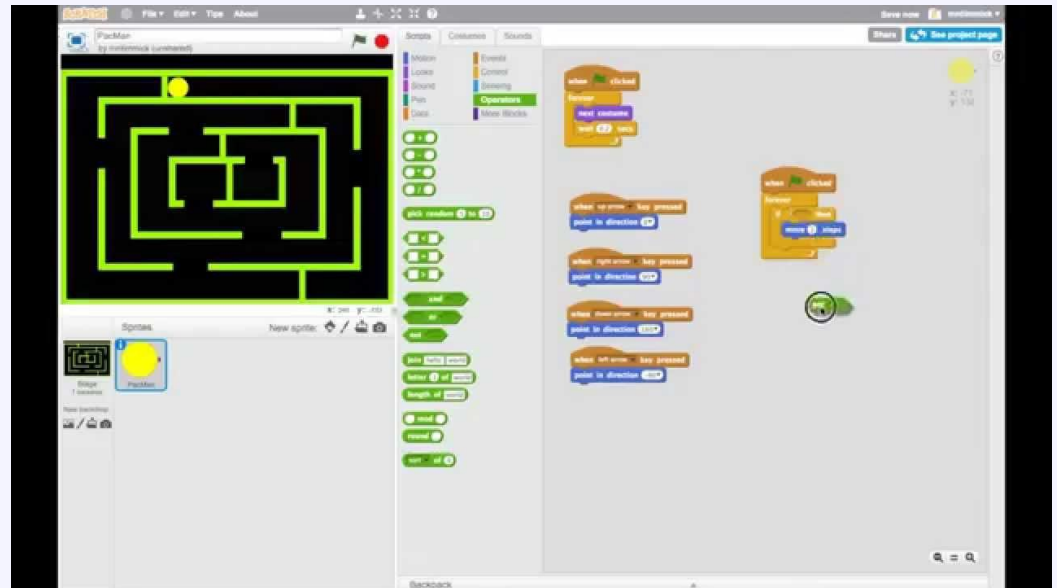


Year 7 Topic 5 – Programming essentials in Scratch – part II

Lesson	Can you?
Lesson 7: You've got the moves!	Define a subroutine as a group of instructions that will run when called by the main program or other subroutines Define decomposition as breaking a problem down into smaller, more manageable subproblems Identify how subroutines can be used for decomposition
Lesson 8: Fly cat fly!	Identify where condition-controlled iteration can be used in a program Implement condition-controlled iteration in a program
Lesson 9: Loop the loop!	Evaluate which type of iteration is required in a program
Lesson 10: Treasure those lists!	Define a list as a collection of related elements that are referred to by a single name Describe the need for lists Identify when lists can be used in a program Use a list
Lesson 11 & 12: Translate this!	Decompose a larger problem into smaller subproblems Apply appropriate constructs to solve a problem

Useful websites

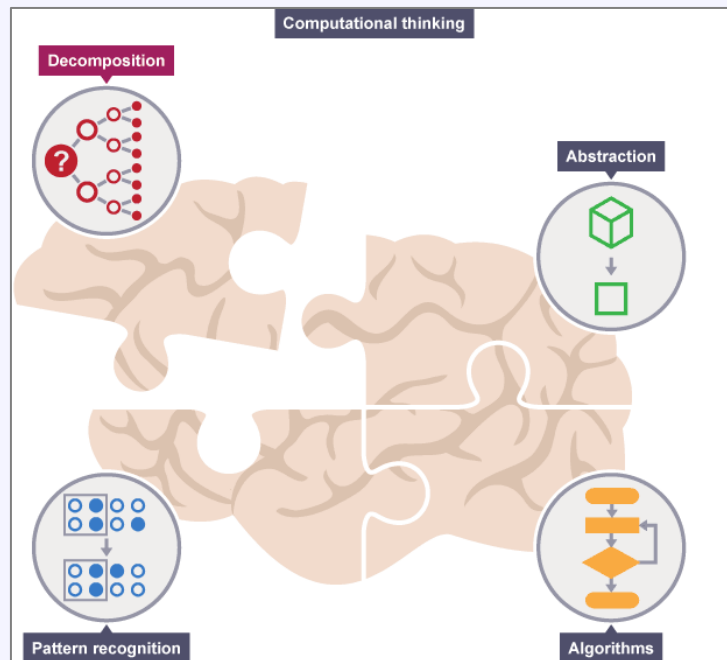
- www.scratch.mit.edu
- www.en.wikipedia.org/wiki/Five_Little_Ducks
- www.en.wikipedia.org/wiki/Software_bug



KNOWLEDGE ORGANISER
Key Stage 3 - COMPUTING

Why is decomposition important?

If a problem is not decomposed, it is much harder to solve. Dealing with many different stages all at once is much more difficult than breaking a problem down into a number of smaller problems and solving each one, one at a time. Breaking the problem down into smaller parts means that each smaller problem can be examined in more detail.



If **computational thinking** techniques are applied to the problem of how to bake a cake, on **decomposing** the problem, the solution would state – among other things – that certain quantities of particular ingredients are needed to make the cake.

The complex problem of the committed crime has now been broken down into simpler problems that can be examined individually, in detail.



Making a plan

It is important to plan out the solution to a problem to make sure that it will be correct. Using **computational thinking** and **decomposition** we can break down the problem into smaller parts and then we can plan out how they fit back together in a suitable order to solve the problem.

