TE HIKO TĀKARO

So you want to teach Digital Technologies?

DT Teachers Guide

How to teach game design in the Digital Technologies Curriculum

Dan Milward and Gerard MacManus

This resource weaves together Game Design & Development, Coding, your Local Curriculum and Tikanga Māori.

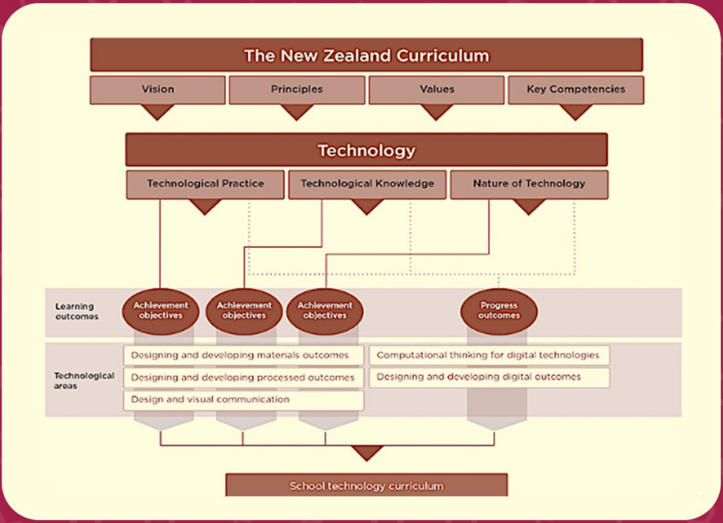
Digital Technologies (DT) covers two technological areas from the New Zealand Curriculum:

1) Computational Thinking (CT)

2) Designing and Developing for Digital Outcomes (DDDO)

Te Hiko Tākaro weaves the two areas together while simultaneously embedding the key competencies from the New Zealand curriculum.

The New Zealand Curriculum



The New Zealand Curriculum

Vision | Principles | Values | Key Competencies

Technology

Technological Practice | Technological Knowledge | Nature of Technology

Learning outcomes: Achievement objectives; Achievement objectives; Achievement objectives; Progress outcomes

Technological areas:
- Designing and developing materials outcomes
- Designing and developing processed outcomes
- Design and visual communication
- Computational thinking for digital technologies
- Designing and developing digital outcomes

School technology curriculum

Version 1.0

HOW IT WORKS

It all starts here. The `DT Teacher Guide'. Follow the steps in this booklet. Learn some code, design a game, then run through the same process with your students.

Step 1

Work through the **'How to code a 2D Video Game'** booklet which covers CT Progress Outcomes 4

TE HIKO TĀKARO
So you want to be a game designer?
GAME DESIGN DOCUMENT WORKBOOK

Step 2

Work through the **'So you want to be a game designer?'** booklet the accompanying **Student Workbook**. Together they cover the DDDO Progress Outcomes 4.

Step 3

Complete the **'DHTM Curriculum alignment'** template. This template will help you weave your local curriculum into your game development learning.

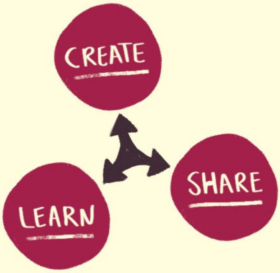
Get them all > https://takaro.gamefroot.com

LEARNING METHODOLOGY

What you are doing is learning, creating and sharing

Your students learn.
With your support they create.
And finally, your students share.

But it needn't be in that order. "Creation to inform learning" and "sharing to inspire creation" show just how easy it is to adjust the order and process.



- Reflecting on your own practice, what order does your delivery and teaching tend to follow? Why is that?
- Think of a successful experience you facilitated for your students. What relationship did learning have to creating and sharing? How fluid was the movement between them?

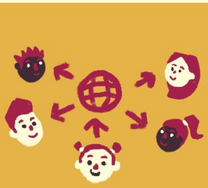
Learn

You and your students

- CT booklet - learn programming concepts.
- DDDO booklet - design your game.

Create

Combine your knowledge from DDDO with your CT skills to develop a unique piece of digital content.



Share

Share your completed Aotearoa-themed game with students, friends and whānau. Do this online, via Gamefroot. **Arcade.gamefroot.com**

BOOKLET 1: CODE A GAME

Now it's time to try your hand at game dev and write some code!

Working through the **"How to code a 2D platformer game"** booklet will give you the hands-on skills you need to make a video game!

You will learn

- Inputs and Outputs
- Variables
- Conditionals
- Loops (iteration)
- Data Types
- Comparative Operators

You will use the cloud-based platform, Gamefroot, to gain first-hand experience with **"Computational Thinking Progress Outcome 4"** from the new and revised Technology Curriculum - Digital Technologies Content.

To get started, open your **"How to code a 2D platformer game"** booklet and work through the introduction section.

The introduction section shows you how to set up a Gamefroot user account and familiarise yourself with the Gamefroot basics.

INPUTS AND OUTPUTS

"A program is a spell cast over a computer, turning input into error messages."
- Dave Barry

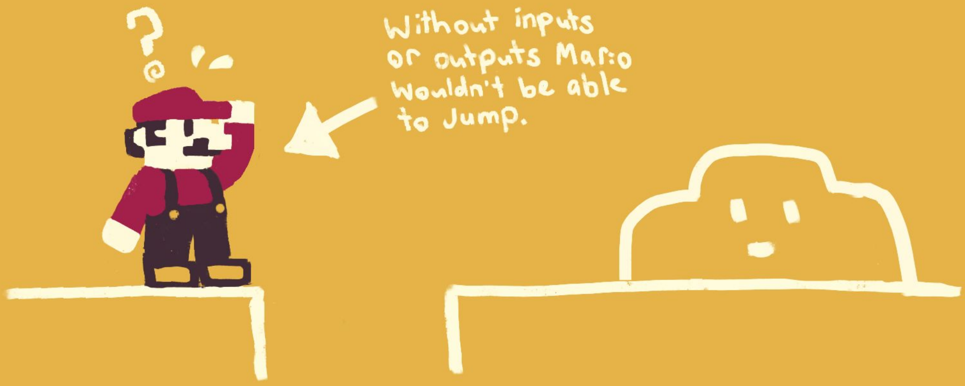
Inputs and Outputs

Inputs are ways to control computer programs. **Outputs** are ways to get information out of them.

When you type on your keyboard, you're creating an input. What you see on your screen is called an output.

In the CT booklet (Chapter 1), you will program inputs that will allow a player to control the Player Character (PC) via keyboard arrow keys.

When you see the PC move in the game, that's the output!



LOOPS (ITERATION)

> "I learned to appreciate repetition. That's why I can dance. It's how I learned to act. I have a high tolerance for repetition."
> - Channing Tatum

Unlike us teachers, computers never get tired of repeating themselves.

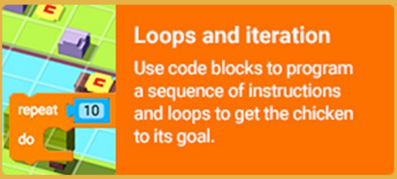
Video games are full of repeated actions. To make sure the computer manages all those repetitions (and not us) it's important that we use **Loops.**

When you want to **repeat** actions in a game, rather than laboriously copying and pasting code, you can put the code you want to repeat inside a loop. This will make your code run multiple times.

WHY DID THE CHICKEN CROSS THE ROAD?
BECAUSE YOU TOLD IT TOO, REPEATEDLY.

For instance, in the CT booklet (Chapter 1), you will use a Constantly loop to follow the PC (player) with the in-game camera. This will ensure the player is always visible in the centre of the game.

To delve deeper into Loops, you could explore the Loops and Iteration tutorial in Gamefroot's: **"Learn to code with Crossy Road"** resource.



Loops and iteration

Use code blocks to program a sequence of instructions and loops to get the chicken to its goal.

VARIABLES

"The secret of happiness is variety, but the secret of variety, like the secret of all spices, is knowing when to use it."
- Daniel Gilbert

Variables are ways that computers can store information.

Variables have a **name** and **value** that you can modify.

Name (Where did I park the car today?) = Value (Space 36)

Variables can also have a **constant value** assigned to them.

Name (Where does the principal park her car?) = Value (Space 01)

This value will never change because it has "Principal" painted on it in big, bold letters.

In the CT booklet (Chapter 4), **"Keeping Score with Variables"**, you'll use a variable that can change to track your player's score.

Consider how many variables you need to store in your mind on any given day. Variables include everything from the results of the latest assessment through to where you parked the car.

Now think about the many pieces of information that need to be stored in even a relatively simple video game.

Eg. High score. Personal scores. Items in an inventory. Level. Speed of attacking aliens.

The more variables your game has, the more varied the gameplay will be.

Just like finding a parking space at school on a rainy day.

DATA TYPES

"If you had an off switch, Doctor, would you not keep it secret?"
- DATA, Star Trek

Data Types are the different sorts of values you can have in your variables. There are three data types a programmer should be familiar with at this level of game development. These are:

String type

String data is a set of characters that may contain spaces and numbers.

In the CT booklet (Chapter 8) you will use Strings for text. You will code a string of data to display text in a multiple choice quiz.

i.e. "What is the meaning of life?"

A) 42
B) Pavlova
C) A good question

Numeric type

Numeric data is anything mathematical.

In the CT booklet (Chapter 7) you will set speed and velocity using Numeric data (numbers).

Name (Taniwha's Health Points) = Value (99)

Boolean type

Boolean data is a True or False value.

In the CT booklet (Chapter 6) you will use Boolean data to establish whether or not a thrown pātū is affected by gravity.

Is the taniwha still alive? True or False?

CONDITIONALS IF, THEN, ELSE

“Success is a science; if you have the conditions, you get the result.”
- Oscar Wilde

Conditionals are a way to tell the code to do one thing based on one condition, and another thing based on another condition.

You can have an **If** statement on its own…

```
If “parking space available” = true
Then “park your car”
```

… or you can pair it with an **Else** statement.

```
If “parking space available” = true
Then “park your car”
Else “park in the Principal’s space
and hope you get away with it”
```

In the in the CT booklet (Chapter 3) you will be using **If, Then** and **Else** statements to determine whether or not the player has reached the **Trophy** (treasure chest) and won the game.

If “you’ve got the gist of Conditionals” **Then** “move onto the next page” **Else** “time for coffee!”

COMPARITIVE OPERATORS

No need to ask
He's a smooth operator
- Sade

Comparative Operators are used to compare one piece of data with another.

To put this in context, in the CT booklet (Chapter 5) you will use a comparative operator to manage the movements of a patrolling guard.

Examples of Comparative Operators

- and / or
- not
- < > =

Let's say Tupaia (above) is starting at Position 1 (far left) and we want him to move until he hits Position 5 (far right). When he gets to Position 5, we want him to turn around and move back to Position 1.

The comparative operators would be something like...

A) If Ghost Cook Position < 5 Then Move Right.

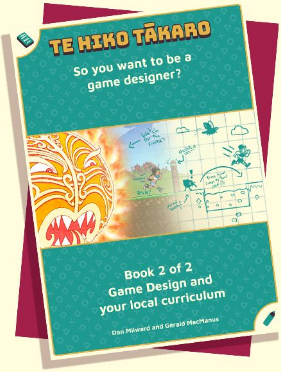
B) If Ghost Cook Position = 5 Then Move Left until Ghost Cook Position = 1

C) Go to A

BOOKLET 2: DESIGNING A GAME

Once you and your students have completed the 'How to code a 2D Video Game' booklet you are ready to design your own game!

The **'So you want to be a game designer'** booklet contains a number of activities that help you think like a game designer.



TE HIKO TĀKARO

So you want to be a game designer?

Book 2 of 2
Game Design and your local curriculum

Dan Milward and Gerald MacManus

The activities in the **"Student Workbook"** will help you and your students come up with your own Game Design Documents!

This is what real life game designers use to develop and pitch their game ideas.

Once your students finish the workbook, they can apply their 'Computational Thinking' skills and turn their game idea into a reality!

GAME DESIGN ACTIVITES

"By playing games you can artificially speed up your learning curve to develop the right kind of thought processes."
- Nate Silver

Never Alone Analysis

The Never Alone analysis activity asks you to describe the game's Setting, Components, Goals, Core Mechanics, Challenges and Rules.

You can purchase and play Never Alone from the Appstore or Google Play, or you could watch these YouTube videos with your students.

Trailer [https://bit.ly/2ZvsTaV]

Gameplay [https://bit.ly/2yM6SZv]

Mashup Activity

This is a good entry point to designing a game. It works by mashing existing things together to make something 'new'.

Wall O' Games Activity

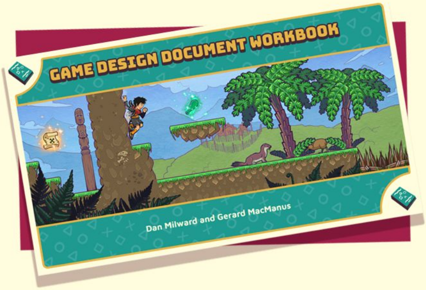
This is an effective group exercise for generating the games and concepts needed for the Mashup Activity.

GAME DESIGN DOCUMENT

> “Design is a funny word. Some people think design means how it looks. But of course, if you dig deeper, it’s really how it works”. - Steve Jobs

Students need to complete the activities in the Game Design Workbook.

A Game Design Document (GDD) is used in the video games industry to help individuals or teams describe their game to others.


GAME DESIGN DOCUMENT WORKBOOK

Dan Milward and Gerard MacManus

A GDD literally keeps everyone on the same page.

The GDD is also used to test whether the game is going to be viable to make, taking into account the skills and resources of the game development team.

Potential problems are identified and solved in the GDD before the game goes into the often expensive and time consuming production process.

Why should students write a GDD before making their Aotearoa-themed 2D platformer?

By writing a Game Design Document, students practise design thinking that can help them overcome the challenges of practical game development.

Games can be complicated. Even video game professionals can get a bit lost in the development process. That's when having a 'blueprint' in the form of a GDD comes in very handy.

The Game Design Document Template

https://takaro.gamefroot.com

To become familiar with the fundamentals of game design, we recommend that you work through the Game Design Workbook. This will give you a strong foundation from which you can guide your students through the game design process.

In the Game Design Document Template you will produce a starting game concept with the **Mashup exercise** and will then be shown how to produce concepts for...

- Goal and Story
- Game Space
- Components (Characters)
- Components (Items & Objects)
- Challenges
- Rules and Mechanics

Once the GDD Template is complete, you will then see how students can share their game design and reflect on their learning through...

- The Elevator Pitch
- Collating Feedback
- Considering Research

LOCAL CURRICULUM

9 How can Game Development work in with your local curriculum?

Think about how other curriculum areas and school contexts could be incorporated into your own game ideas.

Using the coding skills learned in the Computational Thinking booklet, your students could design and develop all sorts of games based on all sorts of meaningful subjects.

Check out these examples made by our own rangatahi and tomorrow's game developers.

https://takaro.gamefroot.com

Here are some ideas that students could base their games on:

- Mauī's legendary adventures
- A local historical figure or event
- The challenges of life in your part of Aotearoa
- A celebration of the unique character of your local area
- An indigenous local story
- Human rights in NZ
- Tiriti o Waitangi
- Rāhui climate change and being a guardian

How would you integrate learning areas into your game? Could your students make a language-learning game? Could they make a game that incorporates concepts of maths or science? Could they have a strong storytelling component to their game?

Play game:
https://bit.ly/nz-squirm-germ

GAMEFROOT

http://make.gamefroot.com

This section is all about using the Gamefroot platform and becoming a master in your classroom for your students.

Looking for additional online support?

Browse to make.gamefroot.com, login and click on the help button.

Looking for additional Tutorials

Browse to make.gamefroot.com, click on the tutorials button to find extra tutorials.

Looking for the teacher dashboard?

Browse to make.gamefroot.com, log in and click on the clubs link.

GAMEFROOT FAQ

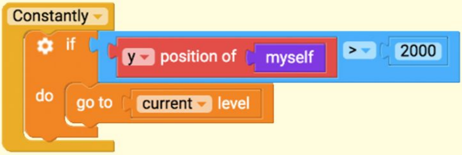
Why do assets get pushed away when I run up to them or jump on them?

All objects have physics by default. If you collide with them, you will impart force to them. You can either disable their physics or make them immovable with a simple script: "When created, set immovable true" or "When created, set physics enabled to false" (blocks from Events and Physics).

Why can I can fall off the outer edges of my game?

Even though it looks like your level has limits, the game world is infinitely large. The visible level size is only a guide. To prevent the player from falling outside your level, block them off with tiles or solid objects.

You could also code a barrier into the player script:



```
Constantly
  if   y position of myself > 2000
  do   go to current level
```

How can I stop seeing white space at the edges of my level when I center the camera?

Use the “Camera boundaries” blocks from Looks: “set camera x minimum to 0” and “set camera y minimum to 0”.
Try different maximums depending on how large your level is.

How do I select multiple assets at once and drag them to reposition them?

Use the selection tool (mouse cursor icon) and drag over the objects to select them. Then click and drag one of them to move them all together.

Can I use one script on multiple objects?

Yes, open the Scripts sidebar (</>) and select the correct script, then click on the objects in your level to attach the script to them. They will all behave in the same way.

Is there a way to copy script blocks from one script to a new script?

In the Script Editor, click on the menu button (☰) in the top left and select Export. Right-click on the code, click Select All, then right-click and select Copy. Close that script, make a new script, then open the menu from the top-left and select Import. Right-click in the text area and select Paste.

If I put in a temporary asset to stand in for the "real" asset, and script the mechanics I want, how do I then transfer the script to the correct asset?

Delete the asset and place the new asset. Open the Scripts sidebar, select your script, then click on the new asset to attach the script.

Can I draw my own assets and use them in Gamefroot so I can add scripts to them?

There's nothing stopping you drawing your own art and importing it into your game! You can take photographs of the landscapes around your area and use these as game backgrounds.

You could use any photo editing or drawing software (e.g. Photoshop). Photopea is a free online alternative **https://www.photopea.com/**

When drawing new game art, you can grab a screenshot of your current game, and then draw your new art over the top. Make a new document 960 x 540 pixels, then import your screengrab and draw over it (or import existing images). A tile is 48 x 48 pixels, and a character is roughly 100 pixels tall.

Make sure to use layers and paint onto a transparent layer so that you aren't painting straight onto your background. That way you can hide your background layer when you go to export your assets as PNGs.

My asset is in the wrong layer and it has a script. How do I move that asset to another layer without losing my script?

You won't lose the script. You need to delete the asset, then select the correct layer and place the asset in your level. Then open the Scripts sidebar, select the correct script, and click on the object to reattach the script.

Can I duplicate/copy and paste objects in the level?

Not yet, but this is coming soon.

Can assets be grouped together?

No, not in the current version of Gamefroot.

How do I use Clubs in my classroom?

You can create a Club to keep track of your students' work. Click on "Clubs" and then click "Create Club". Fill in the fields, and then share the club link with your students. They need to be logged in, then they can browse to your club link to be added to your club. From inside your club you can see members, change their passwords, and play or remix their published and unpublished games.

Is there a teacher dashboard?

Yes, you can access this from within your Club page (see above). From here you can see members, change their passwords, and play or remix their published and unpublished games.

How do I share my game?

Click the Share button in the top-right (⋖) to get a remix link that you can share. To Publish your game online, click File, Publish Online.

GAME DEVELOPERS KUPU

English	Te Reo
play / start	timata
pause	tupaku nei
level up	hekepanuku
epic	tauke
boss battle	whawhai e
game over	te mutunga
help	awhina
exit / quit	putunga
leaderboard	papa whakataetae
campaign	whakatakanga
badge	pine
you win	kua toa
you lose	kua hinga
about	e pa ana
player	kai tākaro
instructions	tohutohu

With special thanks to...

Edwin McRae - Game writer

Tim Harford - Digital teach teacher

Laura Jones - Project manager

Professor Tim Bell - The CT Wizard

Josh Walker - Graphic designer

AKHB - Mystery support person

William Young - Gamefroot advisor

A games industry partnership between

MINISTRY OF EDUCATION
TE TĀHUHU O TE MĀTAURANGA

GAMEFROOT