**Digital Technologies | Hangarau Matihiko**

**Level 6 – New Zealand Curriculum**

# Teaching and learning programme

## Game development

**MINISTRY OF EDUCATION**
TE TĀHUHU O TE MĀTAURANGA

**External links to websites**
The Ministry of Education does not accept any liability for the accuracy of information on external websites, nor for the accuracy or content of any third-party website accessed via a hyperlink from this resource. Links to other websites should not be taken as endorsement of those sites or of products offered on those sites. Some websites have dynamic content, and we cannot accept liability for the content that is displayed.

**Summary**

This programme will develop students' understanding of the requirements of game development and Agile processes. Students will learn to use Agile Scrum methodologies to develop a successful digital outcome.

## By the end of this teaching and learning programme, students will be able to:

- understand the requirements of game development
- follow Agile processes to develop a fun and entertaining game in a programming language of their choice.

## Duration (terms, weeks, teaching periods)

1 term: 9–10 weeks

## Key teaching and learning concepts – the big ideas

Game development principles; Agile software development methodologies (iterative development)

## Alignment to NZC and/or Te Marautanga

### DTHM – Computational thinking for digital technologies: Progress outcome 6

- Areas of modular structured programming.

### DTHM – Designing and developing digital outcomes: Progress outcome 4

- Iterative development of a digital outcome.

### NZC – Technology: Technological Practice, level 6

- Working through the technology process of planning, brief development and outcome development and evaluation.

### Key competencies focus areas: Managing self and relating to others

- Managing their digital work and time management to meet project milestones
- Relating to how others think and feel when using digital solutions.



## Links to other learning areas

- Technology
- Maths
- Design

## Teaching and learning pedagogy

Group work – teachers could allow students to work in groups of 2–3 students on a larger game.

Differentiation – allowing students to work at a pace and level they are capable of.

Shared learning – encouraging students to share their ideas, gain feedback, and seek out appropriate solutions.

## Prior knowledge/place in learning journey

This would ideally be suitable to teach in terms 2 or 3 of a year 11/NCEA Level 1 course after introductory programming concepts have been taught.

Students are expected to have basic understanding of a programming language prior to this programme. For example, Visual Basic, Scratch, Python, Small Basic, Gamemaker.

Teachers will need to work with students to create an initial proposal. This will be updated and modified as part of the iterative process.

## Resources required

- computers
- programming development environment
- file-sharing method
- pen/paper/scanner/camera
- coloured paper/scissors/pens
- Post-it notes

## How you might adapt this in your classroom

Teachers could set a theme for all students to work on. Possible themes could be: historical, futuristic, environmental issues or awareness, sport, water or ocean.

## Assessment

AS91884 (1.8): Use basic iterative processes to develop a digital outcome (6)

# Term outline

**The Learning context:**

| What is being covered | Approximate duration | Specific Learning Outcomes Students will be able to: | Learning Activities | Resources |
|---|---|---|---|---|
| Research game development | 1 week (4 hours' class time) | • Understand the process of developing a game | • Review of brief/feedback<br>• Set up trialling and testing log, understanding the requirements<br>• Case study – Blackout sports (concept designs)<br>• Tangential learning or edutainment<br>• How to be creative<br>• What makes a good game?<br>• Game design elements | • Trialling and testing log template (student use)<br>• Game design template (student use)<br>• 1 Game development teacher notes PPT |
| HCI analysis | ½ week (2 hours' class time) | • Analyse game interfaces to understand what makes a good user interface | • What are heuristics?<br>• Examples of game elements that relate to the heuristics | • 2 HCI teacher notes PPT |
| Planning and modelling | 2 weeks (6–8 hours' class time) | • Have a clear guide to what they are going to make and be open to changing their ideas based on feedback | • Choose sketching/designing tools (trailling)<br>• Get feedback and modifying designs<br>• Implications: what do we need to consider and why?<br>• Create a paper prototype & get feedback | • 3 Modelling teacher notes PPT |

## The Learning context *(continued)*:

| What is being covered | Approximate duration | Specific Learning Outcomes Students will be able to: | Learning Activities | Resources |
|---|---|---|---|---|
| Agile | ½ week (2 hours' class time) | • Understand what Agile software development is and how it can improve team work | • What is Agile software development (Scrum)?<br>• How can we use it for our project?<br>• What tools shall we use (trialling)? Eg: Post-its, Trello<br>• Set up a Scrum board | • 4 Agile teacher notes PPT |
| Game development | 5 Weeks | • Work in an iterative way to develop small components of their game | • Iterative process of building, trialling, testing and refining.<br>• Regular weekly "stand-up" meetings and reviewing/updating of tasks<br>• Consider relevant implications | • 5 Making your game teacher notes PPT |
| Final evaluation and GameCon | ½ week (2 hours' class time) | • Show off their games and reflect on the project, positively and constructively learn from the process. | • Measure against requirements/ specifications in proposal/brief<br>• Hold a game convention (GameCon) – game testing<br>• What lessons have we learnt about:<br> • Agile, working with teams?<br> • breaking down big problems into smaller components?<br> • game and software development?<br>• Make notes for next year's class or next time you make something | |