

# Introduction to Game Development – Fall 2015 Syllabus

## Instructor

Margaret Moser

## Student Assistants

TBA

## Course Description

In this core course for the Interactive Media and Games Division, students will learn the art of creating digital game prototypes. The class is taught in the Unity game development environment using C# scripting.

This combination of tools gets students making games quickly while also teaching the fundamentals of game programming. Unity is a professional tool in widespread use, including on award-winning, high-profile games such as Monument Valley, The Room 1-3, Threes!, Assassin's Creed Identity, and many more. It can be used to create games for many platforms (Mac, PC, web, iOS, Android, etc.)

We will also introduce basics of Agile, the industry-standard production methodology, and the use of version control systems in developing code.

By the end of this course, you will have the skills to create digital prototypes of your own ideas. You should not expect to come out of this course a great programmer, but you will come out a better designer, equipped to explore and test your ideas without needing help.

## Course Rationale

All designers need the ability to communicate their ideas to others. Because games are interactive, a functioning prototype communicates the designer's ideas more clearly than any static description can. A game designer's ability to prototype is thus equivalent to a cinematographer's ability to sketch – while the skill does not directly appear in the final product, it allows the designer to refine his or her ideas and communicate them in a direct way to both team members and test audiences.

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 worth further pursuit. This may appear to be a “failure,” but in fact it is valuable information that helps you make good design choices. The ultimate goal for students in this program – this semester and after – is to develop innovative and compelling games, and prototyping is the fastest, surest way to achieve that goal.

Learning to write code has other significant benefits. Designers who understand code can collaborate more effectively with the engineers on a team, and make more informed decisions when implementation issues come up.

In addition, the rules, patterns, and behaviors that form the game experience are a direct reflection of the underlying code. Code is the raw material with which interactive experiences are built, as pottery is made of clay and paintings are made of paint. The designer who understands code therefore has a much deeper understanding of games as a medium.

## Course Pre-requisites

None, though either CTIN 101 or CSCI 101 is recommended. If you are working with code for the first time, you should expect to commit extra time to classwork each week.

## Class Meetings

### Section 18352

When: Mondays and Wednesdays 10-11:50am  
Where: SCA 356

### Section 18354

When: Tuesdays and Thursdays 2-3:50pm  
Where: SCI L113 (in the basement)

## Course Communications

Most assignments and materials will be made available through the Perforce version control system, which will be explained in class. All email announcements will be from **ctin483@gmail.com**. When available, lecture slides will be posted at **slides.com/margaretmoser**.

All email communications for this course will occur through your @usc.edu email. This includes emails you send; emails from other domains will be ignored. This is because it is good professional practice to separate your personal and professional communications.

Please contact **ctin483@gmail.com** for questions about class content, such as “I need help with this bug” or “I can’t log in to the class repository”. This account is checked frequently by several people, and it will get you the fastest response.

You may contact me at **mmoser@cinema.usc.edu** for administrative issues, such as “I’m sick and will miss class” or “I don’t understand my grade”. Please allow at least 24 hours for a response.

## Office Hours

### Professor Moser

Mondays 2-4pm, Thursdays 11am-noon, or  
by appointment

### SAs

TBA; will provide a link to the calendar  
here

During these times we are available for drop-in support in the department offices in SCI 201, near the arcade machine on the second floor. You can also reserve a 15-minute slot within these hours, or request appointments at other times, by emailing **ctin483@gmail.com**.

## Materials

Required:

*Introduction to Game Design, Prototyping, and Development* – Jeremy Gibson

Optional:

*Learning C# Programming with Unity 3D* – Alex Okita

*Unity for Absolute Beginners* – Suzanne Blackman

*Learn To Code by Making Games* – Ben Tristem & Brice Fernandes, on udemy.com (\$10 through August 21, 2015)

Generally, technical books go out of date very quickly, and it is difficult to justify the investment. However, each of these resources covers not just the details of how to use the tools but important concepts and techniques in programming and game development. Note that Udemy regularly offers deep discounts – up to 95% – to those who wishlist a course.

We will use Unity 5.1 for this semester's class. You can download the free version of Unity from <http://unity3d.com/download>. Many students also choose to spend a small amount (on the order of \$20-30) on tools and art assets from Unity's Asset Store. However, many assets and tools on the store are free, and no paid assets are required.

## Evaluation of student performance

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

You are responsible for understanding assignments.

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 that goes beyond the basic requirements of each assignment and brings something creative or otherwise impressive to the work. While visual 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 make interactive prototypes, and students will not earn credit for art unless all of the programming requirements for the assignment or project are already met. This applies to the pair projects as well; each student must contribute equal effort to the coding.

**All homework and projects must be turned in before the beginning of class.** I will download what is on the class repository at the moment class begins; anything turned in

after that will not be graded. If you have trouble getting your work into the repository, you must contact the class email by 10pm the night before the assignment is due. If you have done so then I may (at my discretion) allow you to turn in your work during class. This is the only exception.

During the tutorial phase of the class, homework will be assigned in almost every class. These assignments will be pass/fail based on whether the student followed instructions and demonstrated an effort to complete the assignment. I will drop your lowest homework grade at the end of the semester.

Most assignments will require you to make things in Unity. When you make something in Unity, you must turn in a **build** with functioning code. We will go over in class what a build is, how to make one, and how to turn it in. If it doesn't do everything it's supposed to, but shows effort, you will still receive credit. If it doesn't run at all, you will not receive credit.

Participation consists of participating in class discussions and exercises and coming to office hours, in whatever combination you like. If you do not raise your hand all semester and never come to office hours, you will lose points here. Participation is also affected by attendance (see below for attendance policy).

## **Course Outline**

This material is subject to change. This course is designed to be adapted to the skills and goals of its students, and this outline reflects that.

### **Week I – Week 6: Introduction to Unity and C#**

Structure: During this part of the semester, students will be instructed in various aspects of game prototyping using C# and Unity. We will go over general syntax and code structures in C#, how to use the Unity editor, and how to work with Unity objects through code.

Assignments: Individual assignments each week. All assignments are pass/fail.

### **Week 7 – Week 10: Classic Game Project**

Structure: We will continue with lectures during the first class meeting of each week. In the second class meeting of each week students will present their in-progress builds.

Assignment: Pair assignment due Week 10. Students will work in pairs to create a game prototype that mimics the mechanics and "game feel" of a classic game from the 8-bit era.

**Section 18352: due Wednesday, October 28**

**Section 18354: due Thursday, October 29**

### **Week 11 – Week 15: Final Game Project**

**Structure:** We will continue with lectures during the first class meeting of each week. Students will also turn in a small prototype for this first meeting. In the second class meeting of each week students will present their in-progress builds.

**Assignment:** Pair assignment due during the final exam period. Students will create a new, unique game prototype. This project should both showcase the skills that they've learned throughout the semester and express a unique game design vision.

**Section 18352: beta due Wednesday, December 2**

**Section 18354: beta due Thursday, December 3**

## **Final Exam**

Students will add polish to their original games and submit them to the class repository.

**Both sections: final due at 11:59pm on Monday, December 14**

## **Absence Policy**

Students are expected to attend every class. This is for your own sake – we will move quickly, and it is easy to fall behind. Unexcused absences will affect your participation grade. You will also lose points on the classic game and final game projects if you are absent without excuse on a day when your team is presenting.

The only excused absences are for illness, family emergencies, and (with advance notice) commitments related to a scholarship you are receiving, e.g. for a varsity sport. You must contact me as soon as possible regarding your absence. Generally I will expect to hear from you before class; in exigent circumstances I would expect to hear from you within 24 hours. If I do not hear from you in a timely fashion you may forfeit your option to make up what you have missed.

All that said:

1. **If you are sick, stay home.** You need to be healthy to learn, and so do your classmates (and instructors).
2. I do not distinguish between mental health and physical health. If you cannot complete an assignment on time or come to class because of mental health issues, you must contact me promptly, just as with physical health problems.

## **Incompletes**

The only acceptable reasons for taking an incomplete in the course are personal illness or a family emergency. Students who wish to take incompletes must present documentation of

the problem to the instructor before final grades are due. Incompletes are not available before the Week 12 withdrawal deadline.

## **Behavior in Class**

Part of the purpose of this class is to understand and practice professional behavior. This includes many areas, from email communications to working in pairs, and it would be impossible to list them all. However, these are the general expectations:

1. Behave respectfully towards everyone.
2. Put forth your best effort.
3. Follow through on commitments, including communicating when you can't meet them.

## **Content Warnings**

This course is intended to support your creative explorations in code. For the original game you make as a final project, you are encouraged to make something that interests you. Generally this takes the form of lighthearted arcade-style games. However, if you make something more thematically adventurous, and you include content in your work which may cause distress to your fellow students, please make a verbal 'content warning' immediately before you present the work in class, and include a written content warning in the readme file of a project, when you submit the work for grading.

Content which requires a content warning includes graphic depictions or descriptions of violence, sexual acts, sexual abuse, torture, self-harming behavior such as suicide, self-inflicted injuries or disordered eating, body shaming, and depictions of the mental state of someone suffering abuse or engaging in self-harming behavior.

Students who ever feel the need to step outside class during the presentation or discussion of work that warrants a content warning may always do so without academic penalty. (You will, however, be responsible for any material you miss.)

If you have any questions about what warrants a content warning, or if you ever wish to discuss your personal reactions to material presented in class, I welcome such discussion as an appropriate part of our coursework.

## **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 an SA) 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 II.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/>.

For this class, you are encouraged to copy and modify code from online sources and from class demonstration projects. You are also welcome to work together. If you use more than 3 lines of code from an external source without modifying it, you must provide a link to the source as a comment next to the copied code. You may use any code presented in class without attribution.

If you use any assets (images, textures, sounds, etc.) that are not your own work, you must name and link to the source, either on a credits screen in your game or in a separate credits text file delivered with the build.

## **Instructor Bio**

Margaret Moser is an Assistant Professor of Practice at the USC School of Cinematic Arts, where she teaches courses on game design, digital prototyping, innovative mobile experiences, and experimental interfaces. She holds an MFA in Design and Technology from Parsons.

Margaret's work has been shown at Come Out & Play, Games4Change, and the Babycastles guerrilla game gallery in Brooklyn. She built web-based games at MTV Networks and has served as lead producer of two commercial iOS applications. She has spoken at AlterConf and IndieCade East, and is currently curating the Digital Selects exhibit at IndieCade 2015.