**CURRICULAR PROPOSAL COVER SHEET**

**Course Proposal (Non-General Education)**

(Note: This form must be printed on *blue* paper when submitted for approval.)

|  |
| --- |
| **Department:** Mathematics and Computer ScienceContact Name: Dan Bennett Contact Phone: 732-1177 Email Address: dbennett@edinboro.edu |
| **[x] Revision of an Existing Course**Existing Course Prefix & Number: CSCI 340Existing Course Title: Game ProgrammingIf changed:  New Course Prefix & Number:  New Course Title:  New Title Abbreviation (20 character limit): | Attach both existing and revised\* versions and an updated Bibliography\* If no old version exists please explain.Revision of (check all that apply):[x] Catalog Course Description [x] Course Outline[x]  Prerequisite(s) [ ]  Credit Hours [x]  Course Objectives [x]  Assessment[x] Bibliography (updated Bibliography required)\* PLEASE NOTE: A course outline and bibliography are *not* required for courses without specific content. |
| **[ ] New Course**Course Prefix & Number:Course Title:Title Abbreviation (20 character limit):Credit Hours: | Attach Catalog Course Description, Course Outline, Course Objectives and Assessments, and Bibliography\*Does this course replace an existing course? **[ ]**  No [ ]  Yes  If yes, specify course:\* PLEASE NOTE: Special topics courses must go through the curriculum process as a **New Course** |

Grading Scheme

 [x] Standard Letter Grades

 [ ] S/U

Has this course been previously approved for distance education and/or ITV delivery or are you submitting this course at this time for distance education and/or ITV delivery?

[x]  No [ ]  Yes

 *If yes, please check the delivery method (check all that apply)*

 [ ] 100% on-line web delivery

 [ ] 80-99% on-line web delivery

 [ ] ITV

1. Briefly explain the rationale for the course/revision including how it will improve the quality of the program:

*The changes to this course are to allow more in-depth study of game programming and are needed to allow for the Game Track’s reorganization and for possible accreditation.*

2. Course is (check all that apply):

 [x]  Requirement for major

 [x]  Elective for major

 [ ]  Requirement for minor

 [ ]  Designed for non‑major

3. Is this course required in or does it affect degree programs in any other department(s)?

 [x]  No [ ]  Yes

*If yes, list the department(s) and the degree program(s), and attach documentation of consultation with each department (e.g., e-mail from Department Chair):*

4. Does the course share subject matter with any other department(s)?

 [x]  No [ ] Yes

*If yes, list department(s) and attach documentation of consultation with each department (e.g., e-mail from Department Chair):*

5. Do you currently have qualified faculty to teach the course?

 [ ]  No [x]  Yes

6. Anticipated frequency of offering (i.e., number of sections per semester, year, etc.):

 *This course is currently offered every spring semester.*

7. Method of instruction (e.g., lecture, seminar, lab, studio):

 *Lecture*

8. Prerequisites:

 *C- or better in CSCI 330 and MATH 211.*

9. Are there needs, equipment, new library holdings or other resources involving budgetary considerations?

 [x]  No [ ]  Yes

 *If yes, list specific needs and cost estimates:*

10. Identify the Academic Level for the course.

 [ ]  Developmental/Remedial Course 050

 [ ]  Introductory/Foundation UG Course 110

 [ ]  Intermediate/Foundation UG Course 120

 [x]  Intermediate/Advanced UG Course 130

 [ ]  Advance UG Course 140

 [ ]  Introductory GR Course 150

 [ ]  Intermediate GR Course (open to UG) 220

 [ ]  Advanced GR Course (open to GR) 230

*These levels are based on PASSHE policy. See UWCC Web page for definitions.*

Dean’s Office Use Only: **Assigned CIP Code:**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Approval Process:**

Department: Department Vote Count: Yes\_20\_\_ No \_\_0\_\_ Abstain \_\_0\_\_

 Department Vote Date: 3/3/30 – 3/6/20

|  |  |  |
| --- | --- | --- |
|  | **SIGNATURE** | **DATE** |
| Department Chairperson |   | \_\_\_\_\_\_\_\_\_\_\_ |
| Dean |   | \_\_\_\_\_\_\_\_\_\_\_ |
| Graduate Dean (if appropriate) |   | \_\_\_\_\_\_\_\_\_\_\_ |
| Provost |   | \_\_\_\_\_\_\_\_\_\_\_ |
| Courses Subcommittee(if appropriate) |   | \_\_\_\_\_\_\_\_\_\_\_ |
| Gen. Ed. Subcommittee(if appropriate) |   | \_\_\_\_\_\_\_\_\_\_\_ |
| Program Subcommittee (if appropriate) |   | \_\_\_\_\_\_\_\_\_\_\_ |
| Curriculum Committee |   | \_\_\_\_\_\_\_\_\_\_\_ |
| Vice President/Provost |   | \_\_\_\_\_\_\_\_\_\_\_ |
| President |   | \_\_\_\_\_\_\_\_\_\_\_ |

FOR REGISTRAR'S OFFICE USE ONLY

**Master Course Data**

|  |  |
| --- | --- |
| Course Identifier: \_\_\_\_\_\_\_\_\_\_\_ | Course Title: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Credits: \_\_\_\_\_\_\_\_\_\_\_\_ | Level:\_\_\_\_\_\_\_\_\_ | Date Added:\_\_\_\_\_\_\_\_\_\_\_\_ |
| Load Hours: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | V‑Load: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Majors Only: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Permission Only: \_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Lab Course: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Prerequisite Required: \_\_\_\_\_\_\_\_\_\_\_\_ |
| Computer Title: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | New or Revised Course: \_\_\_\_\_\_\_\_\_\_\_\_ |
| Replaces this existing course:\_\_\_\_\_\_\_\_\_ | Last Action:\_\_\_\_\_\_\_\_\_\_ |
| New course can be repeated for what course(s): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

EDINBORO UNIVERSITY OF PENNSYLVANIA

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

GAME PROGRAMMING

CSCI 340 3 Credit Hours

Catalog Description:

 This course is designed to familiarize the student with the programming tasks involved in developing computer games using a game engine. Topics will include: 2D games, 3D games, sound, game objects, game worlds, characters, and basic AI. A game development environment will be employed by individual students and teams of students to implement working games. Students will employ an underlying programming language to extend the behavior of the game engine. Prerequisite: CSCI 330, MATH275. Must have earned a “C-” or better in all prerequisite course(s).

Course Outline:

1. Introduction to the Editor Interface
	1. Standard Tools
	2. Programming Language(s)
2. Engine Overview
	1. Sound Engine
	2. Physics Engine
	3. Rendering Engine
	4. Programming Extended Behavior
3. 2D Game Programming
	1. Sprites
	2. Animation
	3. Collision Detection
4. 3D Game Programming
	1. Animation
	2. Collision Detection
5. Game Worlds
	1. Terrain
	2. Foliage & trees
	3. Buildings
6. Basic Artificial Intelligence
	1. Path following
	2. Dialog trees
7. Game Objects
	1. Collectables
	2. Importing assets
	3. Materials
	4. Lights
	5. Particles
8. Game Characters
	1. Third Person Characters
	2. First Person Characters
	3. Non-player Characters
9. Game User Interface
	1. Heads Up Display
	2. Input Devices

Course Objectives:

|  |  |  |
| --- | --- | --- |
| **Objectives** | **Assessments** | **Assessment Tools** |
| 1. Students use a game engine to implement a 2D video game.
 | 1. Students will implement a 2D video game.
 | Exams, quizzes, homework, presentations, essays, class participation, or projects. |
| 1. Students use a game engine to implement a 3D video game.
 | 1. Students will implement a 3D video game.
 | Exams, quizzes, homework, presentations, essays, class participation, or projects. |
| 1. Students create objects for use in video games.
 | 1. Students will answer exam questions about game objects.
2. Students will implement various game objects in a video game.
 | Exams, quizzes, homework, presentations, essays, class participation, or projects. |
| 1. Students create characters for use in video games.
 | 1. Students will answer exam questions about game characters.
2. Students will implement various types of game characters in a video game.
 | Exams, quizzes, homework, presentations, essays, class participation, or projects. |
| 1. Students develop simple AI techniques that would be used in a video game.
 | 1. Students will answer exam questions about game AI.
2. Students will implement various basic game AI techniques in a video game.
 | Exams, quizzes, homework, presentations, essays, class participation, or projects. |
| 1. Students identify the components of a video game engine and their use.
 | 1. Students will answer exam questions about game design.
2. Students will implement some video games.
 | Exams, quizzes, homework, presentations, essays, class participation, or projects. |

**BIBLIOGRAPHY**

Carnall, Benjamin, Unreal Engine 4.X By Example, Packt Publishing, 2016

Godbold, Ashley, Mastering Unity 2D Game Development, 2nd ed., Packt Publishing, 2016

Moakley, Brian, Mike Berg and Sean Duffy, Unity Games by Tutorials: Make 4 Complete Unity Games from Scratch Using C#, Razeware LLC, 2016

Nystrom, Robert, Game Programming Patterns, Genever Benning, 2014

PV, Satheesh, Unreal Engine Essentials, Packt Publishing, 2016

Sapio, Francesco, Unity UI Cookbook, Packt Publishing, 2015

Sewell, Brenden, Blueprints Visual Scripting for Unreal Engine, Packt Publishing, 2015

Smith, Matt and Chico Queiroz, Unity 5.x Cookbook, Packt publishing, 2015

Sherif, William and Stephen Whittle, Unreal Engine 4 Scripting with C++ Cookbook, Packt Publishing, 2016

Tavakkoli, Alireza, Game Development and Simulation with Unreal Technology, CRC Press, 2015

Thorn, Alan, Unity 5.x By Example, Packt Publishing, 2016

EDINBORO UNIVERSITY OF PENNSYLVANIA

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

GAME PROGRAMMING

CSCI 340 3 semester hours

Catalog Description:

 This course is designed to familiarize the student with the programming tasks involved in developing computer games. Topics will include: advanced C++ programming, event-loop programming, artificial intelligence, static and dynamic 3D graphics programming. A game development environment will be employed by teams of students to implement a working game. Prerequisite: CSCI 330. Must have earned a “C” or better in all prerequisite course(s).

Course Outline:

I. Introduction

 A. Standard Tools

 1. Source Code Control Systems

 2. Executable Building Systems

 3. Linking With Multiple Libraries

 B. Object-Oriented Techniques

 C. Overview of Game Design

II. Graphical User Interfaces(GUI) and Event Driven Programming

 A. Introduction to Events

 B. Event Driven Main Loop

 C. GUI techniques

III. Static 3D Objects

 A. Wire Mesh Objects

 B. Texture

IV. Scene Graphs

 A. Hierarchical Scene Representation

 B. Building a Scene Graph

 C. Updating a Scene Graph

 D. Rendering a Scene Graph

V. Dynamic 3D Objects

 A. Physical Simulation

 B. Collision Detection

 C. Character Animation

VI. Sound

 A. Audio Format

 B. Audio Techniques

VII. Advance User Interaction

 A. Input Devices

 B. Picking

VIII. Overall Game Architecture

 A. Actors

 B. Messaging

 C. Integration

IX. Networking and Online games.

**BIBLIOGRAPHY**

Adams, Ernest & Andrew Rollings. Fundamentals of Game Design. Prentice Hall, 2006.

Astle, Dave. Beginning Opengl Game Programming. Course Technology, 2006.

Bateman, Chris & Richard Boon. 21st Century Game Design. Charles River Media, 2005.

Bates, Bob. Game Design, 2nd Edition. Course Technology PTR, 2004.

Buckland, Mat. Programming Game AI by Example. Wordware Publishing, 2005.

Cinnamon, Ian. Programming Video Games for the Evil Genius. McGraw-Hill Professional Publishing, 2008.

Crawford, Chris. Chris Crawford on Game Design. New Riders Publishing, 2003.

Dalmau, Daniel & Daniel Sanchez-Crespo. Core Techniques and Algorithms in Game Programming. New Riders Games, 2003.

Deloura, Mark. Best of Game Programming Gems. Charles River Media, 2008.

Dickheiser, Mike. Game Programming Gems 6. Charles River Media, 2006.

Dille, Flint. The Ultimate Guide to Video Game Writing and Design . Lone Eagle , 2008.

Finney, Kenneth. Advanced 3D Game Programming All In One, Second Edition. Course Technology, 2008.

Fullerton, Tracy, Chris Swain & Steven Hoffman. Game Design Workshop: Designing, Prototyping and Playtesting Games. CMP Books, 2004.

Granberg, Carl. Programming an Rts Game With Direct3d. Charles River Media, 2006.

Grootjans, Riemer. XNA 2.0 Game Programming Recipes: A Problem-Solution Approach. Apress, 2008.

Harbour, Jonathan. Beginning Game Programming . Course Technology, 2006.

Harbour, Jonathan. Game Programming All in One. Course Technology, 2006.

Jacobs, Scott. Game Programming Gems 7. Charles River Media, 2008.

Jones, Wendy. Beginning DirectX 10 Game Programming. Course Technology, 2007.

Lengyel , Eric. Mathematics for 3D Game Programming and Computer Graphics, 2nd Edition. Charles River Media, 2003.

Lobao, Alexandre. Beginning XNA 2.0 Game Programming: From Novice to Professional. Apress, 2008.

Media, Charles River. Ai Game Programming Wisdom 3. Charles River Media, 2006.

Meigs, Tom. Ultimate Game Design: Building Game Worlds. McGraw-Hill Osborne Media, 2003.

Miles, Rob. Microsoft XNA Game Studio 2.0: Learn Programming Now. Microsoft Press, 2008.

Perry, Jim. Essential XNA Game Studio 2.0 Programming . Wordware Publishing, 2008.

Rabin, Steve (editor). Introduction to Game Development. Charles River Media, Inc., 2005.

Rabin, Steve. AI Game Programming Wisdom 4. Charles River Media, 2008.

Rollings, Andrew & Dave Morris. Game Architecture and Design. New Riders Publishing, 2004.

Rouse III, Richard. Game Design Theory and Practice, 2nd Edition. Wordware Publishing, 2005.

Salen, Katie & Eric Zimmerman. Rules of Play: Game Design Fundamentals. MIT Press, 2003.

Salen, Katie & Eric Zimmerman. The Game Design Reader: A Rules of Play Anthology. MIT Press, 2005.

Sherrod, Allen. Ultimate Game Programming With DirectX. Charles River Media, 2006.

Spaulding, Seth. Team Leadership in the Game INdustry. Course Technology , 2008.

Thompson, Jim, Barnaby Berbank-Green & Nic Cusworth. Game Design: Principles, Practice, and Techniques – The Ultimate Guide for the Aspiring Game Designer. Wiley, 2007.

Thorn, Alan. Introduction to Game Programming With C++. Wordware Publishing, 2007.

Yuzwa, Erik. Game Programming in C++: Start to Finish. Charles River Media, 2006.

Zerbst, Stefan & Oliver Duvel. 3D Game Engine Programming. Course Technology, 2004.

Course Objectives:

At the conclusion of the course, the student should be able to:

1. Implement large programs using advanced C++ features and other tools.

2. Utilize trigonometry and vector mathematics to solve game-related problems.

3. Apply artificial intelligence techniques to create autonomous agents.

4. Use a graphics library to render appropriate visual representations of 3D objects and scenes.

5. Perform various roles within a group development environment.

6. Research, implement and present various topics involved with game design and development.

Course Assessments:

1. Presentation of research and current implementations of selected topics in game development.

2. Participation in group reviews and assessments to determine the quality of game design and implementation.

3. Active participation in the development of an interactive game.

4. Examinations and/or quizzes