COURSE SYLLABUS

Course Number: CS179.14B
Title: Special Topics in Multimedia: PC and Console Game Development II
Department/Program: DISCS **School:** School of Science and Engineering
Semester: 2nd **School Year:** 2015-2016
Instructor/s: Wilhansen Joseph B. Li <wil+cs179.14b@byimplication.com>
Course Website: http://http://moodle.ateneo.edu/ls/course/view.php?id=736
May 21

A COURSE DESCRIPTION

The course focuses on the fundamentals of PC game programming continuing from CS179.14a. Students will learn how to build a game from scratch in order to gain a deep understanding of their architecture and components, as opposed to using a pre-made game-making software. The format of the lesson will be a mix of lectures followed-by hands-on implementation. By the end of the semester, the students should be able to produce a networked multiplayer game.

B COURSE OBJECTIVES

At the end of the course, students should:

- structure games with appropriate tradeoffs between speed and maintainability.
- be able to prototype gameplay in a short amount of time.
- know various architectures and apply them to make their game scale.
- design protocols to communicate game state for multiplayer games.

C COURSE OUTLINE AND TIME FRAME

The instructor reserve the right to make adjustments to the course outline as he deems fit.

Week and Topic	Learning Objectives	Activities	Student Output
1-2: Advanced C++ • Operator Overloading • Polymorphism • Constructors, Copy Constructors, Destructors • Templates • Smart Pointers • Multiple Inheritance • Virtual Inheritance	• Know how to implement operator overloading. • Implement pure virtual functions. • Create generic functions and classes • Use smart pointers to minimize programming errors.	• Lecture • Homework	C++ Program
3-4: Program Structure and Object Generation • Update method • Object Generation • Object Pool • Service Locator	• Write a SHMUP	• Lecture • Homework	Top-down Shoot-em-up
5-7: Networking • Addresses • TCP vs. UDP • Multicasting and Broadcasting • Peer Discovery • MTU • Packet Reordering • NAT Punchthrough	• Discover Peers • Transmit data over UDP • Deal with missing packets	• Lecture • Homework	Networked Program

8: Networked Physics • Lockstepping • Dead Reckoning	• Simulate physics using transmitted data • Deal with missing packets	• Lecture • Homework	Multiplayer Bouncy Balls
9-10: Matrices • Definition • Operations • Inverses • Orthogonal Matrices • Application	• Perform Matrix Multiplication • Invert Matrices • Factorize Matrices	• Lecture • Quiz	Recitation and Quiz
11-12: Interpolation • Interpolation definition • Function manipulations • Continuity and differentiability • Bézier Curves	• Manipulate functions to fit interpolation requirements • Use interpolations to smoothen animations. • Fit more complex data data	• Lecture • Homework Quiz	Interpolation library
13: Input and Character Control • Keyboard and mouse input • Gamepad input • Character control/physics	• Query for keyboard and mouse input • Query for gamepad support and input • Create a basic 2D platformer	• Lecture • Hands-on	2D platformer
14: Event Systems • Observer pattern • Generic Observer	• Create an event system • Integrate the event system into the entity framework	• Lecture • Hands-on exercise	Event system
15: Configuration Management • Windows Registry • Windows user and application directories • Configuration files (INI, XML, JSON) • String parsing	• Read and write from the registry. • Know the proper directories to store configuration • Generate and parse configuration files	• Lecture • Hands-on	Configuration framework
16-17: Complex Numbers and Quaternions • Complex Numbers • Quaternions	• Compute product of complex numbers and quaternions • Apply 2D rotation using complex numbers • Apply 3D rotation using quaternions	• Lecture • Quiz	Quiz
18: Finals	Application of lessons	Project	

†Will be taken when there is enough time.

D REQUIRED READING

Any one of the following will suffice:

- Game Engine Architecture, by Jason Gregory, Jeff Lander and Matt Whiting, A K Peters (ISBN-13: 978-1568814131), 2009
- Game Programming Patterns by Robert Nystrom (ISBN-13: 978-0990582908), 2014 (available online for free at `http://gameprogrammingpatterns.com`)

- Game Networking http://gafferongames.com/networking-for-game-programmers/
- Beej's Guide to Network Programming http://beej.us/guide/bgnet/

E SUGGESTED READINGS AND RESOURCES

1. Courseware (for announcements, quizzes, etc.): http://http://moodle.ateneo.edu/ls/course/view.php?id=736
2. Syllabus: https://github.com/CS179-14B-2015-2016/syllabus
3. Course Github Organization: https://github.com/CS179-14B-2015-2016
4. Ten C++11 Features Every C++ Developer Should Use
   http://www.codeproject.com/Articles/570638/Ten-Cplusplus-Features-Every-Cplusplus-Developer
5. C++Now 2014 Presentations https://github.com/boostcon/cppnow_presentations_2014
6. Game Programming Gems series
7. Game Engine Gems series
8. Glenn Fiedler's Game Development Articles and Tutorials: http://gafferongames.com/
9. The Witness: http://the-witness.net/news/
10. Wolfire Games Blog: http://blog.wolfire.com/
11. Gamasutra: http://www.gamasutra.com/
12. Game Physics by David H. Eberly, Morgan Kaufmann
13. Thinking in C^{++} 2nd Ed. (Eckel, Bruce):
    http://www.mindview.net/Books/TICPP/ThinkingInCPP2e.html
14. C^{++} FAQ lite: http://www.parashift.com/c++-faq-lite/
15. What Every Computer Scientist Should Know About Floating-Point Arithmetic: http://docs.oracle.com/cd/E19957-01/806-3568/ncg_goldberg.html
16. A Primer on Bézier Curves http://pomax.github.io/bezierinfo
17. Standard C^{++} https://isocpp.org/

F COURSE REQUIREMENTS

5%	Project Proposal
20%	(non-final) Project Milestones
15%	Final Project Milestone
15%	Class Participation/Recitation
20%	Final Project
25%	Quizzes and Homework

F.1 Class Participation/Recitation

1. Each recitation is worth 3% (out of the total 15%).
2. Depending on your answer, the grade per recitation can drop.
3. There's no limit on how much you can recite.
4. The recitation will be totaled in the end to compute the class participation grade.
5. You may be called after a homework to present it in class. The presentation would last a maximum of 20 minutes and is worth 4%. The process (architectural decisions, bugs encountered, debugging process and solution) of making the game should be explained. Only the game is needed for presentation; no keynotes allowed.
6. Excess points are spilled over to other components based on the following formula:

$$\log_2(x+1)$$

   where x is the recitation points in excess of 15.

F.2 Project

1. The output will be an application of the lessons learned in the course.
2. The project need not be original. Just make sure that you cite your source(s), know what you're doing and don't blatantly copy-paste. Remember, the project defense has a heavier weight than your project.
3. The project need not have original art, you may get artwork from resources from the internet. Be sure to cite your sources.
4. Submit the Group Certificate of Authorship, stating all resources and references used (except for the references listed here), together with the project or during defense. Failure to do so will mean a 0 for the project.
5. Do not plagiarize. Offenders will be dealt severely.

6. Project grade breakdown:

3%	Submission
2%	Proper usage of git
3%	Persistence
3%	"Data-driven"-ness
5%	Networking
4%	Polish

F.3 Project Consultation

1. The project consultation is an (admittedly futile) attempt to get people moving on their projects.
2. The project consultation is four parts, contributing 5% each, one every end of month.
3. Failure to do the consultation before the deadline will result in a 0 for the component.

F.4 Project Proposal

1. The project proposal is to be submitted at the first project milestone deadline.
2. It is to be maintained until the finals.
3. The evolution of the document should be noted, either by Git, or by other means (Google docs, multiple MS Word files, etc.)
4. It should accurately reflect the current game.

F.5 Project Defense

Project defenses are 1.5 hours each. Each member should prepare in detail their contribution to a project and be prepared to be questioned. If a question is directed at an individual, *others may not interrupt; failure to comply counts as not answering the question*.

There is no dress code for project defenses but at least dress properly.

Final Defense grade breakdown:

5%	Presentation
10%	Contribution

F.6 Quizzes and Homework

1. Quizzes are usually unannounced.
2. Quizzes will be given at the start of the class. Latecomers may only catch up.
3. Quizzes are to be written on a A4 bond paper or posted on Moodle.
4. Although not necessary, non-graphing calculators may be used in quizzes.
5. A decent skill on arithmetic and geometry (including trigonometry) is expected.
6. Expect quizzes when a reading assignment is given.
7. Expect homework to come every week.
8. There will be no quiz on the week when a homework is due except when it is a moved deadline.
9. Homework either culminates the activities during class time or puts the students in a situation where they have to discover things. They are to be done by group.
10. Homework are graded on the scale of 0 to 4. (4 is only given for exceptional work)

F.7 Hands-on Exercises

Hands-on exercises are meant to enhance learning by putting theory into guided practice. These are not graded because I believe that grading them while preventing cheating is a futile effort and a waste of time. As such, it is to the student's discretion whether or not to do the hands-on exercises. Be forewarned though that 1) some homework may rely on the exercises and 2) there may be exam questions that are based on the exercises.

F.8 Submissions

All works (homework and final project) should be submitted via Github using the following protocol: (taken from `https://education.github.com/guide/forks`):

1. Fork the original repository (there will be one per homework).
2. Clone the repository to your computer.
3. Modify the files and commit changes to complete your solution (add project comments in the provided README.md file).
4. Push/sync the changes up to GitHub.
5. For any member of the group, create a pull request on the original repository to turn in the assignment.
6. State your group name in the pull request title.

F.9 Late Submissions

The score for late submissions will be reduced by 15% for every hour, or fraction thereof, late. (i.e. score is multiplied by $\max(0.15h, 0)$ where h is the number of hours late rounded up)

F.10 Bonuses

- Up to 8% will be credited for contributions to any FOSS projects during the semester.
  - Multiple contributions will be amalgamated and the total bonus shall not exceed 10%.
  - Deadline for notifications of contributions will be on the last day, 23:59, of the finals week.
  - Documentation revisions are considered but will not guarantee bonuses. Do not expect bonuses for trivial corrections such as spelling or grammatical corrections.
  - Pending or unaccepted submissions are considered but will not guarantee bonuses.
  - Wikipedia edits are not considered.
  - For bonuses that are needed to in order pass the class, only game-related contributions are accepted (i.e. one cannot contribute to a Ruby on Rails project to pass the class).
- Bonuses are added to the final grade.
- Bonuses are privileges, not rights. As such, crediting them is up to the instructor's discretion.

G GRADING SYSTEM

[92%,100%]	A	Excellent	[69%,75%)	C	Sufficient
[87%,92%)	B+	Very Good	[60%, 69%)	D	Passing
[80%,87%)	B	Good	< 60%	F	Failure
[75%,80%)	C+	Satisfactory			

Note: $[a, b)$ means a half-open interval that includes a but excludes b. Rounding is done only in the final grade to two decimal places.

H CLASSROOM POLICIES

The class atmosphere will be relaxed but still orderly. The golden rule here is to respect the instructor and not distract others.

1. Usually, thursday is the "lesson implementation" time.
2. Foods and drinks (except water) are prohibited inside the lab.
3. Attendances will not be checked however, they will be noted (i.e. you will be judged).
4. You are responsible for your absences.
5. If you are running late, don't storm or waltz in the classroom.
6. Permission is not needed to leave the classroom but do so discreetly (i.e. don't rage quit).
7. Cellphone usage is allowed during non-exam class times as long as it is used discreetly.
8. Collaboration on homework, including the problem set, is allowed. However, *copying is strictly prohibited*.
9. Those committing academic dishonesty should be ready face the infinity + 1 banhammer from the department.
10. Excessive noise will not be tolerated.
11. Computers and laptops may freely be used. However, I will not repeat lessons due to divided attention brought about by social networking sites and games.
12. Students are expected to learn C/C^{++} on their own. Questions on the language may be asked during consultation hours.
13. No make-up quizzes. No make-up midterms or homework deadline extension unless you were hospitalized, an immediate member of your family died (grandparents included), or you are whisked away due to some competition. Adequate proof must be provided (e.g. medical certificate, ADSA notice) and the instructor must be notified as soon as possible.

I Consultation Hours

By appointment. If you set an appointment with me, keep it or inform me a.s.a.p. if you cannot make it. You may also email me by the email address stated in this syllabus or by my Ateneo email address. Class-related emails sent to any other email addresses will be ignored.