**Name: Date: Excellence / Merit / Achieved / Not Achieved**

|  |  |  |  |
| --- | --- | --- | --- |
| **Achieved: Use complex programming techniques to develop a computer program involves:** | | | |
| Writing code for a program that performs a specified task | Game code  Student interview | The game is created.  Achieved level programs deviate in a few small ways from the sample code and platformer tutorial.  The student can describe the function of all the code submitted.  *(A tick here indicates that when questioned the student described the code and/or has been observed working with the code)* | ⬜  ⬜ |
| 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 | Game code | In a working arcade game these will have all occurred naturally.  The game uses sprites, sprite lists and numeric data.  The code has actions, if statements and while or for loops.  The code takes userinput and responds to user input.  Graphics are displayed on the screen. | ⬜ |
| Uses two or more complex programming techniques.   * programming or writing code for a graphical user interface (GUI) * object-oriented programming using class(es) and objects defined by the student * using third party or non-core API, library or framework | Game code | In a working arcade game the following will have occurred naturally.   * Writing code for a graphical arcade game. * Object-oriented programming using class(es) and objects defined by the student. * Using the Arcade library. | ⬜ |
| Setting out the program code clearly and documenting the program with comments | Game code | 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 * Function names should be lowercase, with words separated by underscores * Variable names should be lowercase, with words separated by underscores * Constants are written in all capital letters with underscores separating words * Comments have been used * Comments label parts of the code | ⬜  ⬜  ⬜  ⬜  ⬜  ⬜  ⬜ |
| Testing and debugging the program to ensure that it works on a sample of expected cases. | Game code | The game functions as expected most of the time then some testing at an achieved level is assumed to have occured.  Student describes some testing in the Sprint Tracking Documents  OR  Student can verbally describe to the teacher the testing undertaken. | ⬜    ⬜ |
| **Notes on Achieved:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Merit: Use complex programming techniques to develop an informed computer program involves:** | | | |
| Advance programming at merit or above | Game code | Code shows some significant changes from the example code.  Game has new features implemented.  Player is in a class created by the student. | ⬜  ⬜  ⬜ |
| Documenting the program with appropriate variable/module names and organised comments that describe code function and behaviour | Game code | 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. * Comments summarise each function and describe what it does. | ⬜  ⬜  ⬜  ⬜  ⬜  ⬜ |
| Following common conventions for the chosen programming language | Game code | The code follows most of the features of the PEP-8 python conventions. | ⬜ |
| Testing and debugging the program effectively to ensure that it works on a sample of both expected cases and relevant boundary cases. | Sprint Summary Sheets | Sprint summary sheets demonstrate incremental creation of the game.  Sprint summary sheets detail players testing out the game.  Sprites can move in and out of corners of the map or walls.  Scoring works if used and is tested to be working.  The game ends at the correct time and this is demonstrated in the testing. | ⬜  ⬜  ⬜  ⬜  ⬜ |
| **Notes on Merit:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | Game code | The game code is well structured  The game code works in a logical way, following the style of the tutorials.  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 are well named and used.  Variables are well named. | ⬜  ⬜  ⬜  ⬜  ⬜  ⬜  ⬜ |
| Comprehensively testing and debugging the program. | Sprint summary and testing details | Sprint summary shows and describes incremental creation of the game with tests at most increments.  Sprint summaries show players testing out the game and responding to feedback  The game functions as expected.  Testing is logical, follows a structure and demonstrates that the game is functional. | ⬜  ⬜  ⬜  ⬜ |
| **Notes on Excellence:** | | | |