



Unit Overview

Target year group: Year 7

Projected Duration: 8 lessons

This scheme of work aims to teach learners the fundamentals of games programming using Kodu Game Lab, which is a visual game development environment. Using Kodu Game Lab learners will develop a range of key skills which include drawing and sculpting a world, adding character and objects. The use of When and Do instructions to control characters and objects including the use of paths and pages. Once learners have built their skills they are required to design, create, test and evaluate their own game.

Relevant Prior Knowledge

Some learners may have used Kodu Game Lab at primary school.

New Language for Learning

- | | |
|-----------|------------|
| ↘ Run | ↘ Path |
| ↘ Terrain | ↘ Wall |
| ↘ Objects | ↘ Bridge |
| ↘ Player | ↘ Settings |
| ↘ When | ↘ Analysis |
| ↘ Do | ↘ Design |
| ↘ Genre | ↘ Testing |

Resources

- ✓ Teacher's PowerPoint
- ✓ Help Cards
- ✓ Lesson 2 Starter – Match-up Activity
- ✓ Homework worksheet
- ✓ Game Design Template
- ✓ Peer Assessment Sheet

Homework

Create a virtual aquarium to simulate artificial intelligence in fish.

Cross-Curricular Links

Literacy:

- ↘ Plan the storyline to accompany a game and also to describe the environment and the backstory.
- ↘ Use of PEE when writing evaluation of completed game.

Numeracy:

- ↘ Logic and sequencing.
- ↘ Use of timers and scores to track values.

Assessment

Assessment is based on the [Progression Pathways Assessment Framework](#) from Computing at School.

Pathway: Algorithms

- | | |
|----------------|---|
| Level 3 | ☑ Design solutions that use repetition and two-way selection. |
| Level 4 | ☑ Design solutions by decomposing a problem and creates a sub-solution for each part. |

Pathway: Programming & Development

- | | |
|----------------|---|
| Level 3 | ☑ Create programs that implement algorithms to achieve given goals. |
| Level 3 | ☑ Declare and assign variables. |
| Level 3 | ☑ Uses post-tested loop and a sequence of selection statements. |
| Level 4 | ☑ Uses a variable and relational operators within a loop to govern termination. |

Support and Extension

Less able learners may take longer than anticipated on some of the tasks. Lesson 4 may be omitted for low ability groups.

More able learners can be challenged individually – each task gives room for improvement and own game design can be as complex as students wish.

Lesson 1

Activities:

- ↘ Introduce learning objective and outcomes.
- ↘ Introduce what Kodu is and what it does, show example game.
- ↘ Demonstrate how to make a world with different:
 - Heights
 - Types of terrain
 - Varieties of colours
 - Water/liquid
- ↘ Demonstrate how to add objects (i.e. apples) to the world.
- ↘ Learners should create a world of their own and include water and objects to be collected.
- ↘ Demonstrate how to add a character to be controlled by the user and how to make the character collect chosen objects.
- ↘ Learners should add a character to the world and program it to be controlled by the user and can collect objects by eating them.
- ↘ **Plenary:** Review progress against the Learning Outcomes (hands or thumbs up for each one).

Lesson 2

Activities:

- ↘ Introduce learning objective and outcomes.
- ↘ In *pairs* learners should complete the starter activity by matching up Kodu's instructions to the correct boxes.
- ↘ Go through answers to starter activity on the board.
- ↘ Demonstrate how to add paths to a world to make walls and roads (also demonstrate how to add a bridge across the water).
- ↘ Learners should add walls and roads, appropriately, to the world they created last lesson.
- ↘ Demonstrate how to add a plain (invisible) path to be used for controlling a character's motion, explain how to program this along with how to make characters shoot.
- ↘ Learners should add a plain path to their world and program an enemy to follow the path and shoot at the player. They may also wish to program their player to shoot when a certain button is pressed.
- ↘ **Plenary:** Review progress against the Learning Outcomes (hands or thumbs up for each one).

Extension:

- ↘ Students who have completed all activities should be looking at the World Settings and experimenting with changing variables in here.

Lesson 3

Activities:

- ↘ Introduce learning objective and outcomes.
- ↘ Show an example racing game on the board (note there should be a different coloured terrain to mark the end of the track). Demonstrate and discuss how to program the player so that he/she can reach the end of the track and wins the game.
- ↘ Learners should be able to develop their own racing game, independently, using the skills they have developed over the last two lessons.
- ↘ **Plenary:** Review progress against the Learning Outcomes (hands or thumbs up for each one).

Learning Objective:

Understand how to use the terrain tools and control a character's movement using Kodu Game Lab.

Learning Outcomes:

- ✓ GOOD: A world demonstrating different heights, types of terrain and water. Collectable objects added to the world.
- ✓ BETTER: Character added to the world and controlled by the user.
- ✓ BEST: Score added to keep track of the number of objects collected.

Learning Objective:

Understand how to add different types of paths to a world and use these to control enemies or NPCs within a game.

Learning Outcomes:

- ✓ GOOD: Walls and roads added to the world to make it look more realistic.
- ✓ BETTER: Use a plain path to control an enemy or NPC's movement.
- ✓ BEST: World settings tweaked to improve the overall appearance.

Learning Objective:

Understand how to combine programming and design skills learnt so far to use paths and opponents within a racing game.

Learning Outcomes:

- ✓ GOOD: Basic racing game that is won when the player reaches the end.
- ✓ BETTER: Opponent added that races against the player by following a path.
- ✓ BEST: Game Over displayed when enemy reaches the end first.

Lesson 4 (may be omitted for low ability groups)

Activities:

- ↳ Introduce learning objective and outcomes.
- ↳ Demonstrate how to use pages to change the speed when a character bumps into an object.
- ↳ Learners should add to their racing game such that their player can bump into objects to gain speed bursts and slow down when they drive over terrain that isn't part of the track.
- ↳ **Plenary:** Review progress against the Learning Outcomes (hands or thumbs up for each one).

Lesson 5

Activities:

- ↳ Introduce learning objective and outcomes.
- ↳ Explain the use of the System Lifecycle and go through different game genres and potential audiences.
- ↳ Learners should think of three ideas for their game and write down the genre, setting and audience for each idea.
- ↳ Learners should use 30 seconds of SILENT thinking time to choose their favourite idea and think about how it will look, potential storyline and characters. They should then share their ideas with their neighbour in a Timed Pair Share activity.
- ↳ Learners should open the Game Design Template and fill out each section to plan their games.
- ↳ **Plenary:** Review progress against the Learning Outcomes (hands or thumbs up for each one).

Lessons 6 & 7

Activities:

- ↳ Introduce learning objective, outcomes and keywords.
- ↳ Recap on last lesson.
- ↳ Learners should create the game they have designed in Kodu Game Lab (they should work on this independently).
- ↳ **Plenary:** Review progress against the Learning Outcomes (hands or thumbs up for each one).

Lesson 8 (This may take two lessons to complete.)

Activities:

- ↳ Introduce learning objective and outcomes.
- ↳ Question learners on why we are testing the games.
- ↳ Explain what a Test Table is used for and how to fill this in.
- ↳ Learners should create their own test table and use this to fix any problems with their games.
- ↳ Learners should then ask a friend to play their game and complete a Peer Assessment Sheet to give their feedback.
- ↳ Learners should write an evaluation of their game.
- ↳ **Plenary:** Review progress against the Learning Outcomes (hands or thumbs up for each one).

Learning Objective:

Understand how different pages can be used to change behaviour of objects within a game.

Learning Outcomes:

- ✓ GOOD: Additional objects added to the racing game that act as speed boosts, using pages to change the speed.
- ✓ BETTER: Areas of land used to slow down the racers using different pages.
- ✓ BEST: Finished racing game that is visually appealing and includes some extra features of your own.

Learning Objective:

Be able to use previous knowledge of games to produce a concept idea and plan for own computer game.

Learning Outcomes:

- ✓ GOOD: Decide on a game idea and produce a plan of how this will look.
- ✓ BETTER: Plan the background story and characters to the game.
- ✓ BEST: Completed game documentation which clearly covers all features of the game in detail.

Learning Objective:

Understand how to combine programming techniques learnt in Kodu Game Lab to produce a game to meet a chosen design.

Learning Outcomes:

- ✓ GOOD: Own game create using Kodu Game Lab.
- ✓ BETTER: Own game created that matches the design documentation (some bits may not work as planned).
- ✓ BEST: Own game completed using a range of programming techniques and works as expected.

Learning Objective:

Understand the importance of testing and reviewing a game to make it better.

Learning Outcomes:

- ✓ GOOD: Completed test table and fixed any errors highlighted as a result of testing.
- ✓ BETTER: Completed peer assessment sheet giving useful feedback to a friend.
- ✓ BEST: Detailed evaluation explaining results of testing, peer assessment and own opinions.