Westlake Boys High School

Assessment

Digital Technologies

AS91906v1 Level 3 Credits 6

Use complex programming techniques to develop a computer program.

Student Name:

Grade Awarded:

Not Achieved	Achieved	Merit	Excellence

Overall comments:

Teacher Name: _____

Signature: _____

Date: ____/____/2025

The work that you submit for this assessment needs to be your **own**.
You may not use AI to generate the entire program.
See the NEXT page for explicit information.

Assessment Task

Important: Read through **all** the instructions in this task before you start working on it.

In this assessment, you need to develop a computer program by using complex programming techniques.

What you will be assessed on:

You will be assessed on how well you do the following.

- Write code for the program to allow it to perform the specified task Use advanced techniques to develop the program.
- Set out the program code.
- Document the program.
- Test and debug the program.

Use of AI in this assessment:

DO:	DON'T
You may use AI tools to clarify programming concepts, syntax, or debugging approaches.	Do not use AI to generate the entire program or portions of code.
If you encounter an error you can't resolve, AI tools can help explain what the error message means or suggest ways to fix it. *	Do not use AI tools to debug your program without referencing the tool used/outcome in your testing documentation.
You can use AI to brainstorm potential test cases or scenarios to ensure your program meets the brief.	Do not use AI to bypass the learning process or tasks intended to assess your skills and knowledge.
You can use AI as a 'Peer review' tool, asking it to check your conventions of the language. *	

Important Note:

You must be able to explain every line of code in your program. Submitting your assessment without understanding of the program will bring the authenticity of your work into question.

* *If you use AI in this manner, it must be documented in your testing documentation.*

Assessment Task

Arcade Game Development

You will be creating a refined Arcade Game with 3 levels. You will be crafting and refining the gameplay, while adding additional features.

You will incrementally create and develop your game based on the results of your own testing, user testing, and feedback.

It is recommended that your game is an arcade platformer style of game. Other styles may be negotiated with your teacher.

Your game must include at least 3 additional features. These can be selected from the list below, or other features negotiated with your teacher.

Possible features:

- Instruction or introduction screen
- Creating own assets including sound and sprites
- Establishing a life counter with a game over result
- Collecting items to be able to advance to the next level
- Creating a timer
- Creating moving platforms
- Create moving enemies
- Reading from, and writing a high score to a file
- Character effect (for example invulnerability timer or character scale effect)

These are only suggestions. *Your choice of additional features must be discussed with your teacher **before** you start the development of your game* to ensure your game meets the assessment criteria.

Submission

- You must hand in your completed program
- Your completed testing document showing evidence of incremental development from your own and user-testing including valid, boundary, and invalid test types.
- Evidence of your legal right to use the assets in the game (sprites etc.)
- A video showing a successful, full play-through of the game

Assessment Schedule

AS91906v1 Level 3 Credits 6

Use complex programming techniques to develop a computer program.

Achieved: Use complex programming techniques to develop a computer program involves:		
Writing code for a program that performs a specified task	- The game is created to meet the given task.	☐
Using complex techniques in a suitable programming language: • Uses variables storing at least two types of data (e.g. numeric, text, Boolean, object) • Uses sequence, selection, and iteration control structures • Takes input from a user, file, sensors, or other external source • Produces output	- The game uses sprites, sprite lists, and numeric data. - The code has actions, conditional statements, and while or for loops. - The code takes user input and responds to user input. Graphics are displayed on the screen.	☐
Uses two or more complex programming techniques.	- Writing code for a graphical arcade game. - Object-oriented programming using class(es) and objects defined by the student. - Using the Arcade library –(third party or non-core API, library or framework).	☐
Setting out the program code clearly and documenting the program with comments	The code follows the main features of the PEP-8 Python conventions. - Blank lines are used to visually break up the code. - Class names use the CapWords convention. - Variable and function names should be lowercase, with words separated by underscores. - Constants are written in all capital letters with underscores separating words. - Comments have been used to label parts of the code.	☐
Testing and debugging the program to ensure that it works on a sample of *expected* cases.	- The game functions as intended. - Some testing of expected cases have occurred and been documented.	☐

Merit: Use complex programming techniques to develop an informed computer program involves:		
Advance programming at merit or above	- Game has a variety of features implemented. - - The program includes classes defined by the student.	☐
Documenting the program with appropriate variable/module names and organised comments that describe code function and behaviour.	- The code follows most of the features of the PEP-8 python conventions. - Variable names are well chosen and help with the reading of the code. - Function names are well chosen and help with the reading of the code. - Comments have been used to describe the behaviour of the code. - Comments have been used to remind the coder what is happening. - Docstrings summarise each function and describe what it does.	☐
Following common conventions for the chosen programming language	- The code follows most of the features of the PEP-8 python conventions.	☐
Testing and debugging the program <u>effectively</u> to ensure that it works on a sample of both *expected* cases and relevant *boundary* cases.	- Students have evidence of testing **expected** and **boundary** cases. - Students have gained feedback and as a result made relevant changes.	☐

Excellence: Use complex programming techniques to develop a refined computer program involves:		
Ensuring that the program is a well-structured, logical response to the specified task	- The game code is well structured. - The game code works in a logical way. The game code contains a number of user-defined classes.	☐
Making the program flexible and robust • **Using actions, conditions, control structures and methods, functions, or procedures effectively** • **Checking input data for validity** • **Correctly handling expected, boundary and invalid cases** • **Using constants, variables and derived values in place of literals.**	- Functions are well named. - Functions have a clear purpose. - Functions split the tasks of the code up into discrete areas. - Sprites cannot move off the screen or through the walls. - The game finishes at the correct time. - Constants and variables are well named and used. - Constants, variables and derived values have been used in place of literals.	☐
<u>Comprehensively</u> testing and debugging the program. **Testing *expected*, *boundary* and *unexpected* test cases.**	- Evidence of incremental creation of the game with tests and feedback at most increments. - Evidence shows players testing out the game and as a result adapting the game as a response to feedback. - The game functions as expected by testing all possible scenarios, including expected, boundary, and unexpected. - Testing is logical, follows a structure and demonstrates that the game is functional in all test cases.	☐
Comments		