(Digital) Game Design – Syllabus

Fall 2015 – 04:547:415:01

Bill Crosbie

(732) 305-2767
bcrosbie@rci.rutgers.edu
skype: bill.crosbie
twitter:@bcrosbie

Class site will be in Sakai (http://www.sakai.rutgers.edu)

Catalog Description:

Creating a digital game involves merging many the skills of many disciplines to a single cohesive whole. It involves applying principles from computer programming, two and three dimensional digital art, animation, physics, mathematics, artificial intelligence, user interface and experience design, psychology, narrative design and visual communication, to provide an incomplete list.

The purpose of this course is designed to expand upon the work done in Game Design Methodology (04:547:315) to provide an introduction in working with these disparate skill sets by building subsets of a complete game in a digital form. These may include but are not limited to interactive 3d environments, prototypes of game play, physics systems and animation techniques supported by the game engine used in the course. Upon completion the student will have a working knowledge of the different skills needed for digital game implementation.

Instructional Objectives:

1. Discuss the features and trade-offs of different technologies used to implement digital games
2. Develop fluency with one game engine in implementing various subsystems, including game mechanics, player input, user interface, artificial intelligence and other systems common to digital game creation
3. Develop a critical eye toward commercial digital games so that the student is able to dissect features, including technical, artistic and game dynamics of how a particular game play system is implemented
4. Develop the student into a creative problem solver that is able to pause when presented with a setback, analyze the feedback she/he is receiving and develop and implement a plan to attempt a new approach to implementing a technological solution

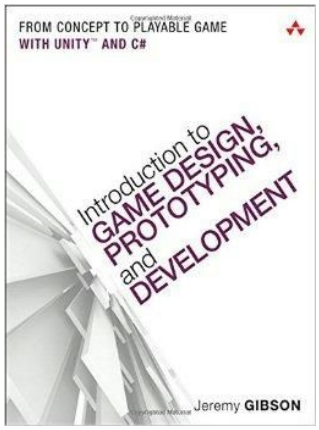
Learning Objectives:

Students will be able to:

1. Discuss and document game creation decisions using industry standard terminology
2. Implement examples of game play systems in the game engine used in the course
3. Assemble multiple game play systems in a cohesive game play experience

Course Materials

Textbooks and online resources:



Introduction to Game Design, Prototyping and Development, Jeremy Gibson
Addison Wesley 1st edition (2014)
ISBN-13: 978-0321933164 ISBN-10: 0321933168

Website: http://book.prototools.net

Lynda.com

Scripting Unity with C# - Kelley Hecker
http://www.lynda.com/Unity-tutorials/Scripting-Unity-C/365280-2.html

Unity 4.3 Essential Training – Adam Crespi

http://www.lynda.com/3D-Animation-Games-tutorials/Unity-43-Essential-Training/150613-2.html

Unity 4 2D Essential Training - Jesse Freeman
http://www.lynda.com/Unity-2D-tutorials/Unity-4-2D-Essential-Training/159243-2.html

Level Design Basics in Unity
http://www.lynda.com/Unity-3D-tutorials/Level-Design-Basics-Unity/133341-2.html

Game Prop Creation in Maya
http://www.lynda.com/Maya-tutorials/Game-Prop-Creation-Maya/97403-2.html

Additional Resources:

lyndaCampus is a school-wide version of lynda.com, an online training library of over 80,000 video based training movies on over 1400 software titles.

lyndaCampus is available for students registered in SC&I courses and can be used to learn software, programming skills, video techniques, etc. Titles include: all of MS Office (Word, Excel, PowerPoint, Access, etc.), all of Adobe CS5 & CS6 (DreamWeaver, Photoshop, Illustrator, InDesign, Flash, etc.), Fundamentals of Video, iMovie, Social Media Marketing, Twitter, SPSS, Google Docs, HTML5, CSS, MySQL, PHP, Python, C/C++/C#, WordPress, etc. Students can watch the videos, follow along with exercise files, and even print a certificate of completion upon completion of a course.

For more information and access instructions, please see **http://lynda.comminfo.rutgers.edu**. For any difficulties accessing the content of lynda.com please contact SC&I IT Services at

Steve Garwood, MLS MCIS
Director of Instructional Design and Technology
732.932.7500 ext. 8080
sgarwood@rutgers.edu

Class structure

This class will be taught in a partially 'flipped' manner in which a significant portion of our time together will be used for active problem solving using the engine. In our (ALMOST) 3 hour session, I will strive to use minimal time for class administration and lecture, although elements of the text are theory based and will necessitate some time. Some portion of the lectures explaining the key points from the readings will take place via videos from the instructor placed in the course shell. We will hold extended discussions of the main theoretical points online. These posts will be graded. We will use a portion of the class time to expand on the discussion and handle questions that arose from it.

There will be a hands on lab where a new technique from the chapter or prototype is worked on collaboratively. The remaining time will be studio feedback and 'code sketch time', where students will be expected to show progress toward completion of the projects and receive suggestions on ways to improve the components they are creating.

Types of assignments

Code Prototypes(60%) – You will be asked to implement 5 different code prototypes from the examples at the back of the book in order to learn different ways of structuring C# code in Unity. Each example contributes 10 points toward your final grade. Part of the prototype grade will be a quiz of the main technique(s) used in implementing that prototype.

Frameworks, Engine and C# language quizzes (30%) –There will be quizzes to ensure that the student understands the main concepts of game systems analysis, the game engine and the C# scripting language.

Final Exam (10%) - The final exam tests the student's knowledge of the terminology and conceptual understanding of working with game creation tools. If a student has a sufficiently high average and has completed all of the code prototypes, the Final exam may be waived if it is determined to be mathematically irrelevant to the final grade. Students will be given sample tests throughout the semester using the Sakai test system as a way of gauging their understanding of terminology throughout the semester in preparation for the final.

Optional Extra credit assignments:
Implement additional prototypes from the back of the book (10 % per prototype) – implement and extend one of the other prototypes from the Gibson text. Please use github.

Analyze and create design document to implement project from 315 as digital game (10%) - take the game that you made from the game methodologies course and design the plan with an analysis of the main systems, artwork, audio etc necessary to create the project in digital form. NOTE: you do not implement the project.

Outdoor Environment with FPS Walker (Extra credit up to 10%) – The game engine is very well suited to creating lush outdoor environments fairly easily. As a way of getting comfortable with the toolset, you will create an outdoor environment with mountains, grasslands, trees, water and if you wish, buildings and other structures. You will add a simple first person control scheme to the camera so that viewers can 'walk' through your environment.

Grade Breakdown – Code Prototypes & Homework –

Every artist needs to sketch to get better. Code prototypes will allow you to practice relevant skills. They are designed to reward consistent practice.

A. Completed the code prototype and extended the examples from the week with your own exploration.
B. Completed most of the code prototype, but was not able to get all elements working
C. Did not complete a code sketch, but did attempt some elements in partial form
F. Did not submit a code sketch

Grade Ranges - Course Final

Grades will be calculated as weighted averages with the code sketches and the final project being weighted more heavily that earlier projects or the final exam. The grade ranges will be

A. Exhibited consistent work at the highest levels. Deep engagement with the material and demonstrable active participation with your peers in studio time and outside of class

meetings. Understands and utilizes proper terminology in development documents and demonstrates this on the final exam.

B+. Exhibited work that either met the acceptable quality level with a major project above B level or developed work at high level but lacked consistency of quality throughout the semester.

B. Exhibited consistent work meeting acceptable quality level. Understands most terminology and demonstrates this on the final exam.

C+. Exhibited work that either met the basic requirements in most instances with one project above C level or worked at acceptable quality but lacked consistency of quality throughout the semester

C. Exhibits consistent work meeting the basic requirements. Projects demonstrate basic competency with the tool but are not suitable for inclusion in online portfolio or for showing to potential employers.

D. Exhibited little understanding of the requirements for projects or projects seriously lacked attention to detail.

F. Course work was consistently unacceptable

These grades roughly equate to the following numerical scale:

A – 100-91 **B+** - 90-87 **B** – 86-81 **C+** - 80-76 **C** – 75-67 **D** – 65-60 **F** – 59-0

However apart from the Code Sketches (0-3) and the tests and final exam(0-100), no numerical grades will be given on assignments.

Attendance/Participation Policy

This class is demanding and requires significant work outside of class to gain familiarity with the language (C#) and the Unity Engine objects. It is important that you come to class having attempted to complete the prototype for that week. It is expected that each student do the assigned readings prior to attending class

This course requires hands on participation with the software and will require you to practice technical and artistic skills. Our class meetings will be the primary time for instructor intervention and direct instruction on techniques requiring assistance. It is expected that you will attend all class sessions. Even if you do not need assistance yourself, sharing your knowledge and skills with your peers and getting feedback on your work is critical to the game creation process.

Students are expected to attend all classes; if you expect to miss one or two classes, please use the University absence reporting website -https://sims.rutgers.edu/ssra/ - to indicate the date and reason for your absence. An email will automatically be sent to me from this system. Note that if you must miss classes for longer than one week, you should contact a dean of students to help verify your circumstances.

Academic Integrity

A university's reputation is only as good as the reputation of the people who matriculate and graduate from the institution. The academy exists to advance human knowledge. Part of

advancing knowledge is being aware of and acknowledging the ideas of those upon which your contributions are based. Because of this enterprise, the university treats dishonesty in this area very seriously.

For this course:

- I expect you to collaborate with your peers
- I expect you to find other sources of scripts that add functionality to your projects
- I expect you to find art assets and textures that you could not otherwise create yourself

As a result of these expectations:

- I expect you to acknowledge your peers contribution to your work in header comments of scripts
- I expect you to acknowledge the source (URL/Author) of scripts you did not write yourself
- I expect you to provide a folder with a readme and any applicable license files for outside art assets

You may find out more information regarding the University policies, your responsibilities and the principles of academic integrity at:

http://academicintegrity.rutgers.edu/academic-integrity-at-rutgers

Serving Student with Disabilities

Students with documented disabilities who wish accommodations in this class must do so through the Rutgers Disabilities Services Office. See http://disabilityservices.rutgers.edu/ for details.

Course Breakdown

Week	Design	Unity work	C#	Self-Tests
1	2	Learn Unity Environment Set up github Unity tut – Roll-a-ball (A)	16,17	
2	1,2	Unity tut – Roll-a-ball (Due) Unity tut – space shooter(A)	18,19	Unity panes
3	3,4	15, 28 Applepicker (A)	20,21, 24	Unity var types
4	5,6,	28 ApplePicker (Due) 29 MissionDemo (A)	22,23, 24	
5	7,8,9	26 – Boids (in class)	25,26, 24	Game Design theory
6	14	29-MissionDemo (Due) 30-Space Shmup (A)		
7	11			
8	11	30-Space Shmup (Due)31- Prospector Solitaire(A)		Use of collections Line renderer component
9	27			
10		31 – Prospector Solitaire (Due)		Use of inheritance,
11	12	Bartok(A)		
12				
		THANKSGIVING		
13	13	Bartok(Due)		
14				
FINAL				Final Exam

Please read the designated chapters for the session listed in the column.