Introduction to Game Development – Spring, 2013 Syllabus

USC School of Cinematic Arts, CTIN 483 *(Sections 18354 & 18355)*

For any class questions, send email to: SAs@ctin483.net

Instructor:	**Student Assistant:**
Jeremy Gibson	Will Hellwarth
(434) 321-8624	(310) 422-3054
jrgibson@usc.edu	whellwar@usc.edu

Course Rationale: Being a game designer requires an understanding of several different things, just a few of which are psychology, systems, fun, and storytelling. Each designer has different levels of expertise in each of these areas, and some only excel in one or two, but one thing that all designers need is the ability to communicate their ideas to others. Because games are a medium which, for the most part, must be experienced to be understood, rapid prototyping is often the best possible way to communicate your design ideas to others, and the ability of game designers to prototype can be thought of as equivalent to the ability of cinematographers to sketch; it's not the core skill of game design, but it does make it drastically easier for a game designer to communicate.

This course will teach you the basic knowledge you need to be able to create digital prototypes of your own ideas. While you should not expect to come out of this course a great programmer, you will come out a better designer. This course is also required preparation for CTIN 484: Intermediate Game Development, in which students work in pairs to create a single, released game by the end of the semester.

Course Description: This production class is focused on rapidly developing game prototypes. In this core course for the Interactive Media Division, students will learn the art of creating digital game prototypes, a practice they will hone throughout their time in the program. Although students are encouraged to have previously taken CTIN 400: Fundamentals of Procedural Media (or CSCI 101), this semester begins with the fundamentals of procedural programming before introducing object oriented programming and is therefore still open to programming novices.

There are three major reasons students of game design learn to program.

1. In games there is no written-word equivalent to the screenplay. In cinema, the screenplay is an excellent conceptual starting point for a film, allowing all team members to begin in preproduction with a shared understanding of the goals and potential pitfalls of the final piece. The central form for preproduction in games is the digital prototype. A concise and clear prototype provides an excellent template from which to engage collaborators, to discover unexpected play patterns, and to receive constructive criticism. If and when such a core game idea is proven to be engaging and interesting, the designer can move forward with confidence when forming a team and spending resources.
2. The second reason the game design student learns to program is to understand—at least in broad terms—what the programmers on a team do on a day-to-day basis. This is essential for facilitating communication amongst team members.
3. Finally, the rule sets, patterns, and behaviors found the game experience are a direct reflection of the algorithms and logic of the code itself. Not just a technical means to an end, computer programming is an art form that informs the software‘s aesthetic experience for the user.

This semester, we will be working with the Unity game development environment and C#. From extensive experience with this class, we have found that this combination of tools provides students with the easiest path to making game prototypes while also teaching them the fundamentals of game programming and giving them a platform which is actually being used to make published games (including Fusion Fall, Rochard, Thomas was Alone, etc.) across multiple platforms (Mac, PC, Linux, Web, iOS, Android, etc.).

You can download a free version of Unity from:

http://unity3d.com/unity/download.

There are also a few different Pro versions of the software which includes better lighting, visual effects for cameras, and the ability to compile to iOS or Android. Amazingly, Unity offers a 1-year educational Pro license exclusively for students for only $93 (a savings of $1907!) at:

http://www.studica.com/us/en/unity/unity-pro-4-student.html (Pro version w/o iOS or Android)

A good prototype is literally an experiment; it asks questions about game design. More often than not, prototypes prove that a given design direction is not viable or worth further pursuit, and such an outcome is a valuable failure. Obviously, the ultimate goal—this semester and after—is to develop innovative and compelling games, and prototyping is the fastest, surest way to achieve that goal.

Class Meeting Information: (all students must attend both lab and lecture)

When:	**Lab: Mondays 10a-12p**	**Lecture: Wednesdays 10a-1p**
or	**Lab: Mondays 2p-4p**	**Lecture: Wednesdays 2p-5p**
Where:	**The XML – RZC 119**	

Office Hours: **You <u>must</u> schedule appointments ahead of time by emailing SAs@ctin483.net**

Jeremy Gibson	– By Appointment (preferably Tuesdays)
Will Hellwarth	– By Appointment

Course Pre-requisites: None, though either CTIN 400 or CSCI 101 is recommended.

Required Textbooks: Sadly, there is no valid textbook for this class. I'm actually writing one now.

Evaluation of student performance:

Homework	40%
Classic Game Project	20%
Final Game Project	30%
Participation	10%
Total:	**100%**

In this class, good work and satisfying all of the requirements of the assignments will earn a student a B. To receive an A, a student must show creative, exceptional work which goes beyond the basic requirements of each assignment and brings something creative or otherwise impressive to the work. While graphical art can contribute to this transition from B to A, graphical ability is not otherwise graded in this class. CTIN 483 is about learning to develop games, and students will not earn credit for art unless all of the programming requirements for the assignment or project are already met.

During the Tutorial Phase of the class, Homework will be assigned almost every Wednesday and due the following Monday. This Monday assignment will be pass/fail based on whether the student demonstrated an effort to complete the assignment and turned the work in on time. The assignment will be examined and demonstrated in class on Monday, and students will then have until Wednesday to finish and polish their assignments for a letter grade.

Course Outline: **This material is subject to change.** This is a very flexible course, and this outline reflects that.

Week 1 – Week 7: Introduction to Unity and C#

Structure: Throughout this half of the semester, students will be instructed in various aspects of game prototyping using C# and Unity. New concepts will generally be presented on Wednesdays, and the students will receive a homework assignment to be turned in before class on the following Monday(usually for a pass/fail grade). On Mondays, the assignment will be explained and demonstrated, and then students will have until Wednesday to make their version of the assignment work properly and to add creative or interesting elements to the assignment to take it from the B they would receive for having a complete assignment to an A which may be awarded for exceptional work.

Assignments: Individual assignments each week. 50% of the grade for each assignment will be pass/fail assigned based on the work turned in on Monday, and the other 50% will be on a letter scale and assigned based on the work turned in on Wednesday.

Week 8 – Week 11: Classic Game Project

Structure: Students will work in pairs to create a game prototype which mimics the mechanics and "game feel" of a classic game from the 8-bit era.

Assignments: Pair assignment due Week 12. Students will recreate a game from the 8-bit era. For this assignment, it is not at all necessary that the graphics look like the original game, rather it is much more important that the game mimic the mechanics of the original as exactly as possible. For instance, if a pair were to recreate Super Mario Bros. for the NES, it would be fine if Mario were visually just a box or capsule as long as the way in which Mario jumped felt **exactly** like the jump in the original game (i.e. pressing the jump button for the same amount of time produces the same upward acceleration, jump height, and downward deceleration as seen in the original game).

Week 12 – Week 15: Final Game Project

Structure: Students will work in pairs to create an original game prototype.

Assignments: Pair assignment due during the final exam period. Students will create a new, unique game prototype. This will be based on their work throughout the semester and should both showcase all of the skills that they've learned throughout the semester and express a unique game design vision.

Final Exam – Monday, May 13 @ 8-10a (18354) OR @ 2-4p (18355)

FINAL GAME PROJECT DUE

Final Exam Time: Students will present their final games to each other and the faculty.

Course Website: http://ctin483.net

While this course has traditionally been available on USC Blackboard (http://blackboard.usc.edu), I have had several critical issues with that site, and this semester, I will attempt to move away from it to the class website.

Missing an Exam / Incompletes:

The only acceptable excuses for missing an assignment or taking an incomplete in the course are personal illness or a family emergency. Students must inform the instructor before the assignment is due and present verifiable evidence in order for a make-up to be scheduled. Students who wish to take incompletes must also present documentation of the problem to the instructor before final grades are due. Incompletes are not available before the Week 12 withdrawal deadline.

Note for students with disabilities:

Any student requesting academic accommodations based on a disability is required to register with Disability Services and Programs (DSP) each semester. A letter of verification for approved accommodations can be obtained from DSP. Please be sure the letter is delivered to me (or to TA) as early in the semester as possible. DSP is located in STU 301 and is open 8:30 a.m.–5:00 p.m., Monday through Friday. The phone number for DSP is (213) 740-0776.

Academic Integrity:

USC seeks to maintain an optimal learning environment. General principles of academic honesty include the concept of respect for the intellectual property of others, the expectation that individual work will be submitted unless otherwise allowed by an instructor, and the obligations both to protect one's own academic work from misuse by others as well as to avoid using another's work as one's own. All students are expected to understand and abide by these principles. Scampus, the Student Guidebook, contains the Student Conduct Code in Section 11.00, while the recommended sanctions are located in Appendix A: http://www.usc.edu/ dept/publications/SCAMPUS/gov/. Students will be referred to the Office of Student Judicial Affairs and Community Standards for further review, should there be any suspicion of academic dishonesty. The Review process can be found at: http://www.usc.edu/student-affairs/SJACS/.

Instructor Bio:

JEREMY GIBSON, M.E.T. — Assistant Professor of Cinema Practice

Jeremy Gibson is an Instructor of Cinema Practice at the University of Southern California who moonlights as an independent game designer and developer. Since 2009, he has taught classes in game design and digital prototyping for the Interactive Media Division of the USC School of Cinematic Arts. In that time, he has refined how design and digital prototyping are taught at USC, has taken an active role in both graduate and undergraduate education, and has created new classes, including Level Design Workshop with adjunct professor Scott Rogers, where students develop their individual artistic voice as level designers, and Immersive Game Design, in which students work in interdisciplinary teams to rapidly develop innovative experiences using cutting-edge technologies like the Microsoft Kinect.

Jeremy Gibson received a Masters of Entertainment Technology from Carnegie Mellon University and a B.S. in Radio, Television, and Film from the University of Texas. While at Carnegie Mellon, his team of students created the multiplayer game Skyrates, which won the Silver Gleemax Award for Strategic Gaming at the 2008 Independent Games Festival and the audience and editor's awards for Best Simulation Game of 2008 from JayIsGames.com. He has worked as a lead programmer and prototyper for companies such as frog design and Human Code and, while in graduate school, worked as an intern for both Walt Disney Imagineering and the Spore team at Maxis.

Prior to joining the USC faculty, Jeremy taught at the Masters of Digital Media Program at Great Northern Way Campus in Vancouver, BC and was an Associate Producer and Designer at Electronic Arts/ Pogo.com, where he designed and produced the game Crazy Cakes. He served as President of Digital Mercenaries, Inc. from 2001-2003 and is currently the Vice President of Airship Studios Corp.

In addition to his recent teaching, Jeremy has also created and taught game design and new media courses for Texas State University in San Marcos, Austin Community College, and the University of Texas.