# (Digital) Game Design – Syllabus

# Summer 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 in 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 (04:547:415) 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

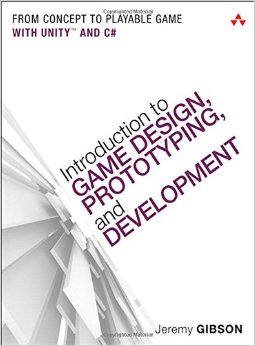
## Learning Objectives:

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 to 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 the bulk 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. At 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 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.

The start of our class time will be question and answer from the HW in scripting and production that was assigned. Then there will be a hands on exercise where a new technique from the chapter 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 & Homework(70%)** – consistency is rewarded in this course. Each class meeting there will be a code prototype due from the book. More advanced prototypes will be completed over multiple weeks. Progress must be made to the level/step indicated.

**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.

**Prototype Tests & Final Exam (20%)** –There will be tests after each prototype to ensure that the student understands the main concepts of the prototype. If the student has an A average on the tests, the student will not need to take the final exam.

**Additional gameplay extending a prototype (10 %) –** There are suggestions at the end of the prototypes to extend the prototype’s gameplay. You will extend either the OmegaMage game the QuickSnap or a card game to include additional gameplay beyond the code provided in the text.

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.

### 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 a code sketch that extended the examples from the week with your own exploration.

**B**. Completed code sketches that were duplicates from the examples of the week. Getting the hang of debugging coding and knowing how to use pre-existing code appropriately is useful

**C**. Did not complete a code sketch, but did attempt some elements in partial form

**F**. Did not submit a code sketch

### Grade Breakdown – Additional GamePlay

The game industry is highly competitive. It values work that is performed to high technological and artistic standards. Game companies also value personal initiative. Your work will be evaluated according to the following criteria:

**A**. Meets and exceeds the requirements of the assignment by combining elements from the lessons in interesting ways to create something new. Project exhibits a consideration of the overall player experience through the integration of technology, art, interface, and interaction design. An example worthy of inclusion in your professional portfolio.

**B**. Meets the requirements of the assignment at an acceptable quality level. Utilizes the techniques from the text and lectures and extends them or combines them.

**C**. Meets only the basic requirements of the assignment by directly mimicking the work presented in the texts or in lecture.

**D**. Does not meet the stated requirements of the assignment. Note that a visually stunning project that does something other than what is asked will earn this grade due to not meeting the specs required.

**F**. Work that is incomplete, not working, buggy, or plagiarized. A submission that shows not legitimate attempt to engage with the task required.

### 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.** Exhibitedconsistent 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+.** Exhibitedwork that either met the basic requirementsin 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 workwas 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 1-What is (digital) game design

#### Design (Ch 1 & 2)

* Extra Creditz – So You want to be a game designer
* Frameworks for thinking about design

#### Implementation (Ch 16-19)

* Setting up a GitHub Account
* Getting accustomed to Unity interface
* Writing our first scripts in C# Unity
* Implementing Roll-a-ball

#### Production

* No in class production work

#### Homework

* Get text book
* **Design**
  + Review Chapters 1-2
  + Read Chapters 3-6 Design
* **Implementation**
  + Review Chapters 16-19
  + Read Chapters 20-24
  + Complete Roll-a-ball
* **Production**
  + Develop a pitch document 2-3 pages that extends Roll-a-ball to make it more interesting

## Week 2- Getting familiar with our tools, GitHub, Unity, Monodevelop

#### Test

Unity panels

Unity game object components & variable types

#### Design (Ch. 3-6)

* Looking at game design from multiple perspectives

#### Implementation (Ch 20-24)

* Working with github
* Working with Unity
* Working with C#
* Implementing the Unity Space Shooter tutorial
* Learning to Debug

#### Production

* No in class production work

#### Homework

* Complete Apple Picker Due Thursday 6/11
* Design
  + Review Ch 3-6
  + Read Ch 7-9, ch 14
* Implementation
  + Review ch 20-24
  + Read Ch 15, 24, 25-26  
      
    Yes ch 24 is that important…

## Week 3-

#### Test –

* Theories of game design

#### Design (Ch 7-9, 14)

* Review paper prototyping
* Thought process of game design
* Digital game industry

#### Implementation(Ch 24-26)

* OOP – Boids – independent objects
* Apple Picker Ch 28 Due at end of class Thursday

#### Production

* Mock up a UI for ApplePicker (pencil sketch, Illustrator, balsamiq, etc)

Homework

* Design
  + Read Chapter 11
* Implementation
  + Tag your final build for Apple picker and submit pull request
  + Start Space Shmup (use code started on github)
* Production
  + Upload your UI mock-up to forum

## Week 8

* Showcase your final design to the class
* Turn in final design document/prototype
* Take final exam

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Design** | **Unity work** | **C#** | **Self-Tests** |
| **5/26** | 2 | Learn Unity Environment | 16,17 |  |
| **5/28** | 1,2 | Set up github  Unity tut – Roll-a-ball (A) | 18,19 |  |
| **6/2** | 3,4 | Unity tut – Roll-a-ball (Due)  Unity tut – space shooter(A) | 20,21, 24 | Unity panes Unity var types |
| **6/4** | 5,6, | 15, 28 Applepicker (A) | 22,23, 24 |  |
| **6/9** | 7,8,9 | 26 – Boids (in class) | 25,26, 24 | Game Design theory |
| **6/11** | 14 | 28 ApplePicker (Due)  29 MissionDemo (A) |  |  |
| **6/16** | 11 |  |  |  |
| **6/18** | 11 | 29-MissionDemo (Due)  30-Space Shmup (A) |  | Use of collections Line renderer component |
| **6/23** | 27 |  |  |  |
| **6/25** |  | 30-Space Shmup (Due)  31- Prospector Solitaire(A) |  | Use of inheritance, |
| **6/30** | 12 |  |  |  |
| **7/2** |  | 31 – Prospector Solitaire (Due)  32 – Bartok OR your prototype idea |  |  |
| **7/7** | 13 |  |  |  |
| **7/9** |  |  |  |  |
| **7/14** |  |  |  | Final Exam |
| **7/16** |  | 32 – Bartok OR your prototype (Due) |  |  |

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