**SOFTWARE DESIGN & DEVELOPMENT**

**ASSESSMENT TASK NOTIFICATION**

|  |  |
| --- | --- |
| **SUBJECT** | SOFTWARE DESIGN AND DEVELOPMENT |
| **YEAR** | Year 12 |
| **WEIGHTING OF TASK** | 30% |
| **DATE TASK ISSUED** | Friday, 17 May 2019 |
| **DATE TASK DUE** | Thursday, 20 June 2019 5:00 pm student’s local time |
| **METHOD OF SUBMISSION** | **Github** |

**Task Description**

|  |
| --- |
| Students will design and develop a software solution based on the following client-developer scenario:  The *Restaurant at the End of the Universe* is losing customers and wants to attract more by providing a simple computer game for people to play while sitting around eating and drinking. They are not really sure what they want, but it has to be simple, easy to learn and engaging. They have in mind a ‘space invaders’ type game but are open to suggestions.  They also have in mind a leader board with rewards for the high score of the month, the high score of the year and an all-time high score.  The leader board should be able to be viewed and sorted on various criteria (eg. score, username, number of games played etc.)  The Software solution will comprise:   * A working program and associated code which *demonstrates a modular approach* * A design brief for the client including:   + work agreement outlining the agreed needs of the client and outcomes of the project   + identification of social, ethical and legal issues that may arise from the project and recommending steps to address them   + proposed solution including storyboards   + licence agreement * A user manual including:   + a guide to playing the game   + installation notes   + system requirements * Evidence of project planning and version management (in GitHub) including:   + regular and well documented commits, appropriate forks and use of a planning tool such as the project tool and issues   + use of debugging output statements (print statements), flags and stubs in early versions of the code * Intrinsic documentation including:   + meaningful variable names   + inline comments   + docstrings * Project documentation including:   + README file   + Structure chart   + Data dictionaries   + Test report |

**Outcomes assessed**

|  |  |
| --- | --- |
| **H3.2** | constructs software solutions that address legal, social and ethical issues |
| **H4.1** | identifies needs to which software solutions are appropriate |
| **H4.2** | applies appropriate development methods to solve software problems |
| **H4.3** | applies a modular approach to implement well-structured software solutions and evaluates their effectiveness |
| **H5.1** | applies project management techniques to maximise the productivity of the software development |
| **H5.2** | creates and justifies the need for the various types of documentation required for a software solution |
| **H5.3** | selects and applies appropriate software to facilitate the design and development of software solutions |
| **H6.1** | assesses the skills required in the software development cycle |
| **H6.4** | develops and evaluates effective user interfaces, in consultation with appropriate people |

Additional detail:

* More marks will be awarded for code which is well structured, readable (easy to understand) and efficient – but if you have to choose between readability and efficiency, choose readability.
* More marks will be awarded for an original and engaging solution
* It is preferable for your user manual and project documentation (apart from the README file) to be created as a wiki in the Github project. If you would prefer to use other software to create your user manual and documentation these files can be uploaded to your Github repository.
* A README file should be present in the root directory of each fork. It should give a brief overview of the project, its current status, and brief instructions – like a mini user manual.
* Evidence of planning and version management should be in the form of:
  + regular, well commented commits for minor changes, for example every time you do some work on the code etc.
  + forks for major changes (versions). It is expected that there will be at least 3 forks corresponding to the following versions: alpha (production), beta (pre-release testing) and release (Version 1.0)
  + The use of the GitHub project tool to raise issues (steps in completing the project). Issues should contain comments describing them in detail, how you plan to solve them, what further issues they raise and any other relevant thoughts.

Marking Criteria

|  |  |  |  |
| --- | --- | --- | --- |
| **Outcome** | **Band 5-6** | **Band 4-3** | **Band 2-1** |
| **Project design**  **H3.2, H4.1, H5.3, H6.1, H6.4**  **12 marks** | Creates a high quality design brief that proposes a well thought out solution which:   * is imaginative, original and meets all of the client’s needs. * identifies and addresses legal, social and ethical issues * includes a storyboard * includes a work agreement * includes a licence agreement | Creates a design brief that proposes a suitable solution which:   * meets the client’s needs. * identifies and addresses legal, social and ethical issues * includes a storyboard * includes a work agreement * includes a licence agreement | Creates a design brief that has some of the following elements:   * meets some of the client’s needs. * identifies and addresses legal, social and ethical issues * includes a storyboard * includes a work agreement * includes a licence agreement |
| **Software Development**  **H4.2, 4.3, H5.3, H6.1**  **18 marks** | Creates a high quality software solution which:   * efficiently fulfills the outcomes of the design brief * is very well structured, readable and applies a modular approach * has a high standard of intrinsic documentation | Creates a software solution which:   * fulfills the outcomes of the design brief * is well structured and applies a modular approach * has intrinsic documentation | Creates a software solution which:   * fulfills some of the outcomes of the design brief |
| **Project Management**  **H5.1, H5.3**  **12 marks** | Applies project management techniques to maximise the productivity of the software development including:   * highly effective use of a version management system (eg GitHub) * detailed, well documented planning using appropriate software (eg GitHub project tool) | Applies effective project management techniques including:   * use of a version management system (eg GitHub) * planning using appropriate software (eg GitHub project tool) | Applies some project management techniques including:   * use of a version management system (eg GitHub) * planning using appropriate software (eg GitHub project tool) |
| **Project Documentation**  **H5.2, H5.3**  **12 marks** | Creates high quality, informative documentation including:   * README * User manual * Structure Chart * Data dictionaries * Test report | Creates documentation including:   * README * User manual * Structure Chart * Data dictionaries * Test report | Creates some of the following documentation:   * README * User manual * Structure Chart * Data dictionaries * Test report |