Index of competency evidence.

CPP Evidence

|  |  |  |
| --- | --- | --- |
| **EVIDENCE** | **Definition** |  |
| 1. A project that runs | Build Successfully and run |  |
|  | Have no errors |  |
|  | Minimal Warnings (Less than 10) |  |
|  | Have no game-breaking bugs on playing |  |
|  |  |  |
| 2. Multiple header and source file pairs | 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 | 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 | Write to a Text File - at least once |  |
|  | Read from a text file - at least once |  |
|  | Perform standard array-processing algorithms - at least once |  |
|  |  |  |
| 5. Debugging and problem solving | Teacher Observation |  |
|  |  |  |
| 6. Source code that adheres to AIE's coding standards | Hungarian or Camel Case with appropriate comments |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

GCD Evidence

|  |  |  |
| --- | --- | --- |
| **EVIDENCE** | **Definition** |  |
| 1. Technical Documentation | A Static Class Diagram |  |
|  | A Collaboration/Sequence Diagram |  |
|  | An activity or state diagram |  |
|  | A detailed UML Document |  |
|  | Commented code |  |
|  |  |  |
| 2. Debugging demonstration & testing report | Use of tracing code |  |
|  | Use of break points and step through |  |
|  | Exception handling (try-catch) - at least once (Jason changed to use ASSERT at least once) |  |
|  | User Feedback - at least two comments from play testers with their names |  |
|  | Improvements - describe at least two improvements made on feedback |  |
|  |  |  |
| 3. High score database functionality | Extract, update & delete data stored in database - at least once |  |
|  | Manipulate database structure (Query, create & delete) - at least once |  |
|  | Maintain data integrity as the updates and deletes occur |  |
|  | Show high scores in a sorted manner |  |
|  |  |  |
| 4.Implemented GUI | Display text to the screen (score etc) - at least once |  |
|  | Responds to user input (menu / etc.) - at least once |  |
|  | Displays game play information (health/lives/etc) at least once |  |
|  |  |  |
| 5.Implemented Object Orientated Code | A release build executable |  |
|  | Multiple source code files |  |
|  | Use at least two collection of data. |  |
|  | Read / Write to binary file - at least once |  |
|  | Overload a function - at least once |  |
|  | Overload an operator - at least once |  |
|  | Have a class use multiple inheritance - at least once |  |