

Career & Technical Education | Information Technology

Game Design

Subject Code: 145090

Outcome & Competency Descriptions

Course Description:

This course will prepare students to design and program games using commercial and opensource programs and applications. Students will learn industry standard programming language constructs to write programs that integrate classes, class methods, and class instances. Students will learn input method handling, animation, collision detection, game physics and basic artificial intelligence.

Strand 1. Business Operations / 21st Century Skills

Learners apply principles of economics, business management, marketing and employability in an entrepreneur, manager, and employee role to the leadership, planning, developing, and analyzing of business enterprises related to the career field.

Outcome: 1.2. Leadership and Communications

Process, maintain, evaluate, and disseminate information in a business. Develop leadership and team building to promote collaboration.

Competencies

- 1.2.7. Use problem-solving and consensus-building techniques to draw conclusions and determine next steps.
- 1.2.13. Identify stakeholders and solicit their opinions.
- 1.2.14. Use motivational strategies to accomplish goals.

Outcome: 1.3. Business Ethics and Law

Analyze how professional, ethical, and legal behavior contributes to continuous improvement in organizational performance and regulatory compliance.

Competencies

- 1.3.4. Identify how federal and state consumer protection laws affect products and services.
- 1.3.7. Identify the labor laws that affect employment and the consequences of noncompliance for both employee and employer (e.g., harassment, labor, employment, employment interview, testing, minor labor laws, Americans with Disabilities Act, Fair Labor Standards Acts, Equal Employment Opportunity Commission [EEOC]).

Outcome: 1.7. Entrepreneurship / Entrepreneurs

Analyze the environment in which a business operates, and the economic factors and opportunities associated with self-employment.

Competencies

- 1.7.13. Protect intellectual property and knowledge (e.g., copyright, patent, trademark, trade secrets, processes).

Outcome: 1.8. Operations Management

Plan, organize and monitor an organization or department to maximize contributions to organizational goals and objectives.

Competencies

- 1.8.2. Select and organize resources to develop a product or a service.

Strand 2. IT Fundamentals

Learners apply fundamental principles of IT, including the history of IT and its impact on society, common industry terms, systems theory, information storage and retrieval, database management, and computer hardware, software, and peripheral device configuration and installation. This base of knowledge and skills may be applied across the career field.

Outcome: 2.10. Equipment

Select, prepare, operate, and maintain equipment.

Competencies

2.10.4. Identify software application requirements.

Outcome: 2.12. Performance Tests and Acceptance

Develop performance tests and acceptance plans.

Competencies

- 2.12.1. Create a written procedure agreed by the stakeholders and project team for determining the acceptability of the project deliverables.
- 2.12.2. Develop a test system that accurately mimics external interfaces.
- 2.12.3. Develop test cases that are realistic, compare with expected performance, and include targeted platforms and device types.
- 2.12.4. Develop, perform, and document usability and testing integration.
- 2.12.5. Make corrections indicated by test results.
- 2.12.6. Seek stakeholder acceptance upon successful completion of the test plan.

Outcome: 2.14. Artificial Intelligence

Understand and apply prescribed methods of using Artificial Intelligence.

Competencies

- 2.14.1. Describe how machine learning and neural networks operate differently than standard decision trees.
- 2.14.2. Analyze how artificial intelligence technology impacts society and the ethical implications of its usage.
- 2.14.3. Write and revise a prompt to generate the desired response from an AI.
- 2.14.4. Evaluate the result of an AI query on a variety of parameters (e.g. validity, relevance, authenticity, potential bias and hallucinations).

Outcome: 2.15. UX/UI Design

Develop basic skills and knowledge of the UX/UI Design Process.

Competencies

- 2.15.1. Understand the UX/UI design process (e.g. vision, journey mapping, wireframing, prototyping, strategizing) for the targeted platform (e.g. graphics, applications, programming).
- 2.15.3. Design user tasks and evaluate results (e.g. use-case scenarios, tabletop exercises, wireframe testing).
- 2.15.6. Design interface elements and experiences that connect concepts with the real world (i.e. Skeuomorphic Design).
- 2.15.7. Implement UI patterns and libraries, such as navigation elements and icons.
- 2.15.8. Draft, design, and utilize design prototypes (low-fidelity, high-fidelity) to guide the design process.
- 2.15.9. Design or select appropriate icons for specific user interaction elements.
- 2.15.10. Understand how the use of appropriate iconography impacts user experience.

Strand 5 Programming & Software Systems

Learners apply principles of computer programming and software development to develop code; build, test, and debug programs; create finished products; and plan, analyze, design, develop, implement, and support software applications.

Outcome: 5.2. Computational and String Operations

Develop code that performs computational and string operations.

Competencies

- 5.2.1. Compare and contrast primitive types of numeric and nonnumeric data (e.g., integers, floats, Boolean, strings).
- 5.2.2. Identify the scope of data (e.g., global versus local, variables, constants, arrays).
- 5.2.3. Write code that uses arithmetic operations.
- 5.2.4. Write code that applies string operations (e.g., concatenation, pattern matching, substring).

Outcome: 5.3. Logical Operations and Control Structures

Develop code that uses logical operations and control structures.

Competencies

- 5.3.1. Explain Boolean logic.
- 5.3.2. Solve a truth table.
- 5.3.3. Write code that uses logical operators (e.g., and, or, not).
- 5.3.4. Write code that uses relational operators and compound conditions.
- 5.3.5. Write code that uses conditional control structures (e.g., if, if-then-else).
- 5.3.6. Write code that uses repetition control structures (e.g., while, for).
- 5.3.7. Write code that uses selection control structures (e.g., case, switch).
- 5.3.8. Write code that uses nested structures and recursion.
- 5.3.9. Write code that creates and calls functions.
- 5.3.10. Code error handling techniques.
- 5.3.11. Write code to access data repositories.
- 5.3.12. Write code to create classes, objects, and methods.

Outcome: 5.4. Integrated Development Environment

Build and test a program using an integrated development environment (IDE).

Competencies

- 5.4.1. Configure options, preferences, and tools.
- 5.4.2. Write and edit code in the integrated development environment (IDE).
- 5.4.3. Compile or interpret a working program.
- 5.4.4. Define test cases.
- 5.4.5. Test the program using defined test cases.
- 5.4.6. Correct syntax and runtime errors.
- 5.4.7. Debug logic errors.

Outcome: 5.5. Programming Conventions

Develop programs using applications security best practices according to information security policies (e.g., cross-site scripting, Structured Query Language [SQL] injection attack, bounds_-checking).

Competencies

- 5.5.1. Develop programs using data validation techniques.
- 5.5.2. Develop programs that use reuse libraries.
- 5.5.3. Develop programs using operating system calls.
- 5.5.4. Develop programs that call other programs.
- 5.5.5. Use appropriate naming conventions and apply comments.
- 5.5.6. Format output (e.g., desktop, mobile, enterprise, reports, data files).
- 5.5.7. Read inputs (e.g., user input, data file, sensors, databases, APIs).

Outcome: 5.6. Software Development Lifecycle

Apply the software development lifecycle (SDLC).

Competencies

- 5.6.1. Determine requirements specification documentation.
- 5.6.2. Identify constraints and system processing requirements.
- 5.6.3. Develop and adhere to timelines.
- 5.6.4. Identify a programming language, framework, and an integrated development environment (IDE).
- 5.6.5. Identify input and output (I/O) requirements.
- 5.6.6. Design system inputs, outputs, and processes.
- 5.6.7. Document a design using the appropriate tools (e.g., program flowchart, dataflow diagrams, Unified Modeling Language [UML]).
- 5.6.8. Create documentation (e.g., implementation plan, contingency plan, data dictionary, user help).
- 5.6.9. Review the design (e.g., peer walkthrough).
- 5.6.10. Present the system design to stakeholders.
- 5.6.11. Develop the application.
- 5.6.12. Compare and contrast software methodologies (e.g., agile, waterfall).
- 5.6.13. Perform code reviews (e.g., peer walkthrough, static analysis).
- 5.6.14. Ensure code quality by testing and debugging the application (e.g., system testing, user acceptance testing).
- 5.6.15. Train stakeholders.
- 5.6.16. Deploy the application.
- 5.6.17. Collect application feedback and maintain the application.

Outcome: 5.7. Configuration Management

Describe configuration management activities.

Competencies

- 5.7.1. Explain version management and interface control.
- 5.7.2. Explain baseline and software lifecycle phases.
- 5.7.3. Analyze the impact of changes.

Strand 7. Digital Media

Learners apply principles of digital media to produce interactive media; develop and produce multimedia applications; integrate typography into media; create 3D models and 2D and 3D animation; and create digital video, audio, and photographs.

Outcome: 7.2. Multimedia Tools

Develop navigational structures, scripts, storyboards, and flowcharts for multimedia applications.

Competencies

- 7.2.1. Develop navigational structures, wireframes, and flowcharts for multimedia applications.
- 7.2.2. Construct and place navigational units.
- 7.2.3. Build in interactive elements.
- 7.2.4. Determine uses and needs for site maps, multimedia scripts, storyboards, and flowcharts.
- 7.2.5. Make preliminary sketches showing placement of images and text on screen.
- 7.2.6. Place buttons and navigational graphics.
- 7.2.7. Select colors based on color theory and psychology.
- 7.2.8. Describe music, video, and special effects to be used.
- 7.2.9. Provide a sample layout to stakeholders for review.
- 7.2.10. Select and create visual design elements appropriate for the intended audience and use.
- 7.2.11. Develop client personas and narratives for intended project outcomes.

Outcome: 7.4. Graphics

Construct and manipulate digital graphics.

Competencies

- 7.4.1. Select and manipulate color profiles (e.g., Red Green Blue [RGB], Cyan Magenta Yellow Key [CMYK], Pantone) for appropriate uses.
- 7.4.2. Select color, shape, size, and texture of objects.
- 7.4.3. Create or acquire graphics.
- 7.4.4. Manipulate and layer objects.
- 7.4.5. Differentiate between vector and raster images.
- 7.4.6. Select graphic software applications based on budget, technical capabilities and hardware specifications to meet intended project outcome.
- 7.4.7. Optimize and export graphics files for intended use.
- 7.4.8. Manipulate graphic objects.
- 7.4.9. Compress and decompress graphic files.
- 7.4.10. Describe and select color profiles (e.g., Red Green Blue [RGB], Cyan Magenta Yellow Key [CMYK], Pantone).

Strand 9. Cybersecurity

Learners apply principles of Cybersecurity to secure and defend information technology systems, selection and implementation of methods and tools to secure physical and digital assets, manage threats, deploy countermeasures, and establish strategies to protect business information using risk and incident management.

Outcome: 9.3. Application Development Security

Develop and maintain application security.

Competencies

- 9.3.1. Identify application vulnerabilities (e.g., Cross-site scripting, SQL injection, LDAP injection, XML injection, Directory traversal/command injection, Buffer overflow, Integer overflow, Zero-day, Cookies and attachments, Locally Shared Objects (LSOs), Flash cookies, Malicious add-ons, Session hijacking, Header manipulation, Arbitrary code execution/remote code execution).
- 9.3.3. Implement secure coding concepts (e.g., Error and exception handling, Input validation, Cross-site scripting prevention, Cross-site Request Forgery, (XSRF) prevention, OWASP).
- 9.3.4. Implement secure application configuration (e.g., Application hardening, Application patch management).