Game Programming *(MIS305)*

Spring Term 2015
M W F 11:45-12:50, Main Lab

Professor: Patrick M. Kelly

Office: 402 Old Main

Phone: 824-2174

E-mail: pkelly@mercyhurst.edu
Emailing will always be the best way of contacting me. Please feel free to email me anytime, and I should typically be able to get back to you within a few hours.

Note: Also, though, if you're emailing me for any reason other than turning in an assignment, **please do not include an attachment**. I have a rule established in Outlook that will send any email from a student in this class directly to a different folder (not my Inbox) if there is an attachment. I try to keep very current on the emails in my Inbox, but I typically won't even look in those other folders until I actually sit down to grade projects.

If you're trying to email me to ask a question and you would need me to look at your file, simply email me an attachment-less message first and explain your question, and then send a second email with the attached file. The second message will go directly to my /GameProgramming folder, but the first message will nag at me in my Inbox so that I know to look in my other folder for your file.

Also: **Always compress/zip your files when you send them to me.**

Office Hours:	Mondays:	4:00-5:30
	Tuesdays:	9:30-11:30
	Wednesdays:	4:00-5:30
	Fridays:	4:00-5:00

Meeting times by appointment may be arranged as well.

Blackboard

I do keep *Blackboard* current. Documents from class, assignments, grades, etc. will all be posted as they become available.

I'd also ask you to please check that the e-mail address listed for you on *Blackboard* (by default, your Mercyhurst account) is the account that you use regularly. You might want to change it on *Blackboard* if this is not the case.

Text

Holistic Game Development with Unity by Penny de Byl,
Focal Press, ISBN 0240819330
holistic3d.com/hgd

Software

We'll be using the free version of the software package *Unity* for this course (current version is 4.6.1). You can install this free software on your own computer by downloading it at http://unity3d.com.

At this same site, you can also download the *Unity Web Player*, which allows *Unity* files to be run within an internet browser. The Web Player is free, and furthermore the free version of the main program that we'll be using this term does allow you to save your creations in the Web Player format. Thus, you will be able to create games that will be playable on the internet.

Course Content

Game Programming is a course geared toward exposing the student to the fundamentals of computer programming in the gaming venue. Specifically, we will be exploring the realm of creating computer games that take place in a 3D environment and have the potential to be as complex and detailed as nearly any mainstream video game on the market today.

The primary tool that we will use for this endeavor is the development platform of *Unity*. This game engine is extremely powerful and flexible. The *Unity* engine also includes a vast amount of integrated components that are highly desirable in scenarios that strive to present realistic gameplay, making realism more easily attained by novice programmers.

However, this course will be about much more than just the how-to's of using the *Unity* software package. A good perspective to have is that the *Unity* program is a tool of the trade and we'll learn about how to use that tool this term . . . but we'll also learn about the trade in a broader sense as well. If you were to explore this avenue in your future, it is important to learn the generalizations of the concepts that we'll be exposed to this term—not the merely the specifics of how those concepts are carried out in *Unity*.

It is also important to realize that game programming is an art and a science. For the latter, there are certain standards in the industry and many, many specifications in the programming; your learning in this course needs to acknowledge these concrete elements and you must know how to incorporate them appropriately in your games. In regard to the art aspect of the discipline, there are many agreed-upon standards, principles, and philosophies that you need to have in mind when you're designing a game; failure to do so can render the game inefficient, unrealistic and unplayable. In this course we'll strive to address as many parts of both faces—the art and the science— of game programming as we can.

Goals & Learning Objectives

The goals for *Game Programming* are:

- to apply your preexisting programming skills in the environment of computer gaming;
- to expose you to the standards used in the industry of game programming;
- to help you increase your problem solving skills; and

The learning objectives for this course include:

- Hone a subset of critical thinking skills by studying modular programming and the logical control structures. Employ these in programming exercises and projects
- Create games and animations of one's own design.

- Demonstrate understanding of the following concepts and employ them in programming exercises and projects:
  variables, functions, parameters, recursion, random numbers, 3D coordinate systems, linear transformations

Grading

Your final course grade will be compiled from projects, quizzes, and two exams (midterm and final). The point distribution amongst these factors will be as follows:

Projects, points will vary = 300 points total
Quizzes, 10 points each = 50 points
Midterm Exam, 100 points = 100 points
Final Exam, 100 points = 100 points
Total: 550 points

Course grades will be determined using the following scale:

A	514-550 points (93.5-100%)	B+	492-513 points (89.5-93.3%)
B	459-491 points (83.5-89.3%)	C+	426-458 points (77.5-83.3%)
C	382-425 points (69.5-77.3%)	D+	355-381 points (64.5-69.3%)
D	327-354 points (59.5-64.4%)	F	0-326 points (0-59.3%)

Projects

The majority of your term-end grade will be based on projects that you will complete using the *Unity* program. You will submit this work electronically, providing me copies of the files and folders for each individual project.
Don't forget: **Always compress/zip your files when you send them to me, please.**

Although the number of projects I assign per week will vary from topic to topic and chapter to chapter, I would say to expect one assignment per week on average. The points the individual projects will be worth will vary as well, totaling to 300 points by the end of the term.

The majority of the 300 points for this component of your grade will come from projects that are either your own creation, guided examples provided by the author, or a combination of the two. For each of these assigned projects, there will be guidelines to follow to ensure that you demonstrate your understanding of the relevant concepts from the course. Creating your game so that it adheres to these guidelines will mark your proficiency with the tools discussed in class. Also, though, you will always be encouraged to infuse your own creativity in these projects . . . so don't *just* follow the guidelines; there will be points awarded for appropriately creative elements included in your projects.
As an example, if a project is going to be worth 20 points total, I might announce when assigning it that it's worth "16 + 0-4 points." This will mean that you'll earn 16 points for doing what's required for the assignment, but then there's also a range of 0 through 4 points that you can earn by adding your own flare to the project.

The projects assigned that are not instances of you creating your own games will be assignments where you will work directly from the textbook, following the author's step-by-step directions. These should very much be "easy points" to earn, because what I'll be asking you to do is just follow the author's hands-on demonstrations and then turn in the file to me as your assignment. To be blunt, this type of project is intended to force you to do the reading. Our author knows that in order to truly learn how to do something on a computer, you need to do it—not just read about it. With this in mind, our author includes a good number of "Unity Hands On" components to the readings, which you should sit down at a computer and do as you're reading it (whether it's something you're turning in for a grade or not).

Quizzes

There will be five quizzes given during the term. The intent of the quizzes is to assess your understanding of some of the more objective knowledge you should have when you're programming games in a 3D environment. These quizzes will typically cover material explained in the textbook and/or discussed in class. In terms of this discipline of game programming being an art and a science, it would likely be your understanding of some of the concepts from the science aspect that would be assessed with a quiz.

The Computer Lab

When courses are taught in a computer lab, there is always the risk that students will be unable to resist the temptation to constantly check email, Facebook, ESPN.com, or other such distractions that sitting in front of a computer can present.

I am actually much less worried about that possibility with this class than I am with other, lower-level courses.

For one thing, I believe that the subject of game programming is just inherently fun and interesting! I truly hope that each of you already possesses some measure of this perspective. I also truly hope that the learning that will take place in this course will be much, much less of a chore than it is something you'll enjoy doing.

Secondly, please realize up front that this will be a very challenging course. You will need to be an active learner in this class if you are going to absorb and understand all of the different aspects of the art and the science of computer programming. If you allow yourself to be pulled away from that active learning you, quite simply, will not succeed in this course.

Realize that much of what I have to show you in this course are things that I have to *show you*—typically on the overhead screen. If you're not paying attention while I'm demonstrating something, you can quickly become lost. Do not expect me to explain a concept to you individually if you weren't paying attention when I demonstrated it to the class as a whole

A Word on Programming

The programming component of this course will be manifested in the scripts we will write to give specific behaviors to the various objects of our game. *Unity* recognizes scripts written in JavaScript, C#, and Boo. It is recommended that you utilize

JavaScript the most (it is possible to mix languages even within the same project) because there is the most support and documentation available to you for JavaScript.

As with most programming languages, it is not expected for you to know every single command and all of the most specific formatting syntax when you're coding for your projects in this course. Hence, you will want to become very familiar with the very best resource available to you for help with the JavaScript language (and, specifically, how it's used in *Unity*): the **Scripting Reference**.

The Scripting Reference is a part of the 'help' documentation provided within the *Unity* software package. When you have a session of *Unity* open, simply click on any of the "?" to access 'help' and there you'll find the Scripting Reference. Alternatively, the Scripting Reference can be found as an html file on the hard drive of the computer you have *Unity* loaded. From a browser, you can access the Scripting Reference at file:///C:/Program%20Files/Unity/Editor/Data/Documentation/Documentation/ScriptReference/index.html (assuming that the program has been installed in the default directory of the computer you're on).

Also, regarding the prerequisite for this course of some programming experience: There is not any particular syntax or formatting that you must remember from your earlier programming course(s), but you should have a firm understanding of the most common programming concepts, such as *if statements*, *while statements* (*loops* in general, actually), *variables*, etc. These concepts will come up frequently during our coding efforts this term; please brush up on them as you need to.

One particular (and particularly easy!) common programming concept that I will insist you utilize is that of **comments**. As you should know, comments in your coding are not processed at all by the computer; they're there just for us humans. An undertaking in creating a computer game will inevitably end up being fairly complex. Because of this, comments will be essential in keeping the various components of your project in order. Furthermore, in the industry of game programming, you would never be working on your own; your inclusion of comments in the coding you write is critical to make that work accessible to the rest of the team.

For this course, you will be required to include meaningful comments in every coding component of your projects. Points will be deducted from projects if there are not relevant, meaningful comments in your code.

Services

Learning Differences

In keeping with college policy, any student with a disability who needs academic accommodations must call Learning Differences at 824-3017 or stop by Old Main room 314, to arrange a confidential appointment with the Disability Services Director during the first week of classes.

(Tentative) Course Schedule:

Day	Sect.	Material	
Wednesday, January 28th		Introduction to the course Introduction to *Unity* Navigating the *Unity* interface	

Friday, January 30th	1.4
	1.5
		Game engines
		Logic;
		Comments;
		Functions;
		Variables;	

Monday, February 2nd	1.5	Revisit ideas from first week
		Operators;
		Conditional Statements;	

Wednesday, February 4th	1.5	Arrays;	
		Objects	
	1.6	Power-of-Two Rule	
Friday, February 6th	2.2	Vectors	
	2.3	Cameras;	
		Coordinate Systems;	
Monday, February 9th	2.3	Translations, Rotation, Scaling;	
		Polygons and Normals	
	2.4	Quaternions	

Wednesday, February 11th	2.5
	
	2.6
	2.7
		Gravity;
		Newton's Laws of Motion
		Squash and Stretch
		Reducing Polygons;	

Friday, February 13th	2.7	Camera Viewing Volume;
		Fog;
		Textures;
		Billboards	

Monday, February 16th	3.2 3.3	Sprites Texture Atlas	

Wednesday, February 18th	3.4 3.5	Animated Sprites Baked 3D Animation	
Friday, February 20th	3.6	Biomechanics	
Monday, February 23rd	3.6	Biomechanics	

Wednesday, February 25th	3.6	Biomechanics	
Friday, February 27th	3.7	Single 2D Sprite Actions Single-Filed 3D Animations	
Monday, March 2nd	3.7 3.8	Blending Secondary Animation	

Wednesday, March 4th	4.2
	4.3
		Game Mechanics
		Searching;
		Matching;
		Sorting;
		Chancing;
		Mixing;
		Timing;
		Progressing;
		Capturing;
		Conquering;
		Avoidance;
		Collecting	

Friday, March 6th	4.3	Searching;
		Matching;
		Sorting;
		Chancing;
		Mixing;
		Timing;
		Progressing;
		Capturing;
		Conquering;
		Avoidance;
		Collecting	

Mon, 3/9	*Mid-Semester Break—no classes!*		
Wed, 3/11			
Fri, 10/8			

Monday, March 16th	4.4	Matching and Sorting;
		Shooting, Hitting, Bouncing, and Stacking;
		Racing;
		Avoidance and Collecting;
		Searching	

Wednesday, March 18th	4.4	Matching and Sorting;
		Shooting, Hitting, Bouncing, and Stacking;
		Racing;
		Avoidance and Collecting;
		Searching	

Friday, March 20th	4.5	Rewards and Penalties	
Mon, 3/23	***Mid-term Exam***	*All material covered up to this point.*	

Wednesday, March 25th	5.2 5.3	Line of Sight Graph Theory	
Friday, March 27th	5.4	Searching through Waypoints	
Monday, March 30th	5.5	Finite State Machines	

Wednesday, April 1st	5.6	Flocking	
Thursday, April 2nd	5.7	Decision Trees	
Fri, 4/3	*Easter Break—no classes!*		
Mon, 4/6			
Wednesday, April 8th	5.8	Fuzzy Logic	

Friday, April 10th	5.9	Genetic Algorithms	
Monday, April 13th	5.10	Cellular Automata	

Wednesday, April 15th	6.2
	6.3
		Game Structure
		Principles of User Interface Design;
		User Profiling;
		Metaphor;
		Feature Exposure;
		Coherence;
		State Visualizations;
		Shortcuts;
		Layout;
		Focus;
		Help	

Friday, April 17th	6.4	Inventories	
Monday, April 20th	6.5	Implicit Teleports; Explicit Teleports	

Wednesday, April 22nd	7.2	Map Design Fundamentals;
		Provide a Focal Point;
		Guide/Restrict Player's Movement;
		Scaling;
		Detail;
		Map Layout;
		Other Considerations	

Friday, April 24th	7.3	Terrain;
		Drawing Terrain;
		Procedural Terrain;
		Procedural Cities;
		Infinite Terrain	

Monday, April 27th	7.4	Camera Tricks;
		Depth of Field;
		Blur;
		Grayscale;
		Motion Blur;
		Sepia Tone;
		Twirl;
		Bloom;
		Flares;
		Color Correction;
		Edge Detection;
		Crease;
		Fish Eye;
		Sun Shafts;
		Vignette;
		SSAO	

	7.5	Skies;
		Skyboxes;
		Sky Domes;
		Clouds	

Wednesday, April 29th	7.6	Weather;
		Wind;
		Precipitation	

Friday, May 1st	7.7	Particles	
Monday, May 4th		Open Lab	
Wednesday, May 6th		Open Lab	
Fri, 5/8	*Reading Day—no classes!*		
Wed, 5/13 at **10:30**	***Final Exam***	*Everything!*	