|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification details** |  | | |
| **Training Package code and title:** | ICT Information and Communications Technology | | |
| **National Qualification Code & Title:** | **ICT40915 Certificate IV in Digital and Interactive Games** | **State code:** | **AWE5** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Student Name** | |  | | | | |
| **Student Declaration** | | I declare that the evidence submitted is my own work:  ………………………………………….. | | | | |
| **Assessor Name** | | Joshua Ferguson under supervision of Brian Blasgund. | | | | |
| **Date Due** | | 24/03/2020 (Week 8) | **Date Received** |  | | |
| **Unit of Competency** | | | | | | |
| **National Code & Title** | **ICTPRG301 – Apply introductory programming techniques** | | | | **State code:** | AUV54 |
| **National Code & Title** | **ICTPRG413 – Use a library or pre-existing components** | | | | **State code:** | AUV59 |
| **Assessment Tool** | **Assessment 1 – Programming Activities (Tasks 1-3)** | | | | | |
| **Demonstration Tasks** | **Objective:** Learners will gain a firm understanding of introductory programming skills and techniques that can be used and developed in order to produce a digital video game. Learners will also gain a firm understanding of developing dynamic scripts and components for use in developing a digital video game.  **Task 1 – Character Controller**  Learners will begin to gain an understanding of developing dynamic components as they construct a character controller that players can use as an avatar in a video game.  **Task 2 – Game Managers**  Learners will consolidate their understanding of developing dynamic components and continue to expand their knowledge of introductory C# programming skills and techniques by constructing game managers to control the various gameplay elements of a video game.  **Task 3 – Small Game**  Learners will use their newly developed skills and knowledge to combine the generated components in order to create playable gameplay for a video game with and prepare it to be published.  **For submission:**  **Include this cover sheet in your submission document. Include references to all sources of information and resources used at the end of this assessment sheet.**  *Must include references to all information and resources. If your work is not referenced it will be considered as PLAGIARISM.* | | | | | |
| **Resource Requirements** | A USB drive to save/backup work on.  A pen, pencil and a notepad for note taking.  Access to the appropriate BlackBoard shell.  Project production documentation such as Hack n Plan (Game design project management software), the GDD (Game Design Document) and team correspondence. | | | | | |

|  |  |
| --- | --- |
| **Skills being assessed** | **Performance Evidence**  Evidence of the ability to:   * Apply programming language syntax, sequence, selection and iteration control structures to the development of an application, or game. * Produce an application, or game, that is designed and built from the program specifications. * Confirm that the created application, or game, meets the original program specifications, and obtain user sign-off for the completed program. * Identify, evaluate, and select the appropriate components that may be re-used. * Modify and integrate the re-used components into a project. * Test the functionality of the re-used components.   **Knowledge Evidence**  To complete the unit requirements safely and effectively, the individual must:   * Outline, and apply, current industry development and design methodologies. * Recognise and apply generic specification. * Define the content and structure of libraries. * Identify and describe common games programming languages, their syntax, and command structure. * Describe the development of small-sized applications or games. |

|  |  |
| --- | --- |
| **INSTRUCTIONS** | |
|  | |
| **TO THE ASSESSOR** | |
|  | |
| **Type of Assessment** | Activity Tasks – Programming Tasks 1-3 |
|  |  |
| **Duration of Assessment** | Within the course time limitations |
|  |  |
| **Location of Assessment** | Classroom, online and home |
|  |  |
| **Conditions** | Students are assessed individually whilst completing a digital game project, students work in production teams.  Their game project provides context for assessment, if their individual games do not cover the performance criteria then individual portfolio items may be used as evidence.  It is recommended that the game project is used as often as possible.  Please check the plagiarism policy available via the South Metro TAFE website. <http://www.southmetrotafe.wa.edu.au/currentstudents/forms>  Gather evidence to demonstrate consistent performance in conditions that are safe and replicate the workplace. Noise levels, production flow, interruptions and time variances must be typical of those experienced in the game development field of work, and include access to:   * The requirement documents. * The site documents. * Software development tools currently used in industry, such as a compiler. * An independent development environment (IDE). * The software development environment. * A reuse library. * The technical requirements. |
|  |  |

|  |  |
| --- | --- |
| **ICTPRG301 – Apply introductory programming techniques** | |
| **Elements and Criteria** | ***1.******Apply language syntax and layout***   * 1. *Apply basic language syntax rules*   2. *Use language data types, operators, and expressions to create a clear and concise code*   3. *Apply the variables and variable scope*   4. *Use the library functions in a program*   5. *Use commenting to create a clear meaning to the code*   ***2.******Apply control structures***   * 1. *Apply the language syntax for sequence, selection and iteration constructs*   2. *Use logical operators to create expressions for use in selection and iteration constructs*   ***3. Code using standard algorithms***   * 1. *Develop algorithms that use the sequence, selection and iteration constructs*   2. *Create and use arrays*   3. *Code the standard sequential access algorithms, for reading and writing text files, including end-of-file detection loops*   4. *Apply string manipulation*   ***4.******Test the code***   * 1. *Use debugging techniques to trace code execution and examine the variable contents to detect, and correct, errors*   2. *Create and conduct simple tests, to confirm that the code meets the design specification*   3. *Document the tests performed and results achieved*   ***5.******Create an application or game***   * 1. *Design an algorithm in response to basic program specifications*   2. *Develop the application or game to meet the program specification*   3. *Test and confirm that the application, or game, meets the initial specifications* |

|  |  |
| --- | --- |
| **ICTPRG413 – Use a library or pre-existing components** | |
| **Elements and Criteria** | ***1.******Identify potential re-use units***   * 1. *Analyse the project design and functionality, to identify re-use components*   2. *Source re-use components with the desired functionality*   ***2.******Evaluate re-use components for suitability in parent project***   * 1. *Evaluate the re-use component, or libraries, for suitability for use within the parent software project*   2. *Compare the functionality of each potential re-use component to the functionality required by the parent project*   3. *Evaluate the cost of implementing the re-use component*   4. *Consider the technical impact on the parent project design*   5. *Consider re-use component vendor licensing issues*   6. *Finalise the selection of re-use components*   7. *Document selection, evaluation and decision processes, as part of the parent project design documentation*   ***3. Incorporate the re-use components***   * 1. *Configure the development environment to include re-use components during the build process*   2. *Construct test programs or use provided example programs, to become familiar with re-use components, in preparation for incorporation into the parent project*   3. *Add re-use components to the parent project incrementally*   4. *Resolve re-use component dependencies*   5. *Assemble and test the parent project, with a focus on the functionality provided by the re-use components* |

|  |  |  |
| --- | --- | --- |
| |  | | --- | | **Assessment 1 – Programming Activity Tasks 1-3** | | You should be a part of a project group, which are assigned in the classroom. Groups are to consist of no more than 5 members. Within your groups you are to brainstorm game pitches and decide upon one which you will work on as a group for the duration of this course.  Once a team of up to five members has been assigned and a game pitch has been decided, this will give you context for all assessments within this course unless clearly stated otherwise. Each group member is to provide the required evidence of individual work being completed for each assessment. |   **TO THE STUDENT** |

**Part 1 – Character Controller**

Learner’s first task is to create a ***character controller*** that can then be further developed and used in projects as required. The controller must be created from a ***first person***, ***third person***, or ***isometric*** perspective and should be designed to be ***dynamic*** – each component of the character controller should be independent and modular so that they may be repurposed or modified without breaking. In order to achieve this, learners will need to ***consider the*** ***technical impacts*** of each component of the character controller as a whole.

The functionality of the character controller must include the following mechanics.

* Player movement
* A heath mechanic
* A level interaction mechanic (buttons/doors)

The scripts created for each component of the controller are to be written in the ***C# programming language***. Ensure that the ***syntax*** of each script is accurate, and each ***block*** of code is ***summarised with a comment***. All scripts must be ***individually*** created by each learner in order to provide evidence of each learner's understanding of data types, variables, operators, and expressions.

In the process of generating the required scripts, learners must also provide evidence of their use of debugging techniques in order to correct programming errors. Learners will need to use various techniques and tools available in order to ***conduct simple tests***, and ***trace the execution of code*** ***and the contents of variables*** while testing each component. Tests performed will need to have their ***results recorded*** via the appropriate form of documentation.

**Part 2 – Game Managers**

Learner’s second task is to create the various ***game managers*** and ***event systems*** that will monitor and control the various elements of the overall game. Some examples include managers for ***scoring***, ***gameplay events***, ***timers***, and ***saving & loading*** game states. As with the previous task, components should be built to be ***dynamic*** where possible. In order to achieve this learners will need to ***analyse the planned components*** in order to ***determine which can be dynamically used***, and ***consider their technical impact*** on the project before finalising the selection of components.

Learners are also to ***build*** the project ***for*** ***testing*** purposes throughout the development cycle. Each build should be ***stored and named appropriately*** to clearly indicate the specific psurpose.

**Part 2 – Game Managers *(Cont.)***

The functionality of the created game managers must provide evidence of the following mechanics.

* Write to and read from a text (.txt) file
* Manipulate the contents of strings
* Make use of C# arrays or lists

The scripts created for each component of the game managers are to be written in the ***C# programming language***. Ensure that the ***syntax*** of each script is accurate, and each ***block*** of code is ***summarised with a comment***. All scripts must be ***individually*** created by each learner in order to provide evidence of each learner's understanding of the required evidence.

In the process of generating the required scripts, learners must also provide evidence of their use of debugging techniques in order to correct programming errors. Learners will need to use various techniques and tools available in order to ***conduct simple tests***, and ***trace the execution of code and the contents of variables*** while testing each component. Tests performed will need to have their ***results recorded*** via the appropriate form of documentation.

**Part 3 – Small Game**

Learner’s third and final task is to ***combine*** the ***previously*** ***created*** ***components*** with ***other*** ***required*** ***components*** appropriately in order to ***create*** ***a*** ***complete***, ***functional*** ***game*** that ***adheres to the design specifications*** of the project. The final result ***must be*** ‘***complete***’ – in other words it must include a ***beginning*** (or tutorial), an ***end/finale*** sequence, and some ***gameplay in between***.

The functionality of the final game must provide evidence of the following mechanics and design specifications.

* Character controller
  + First person, third person, or isometric
  + Controlled by the player
  + At least one level interaction mechanic (buttons/doors)
  + Health
* Gameplay managers & event systems
* Data is read from & written to a text (.txt) file
* At least one complete ‘level’ that can be played from start to finish

Once a final version of the game has been completed learners need to ***create a final build*** of the application, as well as ***store*** and ***name*** ***it*** ***appropriately***.

All of the games scripts are to adhere to the design specifications and to be written in the ***C# programming language***. Ensure that the ***syntax*** of each script is accurate, and each ***block*** of code is ***summarised with a comment***. All scripts must be ***individually*** created by each learner in order to provide evidence of each learner's understanding of the required evidence.

In the process of generating the required scripts, learners must also provide evidence of their use of debugging techniques in order to correct programming errors. Learners will need to use various techniques and tools available in order to ***conduct simple tests***, and ***trace the execution of code and the contents of variables*** while testing each component. Tests performed will need to have their ***results recorded*** via the appropriate form of documentation.

**Assessment Submission Guidelines**

When submitting the assessment to BlackBoard, please put all files relevant to each individual part of the assessment in their own folder.

|  |  |  |  |
| --- | --- | --- | --- |
| **Assessment Decision** | Satisfactory | Not Yet Satisfactory | |
| **Assessor Signature** |  | **Date** |  |
| **Feedback to student** | | | |
|  | | | |
| **Feedback from student** | | | |
|  | | | |
| **Student signature** |  | **Date** |  |

Zip the root folder to compress the overall size of your submission, and ensure you submit your files to the appropriate assessment link.