|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification details** | | | |
| **Training Package Code and Title:** | **ICT - Information and Communications Technology (Release 7.0)** | | |
| **Qualification National Code and Title:** | ICT40120 Certificate IV in Information Technology (Gaming Development) | **State code:** | BFF9 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Assessment Title** | **AT04 Knowledge Questions (Part 3)** | | |
| **Unit National Code & Title** | ICTGAM423 Apply artificial intelligence in game development (Release 1) | | |
| ICTGAM427 Use 3-D software interface and toolsets (Release 1) | | |
| ICTGAM430 Design interactive media (Release 1) | | |
| **Date Due** | **Session 17** | **Date Received** | **16-05-2022** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name** | **Xihao Chen** | **Student ID** | 30053752 |
| **Student Declaration** | I declare that the evidence submitted is my own work:  ………………………………………….. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessor Name** |  | | | |
| **Assessment Decision** | Satisfactory | | Not Yet Satisfactory | |
| **Assessor Signature** |  | | **Date** |  |
| **Is student eligible for reassessment (Re-sit)?** | No | Yes | **Reassessment Date:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Feedback to student** | | | |
| *Via Blackboard (LMS) – Please check [Grade] section.* | | | |
| **Feedback from student** | | | |
| *Via Blackboard (LMS) – Please use [Comment] section during submission.* | | | |
| **Student signature** |  | **Date** |  |

|  |
| --- |
| **Assessment Instructions** |

|  |  |
| --- | --- |
| **TO THE ASSESSOR** | |
| Type of Assessment | *Written Questions* |
| Duration of Assessment | *5 sessions (session 13 – session 17)* |
| Location of Assessment | *Classroom (computer lab), at home* |
| Conditions | *Skills in this unit must be demonstrated in a workplace or simulated environment where the conditions are typical of those in a working environment in this industry.*  *This includes access to:*   * *required hardware and software and peripheral devices* * *human-computer hardware interface devices* * *a range of event-handling systems* * *the internet* * *application libraries* * *widgets* * *graphical user interface software and libraries* * *games engine* * *a range of browsers and digital devices* * *client requirements documentation* * *file storage*   *Learners are required to complete the required tasks and submit the required evidence electronically via Blackboard.* |
| Elements and Criteria | As detailed in the assessment plan.  You are required to make sure that all students meet the elements, performance criteria and foundation skill items as outlined in the provided checklist. |

|  |  |
| --- | --- |
| **TO THE STUDENT** | |
| Purpose of Assessment | You are required to show your understanding of:  *ICTGAM430 - Design interactive media*   * programming techniques applicable to interactive media * specific terminology used by game developers * hardware and software requirements in a human-computer interaction environment * a range of human-computer interaction devices and their application * widgets and their usage * operations in ‘heads-up’ display environments * technical constraints imposed by hardware and software on design and development * file storage and organisational procedures that may be used in designing interactive media   You are required to meet the elements, performance criteria and foundation skill items as outlined. |
| Allowable Materials | Blackboard (Topic by topic) will include the following: Weekly Readings, Class notes, and Weekly Activities.  Internet resources must be recorded as references for the assessment. |
| Required Resources | *Computer with:*   * *Internet Access* * *Word processing software* * *Access to Learning Management System (LMS)* |
| Reasonable Adjustment | In some circumstances, adjustments to assessments may be made for you. If you require support for literacy and numeracy issues; support for hearing, sight or mobility issues; change to assessment times/venues; use of special or adaptive technology; considerations relating to age, gender and cultural beliefs; format of assessment materials; or presence of a scribe you need to inform your lecturer. |
| Assessment Submission | *All activities must be attempted.*  *Use of research tools and peers in formulating answers are acceptable – but work submitted must be your own work and must not be plagiarised.*  *Final files and documentation are to be uploaded to the appropriate area in the Blackboard course created for this unit.*  *If you are marked as NYS (Not Yet Satisfactory) on your first attempt, you will be provided with another opportunity to re-attempt the assessment.* |
| Project contents | This project consists of the following tasks:   * Answer all of the questions for each section |

**Instructions**

To the best of your ability, answer each of the following questions in full. Ensure that you have attempted to answer all questions before submitting.

**Part 3 – ICTGAM430**

|  |  |  |
| --- | --- | --- |
| **Question 1 – Describe the following programming elements (in the context of the C# language) and identify techniques for using them in producing interactive media.** | | |
| ***Class*** | | A class is a container for data with executable functionality; a "blueprint" for constructing objects. |
| ***Variable*** | | A variable is a single instance of a container that stores values. |
| ***Method*** | | A method is a structured block of code that contains statements which provide context and execute specific functionalities of a program. They may take parameters and other values from a class. |
| ***Delegate*** | | A delegate represents references to methods with a specific parameter list and return type. They can be used to pass methods as arguments for other methods to use. |
| ***Event*** | | An event is a variant of a delegate that can only be invoked within its class or structure it is defined within. Classes/structures that subscribe to the event will have their methods called when raised. |
| **Question 2 – Define the following terms as they relate to game development.** | | |
| ***AAA*** | | AAA refers to a category of video games that are developed by corporate game studios/major publishers supported by large teams and resources. AAA developers can create games with short development cycles. |
| ***Asset*** | | An asset is any intangible/tangible component used in the creation of a video game. |
| ***Collision*** | | A collision is when two objects intersect with each other. They usually invoke events and interactions when triggered in-game. |
| ***HCI*** | | Human computer interaction in a video game context refers to how humans interact with a game. HCI elements may include hardware, software, platforms, etc. |
| ***Indie*** | | Indie is a shorthand term for “independent developers”, who work on their own or in small teams without the financial/technical assistance of a publisher. They have a small budget with limited resources and time, but have more freedom with their projects. |
| ***Platform*** | | A platform is a game development context is a foundation used as a base for developing the processes and technologies of a game. It usually refers to the hardware/software platforms of which video games run on. |
| ***UI*** | | A user interface is an interactive space that serves as a visual interaction point with computers. |
| **Question 3 – Explain the hardware and software requirements for a gaming related human-computer interaction environment of your choice.** | | |
| The Oculus Quest 2 is a VR headset with integrated controllers that can run as a standalone unit. It also provides desktop PC support when connected via USB or Wi-Fi with the following minimum requirements:   * Intel i5-4590/AMD Ryzen 5 1500X or greater * NVIDIA GeForce GTX 1060 Desktop, 6 GB or greater * AMD 400 Series or greater * 8GB or more RAM * Windows 10 Operating System * 1x USB Port * Oculus PC app with correct driver software to install/use Oculus Link library.   Many VR sets that include desktop support will require at least a mid-tier PC setup and their related software/drivers to run smoothly. | | |
| **Question 4 – Identify at least three (3) different human-computer interaction devices and explain how they are used.** | | |
| ***1*** | VR headset – A head mounted device that produces a simulated experience outside the real world, projecting images, sounds and tactile sensations in a virtual environment. A VR headset has applications for leisurely and industry uses, such as astronauts training to spacewalk. | |
| ***2*** | Gamepad Controller – A handheld device with various buttons and triggers that execute functionality within a game. Examples include the PlayStation and XBOX controllers for their respective console devices. | |
| ***3*** | Steering Wheel – A peripheral device that simulates a steering wheel found on cars, used in racing games and vehicle simulators for racing drivers. It may include buttons and paddles to imitate the inputs found in a real car. | |
| **Question 5 – Explain what a widget is and describe how they are commonly used in digital applications and video games.** | | |
| A widget is an interactive component of an application with its own graphical user interface (GUI) that displays information and feedback. It represents a specific user-computer interaction (e.g. accessing weather information). Widgets may include any layout of icons, toolbars, buttons, input prompts, menus, windows, etc to convey information.  Widgets may be used in video games as an interactive graphical user interface/companion application via an overlay over the main game screen. An example would be a telemetry widget for racing simulation games, display raw numerical data of a car’s speed, tyre wear, temperatures, etc. | | |
| **Question 6 – Describe the functionality of a ‘heads-up display’ (HUD) in interactive video games.** | | |
| A heads-up display (HUD) is a graphical user interface that displays diegetic/non-diegetic information to the player (Such as health, game map, menus, etc). The HUD includes a mixture of non-interactable and interactable elements that are structured in a way that logically directs the player’s interactions within the game.   * Diegetic UI information exists in the physical game world and exists in the game narrative/lore (Characters are aware), e.g. the Pip Boy from the Fallout game series where the character interacts with the device. * Non-diegetic UI information does not exist in the physical game world or the game narrative/lore (Characters are unaware), e.g. the kill feed popup in multiplayer FPS games. * Spacial UI information exists in the game world, but does not exist in the game narrative/lore (Characters are unaware), e.g. arrow markers in the Need for Speed game series that direct the player to turn. * Meta UI information does not exist in the game world, but exists in the game narrative/lore (Characters are aware), e.g. the stylised speedometer in racing games – The stylised version is a representation of the actual speedometer in a car. | | |
| **Question 7.a – Describe an example of a technical constraint that the hardware used for production can place on the design and development of a video game.** | | |
| A hardware technical constraint that impacts video game design/development is the type of hardware platform that a video game will be played on. Video games that are not designed for cross-platform usage may utilise all features of its target hardware platform (keybinds/mapping and processing power). In contrast, cross-platform video games will need to compromise between functionality and graphical fidelity to ensure compatibility between hardware platforms. | | |
| **Question 7.b – Describe an example of a technical constraint that the software used for production can place on the design and development of a video game.** | | |
| A software technical constraint that impacts video game design/development is the choice of game engine. Some game engines are optimised for 2D games, whereas others are more suitable for 3D games. Different programming languages between game engines may impact the development potential of a video game, as well as the capabilities of engine components. For example, Unity offers cross-platform support and uses the C# language while Unreal Engine 4 uses C++ and has open-source code for full customisability. | | |
| **Question 8 – Explain the organisational standards for version control that will usually apply to file storage procedures when designing interactive media in a professional context.** | | |
| Version control is a form of file management for multiple versions of a document, keeping a record of its creation, development and changes over time. Each version of a file has a unique identifier to distinguish between draft, working and final copies of a document.  For file storage procedures, version control benefits from proper file naming conventions to avoid ambiguity between files. An example would be the file name in camel case, status type and version number such as GameSplashScreen\_Final\_V3.2. Minor revisions would increment the version number by a decimal amount, while major revisions increment the whole number.  The changes would be reflected on a version control table which tracks the history of a file, and a description of the changes. Version control software such as SourceTree and Github automates the table process when it stores versions of files in a repository.  A software repository is a central place to store files that users can pull from when necessary. Repositories allow the “branching” of file versions to allow parallel instances, which is important for collaboration purposes as it minimises file conflicts from the main version of the file.  An example of a repository workflow process is gitflow, which includes the following branches:   * A “main” branch consisting of the final versions of project files, also containing an abridged version of the project’s history. * A “develop” branch that serves as an integration branch for features, also containing the entire history of the project. * A “feature” branch with working files that users can freely add features to, merging back to the “develop” branch with no direct interaction with the “main” branch. * A “release” branch that forks off of “develop”, starting the next release cycle. The “release” branch is merged with “main” when the project is ready for release, then merged back to “develop” for future release work. * A “hotfix” branch that forks directly off the “main” branch, used to quickly patch project releases. Once the patch is complete, the “hotfix” branch is merged into both “main” and “develop” to prevent interruption to the rest of the workflow.   Other organisational standards that apply to file storage with version control include:   * Making final versions of files “read-only” to avoid accidental overwriting. * Retaining drafts of documents if required to keep a record of their context. | | |