

Puzzle Pack 1

Notes for Volunteers



for kids, with kids, by kids.

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



Place the Blue Button in the puzzle area.



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



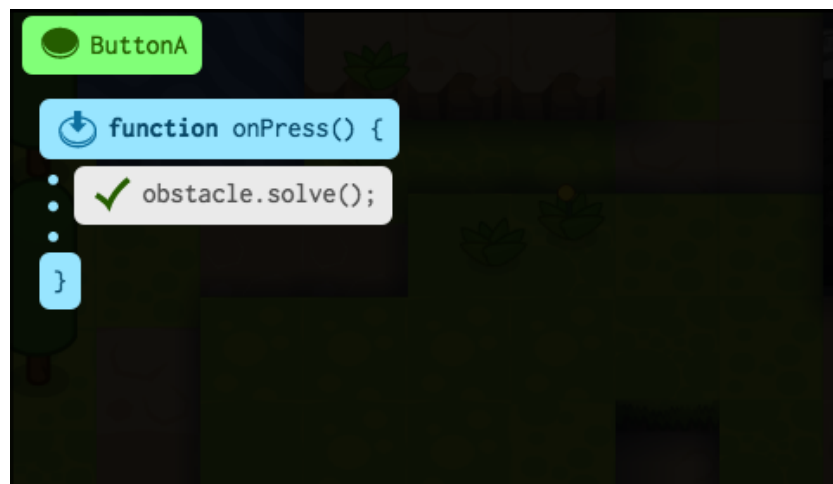
The obstacle should be solved by pressing the Blue Button so the chunk of code needed is `obstacle.solve()`



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:



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.

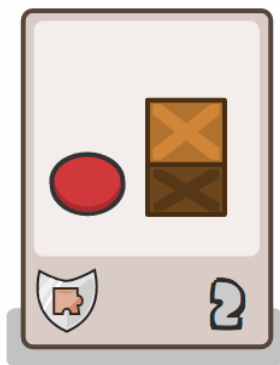
Crate Weight

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



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



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



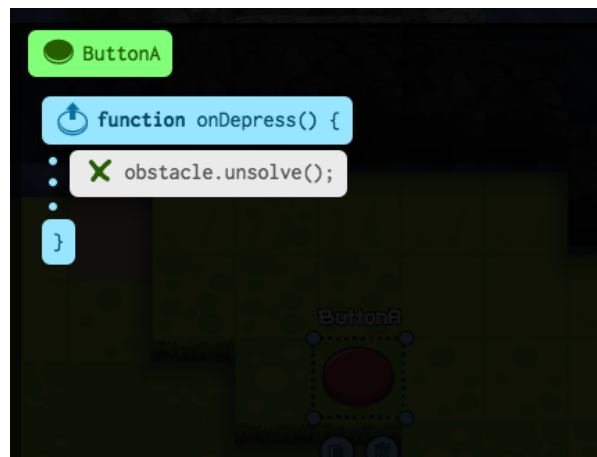
Red Buttons also need code for the event `onDepress`. This event should 'unsolve' the puzzle.



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 to the Blue in Pressy Button and the additional code 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.

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 `onDepress`. 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.

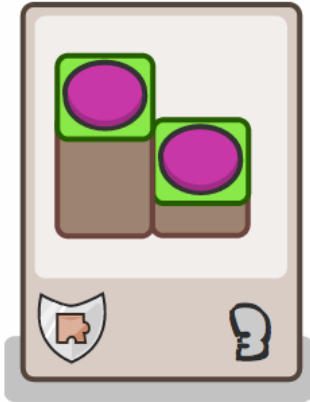
Young's Buttons

Conditional Expressions

A5

YP1

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

1

Build two trenches using the brick and dirt blocks.

2

Place the Purple Buttons and the crates in the trenches.

3

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

Repeat for Button B.

4

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

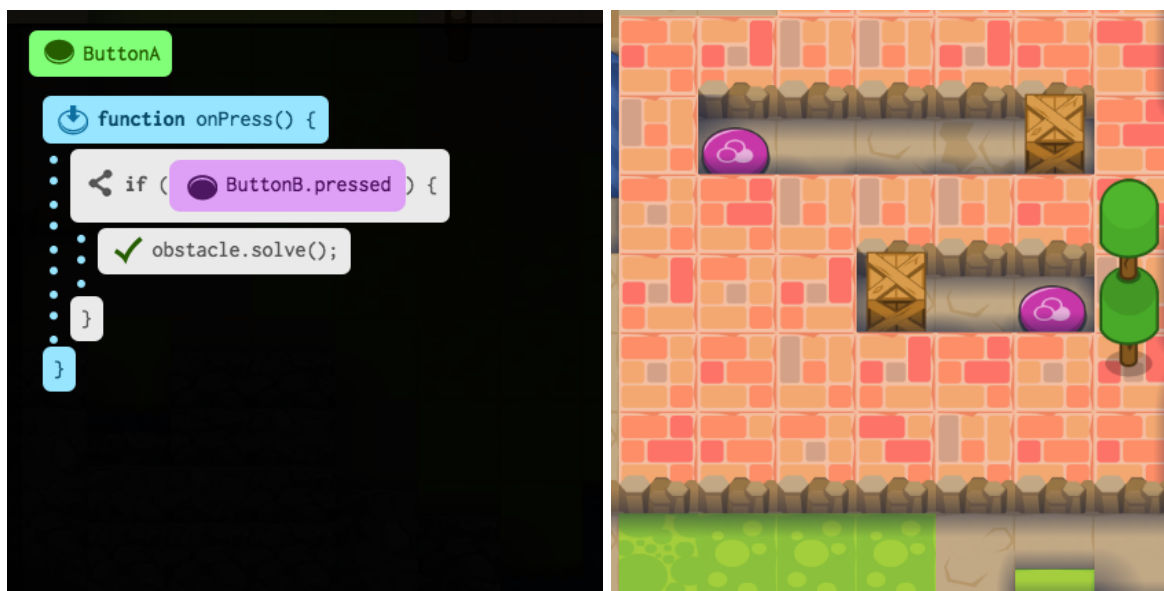
5

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.