|  |  |
| --- | --- |
|  |  |

|  |  |  |
| --- | --- | --- |
| **BTEC National Extended Diploma in** **Information Technology** | | |
| **Assessment Title: Designing, Building and Testing Games** | | |
| Assessor: Trevor Till | Student: | Course code: |
| Unit Number: 22 | Unit Title: Developing Computer Games | Assessment Code: 22.01 |
| Hand out Date :28/2/20 | Hand in date: 20/3/20 | Date handed in: |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Games Development** | P3 | P4 | P5 | P6 | P7 | M1 | M2 | M3 | M4 | D2 |  |  |  |

Please note that the grade that you have been awarded is subject to confirmation by the awarding body.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Scenario**  You are continuing to work on a placement from College with a Games Design company (CoolEdge Games Inc.,) and they want you to demonstrate your understanding of the Games Design process by designing, building and testing a small example game.  You can develop a game using your own user requirments, if you use your own requirments, make sure that there is iteration and selection present on the source code (you can use a timer for the game loop instead of iteration if necessary).  You may prefer to use the following requirements:  the program will have three pictures on the main form which will move across the screen to a finishing post in a number of steps, each moving a random distance across the screen in each step. Three players can select which of the three they think will finish first, and enter their guess into a text box on the screen. They will then click a button which will cause the images to move across the screen in small random steps until one of them crosses the finishing line. The program will then confirm which of the three has won and inform the player who chose that image that they had the winner. Add a button that will display previous winning scores and player names.  **Tasks**  **Task 1 - Design (P3, M1)**  Produce a design for the game, lists its requirements, draw/produce a copy of the interface (wireframe), produce a data dictionary naming and explaining the interface elements, and a data dictionary for the variables required, showing how they are declared in the program and explaining why the particular data types were chosen. (if necessary, produce a storyboard to show how the game will work.)  **Task 2 - Build the game (P4, P6, M2)**  You are now asked to implement the program, you must have used iteration and selection within the program (you could implement the game loop using a timer). Make sure that you add comments to the code and make it easy to maintain (layout, indentation, variable naming, use of functions). It is important that you clearly explain all elements of your program. Add a button or clickable area/link to the program which, when clicked, will show help for the user explaining how to use the program. You may implement this help in an alternate fashion (eg a printed user guide) but it must be present.  **Task 3 (P5, M3)**  Create a black box testing test plan. Carry out the plan with suitable test data and show evidence of your testing by writing the tests up in a table (where useful, also use annotated screenshots).  You will also need to carry out white-box testing. Use a bench testing method to impute values to your variables; show the code snippet for the routine that you bench test and annotate it with the imputed value of the variables in that code snippet.  The black box test plan could be of the form:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | Test | Reason for Test | Expected Result | Actual Result | Reason for Difference | Error Type | Action |   You will also need to carry out UX testing/evaluation, to see if the game is playable, understandable, enjoyable, replayable. Use a minimum of three users, write the test plan for them and submit their completed copies as evidence.  **Task 4 (P7, M4, D2)**  Give a detailed explanation of how the structure and design of a game can assist in the maintenance and capacity for extension with examples from your own design (eg comprehensive and clear, use of functions to separate code routines) and code (eg self-documenting code), Using feedback from users gained from task 3, write a series of recommendations for possible improvements to the game, clearly explaining how these suggestions would make it better. You will then need to write some documentation for the program. If you have completed all the above tasks you will have a good set of technical documentation. Put them together as a report, include a copy of the well laid out, commented code. |

**What to hand in:**

1. An electronic copy of all elements of the assignment: the designs, test plans, results, analysis data tables, code, working project, reports, etc., uploaded to the VLE.

**How to hand in your assessment**

This assessment must be uploaded onto the VLE (or in printed form to the library) by the hand in date on the front of the assignment brief.

If you have a valid reason why you will not be able to hand in the work on time (up to 3 days later), you will need to fill out and hand in an AE1. If you need to ask for a longer extension you will need to fill out and hand in an AE2 to your Head of School at least two days before the assessment deadline.

|  |  |  |  |
| --- | --- | --- | --- |
| **Student declaration:**  I have read the Assignment Submission Procedure. I agree that this is my own work or my own work and that of other members of my group. It has not been copied (plagiarised) from any other source e.g. the internet, a book, another student or group of students. I know that I may FAIL this assignment if my Head of School proves that this is not my own work. | | | |
| Student signature |  | Date: |  |

|  |
| --- |
| **Student feedback on what you liked and/or disliked about this assessment.** |

For staff use only.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Internal Verification** | This assessment has been verified and form IV8 completed: | IV name: | IV signature: | Date: |

|  |  |
| --- | --- |
| SMOG INDEX: | 11.98 |

N.B. Test for readability on scenario and tasks only.

**Assessor marking grid**

**Student name:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Grading criteria | Grading criteria | Achieved (✓)  Not attempted (NA)  Fail (F)  Referral 1 (R1)  Referral 2 (R2) | Comments/Feedback | Target date  for referral. |
| Objectives: Unit 22, Games Development | | | | |
| P3 | produce a design for a  computer game for a given specification |  |  |  |
|  |
|  |
| P4 | develop a computer game for a given specification |  |  |  |
|  |
|  |
| P5 | follow a test strategy to test and debug a computer game |  |  |  |
|  |
|  |
| P6 | produce user documentation  for a computer game |  |  |  |
|  |
|  |
| P7 | produce technical  documentation for a  computer game. |  |  |  |
|  |
|  |
| M1 | determine appropriate data types for a computer game  and show how they are declared |  |  |  |
|  |
| M2 | use appropriate selection and iteration methods for a  computer game |  |  |  |
|  |
| M3 | use a variety of testing tools |  |  |  |
|  |
| M4 | suggest improvements to a computer game following user feedback |  |  |  |
|  |
| D2 | explain how the structure and design of a game can assist in maintenance and capacity for extension. |  |  |  |
|  |
|  |

|  |
| --- |
| **First submission general comments** |
|  |

|  |  |
| --- | --- |
| Assessor signature: | Date |

|  |  |
| --- | --- |
| Learner signature: | Date |

|  |
| --- |
| **Referral and upgrade general comments.** |
|  |

|  |  |
| --- | --- |
| Assessor signature: | Date |

|  |  |
| --- | --- |
| Learner signature: | Date |