CIEE Global Institute – London

Course name: Fundamentals of Computer Game Design and Development
Course number: (GI) CMPS 1002 LNEN
Programs offering course: London Open Campus Block
Open Campus track: STEM and Society
Language of instruction: English
U.S. semester credits: 3
Contact hours: 45
Term: Spring 2020

Course Description

This course introduces both computer game design and development. It includes important computer design concepts and fundamentals to create electronic games using C# and Unity. Students will manage paper and digital prototyping, design iteration, and user testing. They will also use game scripting and programming, including computer graphics and animation. The audience for this course includes current and aspiring game designers and those interested in all principles of the game creation process.

Learning Objectives

Upon completion, students taking this course will be able to:

- Relate the historical evolution of gaming to innovations in computing technology and changing culture.
- Dissect paper and digital games to reveal their major design elements and mechanics.
- Use standard game design processes to propose and develop new games in several genres.
- Critique and consider game design tradeoffs, including storytelling, characters, environment, user interface, sound and general atmosphere.
- Improve games using iterative player feedback and design.
- Create digital games using scripts, objects, rooms and levels.
- Define and use fundamental object programming concepts, including Boolean Operations and Conditionals, Variables and Components, Loops, Lists and Arrays, Functions and Parameters, Debugging, Classes and Object-Oriented Thinking

- Write and execute games with online game makers, C# and Unity platforms.
- Explore how digital game design and development can lead to professional careers.
- Articulate how digital games impacts their life, as well as society.
- Develop intercultural insight to better understand the intersection of games and culture.

Course Prerequisites

None

Methods of Instruction

Students will attend interactive lectures that lead to problem-solving workshops, discussions and interpretation. Instruction emphasizes experiential learning, participation and applications. Students will use critical thinking to connect historical perspectives, contemporary design, programming and player experience. Discussions and written assignments develop a better understanding of computer games, as well as how gaming informs culture and society. Excursions will showcase local application and innovation.

Assessment and Grading

Participation	20%
Weekly Quizzes (6)	15%
Activity Worksheets (5)	15%
Program 1	20%
Digital Games and Culture Essays (2)	10%
Program 2	20%
Total	100%

Course Requirements

Weekly Quizzes

Each week, students will take a quiz on the previous week's course material, including lectures, activities and readings. Quizzes will have True/False, Multiple Choice, filling in blanks and short answer questions. Quizzes will cover only new material from that week but will build on previous concepts.

Activity Worksheets

During and after lectures, students will have a series of tasks, discussions and demonstrations related to the lecture material. They will work both individually and in groups to complete the

tasks, handing in answers to a series of questions, working with simple code or debugging simple program script before leaving the class.

Program 1

Small groups of students will work with instructors and online game construction platforms to create a simple two-dimensional game. This game should include several components of more complex games. It will be debugged and rewritten to improve efficiency and player experience.

Digital Games and Culture Essays

Students will use online resources to explore the intersection between computer games, society and culture. Students will write two 500-word essays: one on how culture is a component of computer games and another exploring how computer gaming impacts the local culture. In each case, computer games and gaming concepts will be used in the analysis and to make major points.

Program 2

Individual students will work with instructors to make their own computer game. This program will be written in C# and Unity. It will include a proposal, peer review, programming and debugging. The end result will be a three-dimensional game.

Participation

Each student is required to attend all sessions of the course and to participate actively in class discussions, class activities, laboratory or field sessions, with invited speakers and during site visits. Be prepared to read approximately 5,000 words per week and take notes from readings and during lectures. Students will also work regularly on computers in class, using a variety of computer languages to write limited programs.

Attendance Policy

Regular class attendance is required throughout the program, and all unexcused absences will result in a lower participation grade for any affected CIEE course. Due to the intensive schedules for Open Campus programs, unexcused absences that constitute more than 10% of the total course will result in a written warning.

Students who transfer from one CIEE class to another during the add/drop period will not be considered absent from the first session(s) of their new class, provided they were marked present for the first session(s) of their original class. Otherwise, the absence(s) from the original class carry over to the new class and count against the grade in that class.

For CIEE classes, excessively tardy (over 15 minutes late) students must be marked absent. Attendance policies also apply to any required co-curricular class excursion or event, as well as to Internship, Service Learning, or required field placement. Students who miss class for

personal travel, including unforeseen delays that arise as a result of personal travel, will be marked as absent and unexcused. No make-up or re-sit opportunity will be provided.

Attendance policies also apply to any required class excursion, with the exception that some class excursions cannot accommodate any tardiness, and students risk being marked as absent if they fail to be present at the appointed time.

Unexcused absences will lead to the following penalties:

Percentage of Total Course Hours Missed	*Equivalent Number of Open Campus Semester classes*	*Minimum Penalty*
Up to 10%	1 content classes, or up to 2 language classes	Participation graded as per class requirements
10 – 20%	2 content classes, or 3-4 language classes	Participation graded as per class requirements; **written warning**
More than 20%	3 content classes, or 5 language classes	Automatic **course failure**, and possible expulsion

Weekly Schedule

NOTE: this schedule is subject to change at the discretion of the instructor to take advantage of current experiential learning opportunities.

Week 1 Introduction to Game Design

Session 1.1 Thinking Like a Designer and the Game Analysis Framework

Students explore the basics of computer design. They begin with a history of computer games and consider how games have changed over time. They relate changes both to technical advances and social changes. Students will play several simple games and begin to analyze them for degree of difficulty, skill, decision-making and general engagement. They examine the rules and consider modifications to them. Students will then consider several definitions for games and why defining games is important. They also define ludology and explore major game analysis frameworks.

Readings: Chapter 1 Thinking Like a Designer and Chapter 2 Game Analysis Frameworks

Watch: Video Games: The Movie. 2014.
https://www.youtube.com/watch?v=LnHRqGHi4bY

Week 2 Game Analysis

Session 2.1 The Layered Tetrad

Students introduce themselves to the Layered Tetrad, a tool to help understand and create games through an analysis of mechanics, play, socialization, meaning and culture. They explore three layers of this framework: Inscribed, Dynamic and Cultural, and how these represent a transfer of ownership from game developer to players. Students spend time drilling down on the Inscribed Layer of game analysis.

Readings: Chapter 3 The Layered Tetrad and Chapter 4 The Inscribed Layer

Watch: Sholwater, J. 2017. Web Design vs. Web Development: Which is right for you? https://www.youtube.com/watch?v=Ujc3yhN9E5Y

Quiz 1 (covers material from Week 1)

Session 2.2 Dynamic and Cultural Layers to Games

Students see that when players start playing a game, the game takes on new dimensions of strategy and meaningful choice. Students explore these emergent properties and how game designers can anticipate and learn from them, including mechanics, aesthetics, narrative and technology. They then go on to examine the cultural layer, where players and society take over control of the game and make it their own. Students work in groups to examine these layers in both popular and unpopular computer games.

Reading: Chapter 5 The Dynamic Dimension and Chapter 6 The Cultural Layer

Due: Activity Worksheet 1

Week 3 Game Design

Session 3.1 Design Goals and Prototyping

Students examine how game designers craft interactive experiences. They revisit iterative design and its importance in game design, including analysis, design, implementation and testing. They consider the importance of innovation in game design, and how this is achieved through brainstorming and ideation. They work in groups to brainstorm, breaking the process down into expansion, collection, collision, rating, discussion, reevaluation and scoping. Students discuss design goals, including their own goals and those of the player.

Quiz 2 (covers material from Week 2)

Watch: Brush, T. 2018. 5 Mistakes I Made When I Started Making Games. https://www.youtube.com/watch?v=RqweVvRfVDg

Readings: Chapter 7 Acting Like a Designer, Chapter 8 Design Goals and Chapter 9 Paper Prototyping

Session 3.2 Game Testing, Math/Game Balance and Guiding the Player

Students will define and explore paper prototyping. They consider the benefits of paper prototyping and several tools to expedite its use. Students work in groups to put paper prototyping to the test with a 2D adventure game. They then move on to game testing, understanding that high-quality testing is essential to good design. They learn various methods for playtesting games, implementing them properly and at what stage of development each method of game testing is most appropriate. Students explore systems of probability and randomness as they relate to both paper and digital game technology. They discuss different mechanisms to guide the player through the game successfully and with appropriate level of challenge.

Readings: Chapter 10 Game Testing, Chapter 11 Math and Game Balance and Chapter 12 Guiding the Player

Due: Activity Worksheet, Computer Science and Culture Essay 1

Session 3.3 Puzzle Design, Staying on Top and the Digital Game Industry

In this session, students define puzzles and their importance in game design. They identify different genres of puzzles, as action, story, construction and strategy. They also consider why puzzles appeal to players, including challenge, distraction, character/environment and spiritual journey. Students explore goals of effective puzzle design and work in groups to examine classic and contemporary digital games for different puzzle types. They also explore how to best approach game projects, break them into sprints, and set task priorities. To do this, they use scrum methodology and burndown charts. Finally, students discuss the digital game industry and how navigate it with an invited local industry insider.

Reading: Chapter 13 Puzzle Design, Chapter 14 The Agile Mentality and Chapter 15 The Digital Game Industry and Game Design is Not for the Faint of Heart. 2019. Geek's Guide to the Galaxy https://www.wired.com/2019/02/geeks-guide-game-design/

Due: Activity Worksheet

Week 4 Digital Prototyping

Session 4.1 Digital Systems Thinking and the Unity Development Environment

In this session, students are introduced to computer programming in the context of digital games. Students explore the mindset needed to approach programming projects. They work in groups on activities to think about the world as systems of interconnected relationships and meanings. They investigate game objects, action lists and flowcharts, as they break digital game programs into sets of very simple decisions and commands. Students then download Unity, the game development environment they will use for the rest of the course. They begin to use its various window panes. They also begin to see why C# is an effective language to use with the Unity interface.

Quiz 3 (covers material from Week 3)

Readings: Chapter 16 Thinking in Digital Systems and Chapter 17 Introducing the Unity Development Environment

Watch: Burroughs, P. 2016. Create New Worlds: a journey through video game design TedxMSU. https://www.youtube.com/watch?v=x313GWFiKgU

Due: Program 1

Session 4.2 Introducing C# and Your First Program

This session introduces students to the basic syntax of C#. They describe the importance of C# as a compiled language vs. an interpreted language. Students consider the importance of C# as managed code, strongly typed, functioned-based and object-oriented. They then work to read and understand C# syntax. Students go on to write their own first program as a working Unity project.

Readings: Chapter 18 Introducing Our Language: C# and Chapter 19 Hello World: Your First Program

Session 4.3 Programming Fundamentals with Unity and C#

Students are introduced to several variables and component types used in Unity/C# programming, some common to computer languages and some unique to the platform. They explore the GameObject/Component structure of Unity and how to use Unity Inspector to set and modify variables. Students learn Boolean operations, like AND, OR or NOT, comparison statements and conditional statements.

Reading: Chapter 20 Variables and Components, Chapter 21 Boolean Operations and Conditionals

Due: Activity Worksheet, Digital Games and Culture Essay 2

Week 5 **Object-Oriented Programming**

Session 5.1 Loops and Collections in C#

Students define loops, consider their use in game programming and type code examples into Unity. They see how standard game loops use player information to redraw the next frame and repeat actions. Students then use loops with lists, arrays and dictionaries. The explore several array types and when to use arrays or lists. Students then begin to work with large numbers of game objects. They apply what they learned to build greater complexity into their game development.

Quiz 4 (covers material from Week 4)

Readings: Chapter 22 Loops, Chapter 23 Collections in C#

Watch: Chiodini, J. 2016. Can a Video Game Save a Life? TedXOldham
https://www.youtube.com/watch?v=6rLJaVohSRQ

Due: Program 1

Session 5.2 Functions, Parameters and Debugging

Students define and investigate functions. They write their own custom functions that take different variable as input arguments and return a single variable at the function's result. They explore special parameters for function input, like function overloading, optional parameters and the params keyword modifier. Students will use functions to write games with more nimble, modular and reusable code. They go on to investigate debugging. They will define it, explore different examples of debugging. They will compare a compile-time error and a runtime error, set breakpoints in code and carefully proofread their code to root out hard-to-find bugs.

Readings: Chapter 24 Functions and Parameters, Chapter 25 Debugging

Session 5.3 Classes and Object-Oriented Thinking

Students define, create and use classes. They come to understand that a class is a collection of both variables and functions in a single object, using C#. They see that classes are an essential building block in modern games, and object-oriented programming, in general. Students take their knowledge of game programming to apply object-oriented thinking. They specifically structure projects for the Unity development environment. Students consider component-oriented design and modularity, how they can make code simpler and easier to manage.

Readings: Chapters 26 Classes and Chapter 27 Object-Oriented Thinking

Due: Activity Worksheet

Week 6 Game Prototypes and Tutorials

Session 6.1 The Apple Picker and Mission Demolition

Students work in groups and individually on a series of focused tutorials to help learn prototypes for various kinds of games. They begin with the Apple Picker, a simple arcade game similar to Kaboom!. They consider the importance of a digital prototype. They go on to explore the game design aspects of the Apple Picker and eventually delivering its coding. Next, they make their own physics game, a la Angry Birds, called Mission Demolition. It includes such physics game staples as collision, mouse integration, levels and game state management.

Quiz 5 (covers material from Week 5)

Readings: Chapter 28 Prototype 1: Apple Picker and Chapter 29 Prototype 2: Mission Demolition

Session 6.2 Space SHMUP

Students use their programming knowledge and examples to build a SCHUMP or "Shoot Em Up" game. This game defines and provides programming for inheritance, static fields and methods, and the singleton pattern.

Readings: Chapter 30 Prototype 3: Space SHMUP and Chapter 31 Prototype 3.5 Space SHMUP Plus

Watch: Engineering Truth. 2016. Game Designer: Reality vs. Expectations. https://www.youtube.com/watch?v=ByIBRktO2wA

Session 6.3 Where to Go from Here: Your Digital Gaming Future

Students will present their programs to one another in a 5-8-minute PowerPoint presentation. They discuss each other's original games and offer suggestions for how to improve them. With their instructor, students will also consider next steps for their game design and development careers.

Final Quiz 6 (covers material from Week 6)

Due: Program 2

Course Materials

Course Textbook

Bond, Jeremy Gibson. 2017. *Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C#*. 2nd edition, Addison-Wesley Professional