Years 3-4

Programming and algorithms and KLA examples

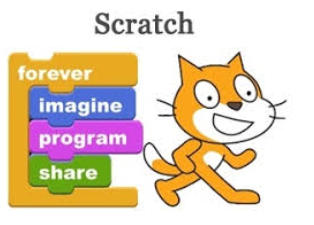
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Specific DT focus** | **Subject area** | **Short focus** | **Summary** | **Title** |
| Pathways | English | Clear, precise communication | In pairs, explore giving and following a sequence of steps and decisions to build a LEGO® toy. | Take a [LEGO® 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**: Design a quiz – Convicts: crime and punishment

***SUB HEADING***: Decision making (branching)

**Summary Text:** Students design and create a simple game/quiz to demonstrate convict crimes and punishments.

**Year Level**: 3-4



# Suggested steps

1. Students use three recommended sources to locate and collect information and data about the punishment of convicts in early Australia.

* <http://education.abc.net.au/home#!/topic/494654/convict-and-colonial-australia>
* <http://splash.abc.net.au/home#!/media/1253347/a-flogging-for-a-pair-of-boots>
* <http://splash.abc.net.au/home#!/media/2443387/female-and-child-convicts>

1. Provide students with focus questions to support their research. For example: What were the most common forms of punishment for the convicts? What were the reasons for the punishment? Were the methods of punishment the same for men and women?
2. Support the students to work in pairs to record, sort and represent their data using a table, infographic or simple graph.
3. Share a Scratch quiz's following optional formats (on the topic of early colony years), and invite students to interact with it.   
   (Quiz with typing choice – easy) <https://scratch.mit.edu/projects/168455003/>

(Quiz with branches – easy) <https://scratch.mit.edu/projects/168455004/>

(Quiz with key press) <https://scratch.mit.edu/projects/168455001/>

1. Encourage students to consider which programming blocks were used to create the quiz.
2. Use the **See inside** option to support student understanding of the purpose and function of the *if/then/else* blocks.
3. Working collaboratively in pairs or groups, support students to use a visual programming language such as Scratch, to design and create their own simple game/quiz to demonstrate understanding of convict crime and punishment in early Australia.
4. Students create their quiz program, focusing on one question first to ensure it is working as expected (debugging). Once that program is tested and refined, then this can be applied to their remaining quiz questions. When working in groups of two or three, employ a strategy that allows all students to get equal opportunity to program.

Explain that students will need to use their research (infographic, table or graph) to help determine the quiz questions and answers. Each game/quiz must include decision-making, eg multiple-choice questions or paths to choose from.

# Discussion

Centre the discussion on data and on the data that was needed to help the players make their decisions.

Suggested questions for discussion and reflection:

* At what stages does the user have control and make choices?
* What programming did you use to enable this?
* When you consider the information (instructions) you provided to the player, was enough information provided to help them decide which path to take? Was the information confusing at any point?
* What do you see when you compare your algorithm to your final programming?

# Why is this relevant?

In years 3–4, students should be provided with the opportunity to implement programs that make decisions on the basis of user input or choices such as selecting a button, pushing a key or moving a mouse to ‘branch’ to a different segment of the solution. This activity can be used to strengthen students' understanding of computer programming as a series of instructions that can change depending on different user inputs or conditions. The focus is on how computers follow instructional pathways.

An interactive game or quiz providing the user with a choice of paths or options is a fun 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 provided. Decisions are an important part of computational thinking. They allow actions to be changed based on the value of data. Algorithms are the step-by-step procedures required for solving a problem.

# Assessment

**Part A**: Assessment task – Implement a digital solution (game or quiz) using a visual programming language

Evaluate the student's ability to:

use a simple visual program with algorithms to implement a solution

design algorithms with branching and user input.

**Part B**: Self-assessment and reflection

Provide students with an opportunity to reflect on their digital solution, using the following prompts:

* Explain reasons for any changes to the algorithm before the final program was completed.
* Did debugging and testing lead to any improvements? In what way/s?
* How did you work together as a group? Now imagine that you had to complete this task on your own. How might the outcome be different?

# Australian Curriculum alignment

## Technologies – Digital Technologies

Implement simple algorithms as visual programs involving control structures and input ([AC9TDI4P04](https://v9.australiancurriculum.edu.au/f-10-curriculum.html/learning-areas/digital-technologies/year-3_year-4/content-description?subject-identifier=TECTDIY34&content-description-code=AC9TDI4P04&detailed-content-descriptions=0&hide-ccp=0&hide-gc=0&side-by-side=1&strands-start-index=0&subjects-start-index=0&view=quick))

Discuss how existing and student solutions satisfy the design criteria and user stories ([AC9TDI4P05](https://v9.australiancurriculum.edu.au/f-10-curriculum.html/learning-areas/digital-technologies/year-3_year-4/content-description?subject-identifier=TECTDIY34&content-description-code=AC9TDI4P05&detailed-content-descriptions=0&hide-ccp=0&hide-gc=0&side-by-side=1&strands-start-index=0&subjects-start-index=0&view=quick))

## HASS

### History

**Knowledge and Understanding**

Stories of the First Fleet, including reasons for the journey, who travelled to Australia, and their experiences following arrival [(ACHASSK085](http://www.australiancurriculum.edu.au/curriculum/contentdescription/ACHASSK085))

**Researching**

Locate and collect information and data from different sources, including observations [(ACHASSI074](http://www.australiancurriculum.edu.au/curriculum/contentdescription/ACHASSI074))

Record, sort and represent data and the location of places and their characteristics in different formats, including simple graphs, tables and maps, using discipline-appropriate conventions [(ACHASSI075)](http://www.australiancurriculum.edu.au/curriculum/contentdescription/ACHASSI075)

## ICT Capability

By the end of Year 4:

**Locate, generate and access data and information**

* locate, retrieve or generate information from a range of digital sources

**Generate solutions to challenges and learning area tasks**

* create and modify simple digital solutions, creative outputs or data representation/transformation for particular purposes

**Generate ideas, plans and processes**

* use ICT to generate ideas and plan solutions