Build a scene key information

Key

Choice

Sequence

Repetition

Variables

VAR

Difficulty

Away from Scratch

Scratch 3.0

Scratch 2.0

Research

Computational Attitudes

Formative Assessment

Summative Assessment

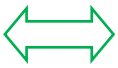
Code-it Gold

Code-it Gold is about using multiple, research supported, methods of teaching programming. It is about giving you choice over delivery and method as each module has four methods to chose from. This means you can vary your approach to adjust the challenge. If you think there is more your pupils can achieve in primary programming but are not sure how then **Code-it Gold** is for you.

Most modules will have an online quiz or test. Help us to learn which methods are most effective by removing any pupil names, indicating which module it came from and emailing it to phil.bagge@code-it.me Ask your head teachers permission first.

The **Code-it Gold** resources on code-it are free but if you need more training or inset then contact Phil Bagge via my contacts page.

Build a scene key information

Choice 

At some points in the planning you will see the choice symbol . This means you can chose an alternative way of teaching and directing learning. There is brief guidance to help you decide if this is the right choice for you in the research sections. Pupils benefit most when we use different approaches across the models.

Concepts  VAR

The earliest modules develop pupils understanding of sequence. The animation modules introduce pupils to repetition and the personalisation module suggests a way that variables could be introduced. The early and middle modules are designed for lower KS2 and the last module for upper KS2.

Difficulty 

Difficulty will refer to both the challenge for pupils and the challenge for teachers. All programming has an element of struggle to it but the author understands that most primary teachers will be non-specialists and will attempt to hold your hand through the process.

Away from Scratch 

When introducing new concepts or planning the best way forward, these are best done away from the programming environment. Research and experience suggests that pupils don't like to plan but planning always improves final outcomes

Scratch 2.0, 3.0  

All planning is created using Scratch 3.0 but where things differ in Scratch 2.0 there are alternate instructions. Just look out for the Scratch 3.0 or 2.0 signs.

Research 

Look out for the research call outs. These will help you understand why some aspects of the planning is offered in different ways. Best evidence suggests that pupils benefit from different approaches so if you decide to use the PRIMM approach in your first

project use a different approach for your second module.

Computational Attitudes

Help pupils to understand what attitudes will enable them to grow in computing stature and recognise their positive choices. There are stickers that you can download, print and give to pupils. See a full range of computational attitudes on the next page.

Assessment

Look out for the assessment symbols to identify aspects you can check during the learning and ways you can enrich or support when needed as well as summative opportunities.

What is inside

Imagine a project that can be followed from cover to cover, or dipped into, adapted, shared across year groups, taught in different ways and which covers multiple computing objectives. Build a scene is designed to do just that.

Mastery

Mastery means very different things to different teachers but if mastery means being able to adapt your teaching approach and learning opportunities to meet differing needs of pupils then the different approaches to the same topic would in time help you to develop a mastery approach.

At a glance

Monologue	Dialogue	Stage & Sound	Animation	Personalising
				VAR
				
Stand alone	Stand alone	Dependant on either monologue or dialogue	Stand alone	Dependant on either monologue or dialogue
				
Versions A Parsons problem B Use modify create C PRIMM D Modelled offline	**Versions** A Parsons problem B Use modify create C PRIMM D Modelled offline	**Versions** A Completion task B Use modify create C PRIMM D Reverse completion E Teacher Demo	**Versions** A Parsons problem B Use modify create C PRIMM D Sequence to Loop	**Versions** A Parsons problem B Completion task C PRIMM

Computational attitudes

I recognise there is more than one way to solve/describe a problem

I don't just accept the first solution

I look for a range of solution to the same problem

I look for how a project can be extended

I can evaluate my solutions against a set criteria

Handles Ambiguity

Open Ended Problem Solver

I can break complex problems into parts

I can design criteria to evaluate my creations

Evaluates

Insert picture of your students here

Copes with Complexity

I can discover / concentrate on the most important part of a problem

I can identify patterns in problems & solutions

I can contribute useful ideas to a partner or group

I can encourage others to share their ideas

Communicates

Adapts

I can adapt existing ideas to solve new problems

I can develop, test and debug until a product is refined

I lead using all the people talent in my group

I learn from setbacks and don't let them put me off

Investigates

Perseveres

I make predictions about what will happen

I can persevere even if the solution is not obvious

I repeatedly experiment through predicting, making, testing & debugging

@baggiepr

Inspired by Behaviour Rubric created with @MarkDorling and linked at http://code-it.co.uk/attitudes/