|  |  |  |  |
| --- | --- | --- | --- |
| **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 1)** | | |
| **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:*   * *the internet* * *research tools* * *required hardware, software and its component* * *path-finding libraries* * *game development testing tools* * *development tools to implement AI strategies* * *game design specifications and documentation*   *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:  *ICTGAM423 - Apply artificial intelligence in game development*   * basic path-finding algorithms implications on game development * major AI terms used in the game industry * the development process of creating AI strategies in non-player characters (NPCs) in a game * applying object-oriented programming practices * overall architecture of a game engine * game development testing processes * industry standards applicable to game development   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 1 – ICTGAM423**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Question 1 – Describe the order of each node in the first complete paths from Lectra City to New Vegas returned using the depth-first search, breadth-first search, and Dijkstra’s path-finding algorithms.** | | | | | | |
|  | | | | | | |
| **DFS:** | | | Lectra City > City 17 > Dunwall > Rapture > Racoon City > New Vegas | | | |
| **BFS:** | | | Lectra City > City 17 > Los Santos > Night City > Dunwall > Stilwater > Arkham City > Rapture > Racoon City > Verdansk > New Vegas | | | |
| **Dijkstra’s:** | | | Lectra City > Los Santos > Stillwater > Racoon City > New Vegas | | | |
| **Question 2 – Provide definitions for the following terminology as it relates to artificial-intelligence in game development.** | | | | | | |
| ***Agent*** | | | | | An agent is an entity that perceives its environment and makes autonomous decisions to achieve a specific goal. An agent perceives its own actions, but not always the consequences of the actions. | |
| ***NPC*** | | | | | A non-player character (NPC) in a video game context is an AI controlled character that serves an in-game purpose (plot, assisting the player or tied to a game mechanic). | |
| ***Navigation Mesh*** | | | | | A navigation mesh (NavMesh) is an in-game framework that defines which surfaces in the world are walkable for AI. It allows agents to find a traversable path from A to B in the game world. | |
| ***Path-finding*** | | | | | Path-finding is a method used in game development to find and traverse the shortest/most efficient route through the in-game world via algorithms, such as A\*. | |
| ***Decision-making*** | | | | | Decision-making in a game development context is the process of an agent selecting an appropriate outcome based on the information it senses through the in-game environment. | |
| **Question 3 – Identify three (3) different development patterns that can be used to control the decision-making of an AI NPC in a video game.** | | | | | | |
| ***1*** | | Finite State Machine | | | | |
| ***2*** | | Behaviour Tree | | | | |
| ***3*** | | Utility Based System | | | | |
| **Question 4 – Describe how the following object-oriented programming principles are applied in practice.** | | | | | | |
| ***Abstraction*** | | | | Abstraction refers to hiding the specific implementation details from the end user, only showing the function but not the process. It allows the end-user to implement additional logic on top of the abstracted logic. For example, a user that presses the = sign on an simple calculator app will receive a numerical answer and not the list of arithmetic operations. | | |
| ***Encapsulation*** | | | | Encapsulation refers to the hiding of values and data states within objects and classes, preventing direct access from non-programmers. By preventing direct access, it promotes the use of public interfaces (methods of an object) to access these values. As such, encapsulation serves to ensure functionality is defined in a single place, as well as preventing unexpected data changes from external code execution in different parts the program. | | |
| ***Inheritance*** | | | | Inheritance refers to the shared use of logic between classes, which provides the child class with the attributes and logic from the parent class. It further extends a program’s functionality, building upon other classes and allowing sections of code to be reused. | | |
| ***Polymorphism*** | | | | Polymorphism allows a child class to share the information/behaviour of its parent class while retaining its original functions. This allows for accessing different objects from the same interface. Polymorphism provides the ability of a single variable to store multiple data types (float, int, etc). | | |
| **Question 5 – Describe the role of each of the following game engine components in relation to the overall architecture of a game engine.** | | | | | | |
| ***Rendering engine*** | | | | | | A rendering engine generates 23/3D animated graphics in the game engine using a rendering application programming interface (API), such as Direct3D/OpenGL. It handles the calculation of lighting and occlusion culling of assets, as well as scene management. |
| ***Physics engine*** | | | | | | A physics engine provides an accurate simulation of physical systems in the game engine, e.g. movement of rigid bodies, fluid dynamics, cloth movement, etc. Gravity, collisions, speed of objects and other physical forces are all handled by the physics engine. |
| ***Audio engine*** | | | | | | An audio engine controls the processing and output of sound effects in a game engine using a combination of algorithms/audio libraries. It loads/unloads audio files, as well as simulating audio effects such as spacial audio (e.g. Steam Audio’s binaural sound module), reverberation and occlusion. |
| ***Artificial intelligence engine*** | | | | | | An artificial intelligence (AI) engine handles the pathfinding, decision-making and simulated intelligence of entities in a game engine. The implementation is done via pathfinding scripts, agents and navigation meshes/behaviour trees. Machine learning modules may also feature in an AI engine. |
| **Question 6 – Briefly describe each of the following testing processes as they apply to game development.** | | | | | | |
| ***Functionality testing*** | | | | | | Functionality testing involves quality assurance testers identifying general problems with the game or user interface while playtesting the game. Observations may include asset integrity, audio/video sync, UI scalability, graphical integrity, etc. |
| ***Compatibility testing*** | | | | | | Compatibility testing involves quality assurance testers highlighting compatibility issues across different types of hardware. The game is run across a variety of hardware to observer if it will run the same on each platform regardless of the hardware and software combination. |
| ***Play testing*** | | | | | | Play testing involves quality assurance testers observing for errors while playing the game as it’s intended in the gold master build. |
| **Question 7 – Identify and describe at least two (2) industry-standards that apply to the development of AI in video games.** | | | | | | |
| ***1*** | The A\* algorithm is an industry-standard pathfinding algorithm which is an extension of Dijkstra’s algorithm. It uses an additional heuristic function (F cost) that improves its computation speed compared to Dijkstra’s algorithm, making it ideal for video games for optimisation purposes. The F cost estimates the cost to move from a node to the destination. | | | | | |
| ***2*** | The Unity engine’s inbuilt navigation mesh (NavMesh) system forms the basis for NPC movement, allowing for and displaying pathfinding technique implementations in a visual manner. A NavMesh assists with separating the paths between moving characters, as well as movement and other scenarios involving pathfinding. | | | | | |