

MrPICT.com

**D.A.R.E.S**

**Dare to Innovate**

*Computing projects that inspire pupils to create and innovate.*

**Programming KS1**

*Animations in ScratchJr*

# What is D.A.R.E.S

D.A.R.E.S is an innovative approach to teaching computing which encourages pupils to be critical thinkers, problems solvers and computational thinkers while creating purposeful content to demonstrate how learning can be applied across the wider curriculum.

Through these projects, pupils are able to deepen their knowledge of computing by applying it to solve problems and create content. These may take place in computing lessons over a half term with any spare lessons used for word processing or e-safety. This will then allow pupils to apply these creative ideas more regularly across the curriculum. We do not want to make these projects too prescriptive as we know some classes may need some more time to complete different elements. As teachers, you will know the ability of your class so please feel free to change or adapt these ideas to fit the ability of your class.

The stages which can be adapted into lessons are as follows:

**D - Design:** Pupils start to discuss the desired outcome for their project and are given time to tinker with the software before planning what they will do to achieve their outcome.

**A - Apply:** Pupils are given the opportunity to create, make and produce content using the app or software explored in the Design lesson(s)

**R - Refine:** Pupils spend time considering ways to modify and improve their projects to get the best results possible.

**E - Evaluate:** Upon completing their desired outcome, pupils are given the opportunity to reflect and consider how effectively they have achieved their goal.

**S - Share:** Learners are given the opportunity to publish and exhibit their work to the world embedding skills from the Digital Literacy curriculum.

# Objectives Covered

The following objectives will be covered in this unit from the knowledge progression document available on [MrPICT.com](http://MrPICT.com)

Strand	
<u>Computational Thinking</u>	<ul style="list-style-type: none"><li>● I understand what algorithms are</li><li>● I know how to write simple algorithms</li><li>● I understand the sequence of algorithms is important</li><li>● I know how to debug simple algorithms</li></ul>
<u>Coding/Programming</u>	<ul style="list-style-type: none"><li>● I know how to create a simple program on a digital device e.g. Bee Bot or tablet</li><li>● I know how to use sequence in programs</li><li>● I know how to locate and fix bugs in my program</li></ul>

<b>Key Vocabulary</b>	Algorithm, sequence, order, bug, fix, precise, Digital, program, follow, code, bugs, fix, order, ScratchJr
-----------------------	--

# What is needed:

Pupils will create their animation in ScratchJr <https://www.scratchjr.org/> This is a free app which can be run on Android or iOS tablets. It can also be installed on laptop or desktop PCs using this link: <https://jfo8000.github.io/ScratchJr-Desktop/> and run on Chrome books using <https://chrome.google.com/webstore/detail/scratchjr/oipimoeophamdcmjcfameoojlbhbgjda?hl=en>

Watch the video on this page if you are unfamiliar with ScratchJr :  
[ScratchJr Video](#)

- You may choose to pair pupils to complete this project as 'paired programming' can be an effective way to support pupils' learning. One pupil acts as the 'driver' controlling the tablet whilst the other 'navigates' instructing them what to do. Pupils then regularly swap roles.

This video is of the final animation shown in the example designs created in ScratchJr.

[Example Animation Video](#)

# Design

## Code Reading

Display ScratchJr and add a sequence of code from the motion and looks blocks for the default sprite, such as the code below for example.



Encourage pupils to read the code to predict what the sprite will do when it is run. Then run the code (by touching it) to explore if they are correct. Repeat this activity exploring different sequences of code with a focus on the importance of the order of the blocks. Encourage discussion throughout.

**Tinker Time** - Show pupils how to add one of your code sequences in ScratchJr so they can then explore the app by tinkering with, and modifying, this code. Share the prompt questions below as a guide, and regularly use mini-plenaries to share what pupils are learning.

- Can you change the way the sprite moves? Can you get it to move further/not as far?
- Can you change what the sprite says?
- What different ways can you get the program to start?

Introduce the topic that will be the focus of pupils' animations. This could link to any area of the curriculum; the example used here links to history with The Great Fire of London.

Give pupils time to design their own ScratchJr animations using the following templates.

# My Project Plan

## What Will I Be Creating?

*What is your key content? Who is your audience?*

## What Will I Need?

*Decompose your project down to think about the different software, apps or resources you will need.*

## What Am I Going To Do?

*What are the steps in completing your project - the algorithm for the overall project?*

## My Own Steps For Success

*How will you know if you've been successful?*

# Example My Project Plan

## What Will I Make?

*What information will be in it? Who is it for?*

I'm creating an animation about The Great Fire of London for pupils in my class.

## What Will I Need?

- Tablet running ScratchJr
- Pen and paper to make my design.

## What Am I Going To Do?

*What are the steps in completing your project - the algorithm?*

1. I'm going to design my animation including an algorithm as a simple storyboard
2. I am going to draw my background and sprites in ScratchJr
3. I am going to code my sprites in ScratchJr

## My Own Steps For Success

*How will you know if you've been successful?*

1. I can design my animation including a storyboard for an algorithm
2. I can draw my sprites and background in ScratchJr
3. I can code my animation in ScratchJr
4. I can fix any bugs in my code

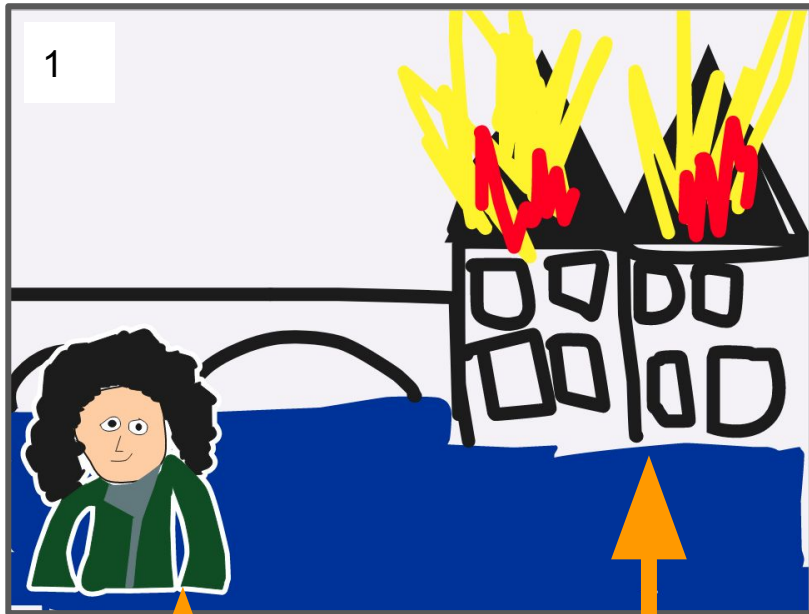
# My Project Design

- *What will your animation look like? What will the background be? What sprites will you need?*
- *What's the algorithm for your project? (This could be created as a storyboard)*



# Example Project Design

- What will your project look like? What will the background be? What sprites will you need?
- What's the algorithm for your project? (This could be created as a storyboard or text)



1

Background showing fire

Samuel Pepys Sprite

This video is of the final animation shown in this example design created in ScratchJr

[Example Animation Video](#)



2

The Great Fire of London started on Sunday 2nd of September 1666

Samuel sprite moves across screen and gets bigger and then shrinks again



3

It destroyed over 13000 house and 87 churches.

# Apply

Provide pupils with the time they need to create their animations in ScratchJr.

If required, these guides can be used for support:

- Interface <https://www.scratchjr.org/learn/interface>
- Paint Editor <https://www.scratchjr.org/learn/paint>
- Block Descriptions <https://www.scratchjr.org/learn/blocks>
- General Tips and Hints <https://www.scratchjr.org/learn/tips>

Pupils should use their designs when coding to check their program matches what they have designed. Pupils may have new ideas they want to incorporate as they code, so encourage them to update their designs to reflect this.

**Debugging** - As pupils are coding, they should regularly run their programs to test them. If they contain bugs (errors), they should find and fix these. This video explains 3 Top Tips for Debugging. The 1st top tip 'Looking closely at your code' particularly applies to KS1 pupils using ScratchJr.



# Refine

Pair pupils, or groups, and give them time to share their designs and animations with each other. Encourage pupils to provide peer feedback using their own steps for success from their project plans.

If required, use prompt questions such as those shown below, encouraging pupils to provide both positives and suggestions for improvements.

- Is the content of the animation appropriate?
- Has a background and sprites been created?
- Do the background and sprites match the design?
- Does what happens in the animation match the algorithm?

Give pupils time to act on the peer feedback to further refine their animations.

To extend the outcome of the animations, you could also introduce how sounds could be added to the animations. This video specifically explores using the sound blocks in ScratchJr



# Evaluate

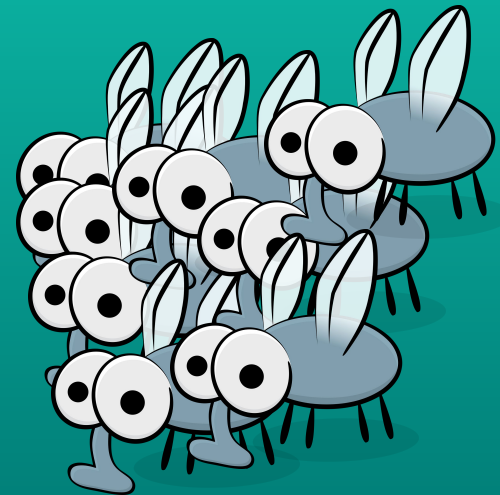
Give time for pupils to watch their completed animations (you may choose to share some with the class) and compare these with their project plan and design.

Ask pupils to complete the self evaluation form to assess how well they believe they have met the objectives of the project. Encourage a selection of pupils to share what they believe worked well and how they would change/improve things if they were to complete the project again.

## Bug Count!

Debugging is an important part of programming but one that pupils often find difficult and discouraging, as they want their program to work first time. In the 'real world' of programming it's estimated that often the same amount of time is spent debugging code as it is writing it in the first place.

As such, pupils should get used to finding and fixing bugs. Creating a 'Bug Count' (a tally of all the bugs found and fixed) as pupils evaluate their projects links to the 'What challenged me' section of the evaluation and helps pupils see bugs as a positive challenge.



# My Computing Project Evaluation

What Was I Trying to Achieve?

How Effective Was My Project?

0 1 2 3 4 5 6 7 8 9 10

What Did I Do Well?

What Challenged Me and How Did I Deal With It?

What Will I Improve Next Time?

# Share

ScratchJr projects can be shared to other iPads with ScratchJr on them either by email or Airdrop. This is explained on this page: <https://www.scratchjr.org/learn/tips/share-projects> To share pupils work further, a screen recording could be taken of their animations. This page includes a video on using the screen record function on Apple devices: <https://www.mrpict.com/apple-features.html>

It is important to make sure the pupils save their work to a safe space online. We recommend Seesaw. The following links will share an in-depth tutorial for how to:

- [Get started with Seesaw](#)
- [How pupils can take ownership of their learning](#)
- [How to give feedback through Seesaw.](#)

[Google Classroom can be used as an alternative, click here for a tutorial.](#)

In uploading their work, it is important to make links to the Digital Literacy Objectives from the Progression Document.

Ask questions such as:

Why do we want to upload this?

What personal information are we sharing?

Who will be able to see this work?

Will this affect our online reputation in a positive or negative way?

- If you want to share the children's work beyond Seesaw to a blog or social media platform, explain to the children what you are doing and why. Again, use this as an opportunity to re-enforce some of the messages from the digital literacy curriculum.

Please feel free to tweet [@ICT\\_MrP](https://twitter.com/ICT_MrP) with your examples and we will feature them on the [MrPICT.com/showcase](http://MrPICT.com/showcase) page.