

Final Evaluation: 40%

|  |  |
| --- | --- |
| **Course Identification** | |
|  |  |
| Name of program – Code: | COMPUTER SCIENCE TECHNOLOGY – VIDEO GAME PROGRAMMING – 420.BX |
| Course title: | **INFORMATION SYSTEM AND PROJECT METHODOLOGY II** |
| Course number: | 420-J12-AS |
| Group: | 00160 |
| Teacher’s name: | ADIN ASHBY |
| Duration: | 3 periods (150 minutes) |
| Semester: | Winter 2025 |
| **Student Identification** | |
| Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Student number: \_\_\_\_\_\_\_\_\_\_\_\_\_\_  Date: April 16, 2025 Result: \_\_\_\_\_\_\_\_\_\_\_\_\_\_  I declare that this is an original work, and that I credited all content sources of which I am not the author (online and printed, images, graphics, films, etc.), in the required quotation and citation style for this work. | |
| **Standard of the Evaluated Competency(ies)** | |
|  | |

**Statement of the evaluated competency(ies) – Code(s)**

1. Develop data exchange services -00SV
2. Collaborate on the design of applications –00SY

|  |
| --- |
| **Instructions** |
|  |
| The use of AI and internet are not allowed during the exam. However, students are allowed to use their class notes only.No break is allowed during this exam. Students are not allowed to exit the examination room before half of the allotted time has passed. Once a student has exited the classroom, he/she may not re-enter (IPEL – Article 5.12.4).The teacher will not answer questions during the exam.Students must remain silent during the exam.It is the teacher’s responsibility to identify language errors. If such errors are found, teachers may apply a penalty of up to 20% of the grade (IPEL – Article 5.7).Plagiarism, attempts at plagiarism or complicity in plagiarism during a summative evaluation results in a mark of zero (0). In the case of recidivism, in the same course or in another course, the student will be given a grade of '0' for the course in question.(IPEL – Article 5.16)  * Students are responsible to make sure they have their login credentials to submit their solution. * Students are responsible to make sure they have submitted the correct version of their exam. |
| |  | | --- | | **Mark Breakdown** | |  |   This evaluation is on 100 points, distributed as follows:   |  |  |  | | --- | --- | --- | | **Question 1** | Implement an Audio Management System | For a total of 50 points | | **Question 2** | Implement an Enemy Spawning System | For a total of 50 points | | **TOTAL: 100 POINTS** | | |   **Instructions**  **Question 1**  Implement an Audio Management System  **Problem Statement**  **Problem Statement**  You are developing an **Audio Management System** for a video game. The system should manage background music, sound effects, and ambient audio in a modular and efficient way. Your solution must reduce resource consumption, and offer a clean API for developers to use.  **Objective**  Use **at least two** of the following design patterns:   * **Flyweight** – Reuse sound assets across scenes. * **Facade** – Provide a simplified interface for managing audio. * **Proxy** – Lazy load audio files or stream large soundtracks only when needed. * **Bridge** – Separate audio abstraction (e.g., volume, channel, effects) from implementation (e.g., MP3, WAV player).   **Requirements**   * Implement at least **two design patterns** from the list above. * Write a **short report** (max one page) explaining your design and pattern benefits. * Apply **OOP** and **SOLID** principles. * Ensure modular, reusable, and readable code.   **Question 2**  Implement an Enemy Spawning System  **Problem Statement**  Design an **Enemy Spawning System** that creates different types of enemies (e.g., melee, ranged, boss) during gameplay. The system must support dynamic instantiation and customization of enemies and allow for efficient memory and logic management.  **Objective**  Use **at least two** of the following design patterns:   * **Prototype** – Clone enemy templates at runtime. * **Factory Method** – Centralize logic for creating different enemy types. * **Builder** – Build enemies step-by-step with optional features (armor, behavior, loot). * **Bridge** – Decouple enemy roles (e.g., attack strategy) from their implementations (e.g., animation or physics engines).   **Requirements**   * Implement at least **two design patterns** from the list above. * Write a **short report** (max one page) explaining your pattern selection and architecture. * Code must follow **OOP** and **SOLID** design. * Ensure clean class structure, naming, and abstraction.  |  | | --- | | Evaluation grid | | **Questions 1 (50 points)**   |  |  | | --- | --- | | **Element of competency:** Develop data exchange services – 00SV | | | **Performance criteria** | **weight** | | 1.1 Accurate analysis of design documents | **/5** | | 1.2 Proper identification of the tasks to be carried out | **/5** | | **Element of competency: Collaborate on the design of applications – 00SY** | | | 2.1 Appropriateness of the recommendations regarding the choice of software architecture | **/10** | | 3.4 Object-oriented modelling compliant with principles of encapsulation, inheritance, composition and polymorphism | **/10** | | 3.5 Proper choice or production of algorithms | **/15** | | 3.6 Compliance with nomenclature rules | **/2.5** | | 4.5 Compliance with application development standards, methods and best practices | **/2.5** | | |
|  |
| **Questions 2 (50 points)** |

|  |  |
| --- | --- |
| **Element of competency:** Develop data exchange services – 00SV | |
| **Performance criteria** | **weight** |
| 1.1 Accurate analysis of design documents | **/5** |
| 1.2 Proper identification of the tasks to be carried out | **/5** |
| **Element of competency: Collaborate on the design of applications – 00SY** | |
| 2.1 Appropriateness of the recommendations regarding the choice of software architecture | **/10** |
| 3.4 Object-oriented modelling compliant with principles of encapsulation, inheritance, composition and polymorphism | **/10** |
| 3.5 Proper choice or production of algorithms | **/15** |
| 3.6 Compliance with nomenclature rules | **/2.5** |
| 4.5 Compliance with application development standards, methods and best practices | **/2.5** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CORRECTION GRID FOR LANGUAGE**   |  |  |  |  | | --- | --- | --- | --- | | Clear Communication | Clear Communication, **most of the time** | Vague Communication | Unclear Communication | | - 0 | - 0,5 | - 1,5 | - 2 | | (Word Choice)  Use of precise and rich vocabulary | (Word Choice)  Use of precise vocabulary | (Word Choice)  Use of imprecise vocabulary | (Word Choice)  Use of inappropriate vocabulary | | - 0 | - 0,5 | - 1,5 | - 2 | | (Format/Type of work)  Respect of norms | (Format/Type of work)  Respect of **most of the** norms | (Format/Type of work)  Non-respect of the norms | (Format/Type of work)  Inappropriate in relation to the required norms | | - 0 | - 0,5 | - 1,5 | - 2 | | (Linguistic Code)  (≤2 mistakes / page) | (Linguistic Code)  (3-7 mistakes/page) | (Linguistic Code)  (8-10 mistakes/ page) | (Linguistic Code)  (>10 mistakes/  page) | | - 0 | - 0,5 - 2.5 | - 2.5 - 3.5 | - 4 | |