Introduction to Game Development – Spring, 2015 Syllabus

USC School of Cinematic Arts, CTIN 483

Instructor
Margaret Moser

Student Assistants
TBA

Course Communications

Most assignments and materials will be made available through the Perforce version control system, which will be explained and documented separately. All email announcements will be from ctin483@gmail.com. Lecture slides will generally be available at slides.com/margaretmoser.

All 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 email 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”. If I receive an email from you, I will verify that I received it, even if I can’t respond to the substance right away. If you don’t receive confirmation, I didn’t get your email.

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.

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 very 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.

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, equipped to explore and test your ideas without needing help.

Students of game design benefit in several ways from understanding code.

1. In cinema, the screenplay is the conceptual document for a film, allowing all team members to begin in preproduction with a shared understanding of the goals for the final piece. The central form for preproduction in games is the digital prototype. A concise and clear prototype provides a
platform from which you can engage collaborators, discover unexpected play patterns, and receive constructive criticism. If a core game idea proves to be engaging and interesting, you can move forward with confidence when forming a team and spending resources to develop it.

2. Designers who understand code can collaborate more effectively with the programmers on a team, and make more informed decisions when implementation issues come up.

3. The rule sets, 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 Description

This production class is focused on rapidly developing game prototypes. In this core course for the Interactive Media and Games Division, students will learn the art of creating digital game prototypes, a practice they will hone throughout their time in the program.

This class is taught in the Unity game development environment using C# scripting. This combination of tools provides students with the easiest path to making game prototypes 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.

Class Meetings

<table>
<thead>
<tr>
<th>Section 18354</th>
<th>Section 18355</th>
</tr>
</thead>
<tbody>
<tr>
<td>When: Mondays and Wednesdays 10-11:50am</td>
<td>When: Tuesdays and Thursdays 2pm-4pm</td>
</tr>
<tr>
<td>Where: SCA 356</td>
<td>Where: SCI 113 (in the basement)</td>
</tr>
</tbody>
</table>

Office Hours

Margaret Moser: Monday 2-4pm, Wednesday 2-4pm or by appointment
Student assistants: TBA

During these times we are available for drop-in support in the department offices in SCI 201. You can also schedule appointments outside of these hours by emailing ctin483@gmail.com.

Materials

No textbook is required, but Jeremy Gibson’s book, *Introduction to Game Design, Prototyping, and Development*, is very strongly recommended for all students. It is available via Amazon in both print and electronic forms. Although it is not required, I will refer to it and suggest readings from it related to whatever we are covering in class.

We will use Unity 4.6.1 for this semester’s class. You can download the free version of Unity from http://unity3d.com/download

This includes some free add-ons, e.g. for publishing to iOS. There is also a Pro (paid) version of Unity, as well as Pro versions of the add-ons. These include better lighting and visual effects, an AI pathfinding system, powerful animation tools, improved team workflow, and better optimization for mobile games.

Unity offers two different licensing schemes for students:

1. **Subscription model.** A 1-year educational Pro license for students (currently $99) includes the Pro versions of Unity’s entire suite of products (iOS, Android, Team, etc.) For comparison, a
commercial-use license for the base Unity Pro license is $900/year, plus $900/year for each add-on.

2. **Perpetual license.** A permanent, commercial license, non-watermarked – the same as you’d get as a professional – is available to students for half price. The add-ons for iOS etc. are sold separately, also at half price. For comparison, the basic Pro license costs $1500 for non-students, plus $1500 for iOS Pro and $1500 for Android Pro. With this type of license, you will generally have to pay a few hundred dollars to upgrade to future versions.

Both license types currently include a free upgrade to Unity 5.0 when it is released (likely around March 2015). Both are available at http://www.studica.com/unity.

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**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework</td>
<td>40%</td>
</tr>
<tr>
<td>Classic Game Project</td>
<td>20%</td>
</tr>
<tr>
<td>Final Game Project</td>
<td>30%</td>
</tr>
<tr>
<td>Participation</td>
<td>10%</td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td><strong>100%</strong></td>
</tr>
</tbody>
</table>

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 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.

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. **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 do something you’re especially proud of, either on an assignment or in one of your project builds, you will have an opportunity to show it in class. If you do this at least once during the semester, you will have an opportunity to drop your lowest homework grade.

You are responsible for understanding assignments. It is best to ask for clarification in class, but you may also send questions to ctin483@gmail.com. Remember that we check that address frequently, but not constantly; you will generally get a response within a few hours.

Some homework 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 demonstrates 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, coming to office hours, and contributing to offline discussions, 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 1 – Week 6: Introduction to Unity and C#

Structure: Throughout 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. There will be frequent, small “quizzes” reviewing concepts and syntax, as well as exercises in Unity itself. All assignments are pass/fail.

Week 7 – Week 10: 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 10. 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).

Due: Monday, March 23 (section 18354) 
      Tuesday, March 24 (section 18355)

Week 11 – 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.

Beta due: Wednesday, April 29 (section 18354)
          Thursday, April 30 (section 18355)

Final Exam

Students will present their final games. The only authoritative source of information about the final is http://classes.usc.edu/term-20151/finals/. The current schedule shows the dates and times below. Take note that they are not our usual class times.

Final due: Monday, May 11, 8-10am (section 18354) 
           Tuesday, May 12, 2-4pm (section 18355)
Absence Policy

Students are expected to attend every class. This is for your own sake – we will move quickly, and it will be 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.

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:
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, abuse (especially sexual abuse or torture), self-harming behavior such as suicide, self-inflicted injuries or disordered eating, eating-disordered behavior or 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 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/.

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 lectures on game design, digital prototyping, innovative mobile experiences, and experimental interfaces.

Margaret's work has been shown at Come Out & Play, Games4Change, and the Babycastles guerrilla game gallery in Brooklyn. She has built web-based games at MTV Networks, lectured on Agile production methods for small teams, and served as lead producer of two commercial iPad applications.

Margaret holds an MFA in Design and Technology from Parsons. Her research interests include persuasive games related to environmental issues, alternative and non-screen interfaces, and formal structures for interactive narratives. She splits her spare time between finding weird new games to play and the quest for the perfect taco.