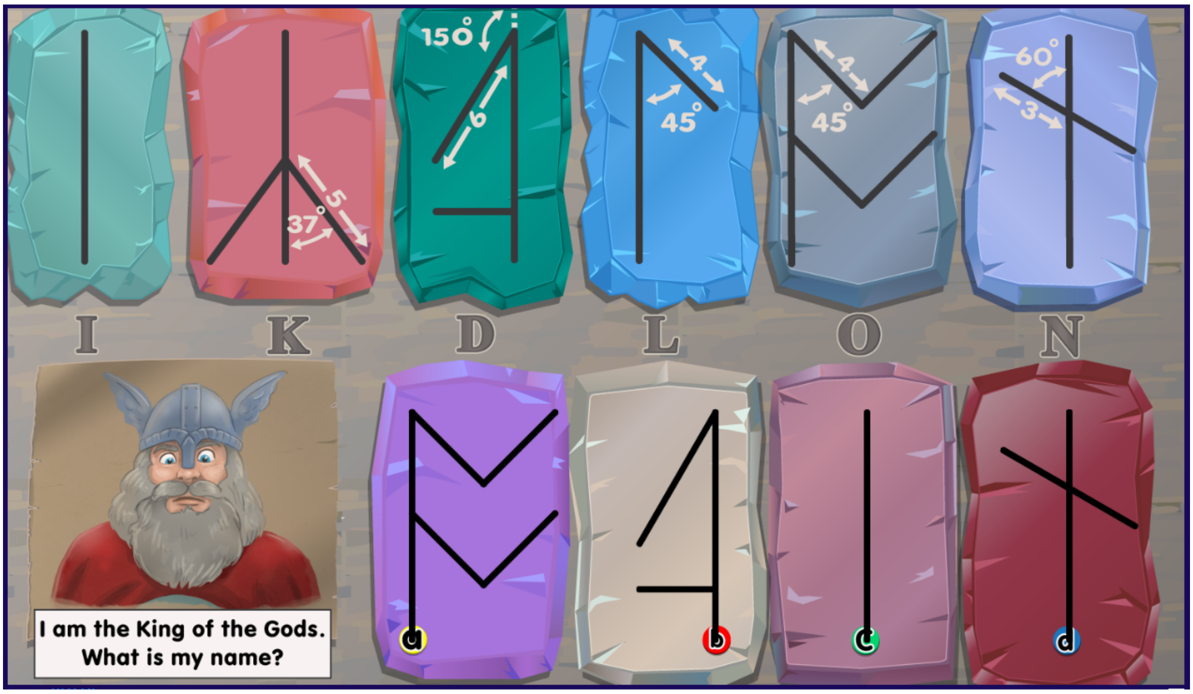
Possible Solution: **NB** The answer is 'ODIN' spelt in runes.

```
;Letter O
gotoa
fd 9 rt 135 fd 4 lt 90
fd 4 bk 4 lt 90 fd 4 rt
45 bk 4 rt 135 fd 4 lt
90 fd 4 bk 4 lt 45
```

```
;Letter D
gotob
fd 2 lt 90 fd 3 rt 180
fd 3 lt 90 fd 7 lt 150
fd 6
```

```
;Letter I
 gotoc
fd 9
```

```
;Letter N
gotod
fd 9 bk 3 lt 60 fd 3 bk
6 pu bk 20
```



7.6 Challenge 6: Dreamtime Drawing

Location and Time: Australia, 30000BC

Story:

This time, Logonator has travelled far back in time, to 30 thousand years ago. He has destroyed some ancient Aboriginal art patterns – this will stop human civilisation from ever developing language and writing. You must fill in the missing shapes to make the Aboriginal art complete.

Teaching points:

Complete the missing shapes. You should use the REPEAT command (this is explained in the video).

Use the ready-made "goto" procedures to jump to the lettered points.

Encourage children to layout their Logo in a readable way using line breaks and comments (to comment put a ';' before the comment so the program doesn't see it as code (see solution)

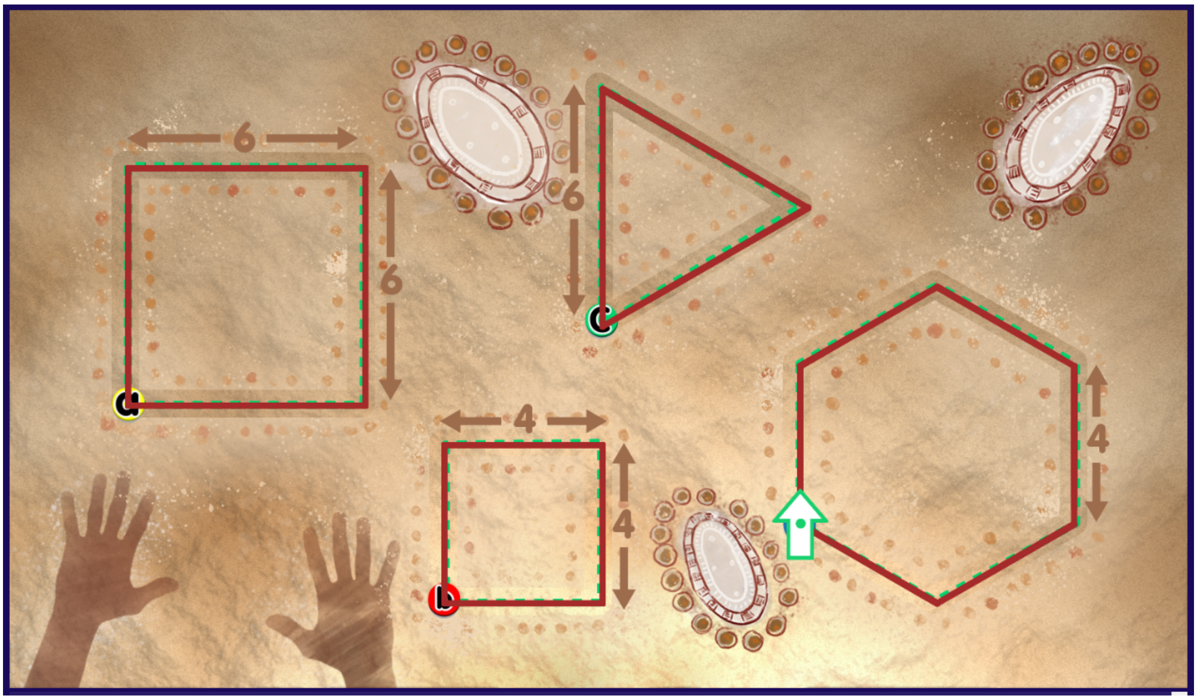
Possible Solution:

```
;big square
gotoa
rpt 4[fd 6 rt 90]
```

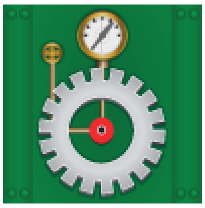
```
;small square
gotob
rpt 4[fd 4 rt 90]
```

```
;triangle
gotoc
rpt 3[fd 6 rt 120]
```

```
;hexagon
gotod
rpt 6[fd 4 rt 60]
```



7.7 Challenge 7: Get in Gear



Location and Time: Manchester, England, 1823 AD

Story:

The industrial revolution is in full swing in England in the 19th century, but Logonator has stolen some crucial pieces of an invention - the first steam engine. This will seriously damage the course of history! You must fix the engine by replacing the missing cogs.

Teaching points:

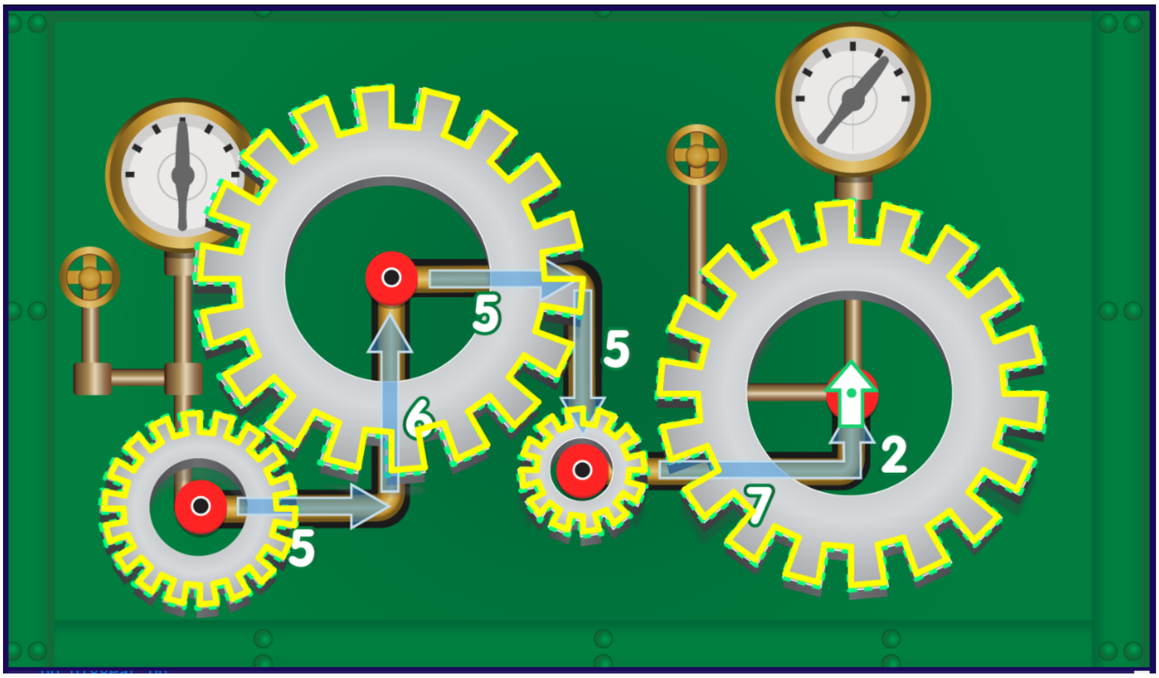
In this challenge you will need to use the penup (pu) and pendown (pd) commands to move to the centre of each cog without drawing a line and then run the procedure for the cog.

There are no goto procedures for you to use.

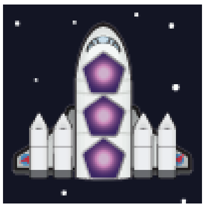
There are three procedures called <biggear>, <mediumgear> and <smallgear>. Call these by clicking on them or typing their name in your Logo code.

Possible Solution:

```
mediumgear pu
fd 6 rt 90 fd 5 lt 90
pd biggear pu
rt 90 fd 5 rt 90 fd 5 rt 180
pd smallgear pu
fd 2 rt 90 fd 7 lt 90
pd  biggear
```



7.8 Challenge 8: Rocket Repair



Location and Time: Earth's moon, 2146AD

Story:

The year is 2146 and this is the space shuttle that first carried human beings to an inhabited planet outside our solar system. Logonator has removed the engines and warpcores from the space shuttle. You need to finish the rocket before take-off time.

Teaching points:

In this challenge you will need to make your own procedures.

One for the <engine>* (rectangle with triangle on the top) and one for the <warpcore>* (pentagons).

You will then need to run these procedures several times to complete the rocket.

There are no goto procedures for you to use so you will need to use the penup (pu) and pendown (pd) commands to move to the correct starting place before running each procedure.

*These names are just examples and not required specifically.

Possible Solution:

warpcore procedure

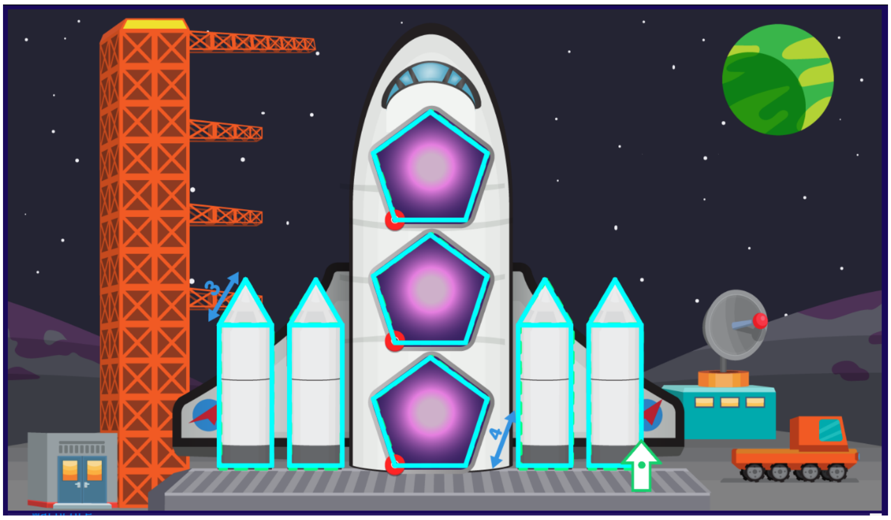
```
LT 90
RPT 5 [rt 72 fd 4]
RT 90
```

engine procedure

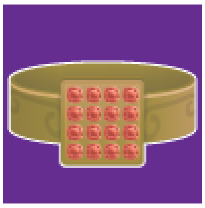
```
RPT 2 [ FD 8 RT 90 FD 3 RT 90]
FD 8 RT 30 FD 3 RT 120 FD 3 RT 30 FD 8 LT 180
```

Solution

```
engine
PU RT 90 FD 1 LT 90 PD
engine
PU RT 90 FD 3 LT 90 PD
warpcore
PU FD 7 PD
warpcore
PU FD 7 PD
warpcore
PU RT 90 FD 7 RT 90 FD 14 RT 180 PD
engine
PU RT 90 FD 1 LT 90 PD
engine
```



7.9 Challenges 9 & 10: Royal Rubies



Location and Time: Hampton Court, 1533 AD

Story:

Logonator's latest dastardly plot is to steal the rubies from Anne Boleyn's wedding ring. She won't be able to marry Henry 8th and give birth to the future Queen Elizabeth I unless her wedding ring is complete.

You must replace the jewels in the ring to get history back on track.

This challenge is split into two parts:

Part 1: Children create a procedure <rowofrubies>* to draw a line of rubies by nesting the procedure for the ruby inside another procedure

Part 2: Children replace the rubies in the ring with several rows of rubies by nesting their <rowofrubies>* procedure within another procedure <ringofrubies>*

*These names are just examples and not required specifically.

7.9.1 Challenge 9 - Part 1 - Ruby Ring

Teaching points:

In this challenge you will need use the <ruby> procedure that has been given to you to make your own procedure for <rowofrubies>*.

This procedure should include the code to draw a ruby, move to the start of the next ruby and draw it again.

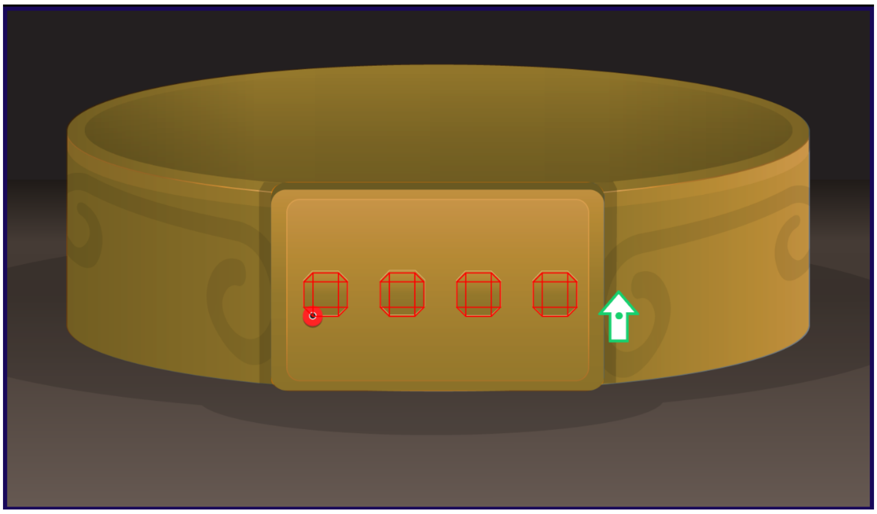
You should use repeat commands so that the procedure draws a whole row of rubies (4 rubies).

You will then need to run this procedure to replace the rubies in the ring.

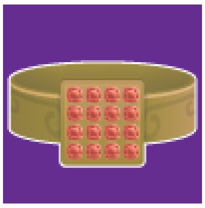
*These names are just examples and not required specifically.

Possible Solution:

RPT 4 [PD ruby RT 90 PU FD 3 LT 90]



7.9.2 Challenge 10 - Part 2 - Ruby Ring



Teaching points:

In this challenge you will need to write a new procedure <ringofrubies>* that uses your <rowofrubies>* procedure.

The procedure should call your <rowofrubies>* procedure then move to the start of the next row and draw the row again.

You should use repeat commands so that the procedure draws a whole ring of rubies (4 rows).

You will then need to run this procedure to replace the rubies in the ring.

*These names are just examples and not required specifically.

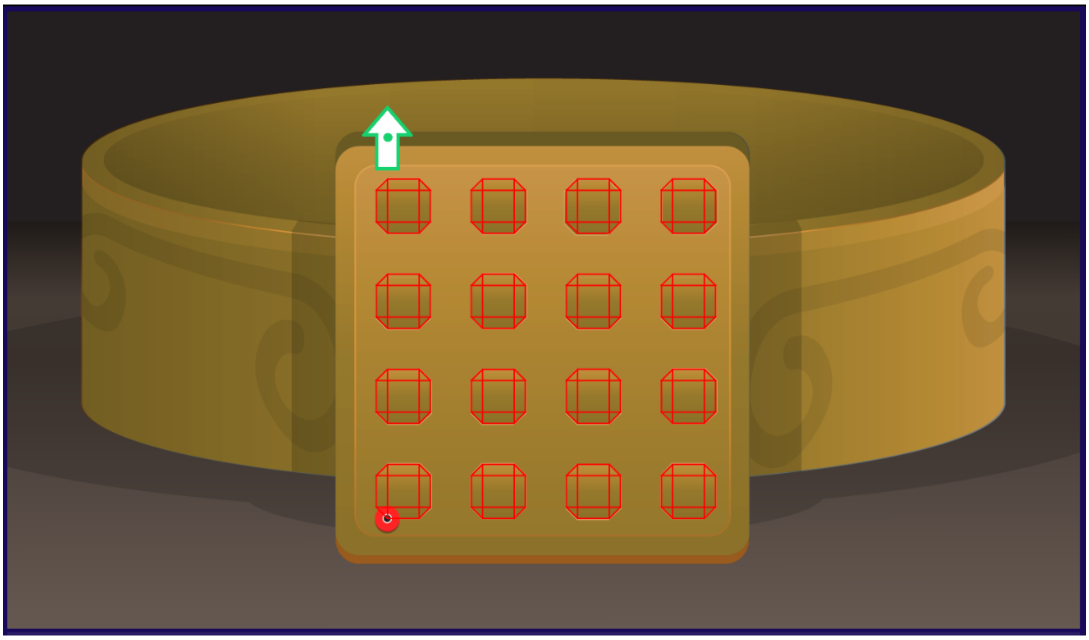
Possible Solution:

rowofrubies* procedure

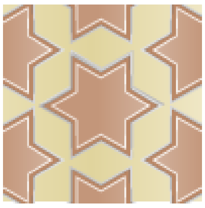
RPT 4 [PD ruby RT 90 PU FD 3 LT 90]

Answer - ringofrubies* procedure

RPT 4 [rowofrubies PU LT 90 FD 12 RT 90 FD 3]



7.10 Challenge 11: Taj Troubles



Location and Time: Agra, India, 1640 AD

Story:

Logonator has demolished part of the most beautiful building in the world; the Taj Mahal. The floor has been destabilized and the building is in danger of collapsing! You must repair it for future generations to enjoy.

Teaching points:

In this challenge you will need to make two procedures; one for a 4-pointed star and one for a six-pointed star.

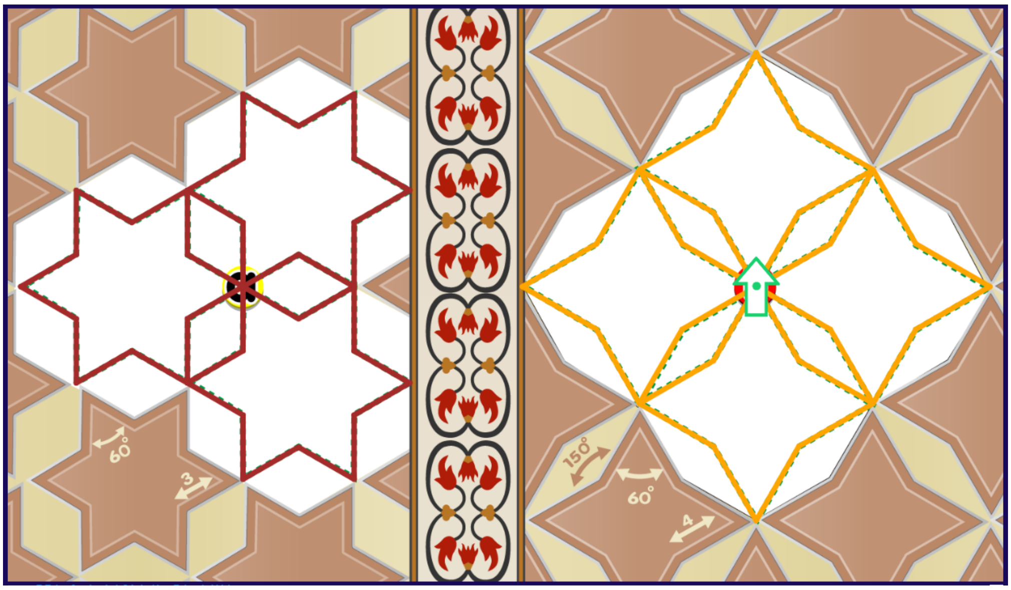
Use the angles marked on the tiles to help you.

You will then need to use a repeat loops to call the procedure and rotate the correct amount.

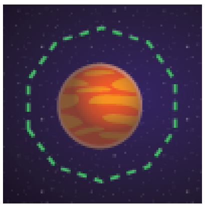
You can use gotoa and gotob to move to the correct part of the floor.

Possible Solution:

4-pointed star procedure	6-pointed star procedure	Solution
RPT 4 [RT 30 FD 4 RT 30 FD 4 LT 150]	RPT 6 [FD 3 LT 60 FD 3 RT 120]	gotoa RPT 3 [sixstar RT 120] gotob RPT 4 [fourstar RT 90]



7.11 Challenge 12: Protect the Planets



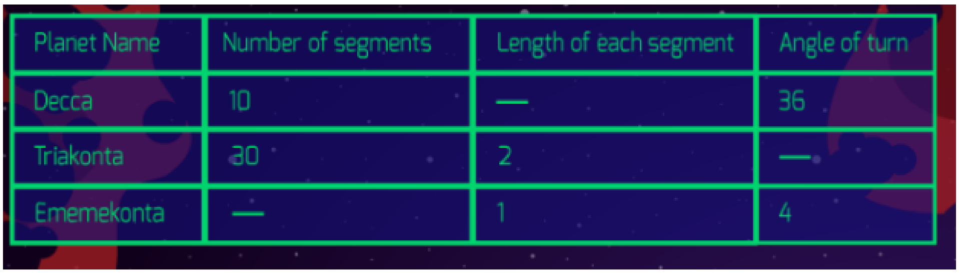
Location and Time: A galaxy far, far away, 2387

Story:

In 2387, the future human beings have spread out across the galaxy and live on many different planets. Logonator is planning an attack on one of these defenceless solar systems. You must create force-fields around the planets to protect them from his attack and save the colonists.

Teaching points:

You need to draw the force fields around the planets using the blueprints on the screen:



Planet Name	Number of segments	Length of each segment	Angle of turn
Decca	10	—	36
Triakonta	30	2	—
Ememekonta	—	1	4

To draw shapes approaching circles in Logo you need to more forward a small amount and rotate a small amount many times.

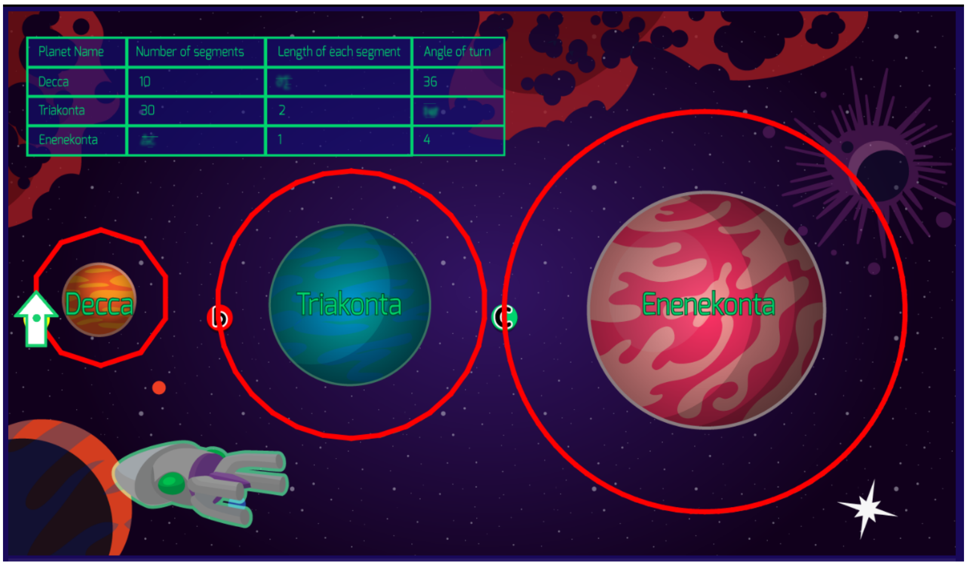
Use the following structure and the information given in the blueprint and experiment with the missing amounts using the structure RPT _ [FD _ RT _].

The number of repeats multiplied by the angle to turn each time must equal 360 so you can use your maths skills to help you.

You can use the goto procedures to move between the different orbits.

Possible Solution:

Decca procedure	**Triakonta procedure**	**Enenekonta procedure**	**Answer**
rpt 10[fd 3 rt 36]	rpt 30[fd 2 rt 12]	rpt 90[fd 1 rt 4]	gotoa Decca gotob Triakonta gotoc Enenekonta



Planet Name	Number of segments	Length of each segment	Angle of turn
Decca	10		36
Triakonta	30	2	
Enenekonta		1	4

7.12 Challenge 13: The Amazing Amazon

Location and Time: Amazon Rainforest, Present day

Story:

Back to the present day, but Logonator isn't finished yet. The Amazon rain forest contains numerous plants that can provide medicines for all kinds of diseases. Logonator is trying to destroy the flowers that will eventually provide a cure for a deadly bacteria that could wipe out humanity. You must grow the flowers again so that they can be used to make an antidote.

Teaching points:

Each flower head is made using a procedure that draws a part of the flower e.g. <daisypetal>* <poppycircle>* and then rotating and repeating this procedure to draw the whole flower.

Use the images on the background to create the initial procedures <daisypetal>* <poppycircle>* and then work out how to rotate and repeat these.

Write a procedure for each type of flower e.g. <daisy>* <poppy>*

In the procedures, write some Logo to draw the stem using the marked lengths then repeat and rotate the <daisypetal>* or <poppycircle>* proceedure the required number of times.

Experiment with the setpc commands to colour your flowers and stems.

Use the goto procedures to move to the bases of each flower in turn.

*These names are just examples and not required specifically.

Possible Solution:

poppy circle	daisy petal	poppy procedure	daisy procedure
RPT 10 [FD 2 RT 36]	RPT 6 [FD 2 RT 15] RT 90 RPT 6[FD 2 RT 15]	setpc green fd 17 setpc purple RPT 8 [poppycircle rt 45]	setpc green fd 27 setpc yellow RPT 6 [daisypetal RT 90 RT 60]

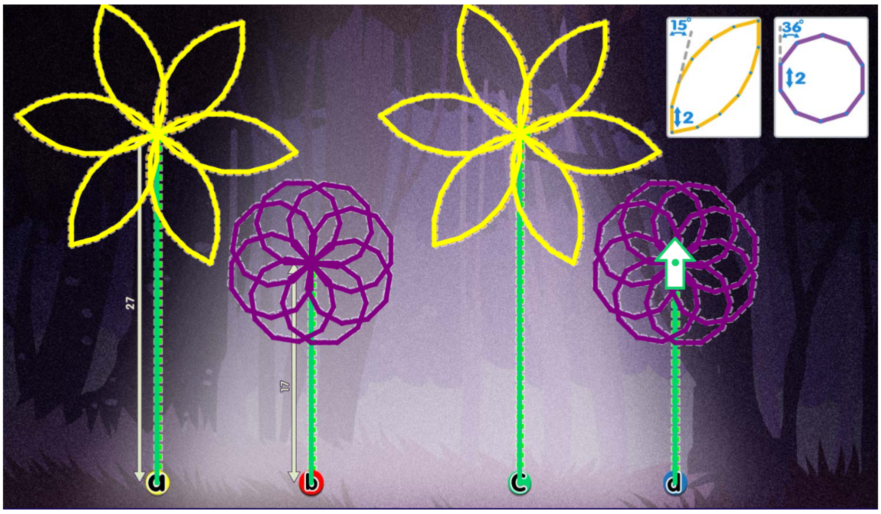
Solution

gotoa pd daisy pu

gotob pd poppy pu

gotoc pd daisy pu

gotod pd poppy pu


15°
2
36°
2
27
17
a
b
c
d