

Puzzle Pack 1



for kids, with kids, by kids.

About this guide

This guide is for teachers and club volunteers and provides a solution to completing the puzzles. For this reason, **this document should not be shared with children**. This document also outlines the learning objectives of each puzzle.



Introduction

In this project, children will learn to use basic JavaScript functions in blocks of code to build game puzzles in their kingdom. To give the activities some context, the children are building puzzles to protect their land from invasion by Glitches. Each puzzle should be solvable so that friendly animals can move around the player's kingdom.

Resources

This project requires the use of Code Kingdoms, which is best used in Google Chrome. You can find the website at codekingdoms.com/codeclub

Learning Objectives

JavaScript

Basic functions: I understand how to use JavaScript functions to achieve my aims.

Conditional expressions: Using my knowledge of comparison operators, values, and booleans, I can write a conditional expression that results in true or false.

Computational Thinking

A1: Writing instructions that if followed in a given order (sequences) achieve a desired effect.

A5: Writing instructions that repeat groups of constituent instructions (loops/iteration) to achieve a desired effect;

Computing Progression

PA2: I know that computers need precise instructions.

PP2: I can create a simple program.

YA2: I can design simple algorithms using loops, and selection i.e. if statements.

YP1: I can use arithmetic operators, if statements, and loops, within programs.

Challenges



Pressy Button

Introductory code a button puzzle



Crate Weight

Code a more complex button puzzle and use other puzzle pieces



Cliffhanger

Incorporate the use of terrain tools into your puzzle



Young's Buttons

Coordinate your button puzzle with the use of IF statements

Frequently Asked Questions

Q. What other help is there for the player?

- a. To help visualise what a puzzle should look like, players can click on the Question Mark icon to see a completed version.



- b. If a player is unsure which steps of the puzzle they have completed or what still needs doing, they can click on the checklist icon to see a list of the required steps.



Q. What is the purpose of building puzzles?

- a. Building puzzles is a defence against invading Glitches. The puzzles, however, must be solvable by friendly animals so they can move freely around your kingdom. Glitches aren't very clever so if a puzzle is coded well they are unlikely to be able to solve it. The general rule is if an animal tester can bypass a puzzle piece (e.g. catapult) to reach the checkpoint then a Glitch will easily get past the puzzle.

Pressy Button

Basic Functions

A1

PA2

PP2



Description

This basic puzzle introduces the mechanics of building puzzles and as such is quite straightforward. It explains how to place puzzle pieces (e.g. a button) and using the coding interface.

Design Tip

Encourage players to place their button in a location that will be tricky for Glitches to find. If their puzzle doesn't keep the Glitches out it should at least hamper their progress.

Steps to complete

1

Place the Blue Button in the puzzle area.

2

Add some code to the Blue Button for when it is pressed down. This is achieved using the `onPress` event.

3

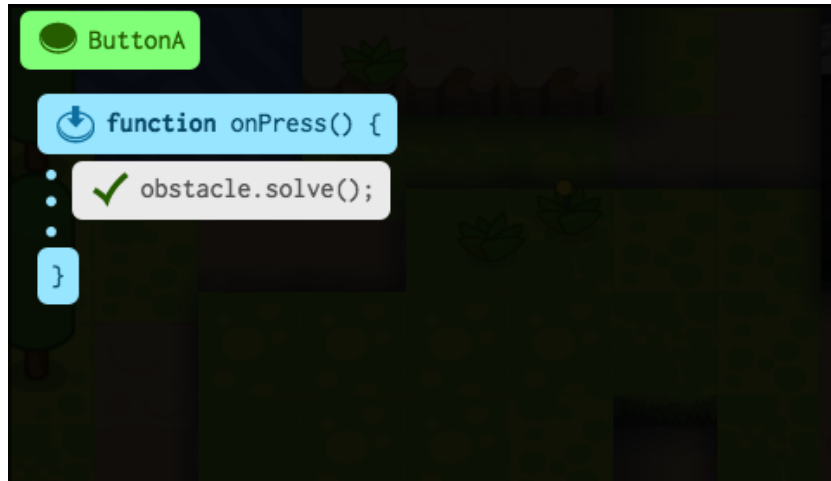
The obstacle should be solved by pressing the Blue Button so the chunk of code needed is `obstacle.solve()`

4

Test the puzzle by pressing the play icon in the bottom right corner of the screen. If the tester animal fails to solve the puzzle, there is an opportunity to edit and improve the puzzle.

Code required to complete the puzzle

When complete, the Blue Button should have the following code:

A screenshot of a code editor interface. At the top, there is a green button labeled 'ButtonA'. Below it, a blue button contains the code for an 'onPress' function. The code is:

```
function onPress() {  
  obstacle.solve();  
}
```

Skill Development

JavaScript

Basic functions: I understand how to use JavaScript functions to achieve my aims.

Computational Thinking

A1: Writing instructions that if followed in a given order (sequences) achieve a desired effect.

Computing Progression

PA2: I know that computers need precise instructions.
PP2: I can create a simple program.

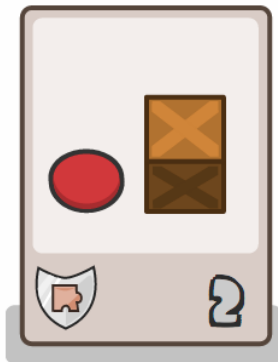
Crate Weight

Basic Functions

A1

PA2

PP2



Description

Crate Weight introduces Red Buttons which are slightly more complex than Blue Buttons. They require two lines of code to work and an object, like a crate, to hold them down.

Design Tip

Ensure that the crate is on the same level as the button as crates can't be pushed onto higher terrain.

Steps to complete

1

Place the Red Button and the crate in the puzzle area.

2

Add some code to the Red Button for when it is pressed down.

3

Red Buttons also need code for the event `onRelease`. This event should 'unsolve' the puzzle.

4

Test the puzzle using the play icon in the bottom right corner of the screen.

Code required to complete the puzzle

The Red Button should be coded similarly to the Blue in Pressy Button. It should look like that shown below:



```

// When the button is pressed.
// Write instructions to run when the button is pressed down.

function onPress() {
  obstacle.solve();
}

// When the button is released.
// Write instructions to run when the button rises up again.

function onRelease() {
  obstacle.unsolve();
}

```

Skill Development

JavaScript

Basic functions: I understand how to use JavaScript functions to achieve my aims.

Computational Thinking

A1: Writing instructions that if followed in a given order (sequences) achieve a desired effect.

Computing Progression

PA2: I know that computers need precise instructions.
PP2: I can create a simple program.

Cliffhanger

Basic Functions

A1

PA2

PP2



Description

Cliffhanger is identical to Crate Weight in the way it must be coded, but also introduces the player to terrain tools. Terrain tools are integral to building difficult to solve puzzles through complementing the player's code.

Design Tip

It is possible to use more than one block of each terrain to make your puzzle more distinct from the example shown by the Question Mark icon.

Steps to complete



Build a hill using the terrain tools tab.



Place the Red Button and the crate in the puzzle area.



Add some code to the Red Button for when it is pressed down.

4

Red Buttons also need code for the event `onRelease`. This event should 'unsolve' the puzzle.

5

Test the puzzle using the play icon in the bottom right corner of the screen.

Code required to complete the puzzle

The Red Button should have identical code as in `Crate Weight`. An example of a built hill is shown below.



Skill Development

JavaScript

Basic functions: I understand how to use JavaScript functions to achieve my aims.

Computational Thinking

A1: Writing instructions that if followed in a given order (sequences) achieve a desired effect.

Computing Progression

PA2: I know that computers need precise instructions.
PP2: I can create a simple program.

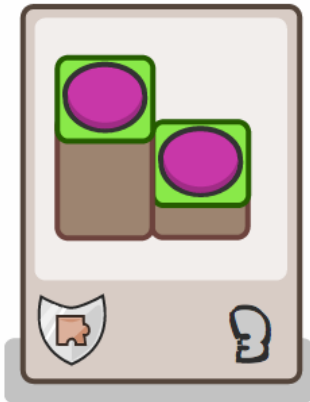
Young's Buttons

Conditional Expressions

A5

YPI

YA2



Description

Young's Buttons introduces the player to IF statements. Both Purple Buttons must be held down at the same time for the obstacle to be solved - a coded IF statement will check for this condition.

Design Tip

Ensure the tester is able to push the crates onto your buttons without getting stuck.

Steps to complete



Build two trenches using the brick and dirt blocks.



Place the Purple Buttons and the crates in the trenches.



Starting with Button A, code an IF statement that requires both buttons to be pressed down to solve the obstacle.

Repeat for Button B.

4

Code the obstacle to be 'unsolved' when the buttons are released (same as for Red Buttons).

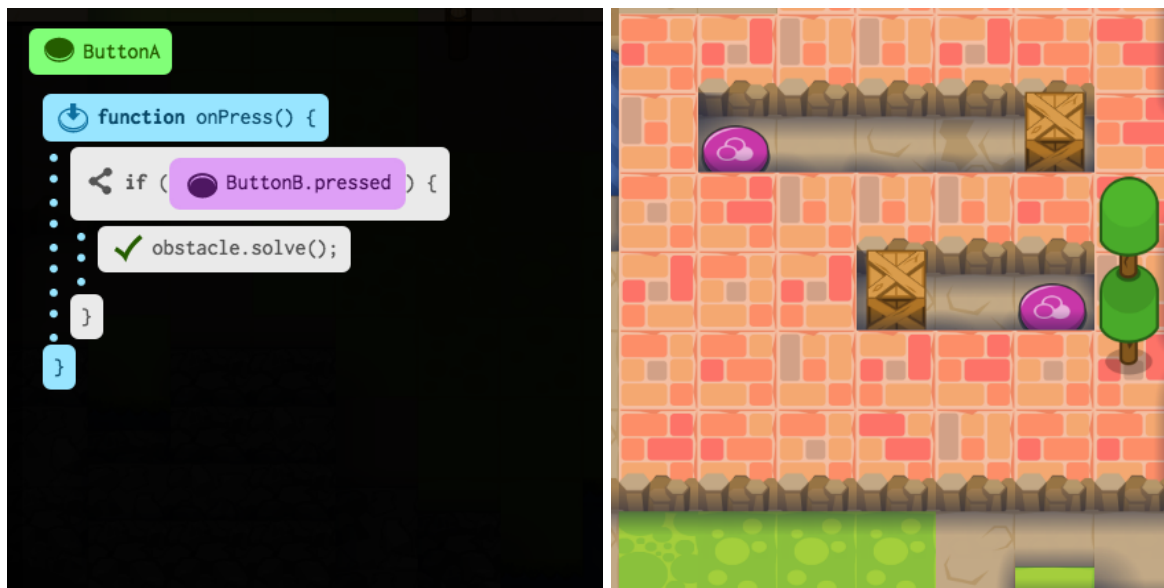
5

Test the puzzle using the play icon in the bottom right corner of the screen.

Code required to complete the puzzle

To create an IF statement for each button you will need to write the code shown below.

IF statements are found under the languages tab when in the Sequencer. Once placed in the Sequencer the condition, in this case `ButtonB.pressed`, can be dropped inside the IF statement.



Skill Development

JavaScript

Conditional expressions: Using my knowledge of comparison operators, values, and booleans, I can write a conditional expression that results in true or false.

Computational Thinking

A5: Writing instructions that repeat groups of constituent instructions (loops/iteration) to achieve a desired effect.

Computing Progression

YP1: I can use arithmetic operators, if statements, and loops, within programs.

YA2: I can design simple algorithms using loops, and selection i.e. if statements.