

Coding with Scratch: Developing Games

Computing | Year 5 | Planning Overview

In this unit about *Coding with Scratch: Developing Games*, children will learn how to design their own backdrop with increasing complexity. They will also learn how to code a sprite to move when certain keys on a keyboard are pressed. Children will also become familiar with if, then statements to create conditions with a game.

Accompanying this unit is a helpful [Knowledge Organiser](#). The Knowledge Organiser collates the subject knowledge for the unit and is used throughout.

Further support with using Scratch can be found in our Twinkl Scratch Adult Guidance.

Hardware and Software

Hardware

- PC devices, such as laptops, Chromebooks and/or tablets

Software

- Scratch Online accessed via <https://scratch.mit.edu/>

Health and Safety

Children should be encouraged to have good posture and sit up to the computer. Children should not spend extended periods of time looking at the screen. Ensure safe and responsible use of portable digital devices, discouraging children moving around the classroom with technology. Make sure that food and drink are kept away from all electronic items.

Scratch Safety

Showing or creating the flashing sprite effect could be problematic for children with conditions such as epilepsy. Discretion is advised.

Home Learning

[Swat Game!](#)

In this task, children will create a variable and use various Scratch coding blocks to create a game where a player must use the arrow keys to avoid another sprite.

[Cheeky Monkeys!](#)

In this task, children should create a game, using **Operator** blocks. They will program sprites to appear randomly on the **Stage** for the player to try to click on the sprite.

Disclaimer

External Links:

We hope you find the information on our website and resources useful. This resource contains links to external websites and/or external apps. Please be aware that the inclusion of any link in this resource should not be taken as an endorsement of any kind by Twinkl of the linked website and/or app, or any association with its operators. You should also be aware that we have no control over the availability of the linked pages and/or apps. If the link is not working, please let us know by contacting TwinklCares and we will try to fix it although we can assume no responsibility if this is the case. We are not responsible for the content of external sites and/or external apps.

Assessment Statements

By the end of this unit...

Working Towards the Expected Level:

- With support and prompts, children can design and create a simple maze game by: designing backdrops and sprites; using relevant coding blocks; programming consequences for actions completed; adding appropriate effects and including backdrop and costume changes.
- Children can begin to understand and use conditional statements in their code, including if...then and if...then...else blocks.
- With support, children can understand simple algorithms by predicting what may happen within their code.
- Children can begin to understand how Operators blocks work and can use these within their code.
- Children understand decomposition and can decompose a problem, with support.
- Children can create variables and implement these variables in code, with support.

Working At the Expected Level:

- Children can design and create a simple maze game by: designing backdrops and sprites; using relevant coding blocks; programming consequences for actions completed; adding appropriate effects to enhance the game by including a backdrop and costume changes.
- Children understand and use conditional statements in their code, including if...then and if...then...else blocks.
- Children understand simple algorithms by predicting what may happen within their code.
- Children understand how Operators blocks work and can use these within their code.
- Children understand decomposition and can decompose a problem.
- Children can create variables and implement these variables within their code.

Working At Greater Depth:

- Children can independently design and create a simple maze game by: designing backdrops and sprites; using relevant coding blocks, including being able to justify their coding choices; programming consequences for actions completed; adding appropriate effects to enhance the game and including purposeful backdrop and costume changes.
- Children understand and confidently use conditional statements in their code, including if...then and if...then...else blocks.
- Children understand simple algorithms by predicting, with confidence, what may happen within their code.
- Children understand how Operators blocks work and can use these within their code, with confidence.
- Children understand decomposition and can independently decompose a problem.
- Children can independently create variables and implement these variables within their code.

Lesson Breakdown

1. A-Maze Game!

Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.

Use sequence, selection, and repetition in programs; work with variables and various forms of input and output.

To design and program a maze game.

Resources

[Lesson Pack](#)

[Knowledge Organiser](#)

[KWL Grid](#)

Highlighters

Hardware:

- PC devices, such as laptops, Chromebooks and/or tablets

Software:

- Scratch Online version accessed via <https://scratch.mit.edu/>



Additional Lesson Information and Possible Misconceptions

This lesson contains an Adult Guidance document and it is recommended that it is read prior to the start of the lesson.

The presentation and activity sheets include a similar step-by-step guide for children to follow, when creating their Maze Game. The presentation slides can be used in conjunction with the activity sheet or can be used as a discussion activity before the children begin to code independently.

2. Level Up!

Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.

Use sequence, selection, and repetition in programs; work with variables and various forms of input and output.

To design and program the next level for my Maze Game.

Resources

[Lesson Pack](#)

Hardware:

- PC devices, such as laptops, Chromebooks and/or tablets

Software:

- Scratch Online version accessed via <https://scratch.mit.edu/>



Additional Lesson Information and Possible Misconceptions

This lesson contains an Adult Guidance document and it is recommended that it is read prior to the start of the lesson. It is also important to note that solutions provided are one possible solution and children may produce their own fully-working code.

The presentation and activity sheets include a similar step-by-step guide for children to follow, when creating their Maze Game. The presentation slides can be used in conjunction with the activity sheet or can be used as a discussion activity before the children begin to code independently.

3. Game Over!

Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.

Use sequence, selection, and repetition in programs; work with variables and various forms of input and output.

To add a final level, further enhancing the code in a Maze Game.

To add sound effects with a purpose.

Resources

[Lesson Pack](#)

Hardware:

- PC devices, such as laptops, Chromebooks and/or tablets

Software:

- Scratch Online version accessed via <https://scratch.mit.edu/>



Additional Lesson Information and Possible Misconceptions

This lesson contains an Adult Guidance document and it is recommended that this is read prior to the start of the lesson.

Alongside the Adult Guidance, you can view comment examples in Scratch. To do so, please access [Lesson 3 Comments \(Game Over!\)](#) within the [Scratch Project Area](#).

The presentation and activity sheets include a similar step-by-step guide for children to follow, when creating their Maze Game. The presentation slides can be used in conjunction with the activity sheet or can be used as a discussion activity before the children begin to code independently.

4. Splat! Game

Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.

Use sequence, selection and repetition in programs; work with variables and various forms of input and output.

To design and program a game within Scratch using Boolean operators.

Resources

[Lesson Pack](#)

Hardware:

- PC devices, such as laptops, Chromebooks and/or tablets

Software:

- Scratch Online version accessed via <https://scratch.mit.edu/>



Additional Lesson Information and Possible Misconceptions

This lesson contains an Adult Guidance document and it is recommended that this is read prior to the start of the lesson.

5. Pop!

Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.

Use sequence, selection and repetition in programs; work with variables and various forms of input and output.

To program costume changes for a sprite in a game.

To add effects that enhance a game.

Resources

[Lesson Pack](#)

Hardware:

- PC devices, such as laptops, Chromebooks and/or tablets

Software:

- Scratch Online version accessed via <https://scratch.mit.edu/>



Additional Lesson Information and Possible Misconceptions

This lesson contains an Adult Guidance document and it is recommended that this is read prior to the start of the lesson.

6. High Score!

Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.

Use sequence, selection and repetition in programs; work with variables and various forms of input and output.

To add a point-scoring system to a game.

To add backdrop changes to a game.

Resources

[Lesson Pack](#)

[KWL Grid](#)

Hardware:

- PC devices, such as laptops, Chromebooks and/or tablets

Software:

- Scratch Online version accessed via <https://scratch.mit.edu/>



Additional Lesson Information and Possible Misconceptions

This lesson contains an Adult Guidance document and it is recommended that it is read prior to the start of the lesson.