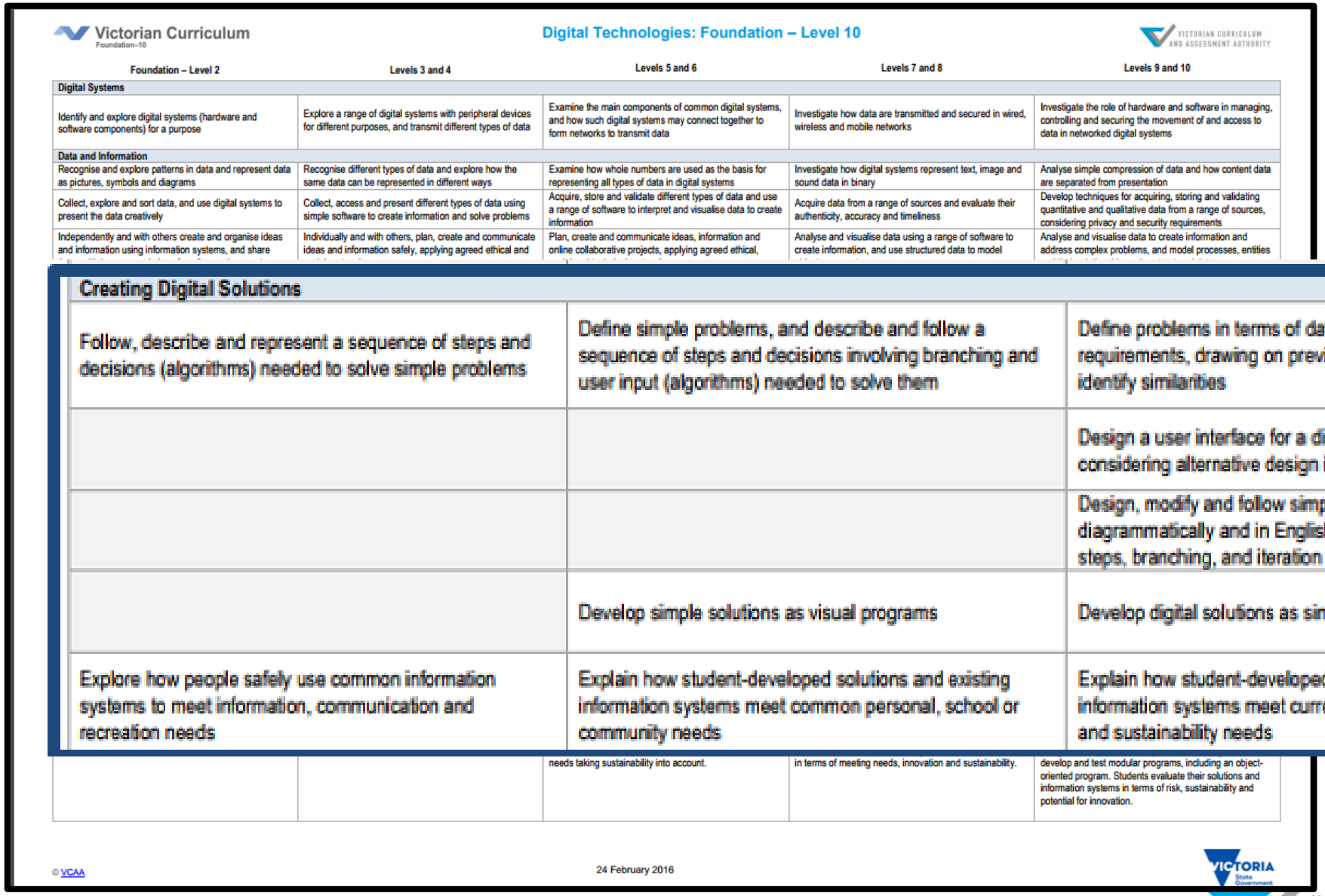
The Digital Technologies Curriculum Creating Digital Solutions F-6

Megan van der Velden
VCAA Digital Coding Specialist Teacher

VICTORIAN CURRICULUM AND ASSESSMENT AUTHORITY
VICTORIA State Government

Scope and Sequence



Victorian Curriculum
Foundation–10

Digital Technologies: Foundation – Level 10

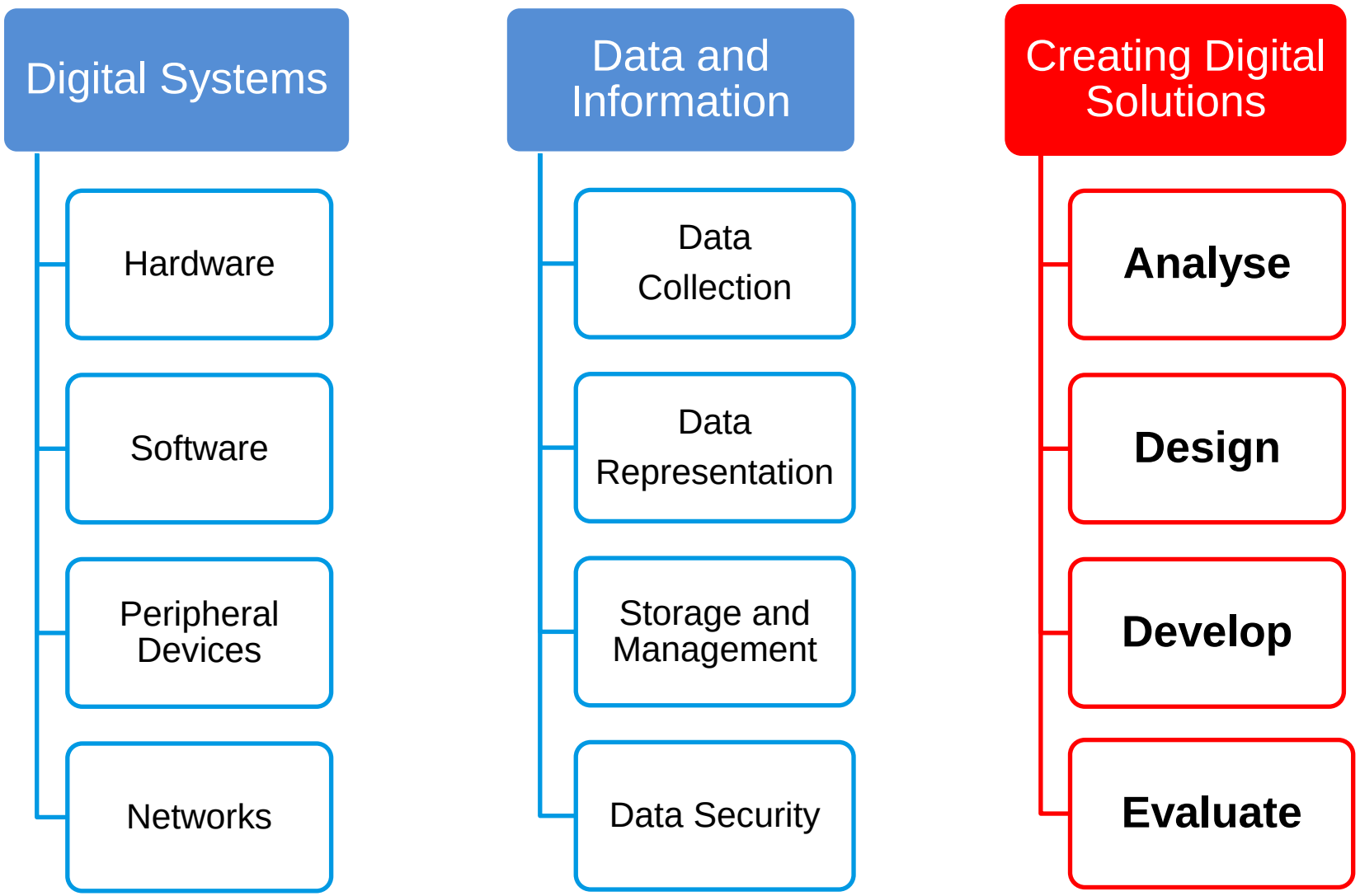
VICTORIAN CURRICULUM AND ASSESSMENT AUTHORITY

Foundation – Level 2	Levels 3 and 4	Levels 5 and 6	Levels 7 and 8	Levels 9 and 10
Digital Systems				
Identify and explore digital systems (hardware and software components) for a purpose	Explore a range of digital systems with peripheral devices for different purposes, and transmit different types of data	Examine the main components of common digital systems, and how such digital systems may connect together to form networks to transmit data	Investigate how data are transmitted and secured in wired, wireless and mobile networks	Investigate the role of hardware and software in managing, controlling and securing the movement of and access to data in networked digital systems
Data and Information				
Recognise and explore patterns in data and represent data as pictures, symbols and diagrams	Recognise different types of data and explore how the same data can be represented in different ways	Examine how whole numbers are used as the basis for representing all types of data in digital systems	Investigate how digital systems represent text, image and sound data in binary	Analyse simple compression of data and how content data are separated from presentation
Collect, explore and sort data, and use digital systems to present the data creatively	Collect, access and present different types of data using simple software to create information and solve problems	Acquire, store and validate different types of data and use a range of software to interpret and visualise data to create information	Acquire data from a range of sources and evaluate their authenticity, accuracy and timeliness	Develop techniques for acquiring, storing and validating quantitative and qualitative data from a range of sources, considering privacy and security requirements
Independently and with others create and organise ideas and information using information systems, and share	Individually and with others, plan, create and communicate ideas and information safely, applying agreed ethical and	Plan, create and communicate ideas, information and online collaborative projects, applying agreed ethical,	Analyse and visualise data using a range of software to create information, and use structured data to model	Analyse and visualise data to create information and address complex problems, and model processes, entities
Creating Digital Solutions				
Follow, describe and represent a sequence of steps and decisions (algorithms) needed to solve simple problems		Define simple problems, and describe and follow a sequence of steps and decisions involving branching and user input (algorithms) needed to solve them		Define problems in terms of data and functional requirements, drawing on previously solved problems to identify similarities
				Design a user interface for a digital system, generating and considering alternative design ideas
				Design, modify and follow simple algorithms represented diagrammatically and in English, involving sequences of steps, branching, and iteration
		Develop simple solutions as visual programs		Develop digital solutions as simple visual programs
Explore how people safely use common information systems to meet information, communication and recreation needs		Explain how student-developed solutions and existing information systems meet common personal, school or community needs		Explain how student-developed solutions and existing information systems meet current and future community and sustainability needs
		needs taking sustainability into account.	in terms of meeting needs, innovation and sustainability.	develop and test modular programs, including an object-oriented program. Students evaluate their solutions and information systems in terms of risk, sustainability and potential for innovation.

VICTORIA State Government
VICTORIAN CURRICULUM AND ASSESSMENT AUTHORITY

VICTORIA State Government

Strands


Digital Systems
Hardware
Software
Peripheral Devices
Networks
Data and Information
Data Collection
Data Representation
Storage and Management
Data Security
Creating Digital Solutions
Analyse
Design
Develop
Evaluate

Key Concepts
and
Ways of Thinking

Key Concepts

Concept	Definition
Abstraction	Hiding details not directly relevant, allows for solutions to be transferred across contexts.
Data Collection	Creating information and utilising it in different ways to extract meaning
Specification, algorithms and development	Sequential and detailed instructions, leads to developing coded solutions
Digital Systems	Connected hardware, software and networks, and methods of communication
Interactions and impacts	How people actually interact with technology and the effect on society and the environment

Computational thinking

A **problem-solving** method that involves various techniques and strategies in order to solve problems that can be implemented by digital systems, such as:

- organising data logically
- breaking down problems into components
- the design and use of algorithms, patterns and models

Design thinking

Purposeful use of strategies for:

- understanding **design problems** and **opportunities**
- **visualising and generating** creative and innovative ideas
- **analysing and evaluating** those ideas that best meet the criteria for success and planning

Systems thinking

A holistic approach to the **identification** and **solving** of problems where

- **parts** and **components** of a **system** are analysed
- **interactions** and **interrelationships** between **information system components** (data, processes, people and digital systems) are analysed to see how they influence the functioning of the whole system
- students **understand systems** and work with **complexity, uncertainty and risk**

Creating Digital Solutions strand

Creating Digital Solutions

Creating Digital Solutions requires:

- skills in using digital systems
- computational, design and systems thinking
- interacting safely by using appropriate technical and social protocols

Problem-solving methodology – Levels F - 6

Analysing

- What is the problem that needs to be solved?

Designing

- The user interface and algorithms

Developing

- The solution development

Evaluating

- Determining if the problem has been solved.

Algorithmic development

F - 2
3 - 4

Linear - one path to take
User input - step through algorithm in response to user
Branching - user input or another condition selects a different set of instructions

5 - 6

Iteration - repeating part of the algorithm a set number of times or until a condition is fulfilled

7 - 8

Functions - discrete group of instructions that are called to action in defined conditions

9 - 10

Modular - reaching out to another set of instructions that have a more specific purpose or focus (methods)

Resource: Steve Allen

Creating Digital Solutions – Foundation - Level 2

Content Description:

- Follow, describe and represent a sequence of steps and decisions (**algorithms**) needed to solve **simple problems**

- Explore how people safely use common information systems to meet information, communication and recreation needs.

Achievement Standard:

- Students design solutions to simple problems using a sequence of steps and decisions.

Foundation - Level 2

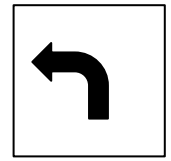
Possible Problem

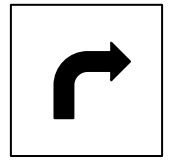
Using the arrow squares, show the steps needed for a Bee-Bot to take to follow a square shape:

Let's look at the problem in a bit more detail (Analyse)

The BeeBot needs to be programmed to move in a square shape

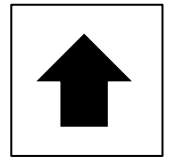
- The steps need to be broken down and shown using some pre-prepared direction cards





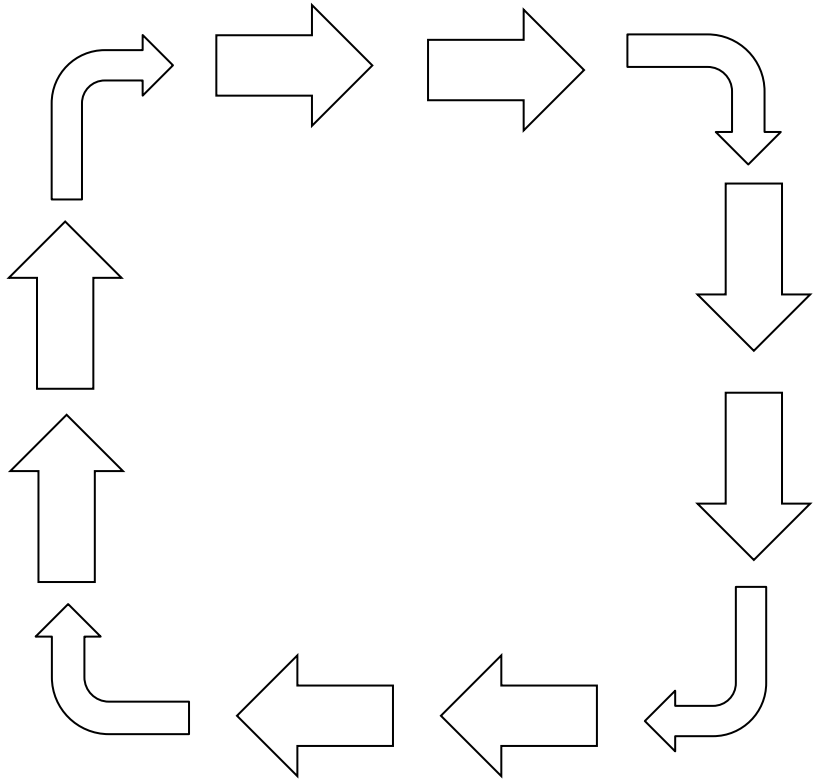
What do we need to know?

- What are the attributes of a square?
- Directions such as forwards, backwards, right turn and left turn



Developing

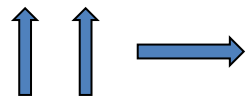
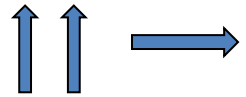
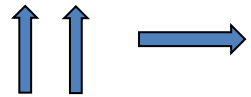
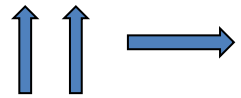
With the cards, a student might develop a solution such as this.



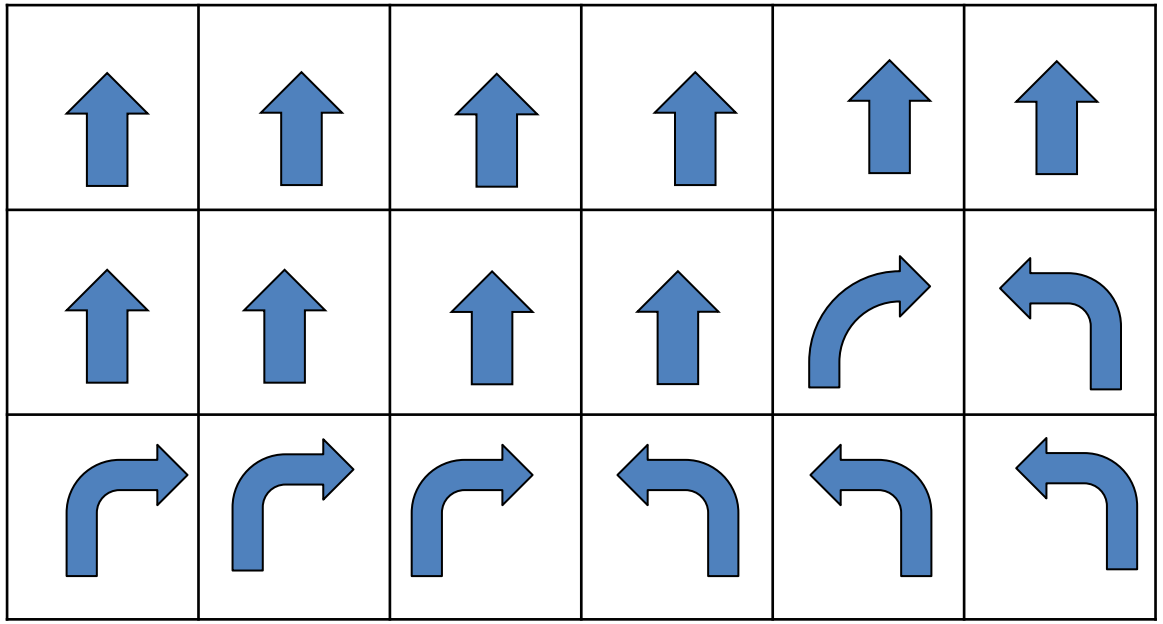
But these instructions are not easy to give to another person to follow – if we have to pack up, then the solution would have to be made all over again.

Students need to consider how they will represent their series of steps

Developing, representing and evaluating

Step 1	
Step 2	
Step 3	
Step 4	

Draw the solution in a table



Cut and paste the arrows

Does it work?

Achievement Standard **Extract** – Level 2

Students design solutions to simple problems using a sequence of steps and decisions.

Creating Digital Solutions – Levels 3 and 4

Content Descriptions

- Define simple problems, and describe and follow a sequence of steps and decisions involving **branching** and **user input** (algorithms) needed to solve them
- Develop simple solutions as **visual programs**
- Explain how **student-developed** solutions and existing **information** systems meet **common personal, school or community needs**

Achievement Standard

- Students define simple problems, and design and develop digital solutions using algorithms that involve decision-making and user input. They explain how their developed solutions and existing information systems meet their purposes.

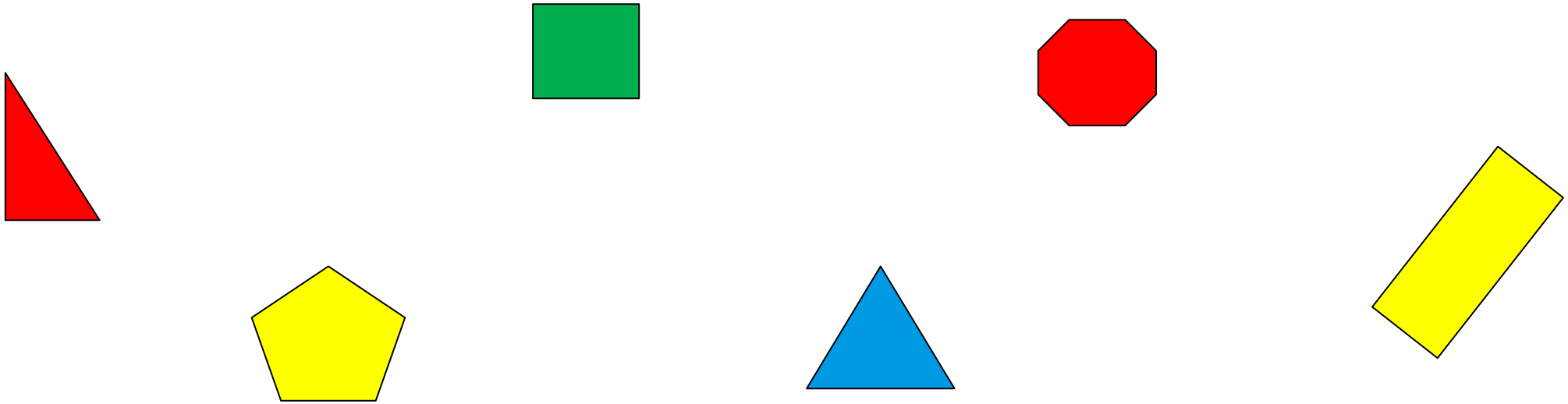
Visual programming – Glossary definition

A programming language or environment where the program is represented and created visually rather than as text. A common visual metaphor represents statements and control structures as blocks that can be composed to form programs, allowing programming without having to deal with syntax errors. Examples of visual programming languages include: Alice, GameMaker, Kodu, Lego Mindstorms, MIT App Inventor, Scratch (Build Your Own Blocks and Snap).

- *Note:* A visual programming language should not be confused with programming languages for creating visualisations or programs with user interfaces, for example, PowerPoint or Visual Basic.

Problem:

The Year 2 students have been learning about 2D shapes. Your task is to use Scratch to make a simple activity that will draw a 2D shape based on the number of sides the student has entered. Think about how you can reinforce to the students the shape that the number of sides has made for them.



Analyse

What is the problem asking?

- For the sprite (the character in Scratch) to draw a **2D shape** in response to **user input** information of the number of sides
- What do we need to know? The attributes of 2D shapes – how many sides, how many angles and the degrees of the angles for each shape
- Reinforce the **name of the shape** they have made
- The Digital Solution should be **easily** operated by a **Year 2 student**

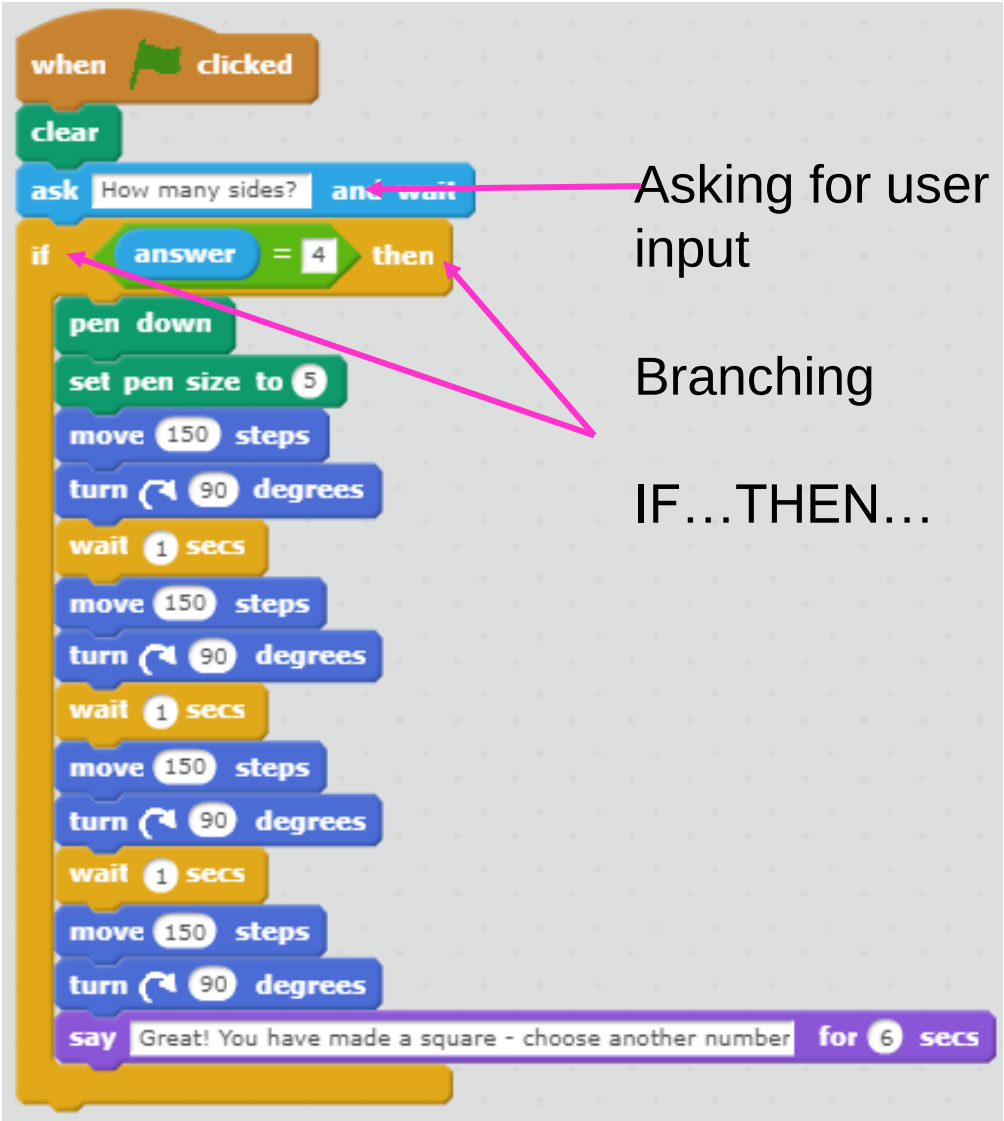
Design and develop

Algorithm - First shape – A square

When green flag is clicked
Pen down
Ask user to input number of sides
IF user input = 4 THEN
    Move 150 steps
    Turn 90 degrees
    Wait
    Move 150 steps
    Turn 90 degrees
    Wait
    Move 150 steps
    Turn 90 degrees
    Wait
    Move 150 steps
    Turn 90 degrees
    say Great! You have made a square – choose another number

Implement the **algorithm** created in the design stage into a **visual programming language** –

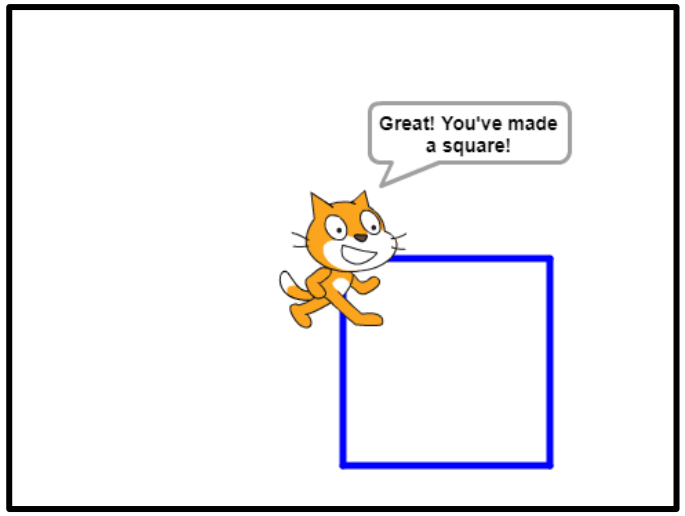
This level includes the addition of **user input** and **branching**

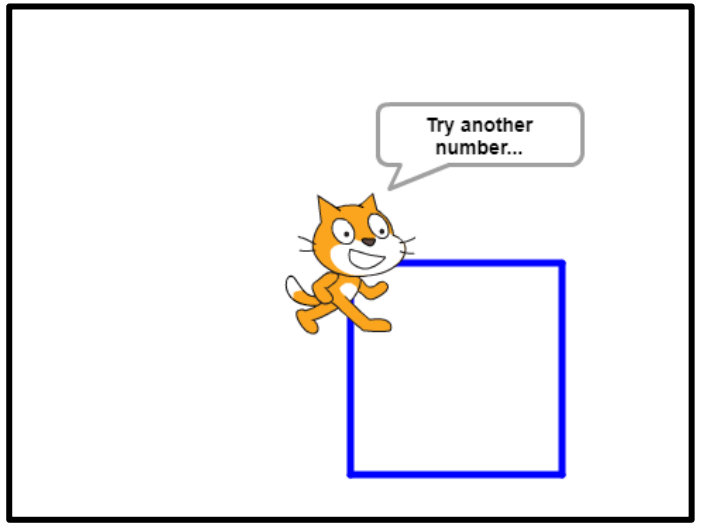


Evaluate

So has the program solved the problem?

- If the user enters the number 4,
- THEN the sprite draws a square on the stage
- Names the shape and encourages the user to try a different number





Achievement Standard Extract Level 4

Students define simple problems, and design and develop digital solutions using algorithms that involve **decision-making** and **user input.**

They explain how their developed solutions and existing information systems meet their purposes.

Creating Digital Solutions – Levels 5 and 6

Content Descriptions

- Define problems in terms of data and **functional requirements**, **drawing on previously solved problems to identify similarities**
- Design a **user interface** for a digital system, generating and considering **alternative design ideas**
- Design, modify and follow simple algorithms represented **diagrammatically** and in English, involving sequences of steps, branching, and **iteration**
- Develop simple solutions as visual programs
- Explain how student-developed solutions and existing information systems meet current and future community and sustainability needs

Creating Digital Solutions – Levels 5 and 6

Achievement Standard Extract Level 6

Students define problems in terms of data and **functional requirements** and **design solutions by developing algorithms** to address the problems. They incorporate **decision-making, repetition and user interface** design into their designs and develop their digital solutions, including a **visual program**. Students explain how information systems and their developed solutions **meet current and future needs** taking sustainability into account.

Creating Digital Solutions – Levels 5 and 6

Problem:

The year 2 students have been learning about 2D shapes. You are to create a program that will test their knowledge of 2D shapes.

Your program needs to respond to user input and let the user know if they are right or wrong. If the student has answered correctly, the shape should be drawn on the stage.

Your design and solution should show that you have considered how the user will interact with the finished product and the suitability of your solution for the end user.

Remember, you are designing and developing a solution for a Year 2 student…

Analyse

Functional requirements?

- Ask a question
- Wait for a response input by a user
- Tell the user if their answer is correct or incorrect
- Draw the shape
- Encourage the user to keep playing

User interface

- Suitable for use by a Year 2 student
  - Game environment
  - Operability

Draw on previously solved solution

- This solution is at level 3-4 but can form the basis of a solution for level 5-6 – we will modify this as we progress

Previous solution Levels 3-4

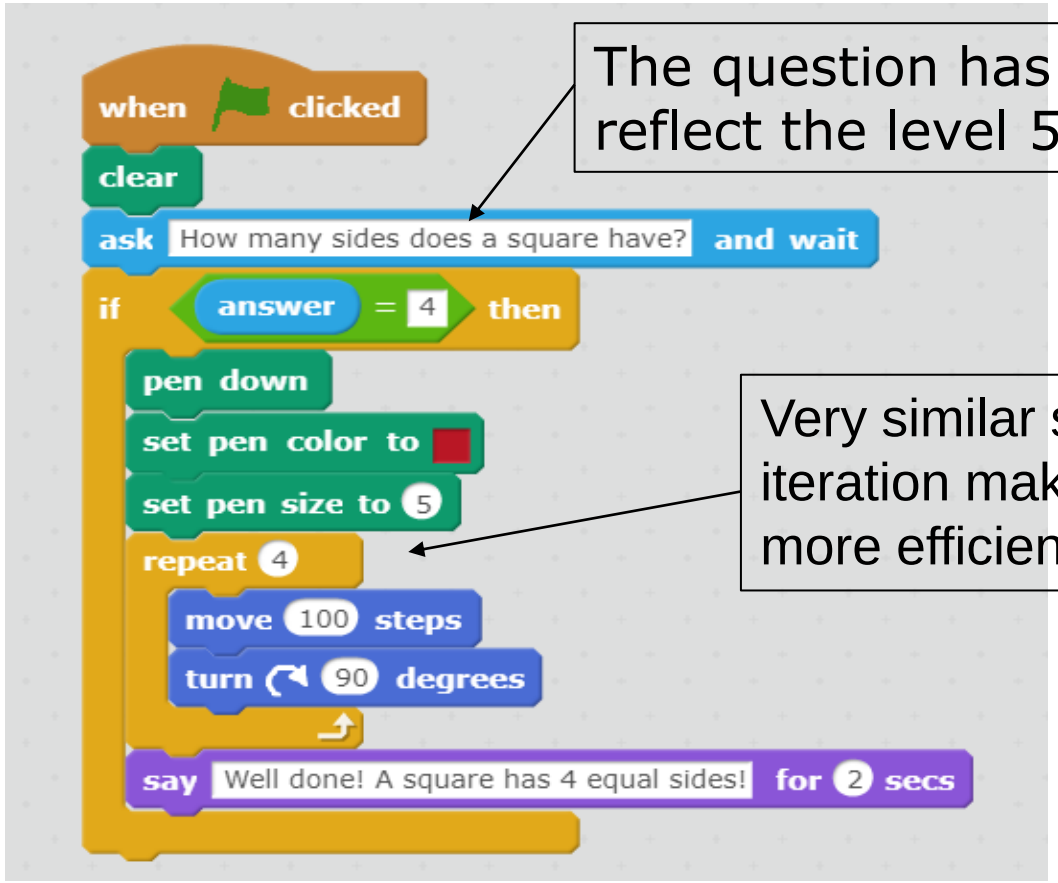
Develop an algorithm

```
BEGIN
        Prompt user to enter in the number of sides in a square
        IF user enters 4 THEN
                REPEAT 4 times
                        Move 150 steps
                        Turn 90 degrees
                Display ‘Well done. A square has 4 sides.’
        ELSE
                Display ‘Incorrect. A square has 4 sides.’

 END
```

Creating Digital Solutions – Levels 5 and 6

Possible Levels 5 and 6 solution



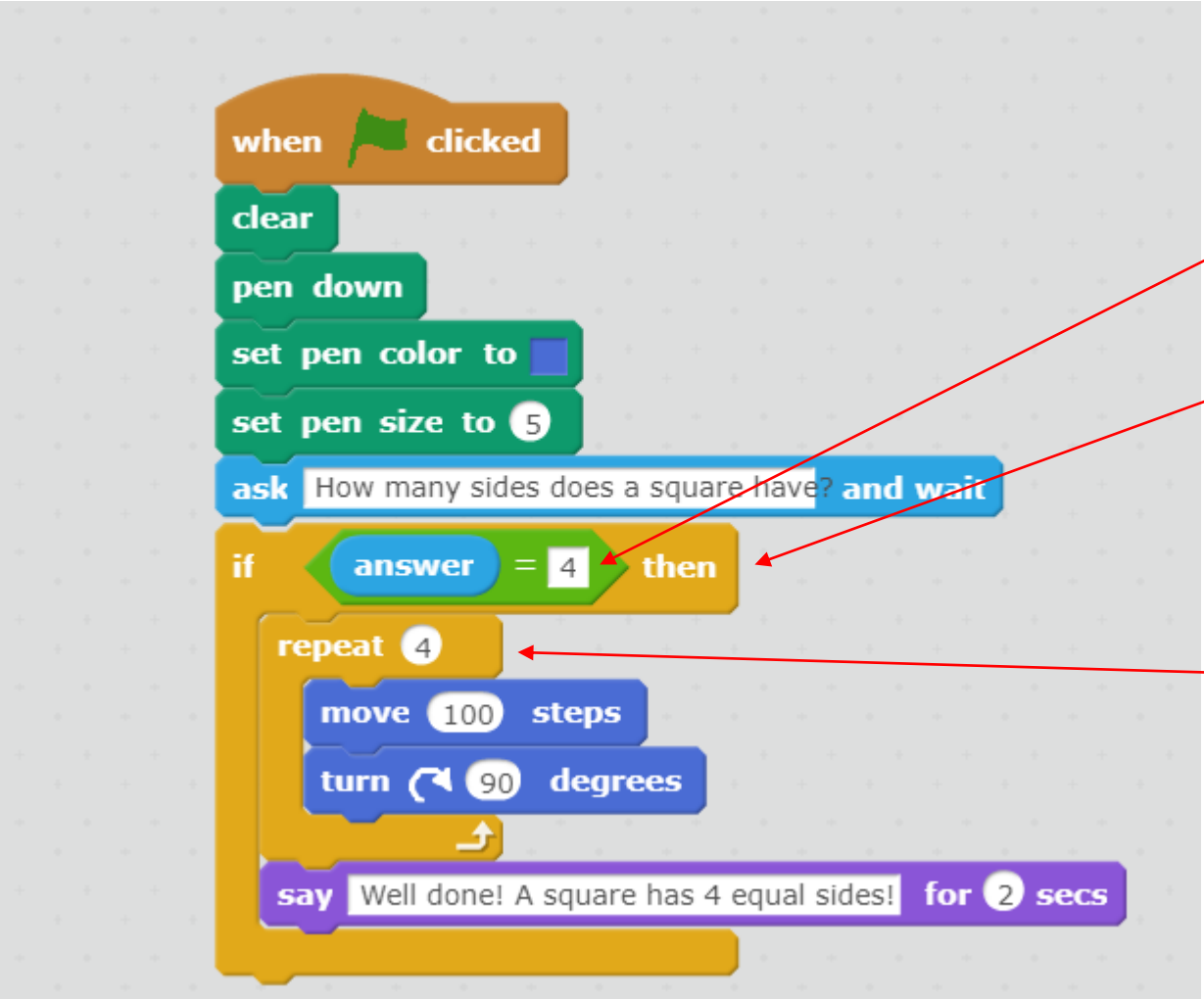
The question has changed to reflect the level 5-6 problem.

Very similar solution, adding iteration makes the solution more efficient.

Previous Levels 3 and 4 solution

Both these solutions would need more questions and shapes to meet the design brief.

Creating Digital Solutions – Levels 5 and 6

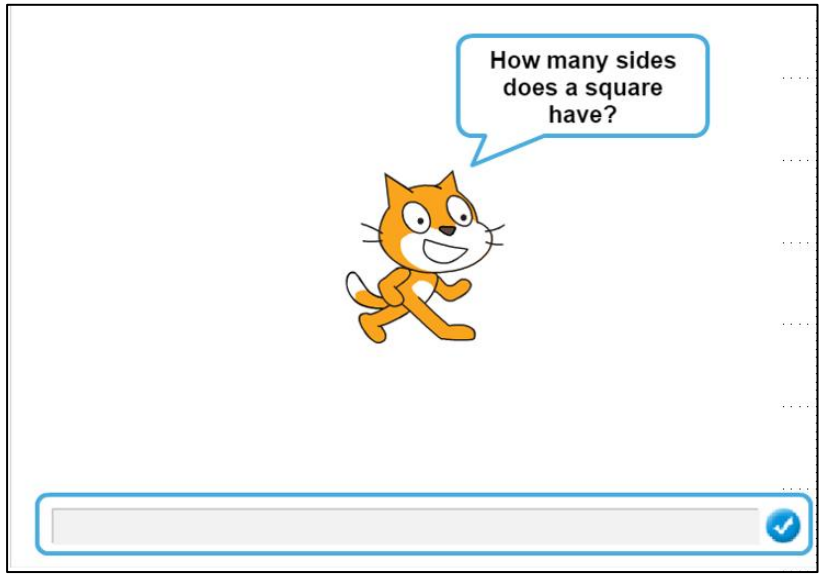


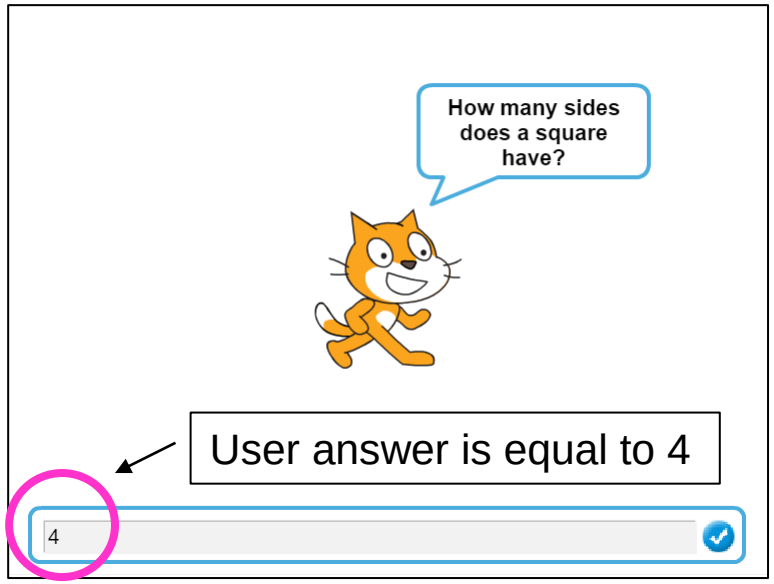
USER INPUT

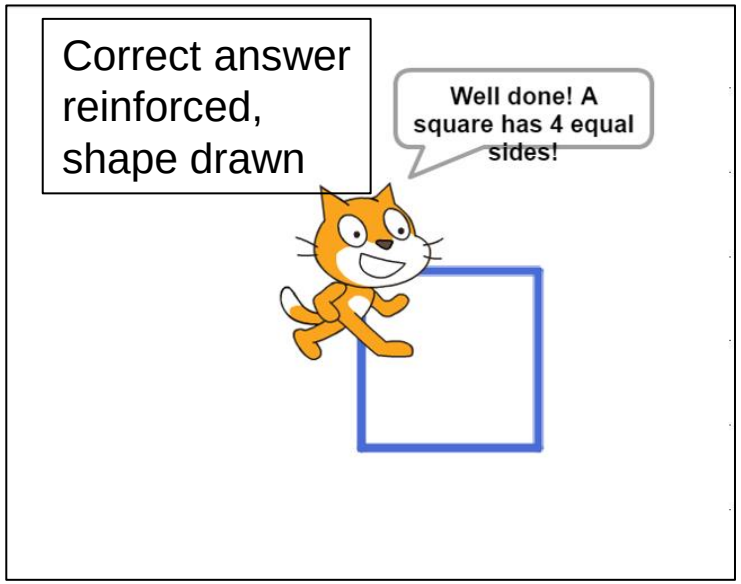
BRANCHING
If answer is equal to 4 then…

ITERATION
Using REPEAT

Possible solution 1 – Correct! – Levels 5 and 6



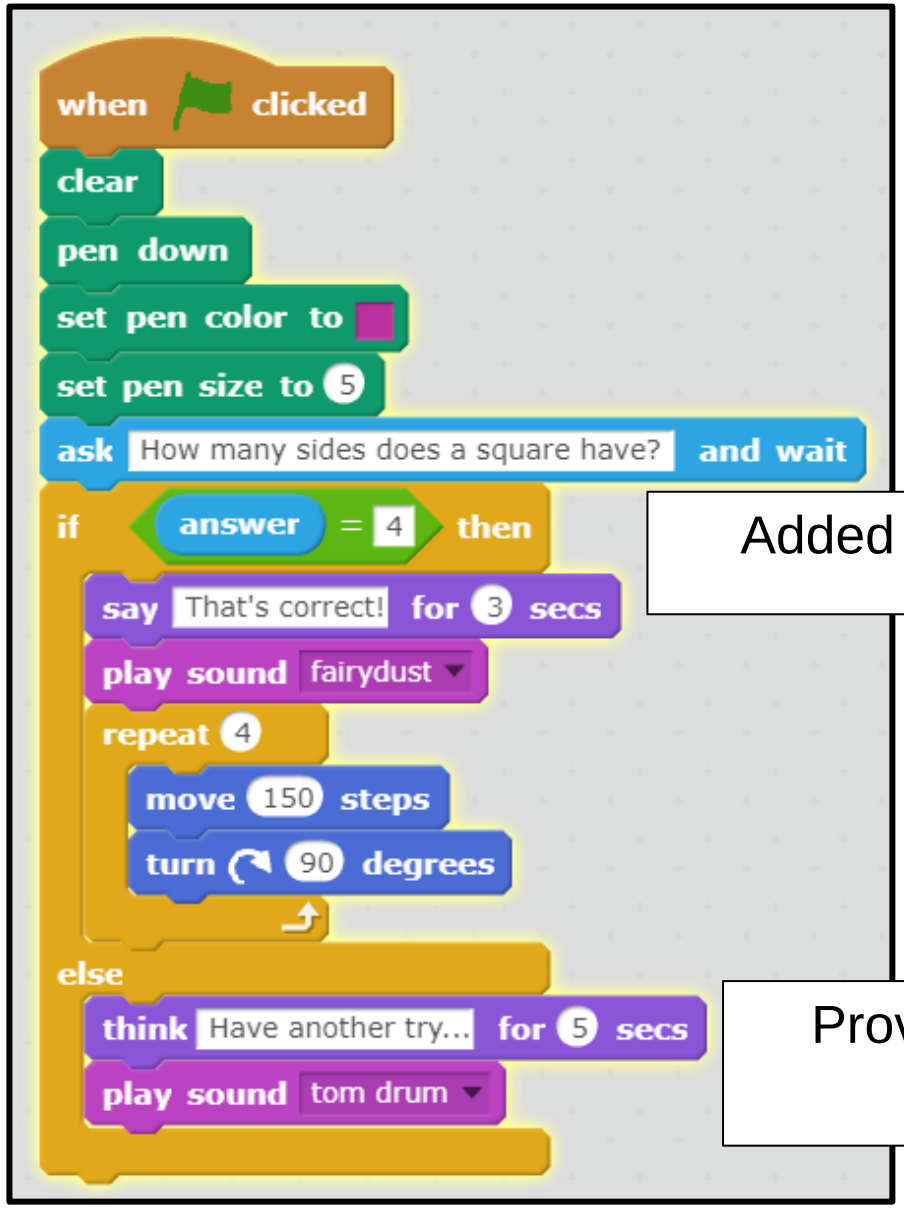


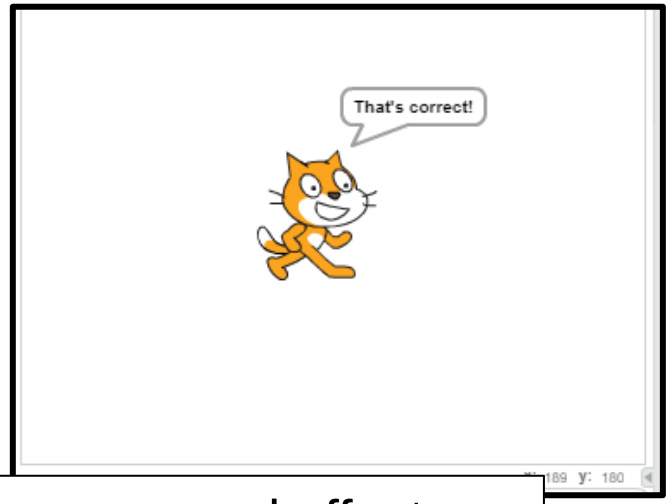


Does this meet all the requirements?
What happens if a wrong answer is input?

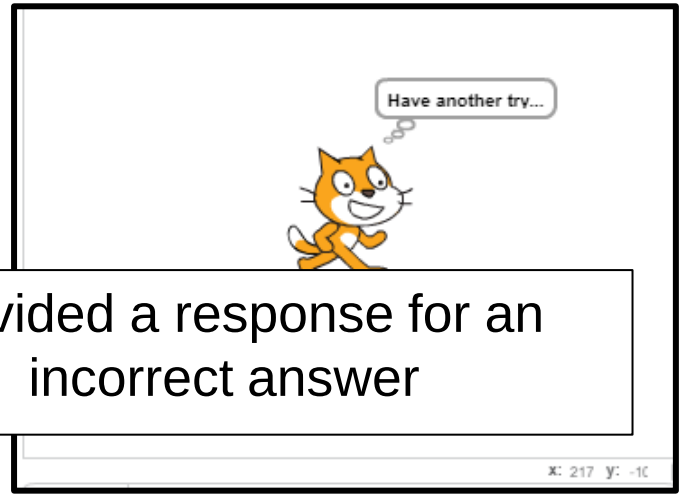
How about interest for a user?

Possible solution 2 – Levels 5 and 6

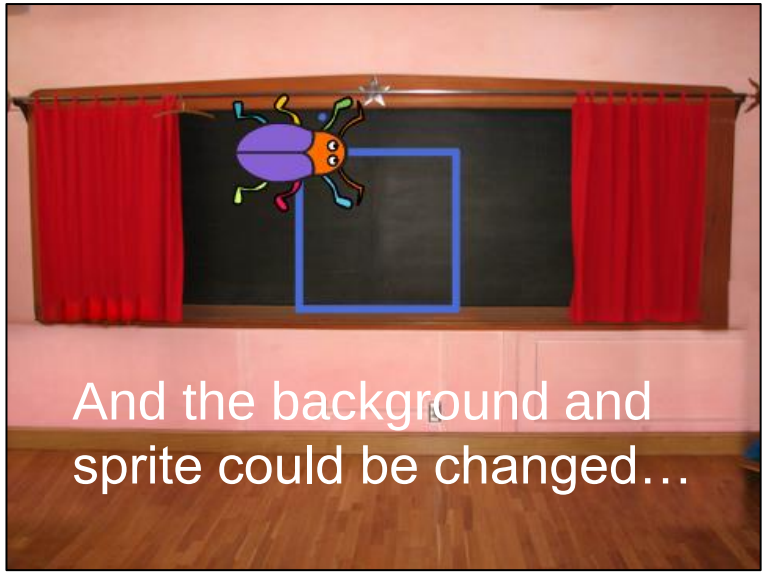




Added some sound effects



Provided a response for an incorrect answer



Evaluating Levels 5 and 6

Has the solution solved the problem?

- Have alternative design ideas been considered and evaluated?
- Have the functional requirements been met?
- What about sustainability?
- How could this solution be extended? (Keeping score – moving the sprite across the screen for a correct answer – limiting attempts to correctly answer the question…)

Functional requirements?

- Ask a question
- Wait for a response input by a user
- Tell the user if their answer is correct or incorrect
- Draw the shape
- Encourage the user to keep playing

User interface

- Suitable for use by a Year 2 student
  - Game environment
  - Operability

Achievement Standards – Level 6

Achievement Standard Extract – Level 6

Students define problems in terms of data and **functional requirements** and **design solutions by developing algorithms** to address the problems. They incorporate **decision-making, repetition and user interface** design into their designs and develop their digital solutions, including a **visual program**. Students explain how information systems and their developed solutions **meet current** and future **needs** taking sustainability into account.

Problem-solving methodology – Levels F - 6

Analysing

- What is the problem that needs to be solved?

Designing

- The user interface and algorithms

Developing

- The solution development

Evaluating

- Determining problem has been solved