COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE

CAS Barefoot

KS1 ScratchJr Knock Knock Joke Activity

Recommended year group: Any key stage 1
Activity duration: 1 hour
Cross-curricular links: English

Concepts and approaches

Creating Programming

Overview

In this activity pupils, in pairs, create a simple animation program of a knock knock joke. They use a storyboard to create their design, write the code in ScratchJr, debug and evaluate.

One of the key learning points in this lesson is that pupils will have to control the timing and order of the two sprites saying the knock, knock joke lines. In this activity the 'wait' command is used to sequence the events.

Pupil objectives

- I can make a design using a storyboard
- I can write code
- I can debug my code

Design introduction (10 mins)

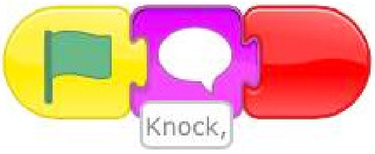
- As a class, ask children to share any knock, knock jokes they know (Slide 2).
- Explain to pupils that they are going to be creating animations of a joke using ScratchJr. Perhaps link this to your English objectives such as 'recognising and joining in with predictable phrases'.
- Lead a class discussion about how they might do this; explore ideas such as they will need to remind themselves of the joke, work out how to code it and debug it. Perhaps look at a couple of examples of animated joke (see resources).
- If it is your normal practice introduce the learning objectives. (Slide 3).
- Explain that first we need to design the program, and that you are going to use a storyboard to work out each of the parts of the joke.
- Ask pupils what a storyboard is. *Pictures of a story that tell us what will happen*. Ask if pupils can recall what a set of steps that make something happen is called, *algorithm*. So our storyboard will be our algorithm for our animation.
- Show children the storyboard template (slide 4) and taking ideas from the class complete it for a chosen knock knock joke. Explain that getting the order of the joke is what is important to start with, stick figures and no backgrounds are what is needed as we only have one lesson, plus their ideas may change once they see what characters and backgrounds are available in ScratchJr. Keep the design simple, perhaps with two characters, speech bubbles, one movement. An example design is shown on slide 5.

Design task (10 mins)

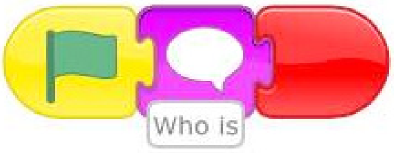
- Provide pupils, in pairs, with a storyboard template and ask them to create their own joke designs.
- During this time remind pupils to keep their designs simple. Ask pairs to talk through their designs with other pairs, so that they are checking they have all the steps they need.

Writing the code and debugging introduction (10 mins)

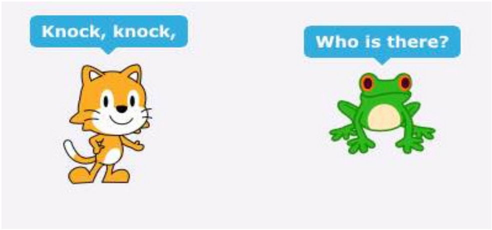
- Model how to open ScratchJr and start a new project, or ask a pupil to demonstrate this.
- Show pupils how to use the storyboard to guide you as you start coding.
- Show pupils how to add a sprite and add a 'say' command to say the first line and test it. Remind pupils that they should write a litte code, test it, write a little more code, test it. (See Slide 6) Model how tick off that part of the storyboard once it is debugged.


Knock,

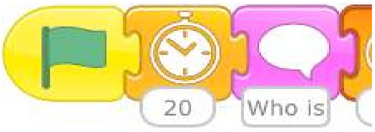
- Add the second sprite and its 'say' command.


Who is

- Again test it. Ask pupils what has gone wrong. The two sprites 'speak' at the same time.



- Remind children of the debugging steps.
  - What should it do? *It should have said the first line, followed by the second.*
  - What does it do? *The two lines are said at the same time.*
  - Where does it go wrong? *The second sprite needs to wait.*
  - Fix it.
- Show how to add the wait command to the second sprites script.


20
Who is

- Test that the code now does as expected and tick off the storyboard step.
- If necessary for your class, model how to code the next part of the storyboard.

Writing the code and debugging task (15 mins)

- Explain to pupils that they will now be given time to work with their partner to program their design in ScratchJr.
- Remind them to take it in turns to program (paired programming). One person will explain to the other what code to add, that person then codes it and together they debug. They could perhaps swap roles after each box on their storyboard.
- Provide support to groups that require it during the main task. A 'ScratchJr commands help sheet' has been provided which includes descriptions of commands which they might use in their program.
- If you see pupils doing work from which the rest of the class would benefit ask them to share their code and explain what they have done and why.
- If pupils finish their animations and have debugged them ask them to present their work to another pair who have finished and discuss what is effective and what might be improved (evaluate).
- Following on from this, ask pupils to add more to their program. See Stretch and Challenge and 'Challenge sheet'.

Mini-plenary (optional): reminder to debug as they program

- Ask children why it is important to write a short section of code, run it, write a little more code, run it? To check it works as we want it to. Can pupils recall the steps for debugging? The following points appear on slide 6 of the presentation to support pupils in debugging the programs.
- Write a little and debug, write a little and debug.
    - What should it do?
    - What does it do?
    - Where does it go wrong?
    - Fix it.

Plenary (5 mins):

- Revisit the objectives for the lesson and ask a selection of pairs to present their animations to the class. Ask pupils to briefly evaluate the work saying what was effective and what might be improved.
- If any pupils undertook the challenge activity ask them to share with the class and explain what they did.

Differentiation

Support: A 'ScratchJr commands help sheet' has been provided. An additional adult, if available, could work with a small group of pupils to provide additional support. They may plan their program as a group and work through implementing this in ScratchJr as a more guided activity.
Stretch & challenge: A 'Challenge sheet' has been provided for pupils who finish the main task.

Assessment opportunities

- Informal, teacher assessment of progress during main task, class discussions and plenary.
- Formal, summative assessment of the designs and ScratchJr projects if required (note however these are completed in pairs).
- Areas to focus on: Creating a simple design with the parts in the correct order. Writing code for each part of the design. Continuous debugging as pupils add code rather than just at the end. Getting the parts of their code to work in the correct order.

Teaching Notes

Concepts and approaches

Creating

Creating is about planning, making and evaluating things. In this activity pupils create a simple animation of a joke using ScratchJr. Pupils plan by storyboarding their joke. They then use this simple design to guide them as they code their animation. They write their code using the ScratchJr programming language. Pupils briefly touch upon evaluation when they think about what has worked well or could be improved in their animation.

Programming

In this activity pupils create an animated joke in ScratchJr. In doing so they use their knowledge of the ScratchJr programming language to implement their algorithm, which in the form of a simple storyboard.They debug their program as they check it does as they expect.

Curriculum links

Computing: create and debug simple programs

English: Possible links could be:

Year 1: develop pleasure in reading, motivation to read, vocabulary and understanding by:

- recognising and joining in with predictable phrases
- spell: words containing each of the 40+ phonemes already taught

Year 2: spell by: segmenting spoken words into phonemes and representing these by graphemes, spelling many correctly; learning to spell more words with contracted forms (e.g.who's)

Prior knowledge

Pupils should already have an understanding of algorithms and be familiar with the programming in ScratchJr, perhaps through tinkering, and the concept of programming through the use of simple programmable toys.

Resources (downloadable from webpage)

- iPads for pupils with ScratchJr app
- A means by which to display ScratchJr and the activity presentation as you work through this lesson.
- Paper for storyboards or printed storyboard templates, one per pair.
- Jokes to be animated.e.g http://www.enchantedlearning.com/jokes/topics/knockknock.shtml (Slide 2)
- ScratchJr commands help sheet – up to one per pair of pupils, if required.
- Teacher examples
- Pupil challenge sheet – up to one per pair of pupils, if required.
- Further help: ScratchJr Wiki , ScratchJr Website
- Reminder on how to use ScratchJr

Related activities

KS1 Introduction to algorithms

KS1 ScratchJr tinkering

KS1 or KS2 Scratch tinkering

KS2 Fossil Formation