Level 3 – Conditional events: Hungry snake

Overview

In this lesson, pupils will consolidate their understanding of conditions in programming and learn how different types of conditions can be used in code for different purposes.

Learning objectives

- To make a simple game that uses conditional hit events to check if one object has hit another.

Links to CAS progression pathways

Programming and development

- Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL)
- Executes, checks and changes programs. (AL)
- Understands that programs execute by following precise instructions. (AL)
- Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL)
- Detects and corrects errors, i.e. debugging in algorithms. (AL)

Links to Curriculum for Excellence

- I understand that sequences of instructions are used to control computing technology. (TCH 0-14a)
- I can develop a sequence of instructions and run them using programmable devices or equivalent. (TCH 0-15a)

Success criteria

ALL I can use a conditional hit event in my code to make an egg disappear when a snake hits it.

MOST I can use several conditional hit events in my code to make different things happen on the screen.

SOME I can design a simple game and explain how the code used to create my program executes as the game is being played.

Key vocabulary

conditional statement, condition, collide, object, hit event

Starter

Show an example of a space maze game made in the first lesson of this unit, and discuss the way in which the code executes as the program runs and the user interacts with it. Recap pupils' understanding of conditional events and how the object responds to the background or environment (e.g. stopping and hiding when it hits particular colours).

Discuss ways to improve the maze game. Suggest using a character as the object to control as it eats or collects things on its way around the maze, or the idea of the maze containing obstacles that cause damage to the character.

Discuss games the pupils have played in which they control a character and make it eat or collect something, and games that contain obstacles to avoid.

Key questions

- What code did we write to control the spaceship?
- What is a ‘conditional’ event?
- How could we change/improve this game?
- What other objects could we include?

Main learning

Show a finished app made in the step 4 of this lesson and give pupils an opportunity to play the game. Break the algorithm for the game into parts: the snake needs to move around the screen with key press events; eggs need to be ‘eaten’ when the snake hits them; the snake needs to disappear if it eats a rotten egg.

Examine what happens when the eggs are ‘eaten’ (they disappear and reappear in a different position). Create a list of conditional statements to describe what the code needs to include.

Work through the lesson steps with pupils, introducing the ‘random position’ block and ensuring pupils understand what this does. Set pupils the challenge of creating the step 4 game themselves, referring to the algorithm you created earlier to help them.

Using the Build step

In the Build step, pupils can practise the coding they have been learning with their own choice of objects.

Key questions

- What objects can you see on the screen?
- What do I want my snake to do?
- What keys will we use to make the snake move?
- What do we want to happen if the snake hits the egg?
- What should happen when the snake hits a rotten egg?

Plenary

Have pupils show each other their apps and explain how they have programmed them using the key words ‘when’ or ‘if’, ‘hit’, ‘then’ and ‘condition’. Ask them to share two pieces of positive feedback, as well as one suggestion for improvement.

Key questions

- Explain how you made your app.
- What did you like about it?
- How could you make your app more challenging?