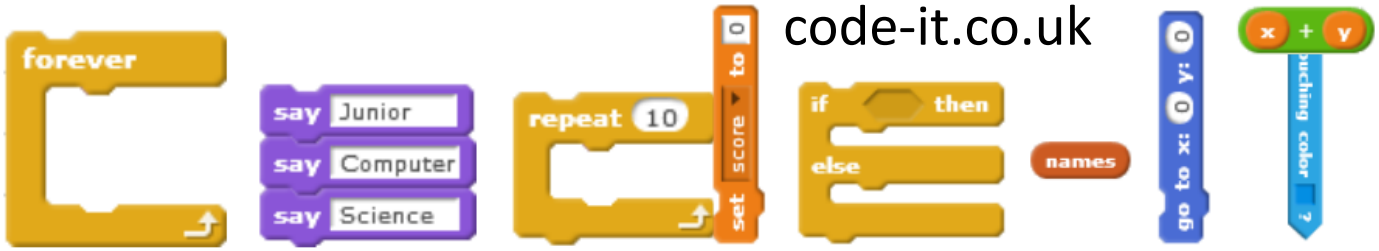


code-it.co.uk

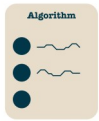
Program Aim: Learn how to decompose and problem solve by creating a simple game where a child travels from country to country in Europe using Scratch 2.0

Travel Europe

Programming Concepts

-Sequence

-Inputs



Computational Thinking

-Algorithmic Thinking

-Generalisation

-Decomposition

National Curriculum Programs of Study

Pupils should be taught to:

-**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**

-**use logical reasoning** to explain how some simple algorithms work and **to detect and correct errors in algorithms and programs**

Formative Assessment

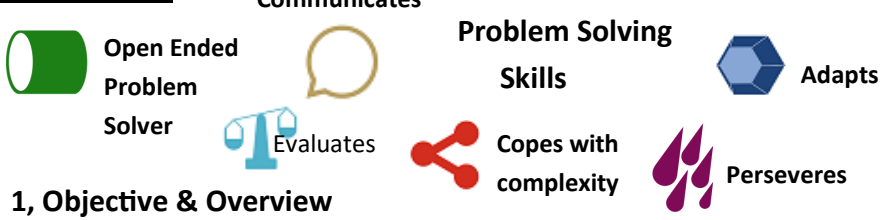
For many activities what constitutes success is defined and how it can be assessed. Pupils that achieve this quickly can, in some circumstances, be moved on to an enrichment activity. Pupils that display a lack of understanding or partial understanding can be directed towards a corrective activity. Teachers can choose to record these on the pupil decompose sheet as **C** (corrective) or **E** (enrichment) helping to build a picture of pupil progress.

Where in KS2

-In my scheme of work it goes in Year 3 (7-8 years old) but it could be used with older pupils as a quick introductory challenge

Decomposing the Game | Programming | Extending the Game

Learning Path



Overlap with other projects

This project covers all the same objectives at the smoking car and magic carpet game so it is not recommended that you do both.

Geography National Curriculum Locate the worlds countries using maps (focus on Europe)

1, Objective & Overview

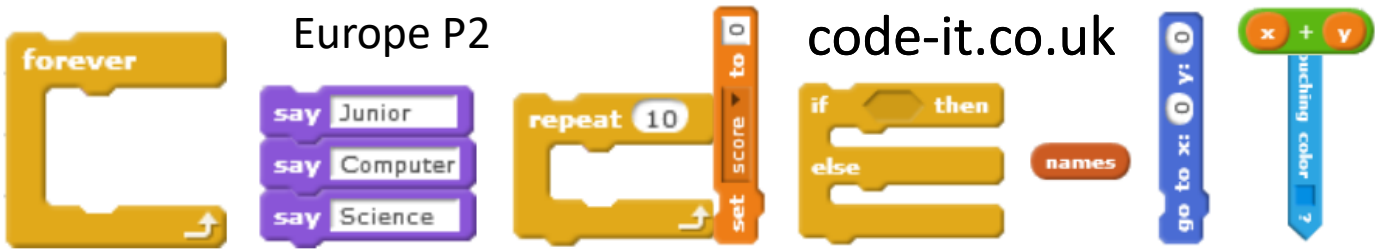
Explain to pupils that they are going to decompose a game by playing it online and working out what objects they need to create and what things the game does.

After they have done this the class is going to build the basics game together before adapting and improving it.

Cross Curricular Focus

This can be adapted to fit in with many topics or projects. It could be a lost cave dweller trying to find their way home without falling from the rock walkway, a bee that has to pollenate each flower or a spaceship that has to visit every planet.

cc BY NC ND



Europe P2 code-it.co.uk -Decomposition

Breaking up a problem into parts and solving the parts separately

I can discover the most important part of the problem

2, Decomposing the Game Either access a copy of the game on the Scratch website or download a copy from code-it. Either demo the game as a whole class or give pupils the link so they can demo it themselves. Don't allow them to look at the code. After a few minutes instruct pupils to fill in the decomposition sheet listing all the objects that they will need to make and circling in pencil all the things that the game does. You may wish to read these out to the class one by one to speed the process up and help poorer readers. Go through their answers using the marking sheet found on the next page.

3, Choose a starting point and justify their choice Ask pupils to decide where they think the best place to start creating the game would be. Get them to discuss it with their neighbours and mark their most important place as a 1 in the box next to the task, 2 for the next most important etc. Can they justify their choice to their friends? You are looking for answers that have clearly considered criteria such as the instructions as *no one will know how to pay the game* or child moving and steering as *there is nothing to do otherwise*. Prioritising is a key part of coping with complexity which is important for decomposition, abstraction and all project management.

4, Programming As you create the game refer back to their decomposition sheet and get them to tick off aspects of the program they have created. In this way you are helping to reinforce the connection between algorithm and code. If pupils are coping really well let them attempt aspects independently.

5, Open the Travel Europe blank game This has a child sprite not available in Scratch 2.0 and a blank map of Europe. You could import your own blank map or let pupils choose their own sprites for greater challenge. Map and sprites were sourced from Pixabay and are copyright free .

Steps You may want to explain that Scratch steps are actually pixels, tiny dots that all screens are made up from

6, Move 10 steps when press the 1 key Explain that they are going to start by making the Europe move when they press the 1 key board input. (It is called an input because it puts information into the computer) Explain that Scratch uses starting blocks which are curved. They can find these in events (Scratch 2.0) or control (Scratch 1.4). Demonstrate how the arrow opens a drop down menu where they can link any key board input to their program. Can they find a move block? Where might it be? Motion blocks. Drag out the move 10 steps and hold it so pupils can see the snap to line. Be slow and deliberate and they will too. Get them all to read the code out loud. Demonstrate how it works by pressing the 1 key. Can they do the same?

List the objects you will need to make

Child, Europe Map, Country names

instructions, aim of the game

Europe Decomposition Sheet Answers

Name _____

Class	I debugged	Teacher

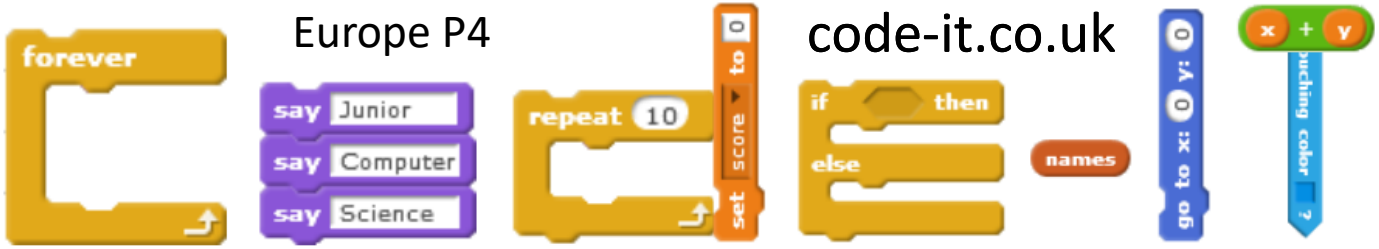
code-it.co.uk

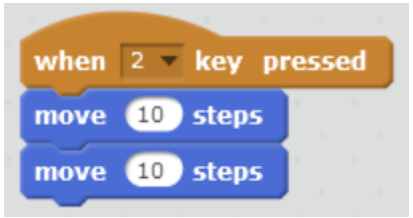
Circle all the things the game does at the moment

- Move when the 1 key is pressed
- Zoom when the z key is pressed
- Move further when the 2 key is pressed
- Darken the background when the k key is pressed
- Turn to the right when right arrow key is pressed
- Spin round when the s key is pressed
- Turn to the left when the left arrow key is pressed
- Grow the child when the g key is pressed
- Leave a trail when d key is pressed
- Stop leaving a trail when the u key is pressed
- Clear all lines when the c key is pressed
- Show instructions when the i key is pressed
- Change to a new background when the x key is pressed
- Shrink the child when the t key is pressed

What would you add to the game?

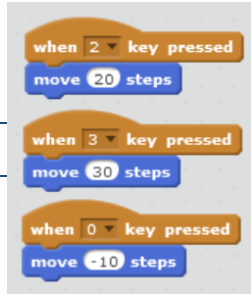
CC BY NC ND

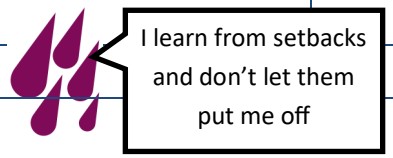




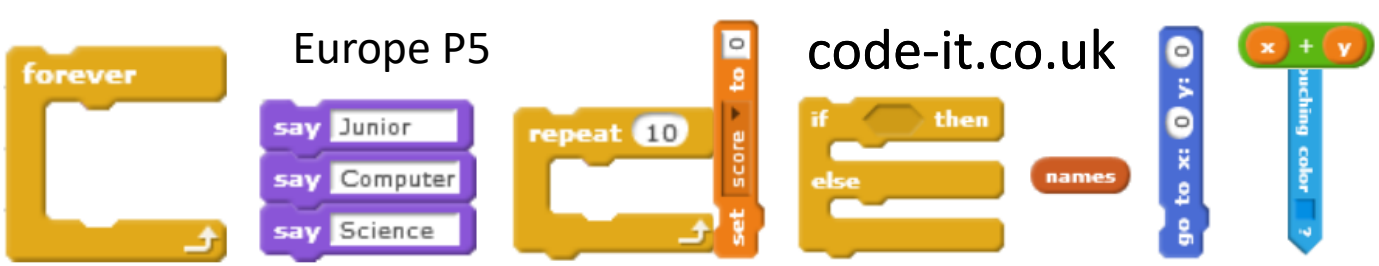
Be sure to point out that there are often more than one way to solve a problem in computing such as move 10 steps twice

Move 10 steps when press the 1 key	**Success? Teacher/Pupil observation** Child moves when the 1 key is pressed. It is not enough to just code, pupils must test this and make the link between the code and the key that activates it.	**Corrective Activity** Can't complete the code because there are no move blocks. They are trying to add the code inside the stage area. Ask them if they are trying to program the stage to move. What are they trying to make move? They need to select the child sprite before programming. Incorrect blocks chosen. Have they read their code out loud? Does it make sense? Is it the same as your example? After giving a hint walk away as this tells the pupil that is their responsibility to debug it.	**Enrichment Activity** Can pupils code the child to move 20 steps when the 2 key is pressed or 30 steps when the 3 key is pressed. Can pupils make the child go backwards when the 0 key is pressed? These new blocks must not be joined to the original when 1 key is pressed move 10 steps block.





7, Steering Can anyone find the steering instructions on the decomposition sheet. Can they read one out? Instruct them to look at the movement blocks as a clue. What parts do they think it will contain? They will need one block to turn right and another to turn left.	**Success? Teacher/Pupil observation** Child turns left when left key is pressed Child turns right when right key is pressed Code has been tested	**Corrective Activity** Ask pupils to read their decomposed instruction out loud. *Turn to the right when right arrow key is pressed.* Ask pupils to underline the key words. Can they find any key words anywhere in the motion blocks (turn)? Point out that right and left can be symbols. You are helping them make the links between algorithm and code.	**Enrichment Activity** Can pupils change 15 degrees and experiment with other numbers? What is the maximum amount of turn that is useful? (360) Before you move on, get pupils to set their turns back to 90 degrees or less.



Europe P5 code-it.co.uk

8, Leaving a trail, stopping leaving a trail and clearing the trail	Success? Teacher/Pupil observation	Corrective Activity	Enrichment Activity
As there is less link between the decomposed instruction and the code, pen down/up are not mentioned directly, it can help to demonstrate the pen up and pen down instructions before challenging pupils to link the C key to clear.	D key puts a pen down. You can tell if this has happened by programming the sprite to move and seeing if a trail has been left behind. U Key stops the pen drawing C Key clears any drawn lines (it won't clear background pictures)	**Common bugs** Not linking it to the correct letter or not changing the letter at all. Instruct pupils to compare their code with yours or read the code out loud. Using the D or U key and thinking that it has not worked because they can't see any line drawn. Can they see the line at the top in Scratch 1.4 or can they drive it away (both versions)? Only a green clear block is available in the pen commands? See move corrective activity on P4	Can pupils create more code blocks to change the colour of the lines. It can help to say, *could you make the g key turn the line green* before extending this into experimentation with other colours I repeatedly experiment through making, testing & debugging

Pen up, down and clear code

D, U & C

You could let pupils choose their own keys as long as a letter carried out one task only .

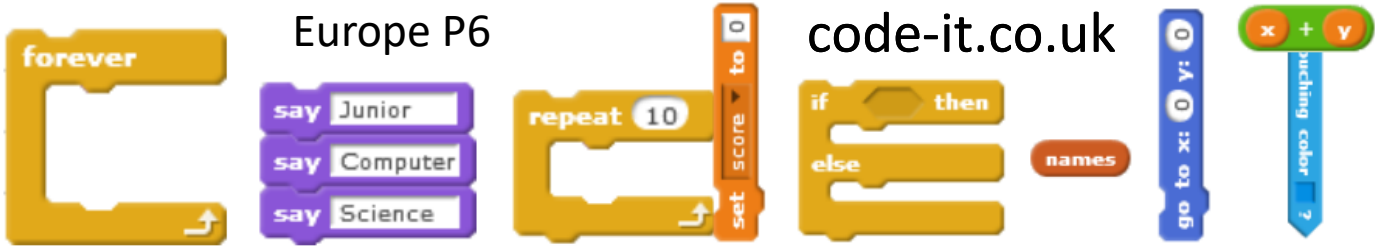
However it helps to draw out the logic of naming linked to what it does and if pupils don't finish the instructions the game can still be played/evaluated by their peers.

In UK note spelling of colour/color

Selecting colour

Selecting Colour

In Scratch 2.0 the arrow changes to a hand. The hand needs to click on another colour found anywhere else to use that colour.



Europe P6 code-it.co.uk Adapt the Map

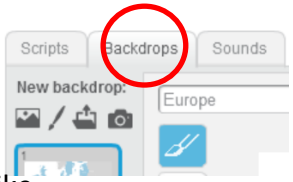
9, Add Country Names to Europe map

Click on stage

Select backdrops

Select the text tool

Click on the map where you would like the text to start

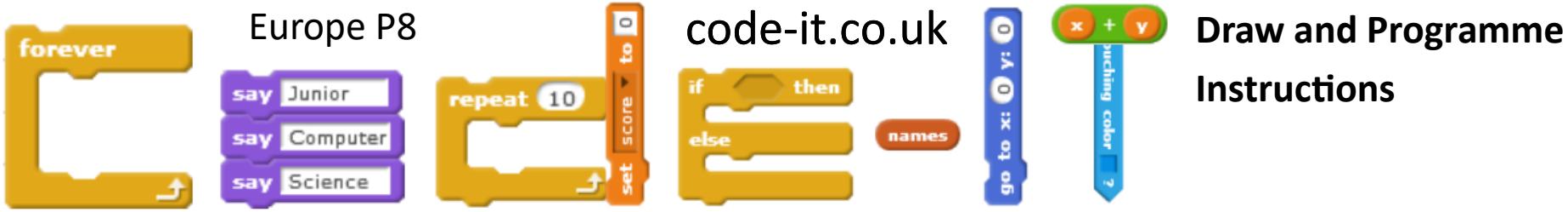


Type name

Click elsewhere on backdrop manipulation points will appear use these to change the size or rotate name

Hold in middle to move name

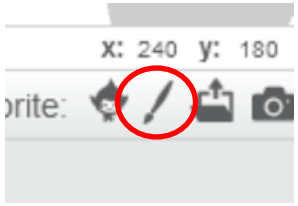
Repeat for next country

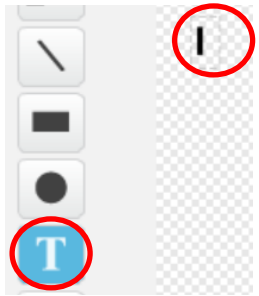


Europe P8 code-it.co.uk

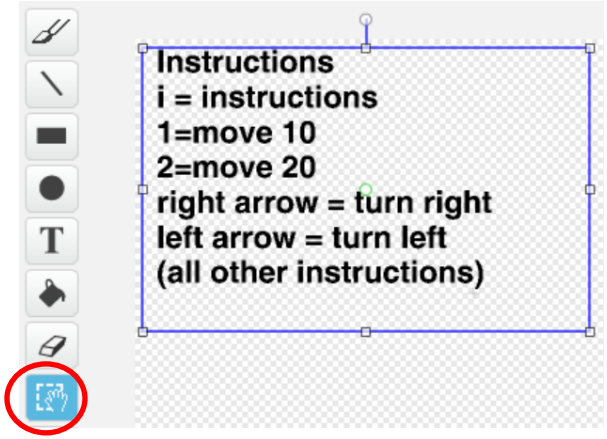
Draw and Programme Instructions

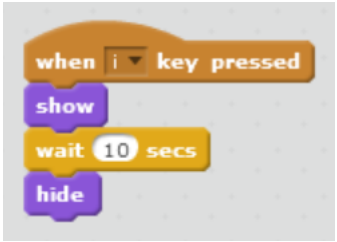
10, Draw instructions Click on the paintbrush icon to paint new sprite,. Select the T text tool. Type in the instructions. If you need to get back to the text at a later date use the select tool.





Select tool



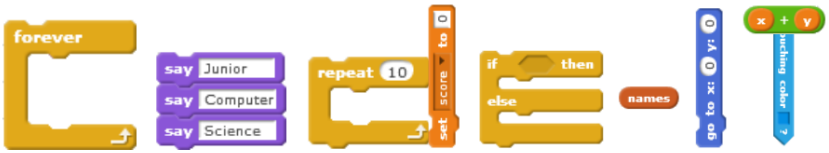
11, Program instructions to appear when 1 key clicked, last for 10 seconds and then hide	**Success? Teacher/Pupil observation**	**Corrective Activity**	**Enrichment Activity**
Ask pupils to listen carefully to the order the instructions are given. Pause between each one. You could display these. When i key pressed show Wait for 10 seconds Hide Give pupils time to puzzle it out without support.	i key brings up the instructions which shows for 10 seconds before hiding automatically. 	Drag out the blocks you would need or provide these in a card format. Code blocks in Scratch 2.0	Could they add a game aim sprite and program it to show itself when a key is pressed, last for x seconds before hiding itself.

12, Extending the game Direct pupils back to their decomposition sheet. Point out the ***what would you add to the game*** section. Explain that these would need to be things that could be triggered by a keyboard input although as they do more programming they will learn more complex skills.

Encourage group discussion with groups of 3 or 4. Remind them of the communication targets. After a time have a class discussion before allowing time to try out their ideas. Before they experiment emphasise that credit will be given for ideas attempted even if they don't ultimately work.

Assessing pupils ability to adapt an idea or try to adapt an idea to create something new	**Success? Teacher observation** Pupils have developed an idea which is different to those taught Pupils can describe an idea they want to create even if they can't fully complete it without support. Praise pupils who try even if they don't fully succeed as computing is all about struggle & persistence in the face of repeated failure.	**Corrective Activity** Drag out a when j key pressed block. Ask pupils what they would like the j key to do? Open the looks or sound blocks. Is there anything in here they could make the j key do?	**Enrichment Activity** Encourage pupils to develop something that has more than one instruction like showing, waiting and hiding the instruction sprite.

14, Summative assessment **Give** each pupil an evaluation sheet and ask them to write their name on the top and leave it on their keyboard. Instruct them to setup their games in presentation mode. They are to move round and sit at someone else's computer, they are not to choose anyone that they are close friend too so they can't be accused of being unfair. Pupils will then have 5 minutes to play and grade aspects of the work leaving a 1 or 2 in each category. 1= Not done or not fully working 2=working . After pupils have moved round three times allow everyone to go back to their computer. Sort out any disagreements. Ask pupils to add up the scores and collect in the sheets to help you form an opinion of the pupils work.



Where Next?

Explore other programming planning at code-it.co.uk/csplanning

Design your own programming planning using computational thinking & problem solving skills

You can follow me on Twitter

@baggiepr

You can find lots of free resources at code-it.co.uk

I enjoy sharing my journey with fellow educators around the world

Summative Assessment Criteria

Move when the 1 key is pressed
Move further when the 2 key is pressed
Turn to the right when right arrow key is pressed
Turn to the left when the left arrow key is pressed
Leave a trail when d key is pressed
Stop leaving a trail when the u key is pressed
Clear all lines when the c key is pressed
Show instructions when the i key is pressed
Change to a new background when the x key is pressed
Background has a start and finish
Something happens when another key is pressed
Instructions are easy to read

Develop Pupils into Computational Thinkers

- Algorithmic thinkers — Think through steps or rules to achieve something
- Algorithmic evaluators — Which algorithm is best, most efficient?
- Decomposers — Break problem up into parts and solve parts separately
- Abstractors — Find the most important part of problem
- Generalisers — Adapt solution from one problem to solve something else

Complexity	I can break complex problems into parts	I can discover / concentrate on the most important part of a problem	I can explain how I used decomposition & abstraction
Ambiguity	I recognise there is more than one way to solve a problem	I recognise there is more than one way to describe a problem	I can explain how I managed ambiguity
Open Ended	I look for a range of solution to the same problem	I don't just accept the first solution	I can describe how a project can be extended
Adapt	I can adapt existing ideas to solve new problems	I can identify patterns in problems & solutions	I can explain how I adapted a solution to solve a new problem
Evaluate	I can evaluate my solutions against a set criteria	I can design criteria to evaluate my creations	I can explain how evaluation helped me improve a project
Experiment & Debug	I can develop, test and debug until a product is refined	I repeatedly experiment through making, testing & debugging	I can explain how using the iterative cycle improves my work
Persist	I can persevere even if the solution is not obvious	I learn from setbacks and don't let them put me off	I can describe how I overcame problems
Communicate	I can contribute useful ideas to a partner or group	I can encourage others to share their ideas	I can lead using all the people talent in my group

Decomposition | Abstraction | Generalisation | Algorithmic Evaluation | Algorithm

Adapted from Behaviour Rubric created with @MarkDorling and linked at http://code-it.co.uk/attitudes/ Adapted with permission