

# Australian Curriculum: Digital Technologies

## Year 3/4 and 5/6 Blockly Wombat Challenge

### Overview

<https://groklearning.com/course/aca-dt-mini-34-bk-wombat/>

#### Table of contents

<b>Introduction</b>	<b>2</b>
<b>Mapping against the Australian Curriculum: Digital Technologies</b>	<b>2</b>
What are students learning?	2
Synopsis	3
<b>Module overview</b>	<b>4</b>
Module 1: Say Hello to Wombat!	4
Module 2: Turning Wombat	4
Module 3: Making Decisions	5
Badges	6
Wombat Playground	6
Offline Activity	7
<b>Next Steps</b>	<b>7</b>
<b>Appendix 1: Wombat Plans and flowcharts</b>	<b>8</b>

## Introduction

**Wombat** is an interactive entry-level Blockly coding challenge, using a modified version of the Turtle/Logo environment.

It is designed as a basic challenge for young or beginner students, and incorporates a very simple set of Turtle blockly code. Students learn to move Wombat around its environment in search of carrots to eat. The challenge addresses a number of the content descriptors of the Australian Curriculum: Digital Technologies and is a suitable entry point into teaching Digital Technologies for year 3/4 students.

## Mapping against the Australian Curriculum: Digital Technologies

Years 3/4 Band

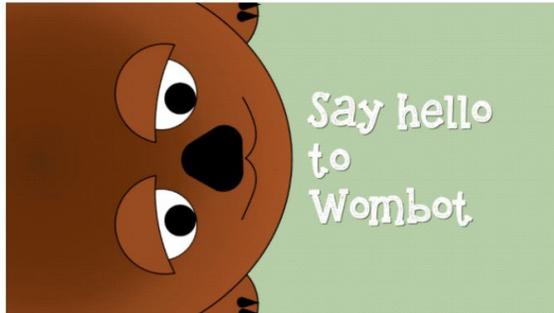
Content Descriptor Code	Content Descriptor	Key Concepts	Addressed by Wombat through:
ACTDIK008	Recognise different types of data and explore how the same data can be represented in different ways	Representations Types of data	Length and distance represented as Integers
ACTDIP010	Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them.	Defining (Specification) Decompose problem Functional Requirements Constraints	Solving problems by following sequences of steps, and describing algorithms
ACTDIP011	Implement simple digital solutions as visual programs with algorithms involving Branching (decisions) and user input	Designing (Algorithms) Tracing	Coding of geometric shapes: line, square, path through a maze. Selection of maze path through input and decisions.

## What are students learning?

In this coding challenge, students learn about programming in Blockly, including data representation, decomposition, design, user input, branching, tracing and evaluation. That's a basic coverage of the key concepts of the Australian Curriculum: Digital Technologies.

## Synopsis

The students meet Wombat, an Australian animal with a taste for carrots.



Students learn to move Wombat around in its environment: first by moving forward a specified number of steps, and then combining forward movement with turning left or right.

Using multiple movements and turns they direct Wombat through simple mazes to reach food. By adding user input, different paths through the mazes can be selected: a simple YES/NO decision allows Wombat to move to one carrot, or another elsewhere in the maze.

The final Wombat Playground allows students to practice what they have learned, and for the very keen it includes some extra code blocks that do not appear elsewhere in the challenge.

## Module overview

The challenge consists of three modules, which are summarised below.

### Module 1: Say Hello to Wombat!

<https://groklearning.com/learn/aca-dt-mini-34-bk-wombat/1/0/>

This module gets the students familiar with the challenge by introducing Wombat and using a very simple instruction: move Wombat forward a number of steps.



By the end of this module students will have a basic introductory knowledge of the blockly code and our problem-solving environment, and will be able to work out simple commands to help Wombat move forward to get a carrot. This leads into Module 2 for more complex movements around the grid.

### Module 2: Turning Wombat

<https://groklearning.com/learn/aca-dt-mini-34-bk-wombat/2/0/>

Moving forward only gets Wombat so far — so the next thing students need to learn is how to make Wombat turn. In the Turtle environment, it's possible to turn the Turtle by any angle; here we limit Wombat to 90-degree turns left or right.

Students begin by exploring turning left and right, and getting used to seeing things from Wombot's point of view. Then they combine moving with turning, to guide Wombot through a simple maze, and around a square path.



By the end of the module students understand how to combine blockly blocks to form more complex programs, and are planning out their algorithms to guide Wombot around. This leads to Module 3, where we introduce simple user input and decision-making.

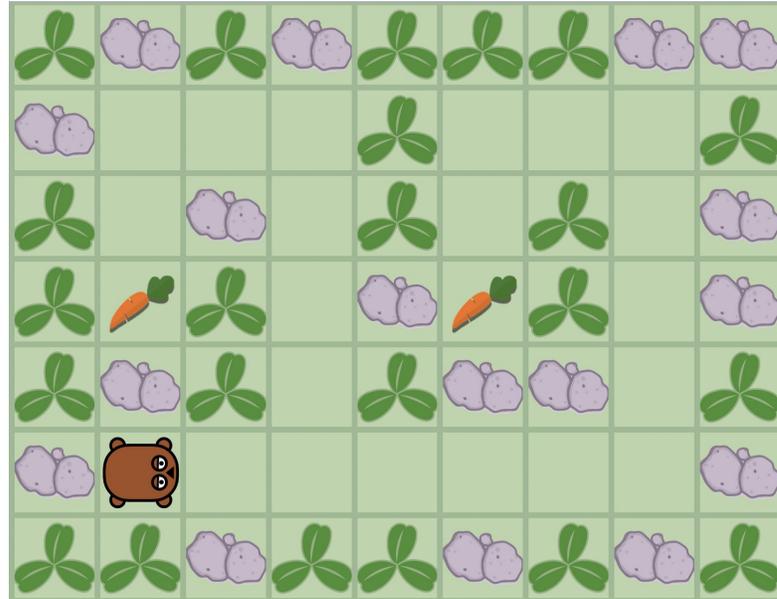
## Module 3: Making Decisions

<https://groklearning.com/learn/aca-dt-mini-34-bk-wombot/3/0/>

What if Wombot has a choice to make: *this* carrot, or *that* carrot? Students are introduced to a simple form of “if”-statement in the form of an “ask”-block: a piece of code that asks a question, and then gives Wombot instructions depending on the students’ *yes/no* answer:



By the end of the module — and the challenge — students can combine moving, turning and decisions to take on the **Final Test**: write code that helps Wombat move through the maze to eat one carrot or the other.



## Badges

At the end of each module students receive a badge to celebrate their success, in the following order:



## Wombat Playground

After the Final Test, the students can play and experiment with code to move Wombat around in the **Wombat Playground**. Here they have access to several new Blockly blocks that are not covered in the rest of the challenge — including moving backwards, turning by different angles, changing Wombat's trail colour (the Turtle pen), and even simple looping using a "repeat" block.

While we don't cover these extra blocks explicitly in this challenge, we have included them in the Playground for keen students (and their teachers!) to discover and experiment — and to encourage them to move on to more coding with the wide range of ACA's blockly, Turtle, python and other challenges!

## Offline Activity



The ACA has an offline Wombat activity, which introduces students to simple algorithms by moving a cut-out Wombat around the floor following a sequence of instructions.

You can download the resource at:

<https://aca.edu.au/resources/wom-bot/>

## Next Steps

The ACA has a load of challenges and resources for students who complete the Wombat Challenge and are looking for more, including:

- **Turtle challenges:** blockly and python versions to explore drawing more complex shapes with the Turtle, like satellites and Christmas trees
- **Blockly challenges:** use code blocks to make a micro:bit rocket, create a cookie clicker or space invaders game, or write a chat-bot program
- **Python, html and javascript challenges:** go beyond blocks to learn different programming languages and create simple interactive web pages
- **Cyber Security:** older students can become white-hat IT security experts, learning about information privacy and security, encryption, networking and web app security
- For **further resources** that might extend, enrich or deepen the classroom experience or support assessment, we recommend visiting the Digital Technologies Hub at <https://www.digitaltechnologieshub.edu.au>.

You can find all of these and many more at <https://aca.edu.au/resources/>

## Appendix 1: Wombat Plans and flowcharts

The following pages provide worksheets to help you and your students plan out their solutions to problems in Module 3: “Which Carrot, Wombat?” and “Wombat’s Final Test”.

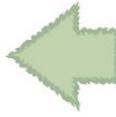
You can also download them as PDF files:

- Which Carrot, Wombat? Plan — [Download PDF](#)
- Wombat’s Final Test Plan — [Download PDF](#)

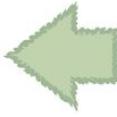


Help Wombat choose  
which **carrot** to eat

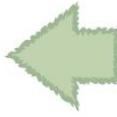
**First** What does Wombat do **first**?



**Then** What does Wombat do **next**?

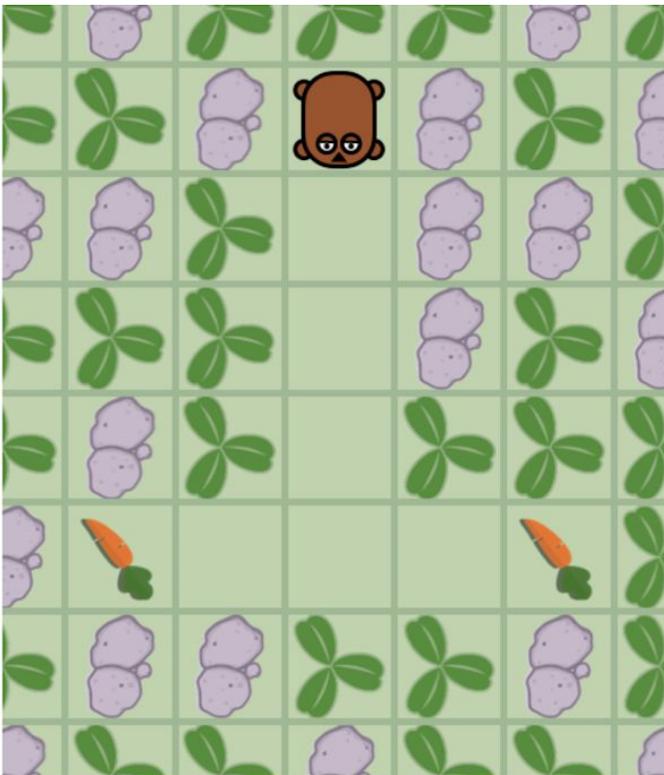


**Then** What does Wombat do **next**?



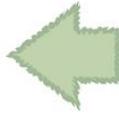
**Then** What does Wombat do **next**?

Which **carrot**,  
Wombat?



Help Wombat choose which **carrot** to eat

**First** What does Wombat do **first**?



Move forward 200 steps

**Then** What does Wombat do **next**?



Decide which way to turn:  
ASK "Turn left?"

**Then** What does Wombat do **next**?

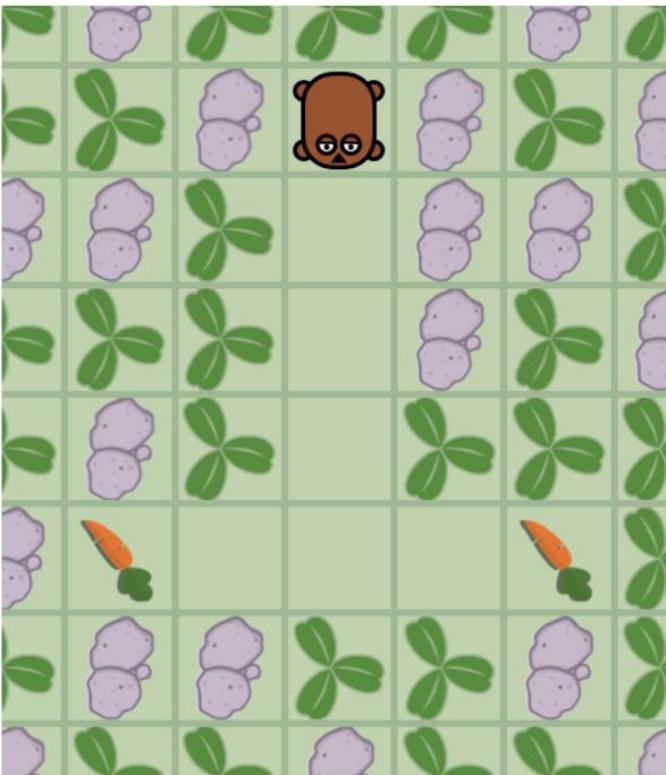


If answer is yes, turn left  
if no (else), turn right

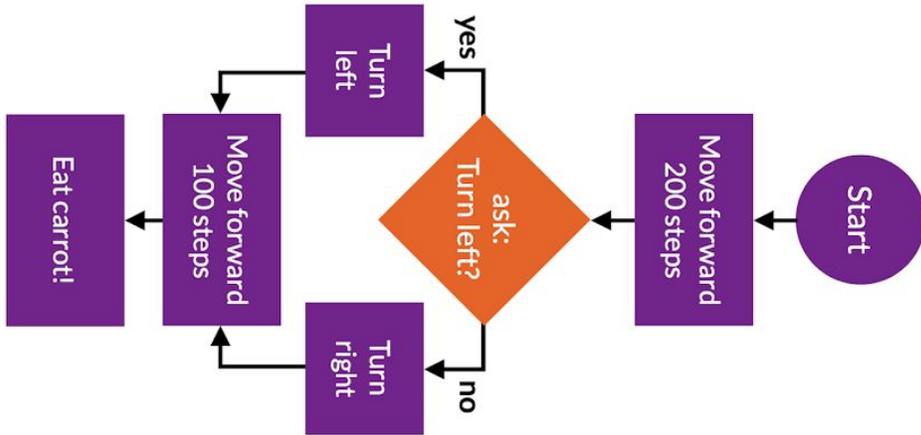
**Then** What does Wombat do **next**?

Move forward 100 steps  
and eat the carrot!

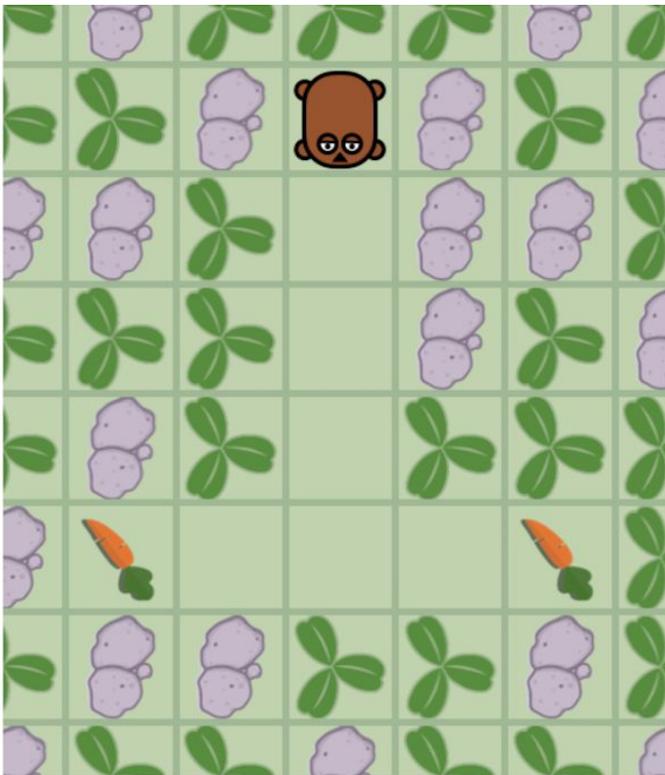
Which **carrot**,  
Wombat?



Help Wombat choose which **carrot** to eat

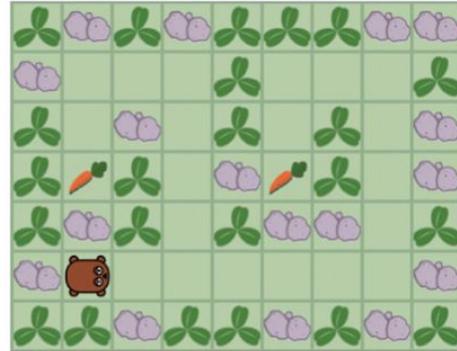


Which **carrot**, Wombat?



# Wombot's final test!

Help Wombot through the maze to find a **carrot**



Here's one way to do it!

Can you think of other ways?

