# Design a quiz – Convicts: crime and punishment

Please refer to the online lesson plan on the DT Hub to access all website links and additional resources.

**Year Level: 3-4:** Students design and create a simple game/quiz to demonstrate convict crimes and punishments.



# Suggested steps

1. Students use two recommended sources to locate and collect information and data about the punishment of convicts in early Australia. (see online lesson)
2. 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?
3. Support the students to work in pairs to record, sort and represent their data using a table, infographic or simple graph.
4. 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)
* (Quiz with branches – easy)
* (Quiz with key press)
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 )
* Discuss how existing and student solutions satisfy the design criteria and user stories (AC9TDI4P05 )

## HASS - History

**Knowledge and Understanding**

* The causes of the establishment of the first British colony in Australia in 1788 (AC9HS4K02)

**Skills: Questioning and researching**

## Locate, collect and record information and data from a range of sources, including annotated timelines and maps (AC9HS4S02)

## Digital Literacy

**Investigating**

* Locate information
* Acquire and collate data
* Interpret data.

**Creating and exchanging**

* Plan
* Create, communicate and collaborate
* Respect intellectual property