Title: Retro Game

Start Date: Wednesday, 26 February 2014

Assessment Day: Wednesday, 11 June 2014

Assessable units of competency

ICAPRG406A - Apply introductory object-oriented language skills
ICAICT417A - Identify, evaluate and apply current industry-specific technologies to meet industry standards

General description

You are to create a Top-Down Shooter Retro style 2D game using the given framework. Examples of this style of game can be found on portal. The major components of the game will be built up through application of programming concepts and through application of class exercises.

Through the completion of this assessment, you will be able to demonstrate the ability to design, plan and build a simple game, create and code "bug and error free" program and have a understanding of C++ programming constructs, classes, functions and data structures.

When the game is loaded, the user should be presented with a splash screen followed by a main menu that contains at least 3 options (Start Game, High Scores and Quit). The game that you have chosen to re/make must be approved by your class teacher, and the final result of this assessment must represent this game, you are able to add new elements to this game however overall the game should be homage to this genre and style of game.

You are to document all code with appropriate comments and include a "how-to-play" text file containing player controls and instructions or alternatively have these presented to the player in game.

Knowledge and skills

Listed here is the knowledge and skills you'll be learning and on which you will be assessed.

- Object-oriented programming concepts.
- Small-size application development
- Basic Syntax and structure of C++
- Use of an IDE (Integrated Development Environment)
- Use of coding standards

Evidence specifications

This is the specific evidence you must prepare for and present on assessment day to demonstrate you have competency in the above knowledge and skills. The evidence must conform to all the specific requirements listed below.

1. A 2D top down shooter retro styled game
2. Multiple header and source file pairs High Scores
3. Source code contains and demonstrates arrays and modular programming
4. Sorted high score table functionality
5. Demonstrated use of your IDE for debugging and problem solving
6. Source code that adheres to AIE's coding standards

Your roles and responsibilities as a candidate

- Understand and feel comfortable with the assessment process
- Know what evidence you must provide during your assessment
- Take an active part in the assessment process
- Be ready for the assessment at the nominated time

Assessment instructions for candidate

METHOD OF ASSESSMENT

Assessment will be conducted by you personally presenting evidence that demonstrates your competence in a short interview with your assessor. The evidence you must prepare and present is described above in this assessment criteria document. Assessments will be conducted on a specific day recorded above in this assessment criteria document.

ASSESSMENT CONDITIONS

You will have approximately 10 mins to present your evidence that demonstrates your competence. It is your responsibility to be prepared. If you have forgotten something or made a small mistake you may correct it, however the assessor may choose to assess other candidates who are better prepared and return to you if time permits. Upon completion of the assessment you will be issued with feedback and a record of the assessment, which you will need to acknowledge that you have accepted the result.

If you are absent on the nominated assessment day (without prior agreement or a sufficient documented excuse) you will be assessed as not yet competent.

GRADING

The assessment you are undertaking will be graded as either *competent* or not *yet competent*.

REASSESSMENT PROCESS

If you are assessed as being not yet competent you will receive clear, written and oral feedback on what you will need to do to achieve competence. You will have one (1) week to prepare your evidence for a reassessment. You will be given only one reassessment opportunity. If you are unsuccessful after your reassessment you will be required to attend an intervention meeting with your Head of School to discuss your progress.

REASONABLE ADJUSTMENTS

We recognise the need to make reasonable adjustments within our assessment and learning environments to meet your individual needs. If you need to speak confidentially to someone about your individual needs please contact your teacher.

Assessment rubric

This table defines exactly what is required to be successfully deemed competent.

Evidence	Definition of Competent for Practice production
1. A runnable project	Submitted project source code should: • build successfully and run • have no errors • minimal warnings (less than 10) • have no game-breaking bugs upon playing
2. Multiple header and source file pairs	Submitted source code should demonstrate definition and use in code of: • Classes – at least three times • Multiple class constructors - at least once • A class with a User-defined aggregation function – at least once • Class inheritance at least two levels deep – at least once • Polymorphism – at least once
3. Arrays and modular programming	Submitted source code should demonstrate: • Separate code into functions – at least twice • Implement an array – at least twice • Logic structures and loops – at least twice • Data types, operators and expressions – at least twice
4. Sorted high score table functionality	Submitted source code should: • Write to a Text file – at least once • Read from a text file – at least once • Perform standard-array processing algorithms (Insert / Search / Sort) – at least once
5. Debugging and problem solving	During the subject you will be assessed, via teacher observation, on your ability to utilise debugging features in your IDE to identify and fix bugs in your code.
6. Source code that adheres to AIE's coding standards	The code that you write should adhere to AIE's coding standards. You should use either Hungarian notation or Camel Case as your coding convention. Your code should be appropriately commented.