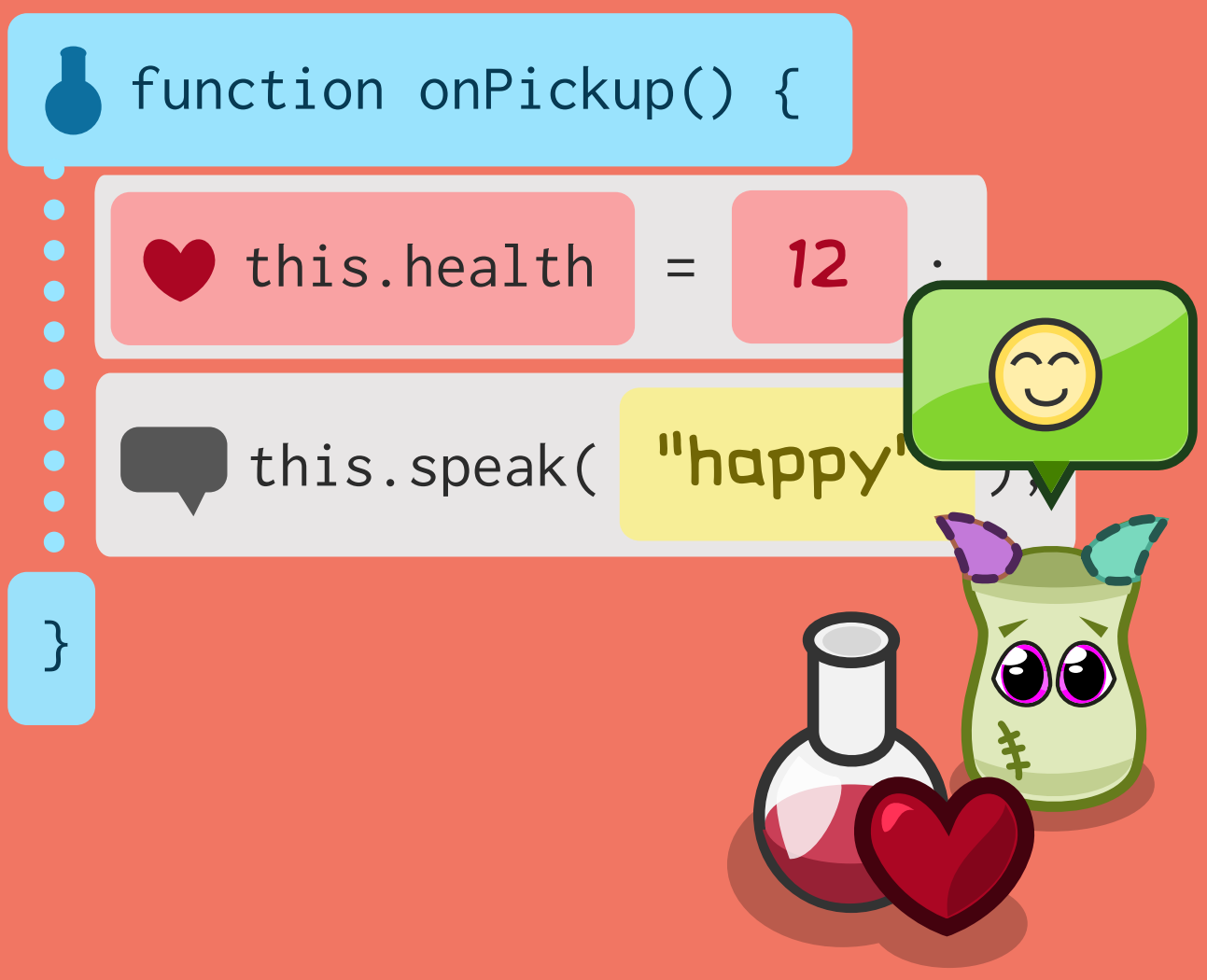
codekingdoms

Code Kingdoms

Unit 2 Learning a Language



for kids, with kids, by kids.

codekingdoms

Resources overview

We have produced a number of resources designed to help people use Code Kingdoms. There are introductory guides to all parts of the product and classroom materials to help teach lessons around Code Kingdoms.

Code Kingdoms Learning: What, where, when and how

A summary of the Code Kingdoms approach to learning.

Teacher Guide

An overview for teachers. Describes the Code Kingdoms learning ethos and details the different parts of the product.

Dashboard Guide

A beginner's guide to using our group management tool. Describes everything from registering for an account to assessing the progress of your kids.

codekingdoms

Sandbox guide

A guide to using our unstructured creation environment. Learn everything from using the menus to making great puzzles.

Unit 1: Introducing Code Kingdoms

An introductory unit of six 'off-the-shelf' lesson plans. Targeted at KS2 kids.

Unit 2: Learning a language

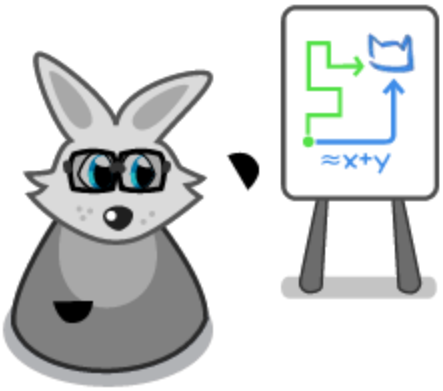
Six 'off-the-shelf' lesson plans designed to teach kids the basic of JavaScript

Puzzle Packs

A guide to building specific puzzles in Creative mode. Step-by-step instructions from start to finish. Four puzzles per pack.

codekingdoms

Contents



codekingdoms

About this guide

This scheme of work contains six one hour lesson plans, a medium term plan for the unit and some accompanying assessment and planning grids. It is designed to focus on learning the rudiments of JavaScript and is best used with those who have done some coding previously. The lesson plans are meant to be flexible, rather than prescriptive, to allow teachers to adapt it as they see fit.

This unit develops skills in three strands of learning: JavaScript, Computational Thinking & Computing. The learning outcomes for each lesson are categorised in these three areas. The unit culminates in the kids building their own map which will allow them to demonstrate their learning throughout the unit.

Resources

To teach this unit you will require the following resources:

- CK School - school.codekingdoms.com
- CK Dashboard - dashboard.codekingdoms.com

You may find our supporting guides useful. They are found at codekingdoms.com/teachers

- Dashboard guide - how to use the teacher management tool to plan and set activities
- Sandbox mode guide - how to use the creation environment to complete lesson plan activities

codekingdoms

Unit Plan

Prior knowledge

This unit assumes that kids have had some minimal prior exposure to Code Kingdoms. They will have registered for an account and will be familiar with the Sandbox mode.

Curriculum-linked objectives

1. Understand the opportunities networks offer for communication and collaboration.
2. Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems.
3. Understand how data of various types (including text, sound and pictures) can be represented and manipulated digitally.
4. Design, write and debug programs that accomplish specific goals; solve problems by decomposing them into smaller parts.

codekingdoms

Individual Lesson Outcomes

	Lesson Title	Outcomes	
1	Job of a coder	YA1	I know that algorithms are implemented on digital devices as programs. Recognise coding as the use of programming languages to make games, programs and websites among other things. Appreciate how diverse the use of code is. Summarise what coding is in one clear sentence.
2	JavaScript: Understanding the basics		Describe what JavaScript is and what it can do. Understand the term “object-oriented”. Explain where and how JavaScript is used. Explain why JavaScript is a good language to learn.
3	JavaScript: Using values and functions	1.3	I understand how to use JavaScript functions to achieve my aims.
		1.4	I understand the difference among different types of values and can explain in what situations I might use each.
		D1	Breaking down artefacts into constituent parts to make them easier to work with;

codekingdoms

	Lesson Title	Outcomes	
4	JavaScript: Adapting a program	1.3	I understand how to use JavaScript functions to achieve my aims.
		1.4	I understand the difference among different types of values and can explain in what situations I might use each.
			Plan how to use algorithms in their own programs.
		YA3	I can use logical reasoning to predict outcomes.
		YP2	I can use logical reasoning to predict the behaviour of programs.
5	JavaScript: Writing a simple program	1.3	I understand how to use JavaScript functions to achieve my aims.
		1.4	I understand the difference among different types of values and can explain in what situations I might use each.
		E1	Assessing that an algorithm is fit for purpose;
		E2	Assessing whether an algorithm does the right thing (functional correctness);
		YA3	I can use logical reasoning to predict outcomes.
		YP2	I can use logical reasoning to predict the behaviour of programs.

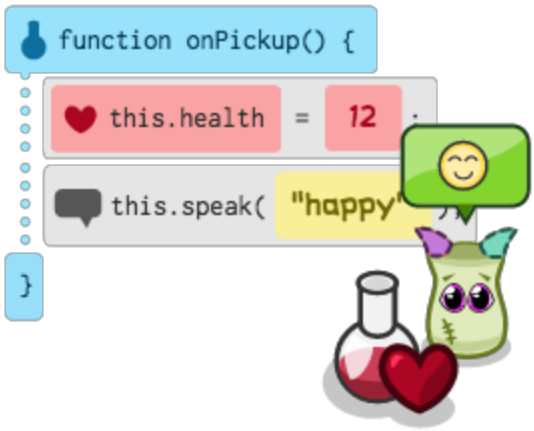
codekingdoms

	Lesson Title	Outcomes	
6	JavaScript: Debugging a program	1.3	I understand how to use JavaScript functions to achieve my aims.
		1.4	I understand the difference among different types of values and can explain in what situations I might use each.
		E2	Assessing whether an algorithm does the right thing (functional correctness);
		E3	Designing and running test plans and interpreting the results (testing);
		YA3	I can use logical reasoning to predict outcomes.
		YA4	I can find and correct errors i.e. debugging, in algorithms.
		YP3	I can find and correct simple semantic errors i.e. debugging, in programs.

Lesson One

Job of a coder

	Lesson Title	Outcomes	
	Job of a coder	YA1	I know that algorithms are implemented on digital devices as programs. Recognise coding as the use of programming languages to make games, programs and websites among other things. Appreciate how diverse the use of code is. Summarise what coding is in one clear sentence.

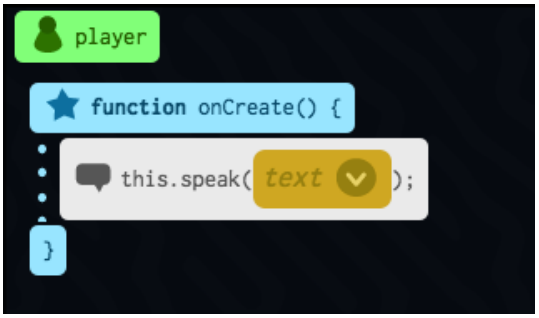


codekingdoms

Lesson Activities

Activity	Notes
Introduction *(15-20 mins)* 1. Discussion: How important is coding? Ask the kids to think about whether coding is the most important language in the world and why before watching: "Is Code the Most Important Language in the World?" https://www.youtube.com/watch?v=Vxv0-sggnqA 2. Discuss the following key questions after viewing: ○ What things did they say and show to change people's understanding of coders and coding? ○ Can you list as many cool places you can work or cool jobs you can have that use coding? 3. Ask the kids to summarise the importance of coding using the following framework: "Learning to code is as important as learning to read and write." I agree/disagree because...	○ This can be constructed as you see fit. It can be a whole class discussion, some discrete think, pair, share activities or part of a presentation to the class.
What types of people code? *(20 mins)* 1. How do we feel about the stats in the film? a. Only 20% of US developers are female b. Only 12% of US Computer Science degrees are awarded to women c. Just 5% of all developers are African American 2. Should this be the case or should there be more variety in the people that code? 3. Why do we think the majority of coders are white males? 4. Is there any reason why other types of people shouldn't code?	○ You may want kids to have written responses for these questions before discussing as a class.

codekingdoms

Activity	Notes
Plenary *(10 mins)* 1. Using the Sandbox mode of Code Kingdoms, the kids summarise what coding is in one clear sentence by making their character speak the sentence. 2. The kids type their sentence in place of "text".	This would be structured in the following way: 

codekingdoms

Lesson Two

JavaScript: Understanding the basics

	Lesson Title	Outcomes
	JavaScript: Understanding the basics	Describe what JavaScript is and what it can do. Understand the term “object-oriented”. Explain where and how JavaScript is used. Explain why JavaScript is a good language to learn.

codekingdoms

Lesson Activities

Activity	Notes
Introduction *(10 mins)* What is JavaScript and what can it do? 1. Introduce JavaScript as the programming language of Code Kingdoms and one of the most commonly used languages in the world. 2. In Sandbox mode, ask the kids to explore the Pieces tab, the Animals tab and the Sequencer to find out all the different things JavaScript can do in Code Kingdoms. 3. Note down a few examples each and then pool them as a class. Some examples include: ○ Make characters move in different directions ○ Make characters speak and jump ○ Make objects control others (e.g. buttons and spikes) ○ Play music and sounds	
Object-oriented programming *(10 mins)* 1. Get the kids to think about how objects in Code Kingdoms interact by asking the following questions: ○ What objects do we have in Code Kingdoms? (List 5 of them) ○ How might they interact with each other? (List 3 interactions - e.g. walk towards each other, Glitches attack player, open a cage)	○ We are leading them to the idea that object-oriented programming is all about objects interacting with each other. ○ In CK these objects are the characters and pieces placed in the worlds, which makes it easy to demonstrate how this works.

codekingdoms

Activity	Notes
Coding interactions *(20 mins)* 1. The kids should write the code shown below, so they end up with a character patrolling back and forth. Instead of using drag and drop they should attempt to write the lines of JavaScript with the slider in the Sequencer on the far right. 2. Where they have to include the 2 directions of patrolling (e.g. North and South) the first might be done as drag and drop and the second written using the first as a guide.  	○ This activity can be adjusted based on the prior experience of your class. ○ The syntax (use of symbols) will likely cause difficulties so perhaps spend plenty of time highlighting how they are used before allowing them to begin writing their own lines of code.
Plenary: Reflect upon using JavaScript (10 mins) 1. What was easy / hard when using JavaScript? 2. How did drag and drop compare to text-based input? 3. Why is JavaScript a good language to learn, especially when making games?	○ Lead them to the idea that the effects of their code are easily observed when the characters interact during play.

codekingdoms

Lesson Three

JavaScript: Using values and functions

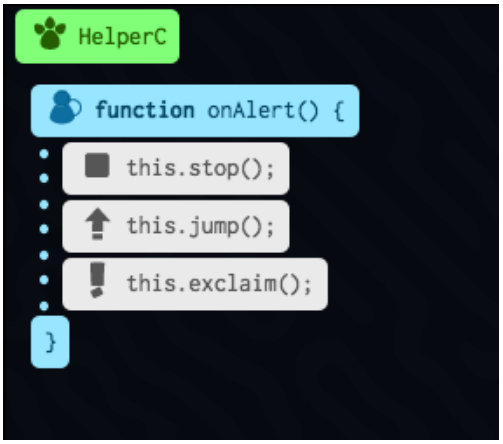
	Lesson Title	Outcomes	
3	JavaScript: Using values and functions	1.3	I understand how to use JavaScript functions to achieve my aims.
		1.4	I understand the difference among different types of values and can explain in what situations I might use each.
		D1	Breaking down artefacts into constituent parts to make them easier to work with;

codekingdoms

Lesson Activities

Activity	Notes
Introduction *(5-7 mins)* 1. Introduce the concept of values by defining each one with real-world examples: a. String - written text e.g. "Welcome to Code Kingdoms" b. Number - any real-world examples of numbers! c. Booleans - these set variables to either true or false e.g. light on = true or door closed = false - named after Mathematician George Boole	
Manipulate values in Sandbox *(10 mins)* 1. Go into Sandbox and ask the kids to create scenarios with each value type. ○ E.g. walking speed, visible = true / false and spoken text by a character. 2. The kids should observe the effects of manipulating number, string and Boolean values and describe the results.	○ Once kids are manipulating their existing code they could be encouraged to do this with the Sequencer slider over to the far right.

codekingdoms

Activity	Notes
Explaining functions *(15 mins)* 1. Outline what a function is and a couple of examples of what they do in Code Kingdoms. 2. The kids identify and list some of their own Code Kingdoms functions - they can attempt to explain what each one does if possible.	○ A function is an instruction - you can think of it like a verb. Functions break down big tasks into smaller instructions, your game maps will be made up of lots of functions.
Plenary: Deconstructing Behaviour *(15 mins)* 1. In Sandbox, create a character with the behaviour shown on the right. 2. Demonstrate the behaviour of the character and ask kids to break it down into its different functions. They can do this by viewing the code attached to the character in the Sequencer. 3. Functions: a. Stop b. Jump c. Exclaim	 ○ Explanation of the term deconstruction may also be required here.

codekingdoms

Lesson Four

JavaScript: Adapting a program in Code Kingdoms

	Lesson Title	Outcomes	
4	JavaScript: Adapting a program	1.3	I understand how to use JavaScript functions to achieve my aims.
		1.4	I understand the difference among different types of values and can explain in what situations I might use each.
			Plan how to use algorithms in their own programs.
		YA3	I can use logical reasoning to predict outcomes.
		YP2	I can use logical reasoning to predict the behaviour of programs.

codekingdoms

Lesson Activities

Activity	Notes
Introduction *(5-7 mins)* 1. To recap the value types ask the kids to identify different value types as either number, string or Boolean: a. A character's ability to swim - Boolean b. A character's speed - Number c. A character's spoken dialogue - String d. Whether a character is visible - Boolean e. A character's nearDistance - Number	
Build puzzles containing values *(10-12 mins)* 1. In Sandbox mode, kids should create three mini puzzles / behaviours that demonstrate functions and values.	○ The map doesn't have to be a coherent game but should demonstrate each value type
Altering an existing map *(15 mins)* 1. The kids should share their map with a partner and ask them to alter the code to make observable changes. They should edit: ○ 3 different types of values ○ 2 different function types 2. They should be able to describe to a partner what they changed and how that impacted the map.	○ Because the kids are manipulating existing code they could be encouraged to do this with the Sequencer slider over to the far right.

codekingdoms

Activity	Notes
Plenary *(20mins)* 1. The kids will need to build their own map in the next lesson. They should begin planning how they will use different functions and values and what will be included in their map.	○ They should complete the lesson with a written plan that may include a sketch of the map. There are a number of documents found in the appendix of this pack to assist with planning maps and algorithms: ○ Planning algorithms sheet ○ Planning maps sheet ○ The 3Ws ○ The OOSY Method

codekingdoms

Lesson Five

JavaScript: Writing a simple program

	Lesson Title	Outcomes
5	JavaScript: Writing a simple program	**1.3** I understand how to use JavaScript functions to achieve my aims. **1.4** I understand the difference among different types of values and can explain in what situations I might use each. **E1** Assessing that an algorithm is fit for purpose; **E2** Assessing whether an algorithm does the right thing (functional correctness); **YA3** I can use logical reasoning to predict outcomes. **YP2** I can use logical reasoning to predict the behaviour of programs.

codekingdoms

Lesson Activities

Activity	Notes
Introduction *(10 mins)* 1. The kids should review their plan from the previous lesson and add to it if necessary. 2. Once they have their plan in place they can begin building their map.	○ If the kids are struggling to think of the algorithms they will need then it may be helpful for them to create their map with the environment and the objects first and then plan their algorithms. ○ Seeing the objects all placed in the world in front of them may help them to better visualise the algorithms that will be required.
Building their map *(20-25 mins)* 1. Their map should demonstrate behaviours using different values and functions. 2. Values that could be used are: a. Visibility b. Speeds c. Help text	
Writing their own lines of code *(15 mins)* 1. Kids should attempt to write a few lines of code using a text input rather than drag and drop. This is done with Sequencer slider moved to the far right. 2. This could be done once the majority of the map is in place, perhaps as an extension activity.	○ The kids will likely need this scaffolded for them. To do this they could create a line in drag and drop and use that as an example when writing their own underneath.

codekingdoms

Activity	Notes
Plenary *(15 mins)* 1. The kids evaluate their map using the map checklist and then peer review a partner's work.	

codekingdoms

Lesson Six

JavaScript: Debugging a program

	Lesson Title	Outcomes
6	JavaScript: Debugging a program	**1.3** I understand how to use JavaScript functions to achieve my aims. **1.4** I understand the difference among different types of values and can explain in what situations I might use each. **E2** Assessing whether an algorithm does the right thing (functional correctness); **E3** Designing and running test plans and interpreting the results (testing); **YA3** I can use logical reasoning to predict outcomes. **YA4** I can find and correct errors i.e. debugging, in algorithms. **YP3** I can find and correct simple semantic errors i.e. debugging, in programs.

codekingdoms

Lesson Activities

Activity	Notes
Introduction *(7 mins)* 1. Introduce the concepts of bugs and debugging with comparisons with proofreading written pieces of work and making improvements. 2. Tell kids they will be creating some bugs in a map and then attempting to debug each other's code. Ask them to think why debugging might be important?	○ Debugging is the process of finding errors or problems with your code and trying to fix it. ○ Sometimes code will be in the wrong order or there could be bits of code missing, the process of fixing the code is called debugging.
Creating Bugs *(10-15 mins)* 1. The kids need to open the map they created in Sandbox during the previous lesson and change parts of the code so that it no longer works as it should - much like a hacker would. 2. This could be by altering or adding functions, values or parameters - it should not simply be deleting lines of code as this will prevent debugging. 3. Once complete, ask the kids to think about how bugs might be created (they are not normally deliberately added). Lead them to the idea that it could be through typing or copying errors by the coder.	○ They could be limited to 5-7 bugs so that they can all be found by the debugger. ○ The kids should make a note of bugs they've added to check whether the debugger has found them all later in the lesson.
Debugging Challenge *(20-25 mins)* 1. The kids should swap their newly-bugged map with a partner and ask the partner to try and find and correct the bugs in their code. Success will be determined by the map working as it was originally designed.	

codekingdoms

Activity	Notes
Plenary *(5 mins)* 1. Ask the kids to think why it might be useful for coders to collaborate when debugging and what was easy / hard about the process of debugging.	

codekingdoms

Map Creation Checklist

Tick off the following when you have included each one in your plan and your completed map design. The peer review column is for the person checking your map.

	In your plan?	In your map?	Peer review
You have chosen your terrain			
There is a cage to end the map			
You have an obstacle that uses one line of code ○ E.g. spikes lower when a button is pressed			
You have an obstacle that uses more than one line of code ○ E.g. spikes lower when a button is pressed and raised when depressed			
You have an obstacle that uses a while loop ○ E.g. a patrolling Glitch that doesn't stop			
You have sequenced your code commands correctly			
All your obstacles have solutions ○ E.g. Glitches can be chased away			
You have decorated your map using trees, shrubs, etc.			
Your map can be completed by a player			

What do you think are the best parts of your map?

What do you think could be improved?

What was the most complicated part of your map?

Describe 3 great things about this map.

Describe 3 things that could make this map even better.

codekingdoms

Map Creation Assessment Sheet

Student:	Red	Amber	Green
has designed their terrain well			
has placed a cage at the end of their map (Objective)			
has coded an obstacle using a simple algorithm ○ E.g. spikes lower when a button is pressed			
has coded an obstacle using a more complex algorithm ○ E.g. spikes lower when a button is pressed and raised when depressed			
has coded behaviour using a while loop ○ E.g. a patrolling Glitch that doesn't stop			
has correctly sequenced commands in an algorithm			
has provided solutions to ALL their obstacles			
has considered the aesthetics of their map			
has created a map that can be played and completed			

The really great things about your map are:

Next time you need to improve:

codekingdoms

Planning Your Own Map

You can use the OOSY Method poster to help you design your map.

What is the objective of your game?

What objects (characters) will there be in your game and what will they do?
E.g. Giant Glitch – chases player around the map and aims to destroy it.

What obstacles will there be in your game?

What solutions will there be for the obstacles?

codekingdoms

Planning Algorithms

When you create your map, the different objects within your game will need instructions to tell them what to do. Use this worksheet to help you consider what those instructions might be. An example has already been done for you.

Object	When?	Who?	What?
ButtonA	onPress	SpikesA	Lower

codekingdoms

The OOSY Method

When faced with a blank template and the task of creating a Code Kingdoms map many students will find it difficult to create a playable map.

The OOSY Method will assist students' planning by scaffolding the method of building Code Kingdoms maps.

OBJECTIVE	First a student must think of an objective for their map. For simplicity in our maps the player always has to reach the rocket.
OBSTACLE	Now the pupil should put an obstacle in the way of reaching the objective that makes the map impossible to complete. E.g. placing spikes on a bridge as an impasse.
SOLUTION	Now think of a solution that will let the player get round the obstacle. E.g. if the player places a lot of Glitches then they could now place a net which the player can pick up and catch them with.
YOUR TURN!	Share the map with friends to see if they can complete it!

codekingdoms

The 3Ws

To assist students in thinking about structuring lines of code, we recommend using the 3Ws (When? Who? What?)

When?	onCreate
Who?	Glitch A
What?	walk towards player

This will allow students to create an algorithm where a Glitch will walk towards a player when it is created, which will look like:

```
GlitchA.walkTowards(player);
```

This method help students to sequence their commands logically.