Unit 5.1: We are game developers

5 words to remember

algorithm: a sequence of precise instructions or steps (or set of rules) for achieving a goal

debug: to identify and correct mistakes in a computer program or algorithm

iterative development: a trial-and-error approach where each version builds on the previous by fixing bugs or adding features

logical reasoning: a step-by-step approach to solving problems, where the conclusion can be explained using accepted and reliable rules

sprite: a graphical character in a program that can be given its own sequence of instructions (the default sprite in Scratch is the cat)

Knowledge check: Game variables

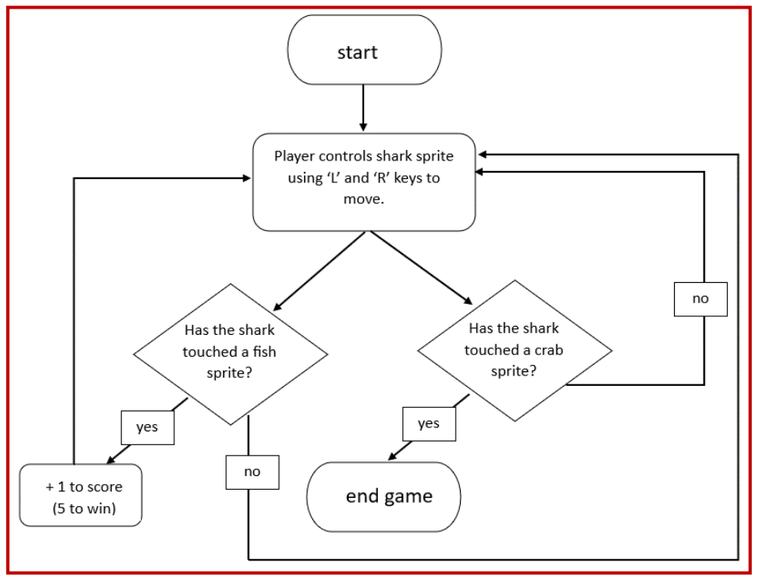
Variables are values that change in a program, for example scoring and timers in games. In Scratch, variables need to be made, named and programmed.

Test yourself: The code in this box shows that a scoring and a timer variable have been used. Identify and explain how this game ends.

Knowledge check: Algorithms and flowcharts

Planning the **algorithm** for a game before creating it supports understanding and **logical reasoning**. Flowcharts can be a useful way of mapping out the steps for playing the game.

Test yourself: Look at the flowchart below, which explains a simple game involving a shark, a fish and a crab. Can you explain the game?

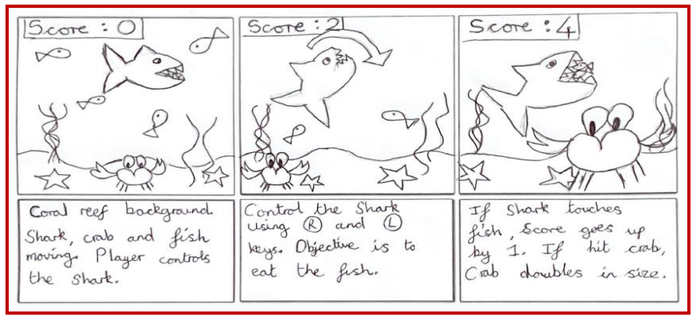


Test yourself: Next, see whether you can recreate the flowchart to add these features:

- Sound when the game ends with a loss
- A timer that ends the game after 60 seconds.

Key takeaways

- Games usually involve both a set of rules and a clear objective (goal). Many, although not all, involve some form of competition. In the case of computer games, the competition can be between the player and the computer, or against other players.
- The most successful games involve some aspect of progression, with the game getting more and more challenging as the player successfully moves through the game.
- When designing computer games, storyboards and flowcharts can be used to support planning, algorithm design and logical reasoning to explain how the game works.



- Scratch programming software has a library of assets, including backgrounds, **sprites** and sounds, but users can also create their own. This is particularly useful when designing games that need a specific background, such as a maze or ping-pong style layout.
- Competitive elements such as timers and scoring can be added to games. In Scratch, these are created using the variable blocks.
- Computer games can contain long sequences of code; therefore, **debugging**, ongoing evaluation and **iterative development** are needed. Working collaboratively allows others to test games and provide valuable feedback.
- Players need clear instructions for games at the start. A splash screen can be designed in Scratch as an additional sprite that appears on screen for a few seconds before the game starts. Here is an example of a splash screen and code: