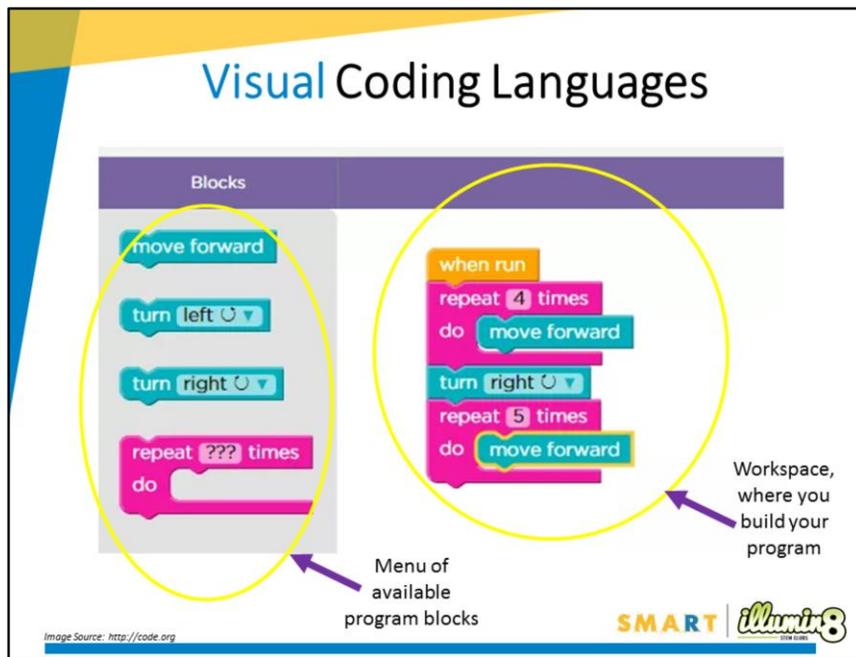


Welcome back!

In the last session, we started to look at computers and coding, and explored the computer language binary. Can anyone remember what binary code looks like? (Binary code is a series of digits, that are either one's or zero's. Each digit is called a Bit. A Byte is made up of 8 Bits.) We also had a go at writing lines of code in English, as instructions for a human robot to move through a maze. We found out our instructions needed to be very specific to solve the task! Today, we'll explore some other coding languages.



## Visual Coding Languages



Scratch and Blockly are visual coding languages. They use “blocks” which can be put together to write a program. Underneath/behind the simple blocks are the written coding languages. In visual programming languages, you have a menu of available “blocks”, or sets of instructions, which you can drag around and place next to other instructions and then you’re good to go!

It’s kind of like connecting pieces of a jig-saw puzzle together. The code blocks need to fit together, to form the whole program.

# MINECRAFT and Blockly coding



Video and Activity: <https://studio.code.org/s/mc/stage/1/puzzle/1>

Video Source: <https://studio.code.org/hoc/1>

SMART | *illumina:3*

Has anyone heard of Minecraft?

**Minecraft** is a computer game. It's a virtual land where users can create their own worlds and experiences, using building blocks, resources discovered on the site and their own creativity.

This is a short video about Minecraft programming, using Blockly.

<https://studio.code.org/s/mc/stage/1/puzzle/1>

We'll then do a fun activity to practice coding and writing algorithms in Blockly, within the Minecraft world.

Refer to coordinator notes for Activity 6.2.1

## Coding for Minecraft with Blockly



Videos: Loops [https://youtu.be/\\_X3FPKyVV1g](https://youtu.be/_X3FPKyVV1g)  
'If' statements <https://youtu.be/WXlbnlX5Rw>

Video Source: <https://studio.code.org/s/mc/stage/1/puzzle/1>



Tips for this activity:

Your screen is split into three main parts. On the left is the Minecraft play space where your program will run (you'll see the character move and follow your programmed instructions).

The middle area is the toolbox and each of these blocks is a command that directs Alex's actions.

The white space on the right is called the work space and this is where you'll build your program.

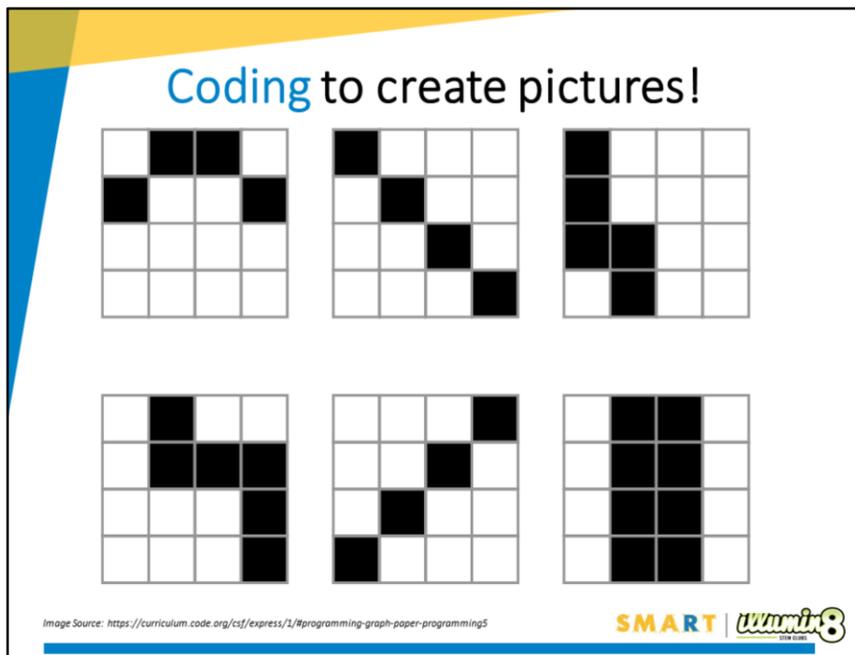
Step through the activities to practice dragging and dropping blocks of code (commands) and then run your programs to test them out. You can edit and change your code to keep improving.

At activity 5, there will be another video, this will introduce a cool coding tool called a 'repeat loop'.

Video: [https://youtu.be/\\_X3FPKyVV1g](https://youtu.be/_X3FPKyVV1g)

At activity 11, there will be another video, this will introduce a cool coding tool called 'if statements'.

Video: <https://youtu.be/WXlbnlX5Rw>



No access to computers, or want to include an unplugged extension activity?

By "programming" one another to draw pictures, students will begin to understand what coding is really about.

Try this great activity from CS Unplugged. Worksheets and instructions are available at:

<http://csunplugged.org/image-representation/>

Alternately, try this great coding activity from code.org. Worksheets and instructions are available at: <https://curriculum.code.org/csf/express/1/#programming-graph-paper-programming>

Refer to coordinator notes for Activity 6.2.2

# Human Interface Design



Image Source: pixabay.com

**SMART** | **illumina:3**

A human interface device (HID) is a method by which a human interacts with an electronic information system either by inputting data or providing output.

As new computers become more and more complex, their human interface design becomes more and more important.

Imagine if using a new computer program was so complex that you had to carry around the manual for instructions. That wouldn't be very efficient!

We need to design the interface between humans and computers to be as simple to use as possible.

Imagine if... the button labelled 'on' on a computer actually turned it off! That would be confusing!

Imagine if... the 'up arrow' on a keyboard actually moved an object downwards! That would also be very confusing!

We would call these poor interface designs!

## Human Interface Design

BLUE	GREEN	RED
YELLOW	ORANGE	GREEN
GREEN	RED	BLUE
PURPLE	PURPLE	BLUE

SMART | *illuminate* 3

Let's test out what would happen if all the names of colours on and the displayed colour of the name were all mixed up!

Can you read out loud what colour word you see?

For example, the first word in the table is the colour "GREEN", even though the word itself is spelt "BLUE".

I'll point to the words, and you say them out loud.

Let's go as fast as we can, and see how many we get right!

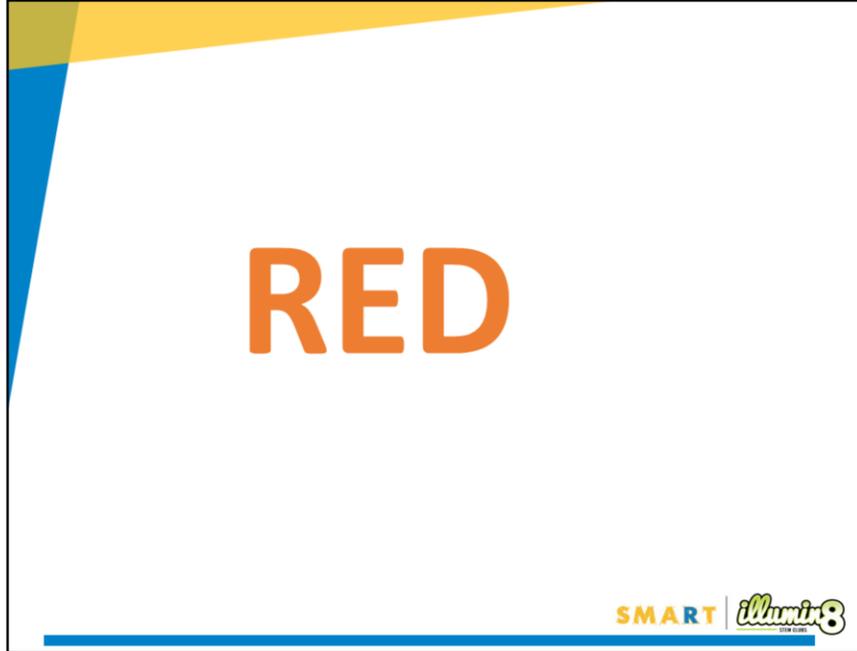
Refer to coordinator notes for Activity 6.2.3.



Answer: green



Answer: green



Answer: orange



Answer: blue



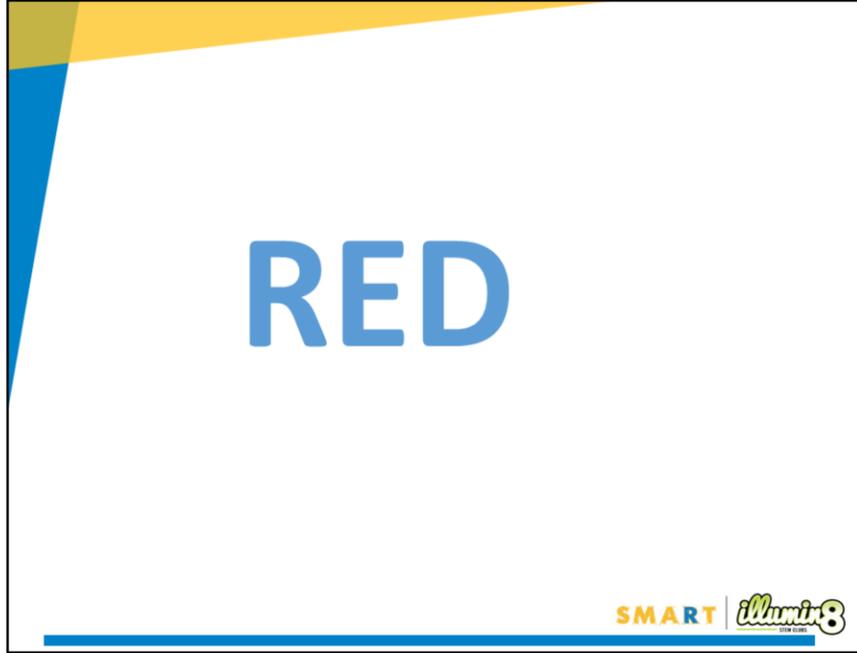
Answer: green



Answer: green



Answer: red



answer: blue



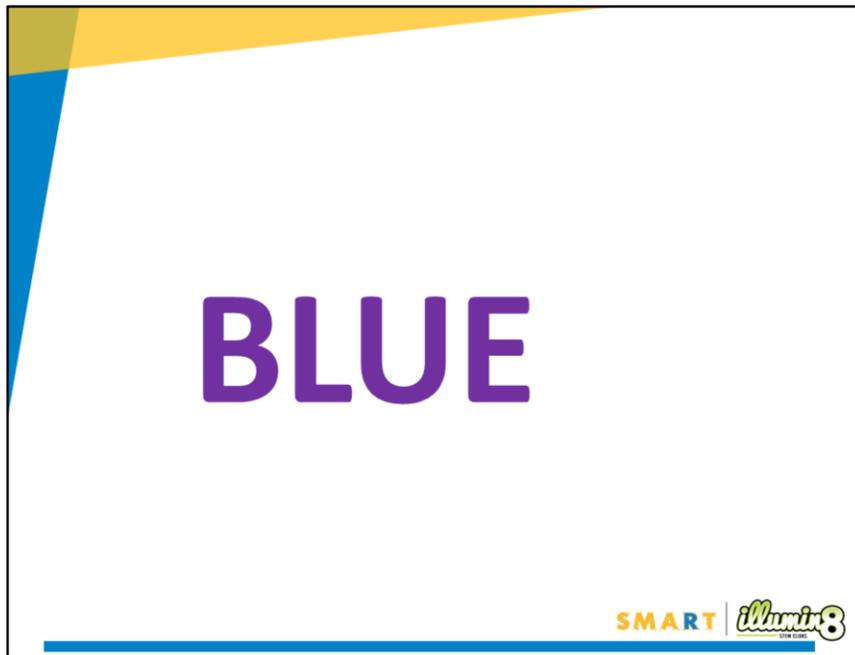
answer: red



answer: purple



answer: blue



Answer: purple!

.....

That was hard wasn't it!

The written words themselves have a strong influence over your ability to say the colour you see. The interference between the different information (what the words say) and the colour of the words your eyes see, causes a problem.

This is actually called the "Stroop Effect", named after John Ridley Stroop who discovered this brain-eye challenge in the 1930's.

Human Interfaces with Computers need to be carefully designed to reduce brain-eye confusions like this challenge!

## Human Interface Coding



Image Source: pixabay.com

**SMART** | **illumina:3**  
THE CODE

An interface for a computer might be a keyboard, a button, a mouse, a game controller, or a touch screen.

This is how we as humans can interact with computer programs.

Any time someone clicks, taps or types on the interface of a computer, that generates an **'event'**.

Coding enables the computer to decide what to do when an **event** occurs.

For example, if we are using a word processing program, pressing the letter "a" button on our computer keypad may display the letter "a" on the screen.

If we are playing a computer game, pressing the letter "a" might make our game start, or stop, or even make a game character jump, kick a ball, drive or run!

# Make a Flappy Game!

Using the coding skills you've developed, can you create all sorts of programs!

Let's try and make a new version of the game "Flappy Bird" !



Image Source: <https://studio.code.org/flappy/1>

SMART | **illuminate** 3

Has anyone heard of Flappy Bird?

**Flappy Bird** is a computer game.

We can create code to customise the game.

For example, we can code to decide what will happen if the mouse button is clicked.

Our bird might flap, or make a sound! We get to decide, by writing the code behind the game.

Video: <https://youtu.be/VQ4lo6Huylc>

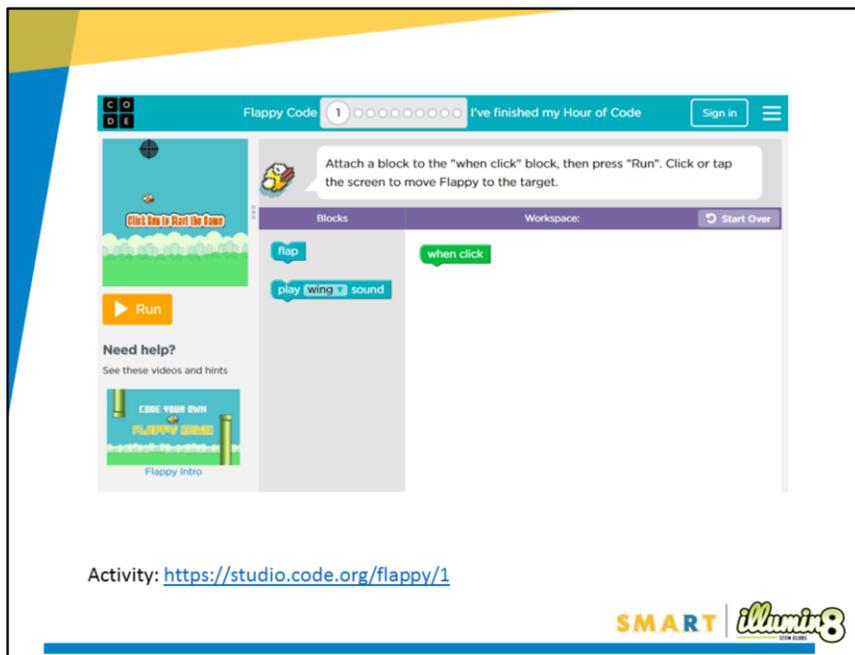


The screenshot shows a Scratch project window titled "Code your own Flappy Game". The stage features a blue sky background with a green ground and a city skyline. A yellow bird is positioned in the center, and two green pipes are on either side. The text "CODE YOUR OWN FLAPPY GAME" is displayed in the center. A play button icon is overlaid on the word "FLAPPY".

Video and Activity: <https://studio.code.org/flappy/1>

SMART | *illumina:3*

<https://youtu.be/VQ4lo6Huylc>



Activity: <https://studio.code.org/flappy/1>

This activity builds on the concepts introduced in the Minecraft exercise. Students will drag and drop blocks, and this time, add coding for responding to user interface inputs, or 'events', such as instructions for what to do if the mouse button is clicked, or the space bar is pressed.

## References

Videos and online activities sourced from <https://code.org/>

© Code.org, Used as per copyright agreement.

Unplugged activities sourced from Computer Science Unplugged,  
<http://csunplugged.org/> per Creative Commons licence