**Design a flag with Pencil Code**

*Please refer to the online lesson plan on the DT Hub to access all website links and additional resources.* **Years 5 -6; DT+ HAAS: Civics and Citizenship**

**Summary**

Design your own Australian flag by firstly examining common elements of flags, creating a step by step process (algorithm) to program your design after exploring a ‘block-based’ turtle drawing program such as Pencil Code.

## Learning hook

Investigate designs of flags from around the world and look specifically at the features of these flags. What are common elements? What geometric shapes are used? How could you create a computer program to recreate a flag design?

Examine the Australian Flag, the Australian Aboriginal Flag and the Torres Strait Islander Flag, and discuss the significance of the symbols on all of them. Centre discussions around reconciliation. Ask students to design a new Australian flag that would foster inclusivity and togetherness and have relevance for future generations.

**Note**: You may decide to integrate this task with HAAS: Civics and Citizenship

## Learning map and outcomes

Explain that students will:

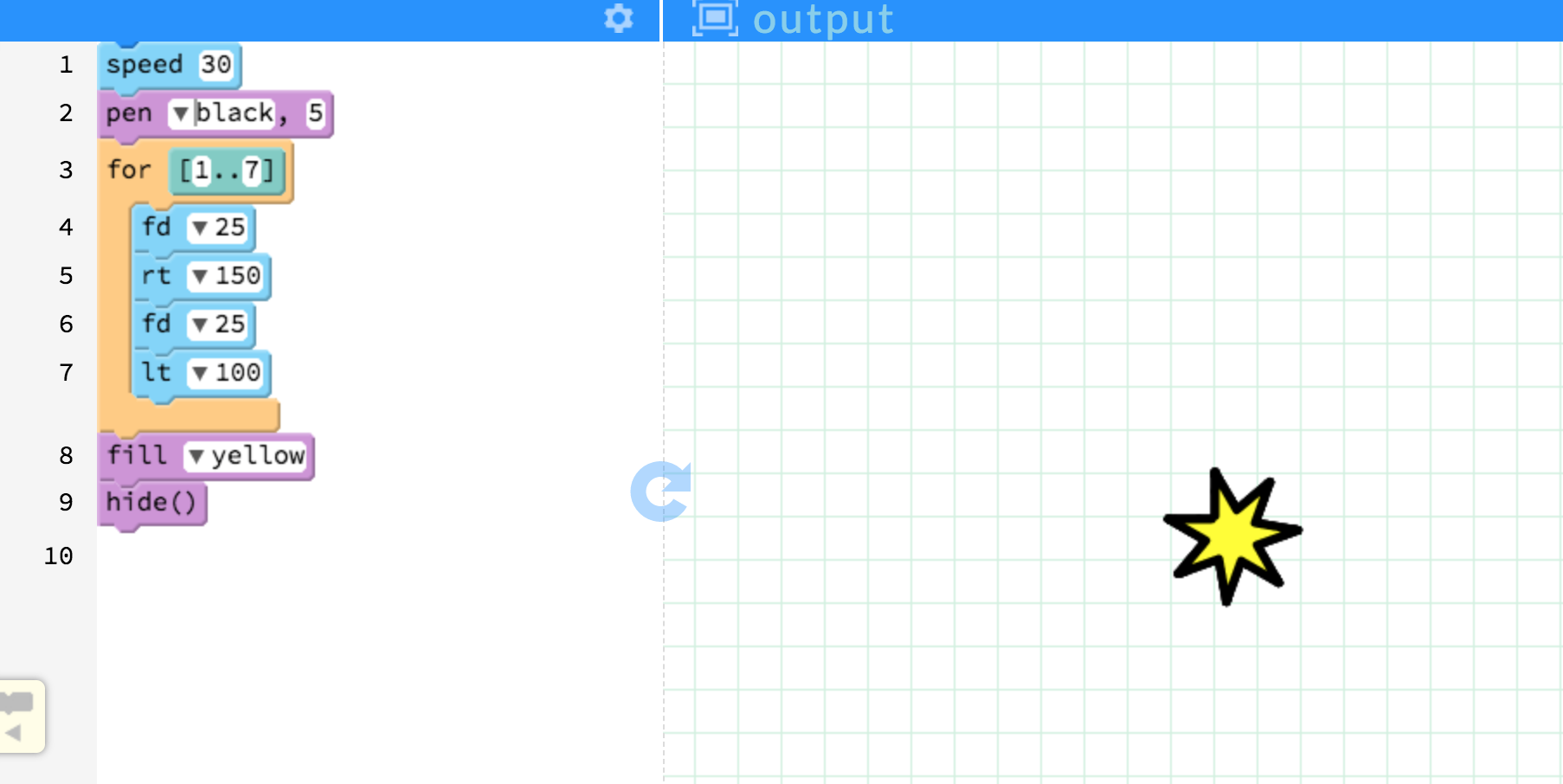
* develop their understanding of iterations and how these can be used in programming
* understand that **iteration** represents repeated sequences that are taken when developing algorithms
* apply data input and output in algorithms for iterations.

Suggested steps

1. Students visit Flags of the world or other similar websites to become familiar with the design of different flags from around the world.
2. Before programming their own flag design, students design their flag. They could use one of the more basic interactive flag design websites (see online lesson resources) to become familiar with the use of colours and features used in the design of most flags around the world. It may also help students become familiar with how to work within the available spatial dimensions if they draw the Australian Aboriginal Flag and the Australian Flag. Examine features on flags and consider proportion.
3. Ask students: If you were to program making this flag you have designed, what functions could you use?
4. Mention some of the terminology the students will need to know or come across using Pencil Code: repetition, iteration, loops.
5. Using Pencil Code, further introduce students to the term ‘iteration’. Create some basic shapes so they become familiar with the function.
6. Go to the Pencil Code Guide and look at loops and how they function with data input and output. If you can make a loop for a star, how can you use this idea to design a flag? (Allow the students to use a protractor to help them when working out angles.)
7. To familiarise themselves with Pencil Code, students could recreate the Australian Aboriginal Flag. Use the iteration for a rectangle. Add in the conditional data to make the rectangles in the flag. This also presents an opportunity to create an algorithm of the process that describes the steps. Talk about creating the program with the least number of steps.

|  |  |
| --- | --- |
|  | Screenshot 2017-10-26 at 1.45.48 PM.png |
|  | NB: The Australian Aboriginal Flag is © Harold Thomas. |

1. It may be useful to follow a similar approach with the Australian Flag. Using the same iteration functions from Step 7, create the basic outline for the Australian Flag. Take a look at how to create a star.



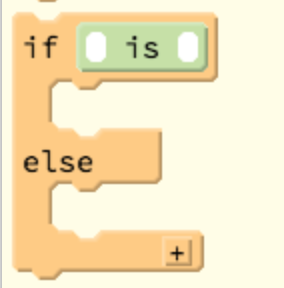
1. Once students have become familiar with these functions, you may also like to investigate the Torres Strait Islander Flag.
2. Using Pencil Code, design what you think a **new** Australian flag could look like.
3. Share designs, with student pairs swapping feedback based on agreed criteria.

**NOTE: In this task, Pencil Code is suggested as a relevant tool; however, you may decide to use a different turtle drawing program. Pro-Bot** is **also an** option**.**

Discussion

Programmers use iterations or loops to save time, repeat steps and simplify algorithms.

* Have you programmed the code for your flag in the least amount of steps possible?
* In what ways have you simplified your program? What blocks did you use?
* If you used loops, how did you work out how many times to repeat the steps?



Why is this relevant?

Using iterations or loops is a more efficient way of coding. Programmers use this approach to repeat a sequence of steps until a condition is met. Being able to repeat a process can help to simplify algorithms and remove unnecessary steps. Iterations give the user the ability to program a short sequence of actions rather than using longer code. Loops tell the program how many times to repeat a sequence and when to stop. They can have conditions with variables in the code that tell the output to change depending on the user input. This video CS Principles: Using Loops provides further detail about how it works.

Curriculum links

**Processes and Production Skills**

* Implement algorithms as visual programs involving control structures, variables and input (AC9TDI6P05)

Links with HASS

**Knowledge and understanding: Civics and Citizenship**

* How citizens (members of communities) with shared beliefs and values work together to achieve a civic goal (AC9HS5K07)

ICT Capabilities:

#### Typically, by the end of Year 6, students:

*Generate solutions to challenges and learning area tasks*

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

Assessment

#### Peer assessment

Students test and give feedback on each other’s flags:

* Has the least amount of blocks been used to replicate the flags?

|  |  |  |  |
| --- | --- | --- | --- |
| Design a new Australian flag | | | |
| One | Many | Relate | Extend |
| *Identify and define* | *Combine and perform serial skills* | *Integrate* | *Create and evaluate* |
| I can identify the blocks for iterations/loops  I can create a repeated process in a program by entering the same blocks over and over | I can describe why programmers might use iteration  I can use the iteration/loop blocks in the *Control* toolbox and enter code to repeat an action; for example, creating an rectangle | I can explain how the iteration/loop uses conditions to tell the ‘turtle’ what to do and when to stop  I can create an algorithm which highlights the use of iterations/loops  I can include iterations/loops to create my own flag design to simplify my program | I can debug my program to ensure it is valid as I test it  I seek feedback on my design and incorporate ideas of others to improve the functionality of the program |