Computing Scheme of Work
Y2 Coding Crash Course Lesson 3

Lesson 3:

Algorithms

Aims

- To understand what an **algorithm** is.
- To create a computer program that follows an **algorithm**.

Success Criteria

- Children can explain that an **algorithm** is a set of instructions.
- Children can describe the **algorithms** they have created.
- Children can explain that for the computer to make something happen, it needs to follow clear instructions.

Vocabulary

- Have a go at the Coding Vocabulary Quiz.
- What can you remember?

Algorithms

- In lesson 1, your teacher was a programmer and you were all robots that had to follow instructions that were written as symbols.





- In computing, a set of instructions is called an

Algorithm

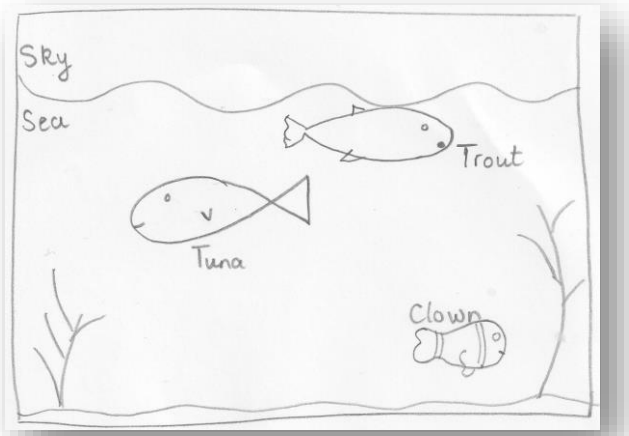
Need more support? Contact us:
Tel: +44(0)208 203 1781 | Email: support@2simple.com | Twitter: @2simplesoftware

Algorithms

An **algorithm** is a precise step-by-step set of instructions used to solve a problem or achieve an objective.

- The **algorithm** is **not** the same thing as the code.

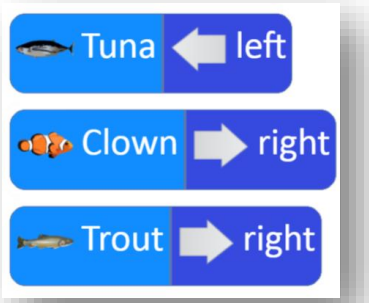
- This is the **design**.



- This is the **algorithm**.

1. Tuna should swim left.
2. Clown fish should swim right.
3. Trout should swim right.

- This is how the algorithm could be represented in **code** using 2Code.



- In other programming languages, the code to program the same algorithm would look different. Here is the same algorithm programmed in a language called JavaScript.

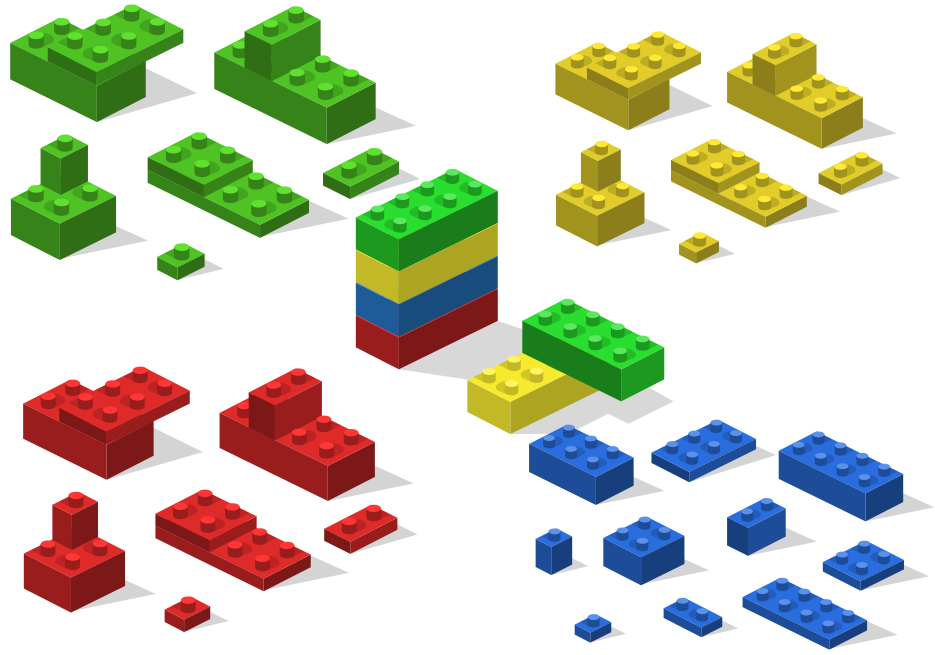
```
Tuna.left();
Clown.right();
Trout.right();
```

Activity 1: Lego Models

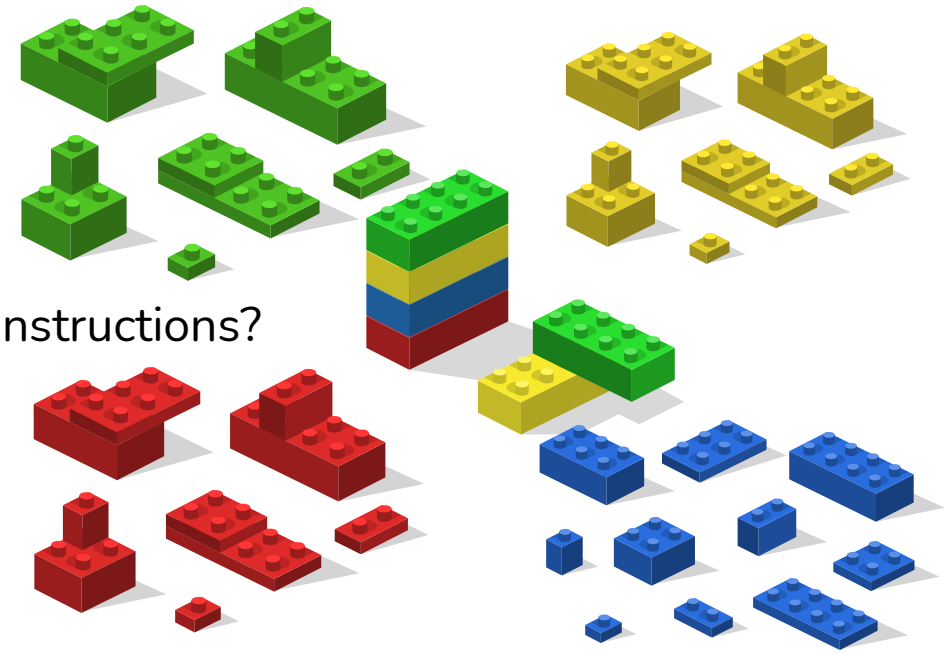
- Look at the two models.
- Which is correct?
- The correct answer could be,

  ‘They are both correct.’

  How can we know which is correct?’
- There is no such thing as correct when building creatively.
- You might ***prefer*** one over the other, but both are ‘correct’.



Activity 1: Lego models



- Look at the two models again.
- Now, look at the instructions.
- Which model is built the correct way now you have seen the instructions?

- Now, you ***can*** answer the question.
- The step-by-step instructions are the **algorithm** for building the model.

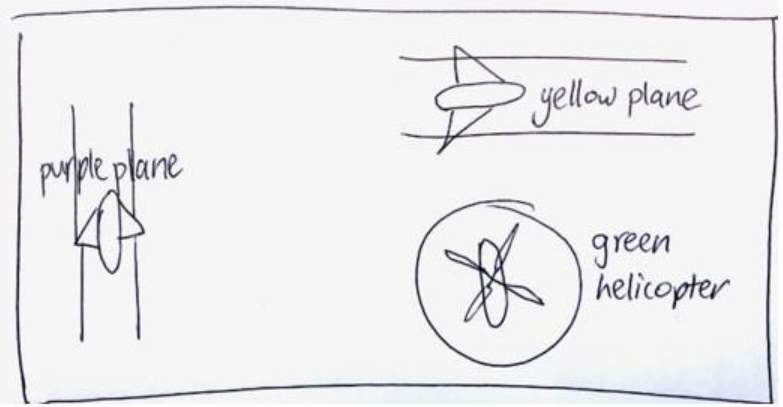
- Why is it important to follow the steps in the **algorithm** in order? Now, look at the instructions.
- What would happen if we completed the **algorithm** in a different order?

Making a Computer Program

- When making a computer program a coder will:
  - **Design** a the visual look of the program.
  - Write the **algorithm** that would make the program accomplish its aim.
  - Create the design on the computer.
  - Write **code** that instructs the computer in line with the **algorithm** and make the program work.
- Can you **predict** what will happen when this program is **run**?
- Which plane will take off first?
- What are the **events**, **objects** and **actions** you would need in 2Code to program this algorithm?

- This shows the **design** and the **algorithm**.

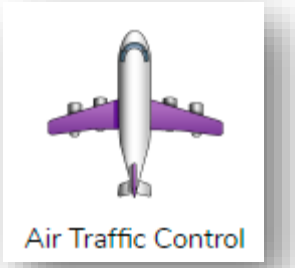
Task: To make an airport program where the planes take off.



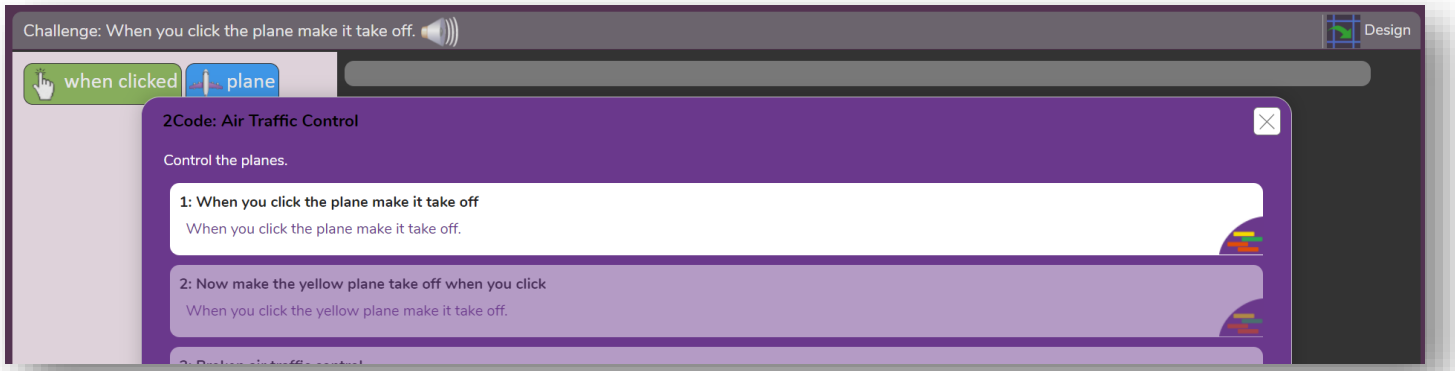
1) Click purple plane to make it take off.
2) Click yellow plane to make it take off.
3) Click green helicopter to make it take off.
4) If green helicopter crashes with yellow plane; make crashing sound.

5) Optional step: If green helicopter crashes with yellow plane make it change direction

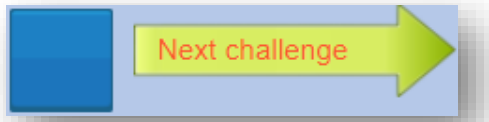
Activity: Air Traffic Control



- Open Air Traffic Control, click on stage 1, watch the video and add code for stage 1 together.



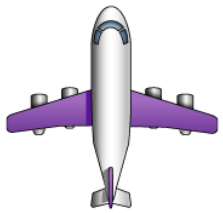
- What is the **event**, **object**, **action**?
- Click on run  to test the **code** and see if it works.
- Click on stop and then **run** again, notice how the **code** goes orange when it **executes**.
- Click on the plane and notice what happens – which bit goes orange?
- Remember to click on stop to make changes or Next Challenge when you have been successful.



Need more support? Contact us:
Tel: +44(0)208 203 1781 | Email: support@2simple.com | Twitter: @2simplesoftware

Activity: Air Traffic Control

- Have a go.



2Do: Air Traffic Control

Program the planes to take off.

Set by: Mrs Runham

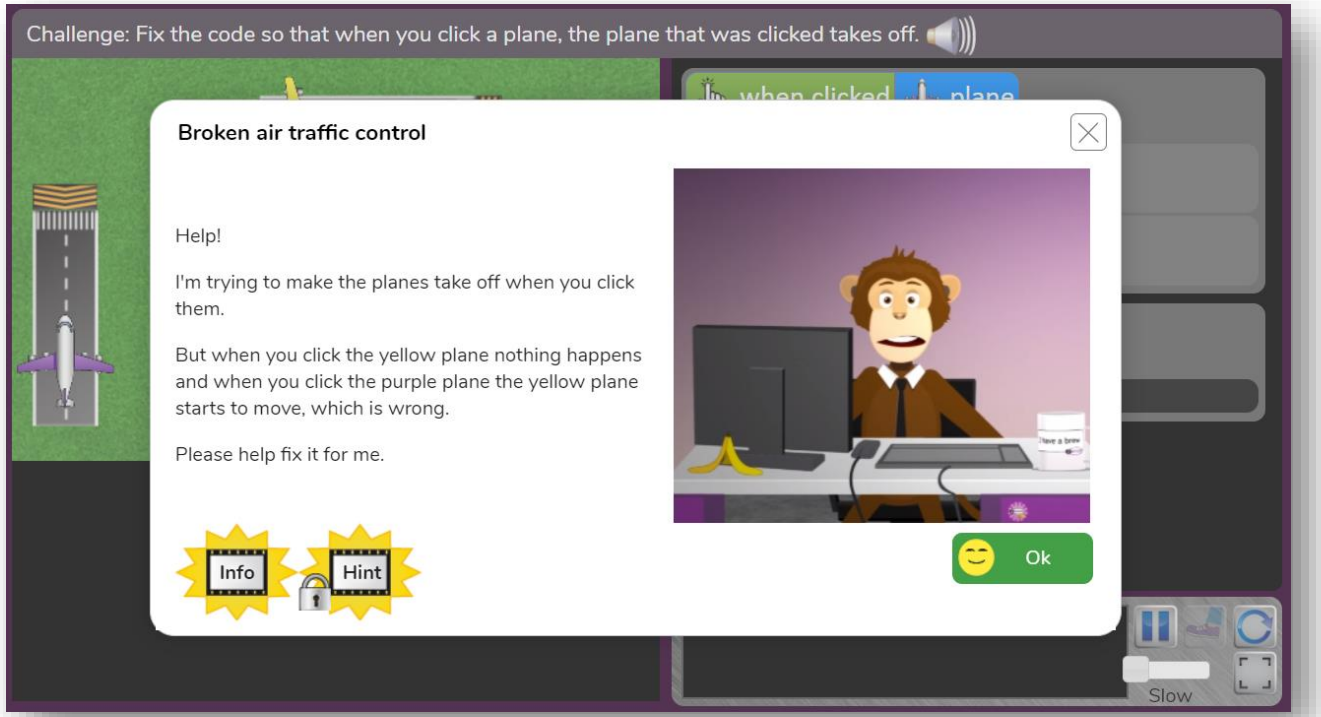
No due date

Start

- Open Air Traffic Control from your 2Dos and complete Stages 1 and 2.
- Notice how the **code executes** when you click on **run**  .

Activity: Air Traffic Control

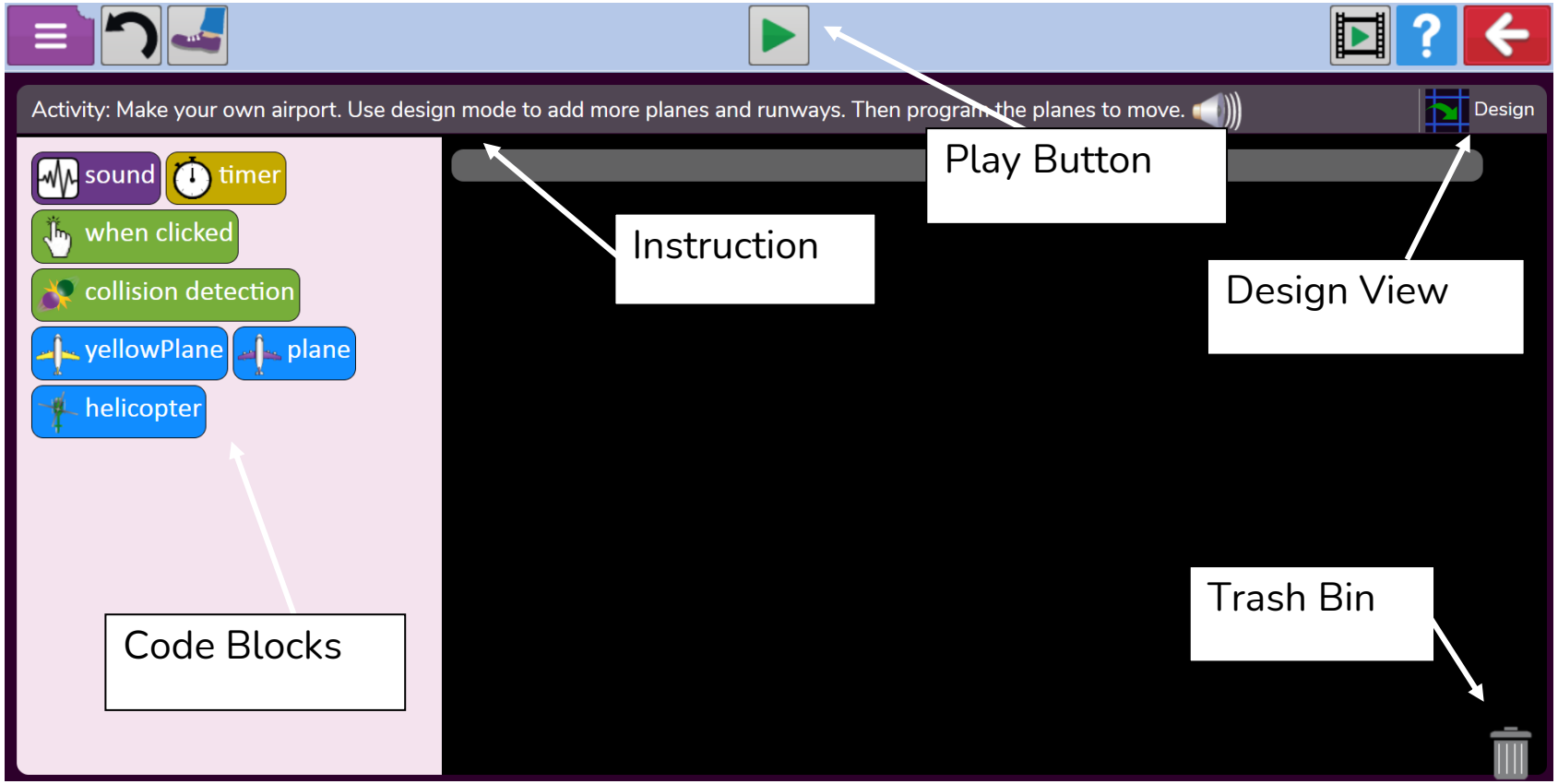
- In Stage 3, the program does not work the way the Coder wants it to.
- We need to **debug** the code.
- What is **debugging?**



- Can you help the Code Monkey fix the code that isn't working properly?

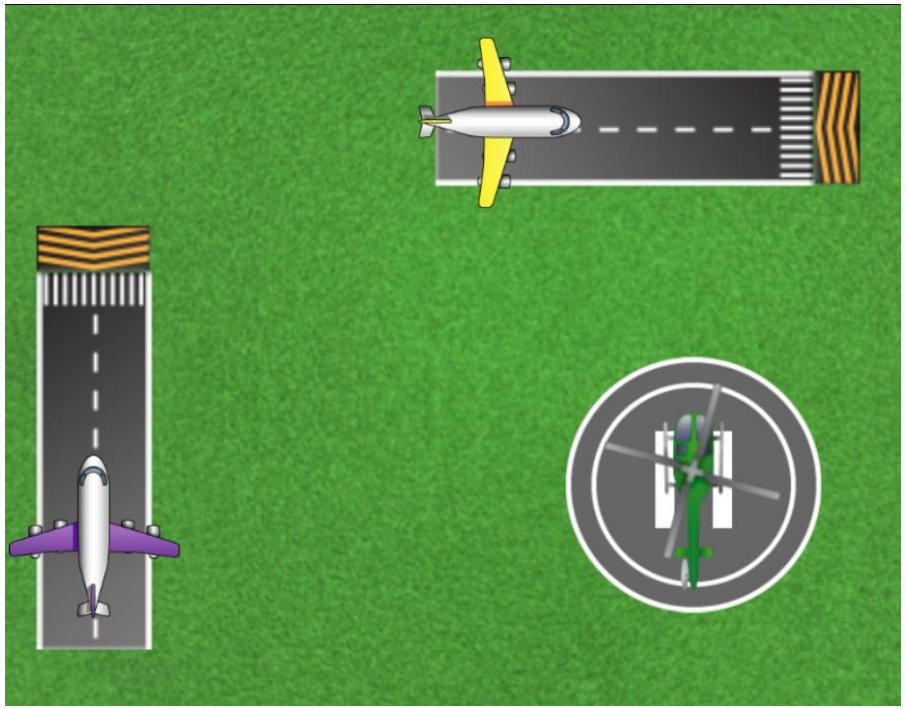
Air Traffic Control

- Can you explain what all these different parts of the code view do?

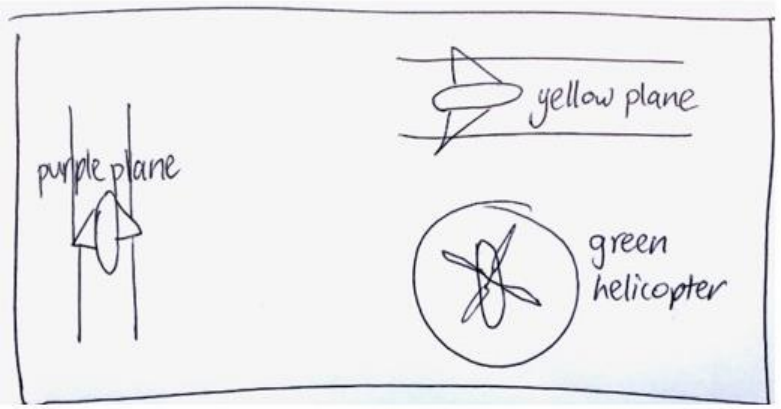


Air Traffic Control

- This is the ‘Design’ for Air Traffic Control final stage. This is the **scene** that has been made in 2Code.
- What is the **background** and what are the **objects**?



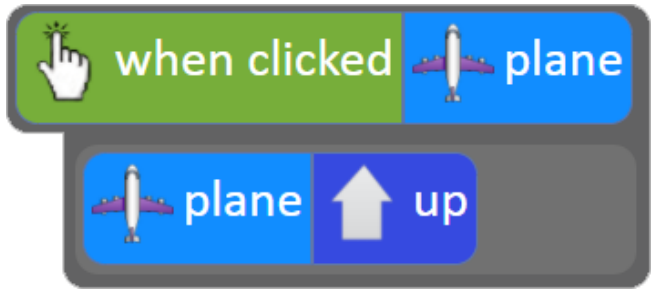
- The grass is the **background**.
- The aircraft and runways are the **objects**.



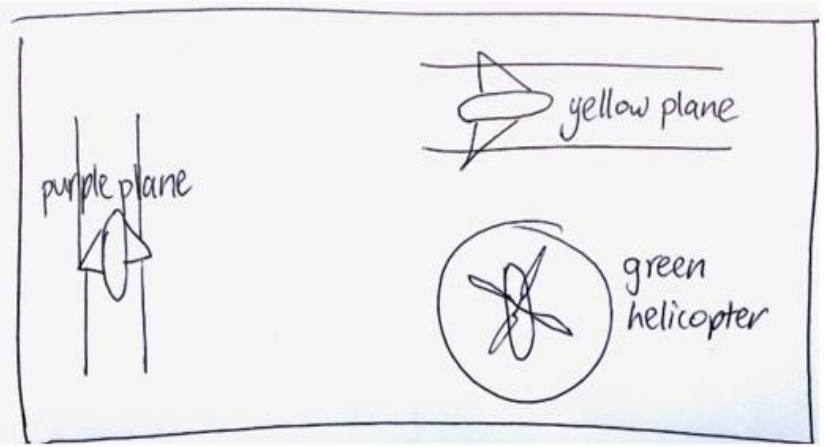
- Remember our program on slide 9: Is it like the 2Code design?

Algorithm → Code

- The task outlines what the program should do.
- The **algorithm** gives us the step by step instructions for how to achieve this.
- We need to program this algorithm using code.
- Look at the algorithm.
- It starts with 3 **click events**.
- Open the [final stage of Air Traffic Control] and add the code for the first one together:



Task: To make an airport program where the planes take off.

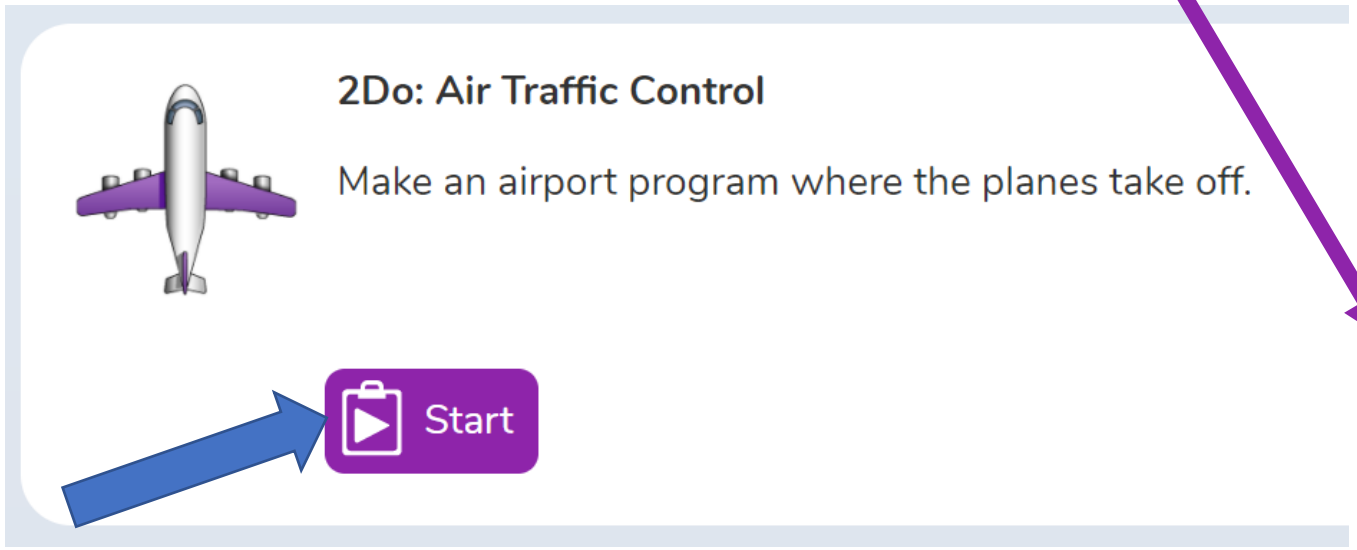


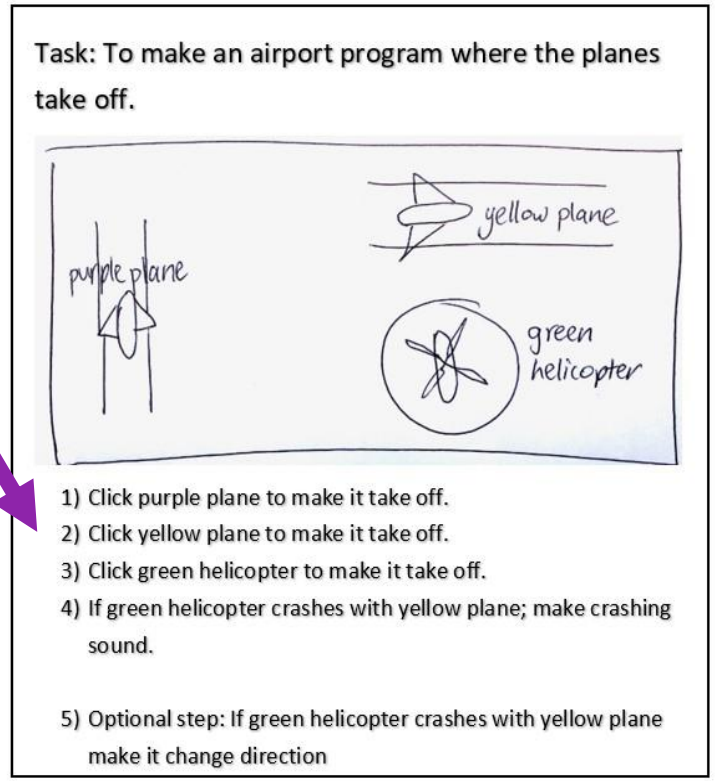
1) Click purple plane to make it take off.
2) Click yellow plane to make it take off.
3) Click green helicopter to make it take off.
4) If green helicopter crashes with yellow plane; make crashing sound.

5) Optional step: If green helicopter crashes with yellow plane make it change direction

Activity 2: Air Traffic Control

- Continue through the Air Traffic Control 2Do until you get to the final stage.
- In the final stage, have a go at adding the correct code to make the first 3 steps of the **algorithm** work.



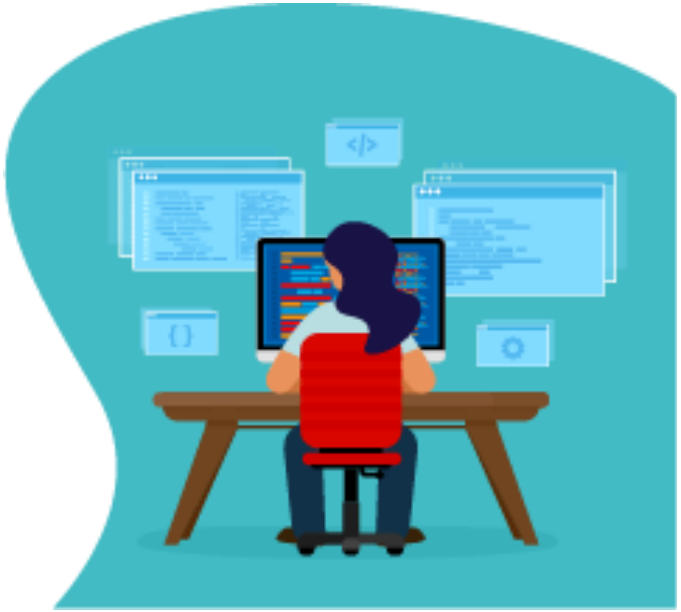


Saving Your Code

Why is saving important?

- If you accidently close the program, you will lose your work if you haven't saved.
- Save each time you make a change and test it.
- You will always have a good version to go back to.

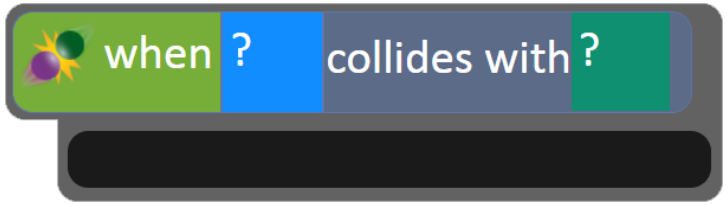
How do you save? Save your program now.



Air Traffic Control

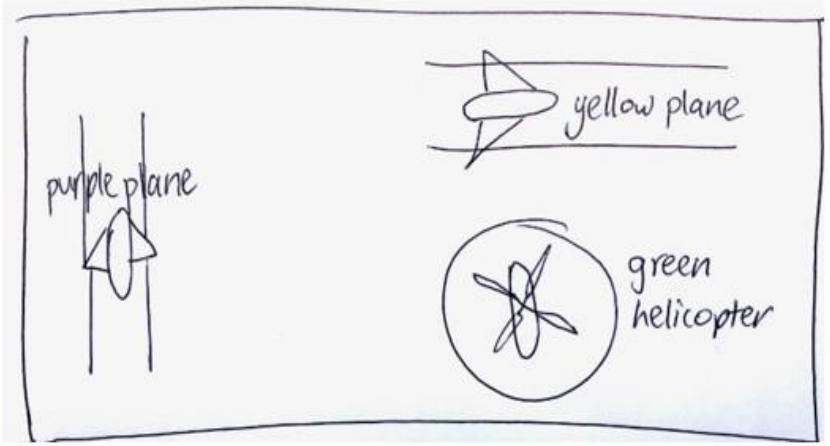
- Read step 4 of the **algorithm**.
- How will the computer know when the helicopter has **collided** with the yellow plane?
- What **event** will we need?

  Collision detection.



- What do we want to happen when the helicopter hits the yellow plane?

Task: To make an airport program where the planes take off.

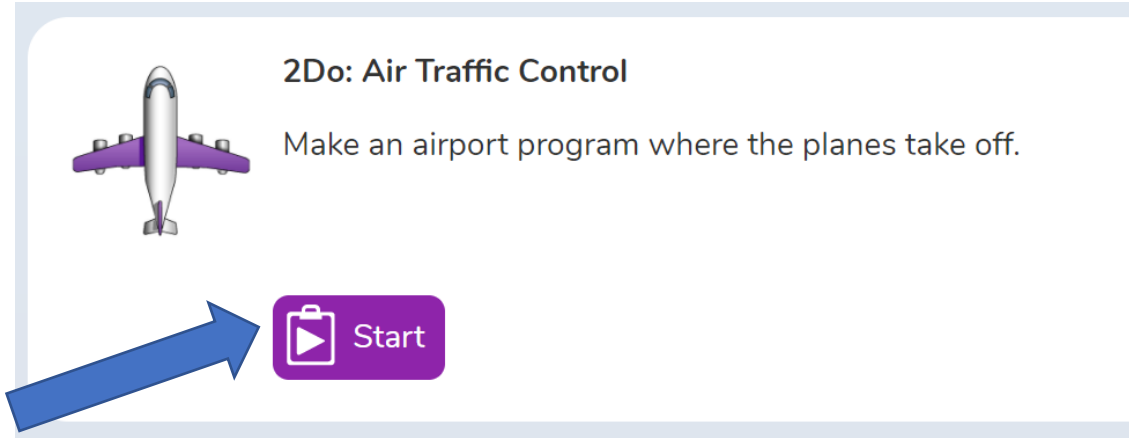


1) Click purple plane to make it take off.
2) Click yellow plane to make it take off.
3) Click green helicopter to make it take off.
4) If green helicopter crashes with yellow plane; make crashing sound.

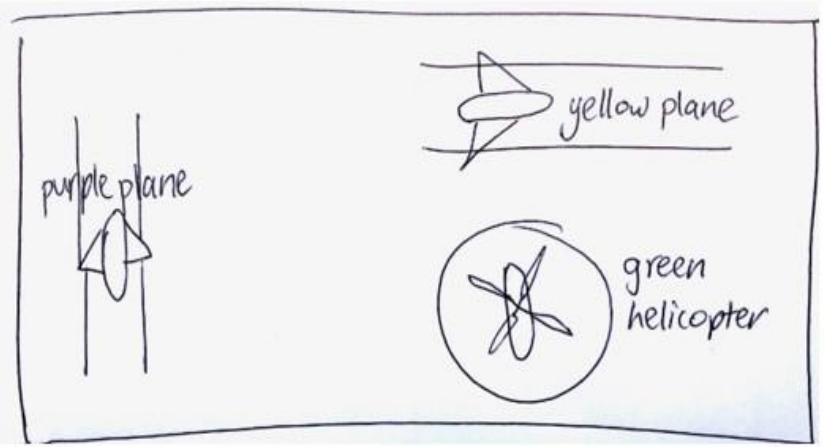
5) Optional step: If green helicopter crashes with yellow plane make it change direction

Extension: Air Traffic Control

- Have a go at adding the correct code to make the 4th step of the **algorithm** work.
- Can you program the optional final step?
- If you finish, see if you can add more **objects** to the **scene** and further develop the program.



Task: To make an airport program where the planes take off.

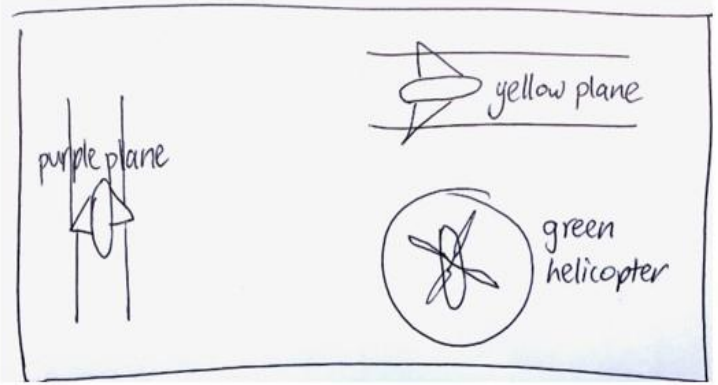


1) Click purple plane to make it take off.
2) Click yellow plane to make it take off.
3) Click green helicopter to make it take off.
4) If green helicopter crashes with yellow plane; make crashing sound.

5) Optional step: If green helicopter crashes with yellow plane make it change direction

How Did You Get On?

Task: To make an airport program where the planes take off.



1) Click purple plane to make it take off.
2) Click yellow plane to make it take off.
3) Click green helicopter to make it take off.
4) If green helicopter crashes with yellow plane; make crashing sound.

5) Optional step: If green helicopter crashes with yellow plane make it change direction

- Did you manage to turn the **algorithm** into code?
- Did you have time to further develop the airport program?
- Work with a partner, read through each other's code and **predict** what will happen when you **run** the program.
- Save your final program and hand in your 2Do.

Need more support? Contact us:
Tel: +44(0)208 203 1781 | Email: support@2simple.com | Twitter: @2simplesoftware

Review Success Criteria

- I can explain that an **algorithm** is a set of instructions.
- I can describe the **algorithms** I created.
- I can explain that for the computer to make something happen, it needs to follow clear instructions.