Years 3-4

Programming and algorithms and KLA examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Specific DT focus** | **Subject area** | **Short focus** | **Summary** | **Title** |
| Pathways | English | Clear, precise communication | In pairs, explore giving and following a sequence of steps and decisions to build a LEGO® toy. | Take a [LEGO® building challenge](#LEGOBUILDING) |
| Decision making (branching) | Create a storyboard to plan a story where the reader is provided with a number of decisions that lead to alternative endings. | Plan a ['choose your own adventure' story](#PLANCYOA) |
| Mathematics | Decision making (branching) | Create a flowchart to represent a sequence of (branching) steps and decisions needed to solve a mathematical problem. | Have [fun with flowcharts](#FunWithFlowcharts) |
| Choices | English | Implementing a digital solution using visual programming | Based on your storyboard for a 'choose your own adventure' story, use a visual programming language to implement a digital solution. |  |
| Create a multimodal game board where the player is provided with a number of decisions. Using Scratch and Makey Makey, add multimodal elements to the story. These are activated using an Ozobot. | [Create a game board that uses an Ozobot](#Ozobot) |
| Sequencing instructions to complete a task | Plan and create a computer program to demonstrate grammar or spelling rules, eg changing nouns from singular to plural; adding ‘ing’. |  |
| HAAS: History | Implementing a digital solution, using visual programming | Design and create a simple game/quiz to demonstrate convict crimes and punishments. | Design a quiz –[Convicts: crime and punishment](#Convict) |
| HAAS: Geography | Implementing a digital solution, using visual programming | Create a computer program for learning a traditional Aboriginal or Torres Strait Islander language. | [Create a language -learning program](#languagelearning) |
| Science | Implementing a digital solution, using visual programming | Design and create a simple quiz to explore the difference between living and non-living things. |  |
| Mathematics | Implementing a digital solution, using visual programming | Modify an existing program or create a program to design a geometric shape or design using Pencil code or similar application. |  |
| Programming a robotic device to follow a path | Create a maze or route for a programmable robot to travel. Estimate and calculate angles and distances. |  |
| Working together | HPE | Collaboration | Collaborate to decide the rules for a new game. Use a flowchart to explain the consequences of unfair play. |  |

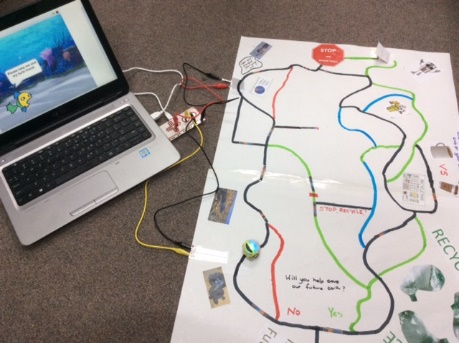
**Title**: Create a board game that uses an Ozobot

***SUB HEADING***: Decision making (branching), coding and robotics

**Summary Text:**

Create a game board where the player is provided with a number of decisions. Using Scratch and Makey Makey, students add multimodal elements to the story. These elements are activated using an [Ozobot](https://ozobot.com/).

**Year Level**: 3-4



*Photo courtesy of Jo Klein*

# Suggested steps

### Investigate and define

Introduce students to the Ozobot, using the [Cloud Ozobot Maze](http://play.ozobot.com/print/ozoluck/ozobot-the-cloud.pdf) or [Space Butterfly Maze](http://play.ozobot.com/print/ozoluck/ozobot-the-space-butterfly.pdf) (available at ozobot.com). Organise students to [draw their own mazes](https://ozobot.com/play/drawing-games) using colour codes and decisions (algorithms).

### Generate and design

1. Discuss how a *‘*choose your own adventure' story works. Explain the concept of this type of story is to allow the reader to choose different pathways leading to alternative events within the plot.
2. Model the construction of a storyboard that will portray graphically the choices and subsequent outcome of each choice as an algorithm with branching. Relate this to how the Ozobot is going to choose different paths within their story/adventure.
3. Students:

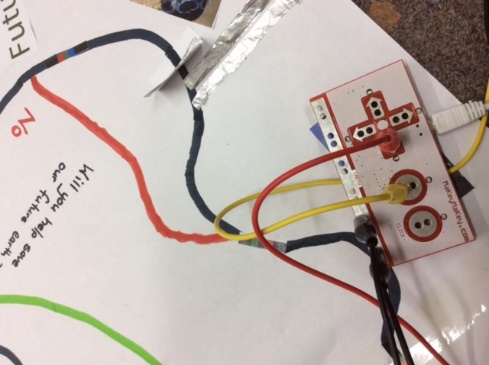
* plan out their story using a graphical algorithm model that includes simple branching options
* draw the story on paper to display the plot, including all branching choices offered and alternative endings (if included).
* on the large sheets of paper, connect story paths for the Ozobot to follow by using black and coloured lines. They include intersecting lines to include branching options (by default, at an intersecting line the Ozobot will randomly decide which path to take unless a code is used).

### Produce and implement

Using Scratch (or similar visual block-programming software), students create simple animations for the key plot points or events in their story, using sprites and/or speech/audio blocks.

## Extension

* Include conditionals, branching and user input (eg, imagining that the player is asked which path they want to take).
* Connect the computer to the story mat using a Makey Makey and use a ‘switch’ to activate multimodal elements created within Scratch.   
  **Hint**: To create the circuit, include conductive tape or aluminium foil under the paper switch (see it at the top middle part of the picture).



*Photo courtesy of Jo Klein*

Makey Makey activates when the Ozobot proceeds over the paper, completing the circuit. See the aluminium foil at the top of the image.

### Evaluate

Invite students to share their interactive stories with their peers and discuss the alternative endings or variations in endings.

# Discussion

* Students explain how their story meets the audience's needs and how the concept of ‘branching’ or decision making helped the audience make their decision.
* Was enough information provided to help them decide which path to take?
* Was the information confusing at any point?
* Centre the discussion on data and ask students to suggest what data was needed to help readers make their decisions.

# Why is this relevant?

Decisions are important within computational thinking. They allow actions to be changed, based on the input of data. This input could be:

* user-input; for example, selecting an onscreen value or button, typing in an answer
* sensed from the immediate environment; for example, collected via a sensor on a robotic device sensing an obstacle and being programmed to avoid it (or, in the case of Ozobots, different colours when sensed will change the speed).

Algorithms are the step-by-step procedures required for solving a problem. Algorithms may be described either diagrammatically or in structured English. Flowcharts are often a good way of visualising algorithms and can be an effective way to teach the concept of ‘branching’. Branching involves making a decision between one of two or more actions, depending on sets of conditions and the data being inputted.

This activity can be used to strengthen students' understanding of computer programming and robotics. It can be used to combine computer programming and robotics to create engaging and innovative solutions to simple digital problems, using multimodal texts.

To create a STEM-based unit, include:

* construction of a story background, using bridges and 3D elements to add in elements of engineering design
* focus on a science concept, eg create a space game board or design a game with data on ways to create a future Earth.

# Assessment

Evaluate students’ understanding using a simple checklist. For example:

* The storyboard represents a sequence of events that follow a well-defined plot.
* The storyboard provides the reader with a choice of options leading to alternative endings.
* Scratch animation includes branching and user inputs.
* The interactive story is correctly connected to a Makey Makey and includes visual data and information directing the reader to the correct path chosen.

# Australian Curriculum alignment

## Technologies – Digital Technologies

Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them [(ACTDIP010)](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP010)

Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input [(ACTDIP011)](http://www.australiancurriculum.edu.au/curriculum/contentdescription/ACTDIP011)

Explain how student solutions and existing information systems meet common personal, school or community needs [(ACTDIP012)](http://www.australiancurriculum.edu.au/curriculum/contentdescription/ACTDIP012)

## English

**Creating literature**

Create imaginative texts based on characters, settings and events from students’ own and other cultures using visual features, for example perspective, distance and angle [(ACELT1601](http://www.australiancurriculum.edu.au/curriculum/contentdescription/ACELT1601))

**Creating texts**

Plan, draft and publish imaginative, informative and persuasive texts demonstrating increasing control over text structures and language features and selecting print, and multimodal elements appropriate to the audience and purpose [(ACELY1682)](http://www.australiancurriculum.edu.au/curriculum/contentdescription/ACELY1682)

Use software including word processing programs with growing speed and efficiency to construct and edit texts featuring visual, print and audio elements [(ACELY1685)](http://www.australiancurriculum.edu.au/curriculum/contentdescription/ACELY1685)

Create literary texts by developing storylines, characters and settings [(ACELT1794)](http://www.australiancurriculum.edu.au/curriculum/contentdescription/ACELT1794)

## ICT Capability

**Level 3**

Typically by the end of Year 4, students:

**Select and use hardware and software**

* identify and independently operate a range of devices, software, functions and commands, taking into consideration ergonomics when operating appropriate ICT systems, and seek solutions when encountering a problem

**Collaborate, share and exchange**

* use appropriate ICT tools safely to share and exchange information with appropriate known audiences