CS 3540 – Game Programming

Spring 2016 Course Syllabus

Course Information

Professor: Nik Bear Brown

Email: nik@ccs.neu.edu

Office: 451 West Village H

Office Phone: 617 373-5621 (x5621)

Office Hours : M,W 2-5PM, Tu,F 4-5PM

or by appointment

or just drop by my office to see if I’m there.

Lecture time:

1:35PM- 3:15PM Tuesday, Friday

Lecture location: Behrakis Health Sciences Cntr 105

Course website: Blackboard (for raw scores, uploading assignments)

Piazza: <https://piazza.com/northeastern/spring2015/3200> (Notes, Materials, Questions)

Course Structure

This course consists of programming assignments (every 10-14 days), a group project and an individual project. Students have the choice of their programming languages and game engines.

The topics are:

Game Engines

- Scene

- Hierarchy

- Project

- Inspector

- Game Objects

- Selecting

- Moving

- Rotating

- Scaling

- Snapping

- Parenting

- Character Controller

- Adding scripts

Navigation & Terrain

- Navigation and Functionality

- Cursor Control

- Terrain Generation

- . Add terrain to the project

- . Sculpt terrain using a height map

- . Sculpt terrain using Unity tools

- . Download and import the terrain toolkit

- . Utilize the terrain toolkit

- . Add textures to a terrain

- . Add foliage to a terrain

- . Add water to a terrain

- . Add ambient details to a terrain

- . Navigate and refine our world

Procedural Terrain

Assets, Actions & State

- Importing 3D Models

- Importing Audio

- Importing Textures

- Importing Scripts

- Asset Packages

- Unity Audio Basics

- Building and Playing Game Music

- Imported Assets

- . Add game objects to our project

- . Use meshes in our project

- . Apply textures to meshes

- . Work with cameras in Unity

- . Utilize layers to organize objects

- . Illuminate your world with lights

- . Create and use prefabs

- Action Objects

- Managing State

Physics and Special Effects

- Physics

- Rigid Bodies and Colliders

- Physics Material

- Cloth

- Rag Doll

- Special Effects

Menus, Inventory, Dialogue & Levels

- Menus

- Building a Title Screen

- Building the Main Menu

- Inventory

- Dialogue

- Levels

Game Artificial Intelligence

- Representing state

- State search

- Probability

- Examples of AI for common genres: Pong game, Match 3 (Tile-matching), Shooter (Asteroids, Defender, Space Invaders), Maze game, Fighting game (Street Fighter, Tekken), Platformer (Donkey Kong, Sonic, Mario, Temple Run), Puzzle (Tetris, Sudoku)

CgFX and Shaders

- CgFX

- Unity Shaders

- Unreal Shaders

Network Programming

- Network API

- .NET Sockets

- Implementing Master Server semantics

- Hosting a game and registering with the LAN

- Joining a LAN-Hosted game

- Issues with NAT and Multi-Player Network Gaming

- Issues with LAN and Multi-Player Network Gaming

Game Server and Games with Friends

- Building a game server

- Testing on the Unity Master Server

- Running your own Master Server

- Managing a Master Server

- Hosting a Game and Registering with your Master Server

Mobile Programming

- Mobile Programming

- . Design for mobile devices

- . Set up the Mobile Development Environment

- . Use a phone's accelerometer for game input

- . Use a phone's multi-touch screen for game input

Game Analytics

- High level overview of Game Analytics.

- Game Analytics tools

Course Description and Prerequisites

This course examines how to program for games in Unity 3D or Unreal Engine. Introduces the different subsystems used to create a 3D game, including rendering, animation, collision, physics, audio, trigger systems, shading, game logic, behavior trees, and simple artificial intelligence. Offers students an opportunity to learn the inner workings of game engines and how to use libraries such as physics and graphics libraries to develop a game. Discusses graphics pipeline, scene graph, level design, behavior scripting, object-oriented game design, world editors, and game scripting languages. Game servers and mobile game programming (Android and iPhone) will also be introduced. Procedural level generation for Unity 3D or Unreal Engine using the Houdini Engine for Unity (<http://www.sidefx.com/unity>). Social/network gaming and game analytics and visualization will also be discussed.

* There will be weekly assignments creating small games or tools related to games.
* There will be weekly in-class progress presentations.
* There will be a research paper about (Due week 10).
* There will be a group project and an individual project. (Due week 13 – exam week).

The first half of the class will primarily involve the mechanics of game programming in Unity 3D or Unreal Engine by building a series of small games. Students will be given a simple skeleton and then expected to enhance/extend the skeleton for each assignment. Programming for the following game types will be discussed:

* Pong game
* Match 3 (Tile-matching)
* Shooter (Asteroids, Defender, Space Invaders)
* Maze game
* Fighting game (Street Fighter, Tekken)
* Platformer (Donkey Kong, Sonic, Mario, Temple Run)
* Puzzle (Tetris,Sudoku)

The second half of the class will primarily involve social/network gaming, converting a game to a mobile platform, setting up game servers, procedurally generating content, game artificial intelligence, game analytics and visualization.

The class is very interactive so bring a laptop if possible. Mondays will consist demos and lectures. Wednesdays will consist of students presenting progress and brainstorming. Students will be expected to present their progress EVERY Wednesday. While informal, the Wednesday brainstorming session is a crucial part of your participation grade (15% of overall grade). This is a VERY HANDS ON class. You will learn game programming by building games.

Prereq. CS 3500 and CS 3520.

Learning Objectives

Learning objectives for the course are:

* Learn how to game program C# in a clear, robust, efficient, and safe manner.
* Unity 3D rendering
* Animation
* Unity 3D collisions
* Unity 3D physics
* Unity 3D audio
* Unity 3D trigger systems
* Unity 3D shaders
* Game logic
* Social/network gaming
* Mobile gaming
* Game servers
* Procedural content
* Artificial intelligence
* Game analytics and visualization.

Course Materials

The textbooks are all available for free to NEU students via SpringerLink (<http://link.springer.com/>). the *required* textbooks we will be using in this class are:

**Beginning 3D Game Development with Unity: All-in-One, Multi-Platform Game Development**

Author: Sue Blackman, *Apress* 2013

ISBN: 9781430248996

SpringerLink: <http://link.springer.com/book/10.1007/978-1-4302-4900-9>

**Learn Unity 4 for iOS Game Development**

Author: Philip Chu, *Apress* 2013

ISBN: 9781430248767

SpringerLink: <http://link.springer.com/book/10.1007/978-1-4302-4876-7>

**Learn Unity for 2D Game Development**

Author: Alan Thorn, *Apress* 2013

ISBN: 9781430262305

SpringerLink: <http://link.springer.com/book/10.1007/978-1-4302-6230-5>

**Game Analytics - Maximizing the Value of Player Data**

Editors: Magy Seif El-Nasr, Anders Drachen, Alessandro Canossa, *Springer* 2013

ISBN: 978-1-4471-4768-8 (Print) 978-1-4471-4769-5 (Online)

SpringerLink: <http://link.springer.com/book/10.1007/978-1-4471-4769-5>

**Introduction to Artificial Intelligence**

Author: Wolfgang Ertel, *Springer* 2011

ISBN: 978-0-85729-298-8 (Print) 978-0-85729-299-5 (Online)

SpringerLink: <http://link.springer.com/book/10.1007/978-0-85729-299-5>

Recommended books (free to NEU students via SpringerLink):

**Beginning C# Object-Oriented Programming**

Authors: Dan Clark, *Apress* 2013

ISBN: 978-1-4302-4935-1 (Print) 978-1-4302-4936-8 (Online)

SpringerLink: <http://link.springer.com/book/10.1007/978-1-4302-4936-8>

**C# Quick Syntax Reference**

Authors: Mikael Olsson, *Apress* 2013

ISBN: 978-1-4302-6280-0 (Print) 978-1-4302-6281-7 (Online)

SpringerLink: <http://link.springer.com/book/10.1007/978-1-4302-6281-7>

**Learning C# by Programming Games**

Authors: Arjan Egges, Jeroen D. Fokker, Mark H. Overmars, *Springer* 2013

ISBN: 978-3-642-36579-9 (Print) 978-3-642-36580-5 (Online)

SpringerLink: <http://link.springer.com/book/10.1007/978-3-642-36580-5>

Hardware:

You must has access to a computer that runs Unity 3D or Unreal Engine. If you don’t have a laptop or home computer that can run Unity 3D or Unreal Engine. If you don’t have this then contact me, and we’ll set you up with an account in the game lab.

Mobile device - You must has access to a mobile device (iPhone/iPad/Android) around week nine of the course. The mobile device need not have a working phone number; just the accelerometer and touch pad functionality for testing a port of a game to a mobile device. If you don’t have this then contact me and we’ll partner you with somebody with a mobile device.

Software:

Unity 3D (<http://unity3d.com/unity/download>)

Houdini Engine for Unity (<http://www.sidefx.com/unity>)

Unreal Engine (<https://www.unrealengine.com/blog>)

Participation Policy

Participation in discussions is an important aspect on the class, and contributes towards your grade. Participation on the Piazza discussion forums (e.g. asking and answering questions about assignments, discussing readings, midterm review) also counts towards this grade. It is important that both students and instructional staff help foster an environment in which students feel safe asking questions, posing their opinions, and sharing their work for critique. If at any time you feel this environment is being threatened—by other students, the TA, or the professor—speak up and make your concerns heard. If you feel uncomfortable broaching this topic with the professor, you should feel free to voice your concerns to the Dean’s office.

Collaboration Policies

Students are strongly encouraged to collaborate through discussing strategies for completing assignments, talking about the readings before class, and studying for the midterms. However, all work that you turn in to me with your name on it must be in your own words or coded in your own style. Directly copied code or text from any other source is not allowed. Feel free to discuss general strategies, but any written work or code should be your own, in your own words/style. If you have collaborated on ideas leading up to the final solution, give each other credit on what you turn in, clearly labeling who contributed what ideas. Individuals should be able to explain the function of every aspect of group-produced work. Not understanding what plagiarism is does not constitute an excuse for committing it. You should familiarize yourself with the University’s policies on academic dishonesty at the beginning of the semester. If you have any doubts whatsoever about whether you are breaking the rules – ask!

To reiterate: **plagiarism and cheating are strictly forbidden. No excuses, no exceptions*.*** *All incidents of plagiarism and cheating will be sent to OSCCR for disciplinary review.*

Assignment Late Policy

Assignments are due by 11:59pm on the due date marked on the schedule. Late assignments will receive a 20% deduction per day that they are late, including weekend days. It is your responsibility to determine whether or not it is worth spending the extra time on an assignment vs. turning in incomplete work for partial credit without penalty. Any exceptions to this policy (e.g. long-term illness or family emergencies) must be approved by the professor.

Grading Policies

Students are evaluated based on their participation in class (both online and during lecture), responses to supplemental readings, performance on programming assignments, and both the execution and presentation of a group midterm and final project (which take the place of exams). If a particular grade is required in this class to satisfy any external criteria—including, but not limited to, employment opportunities, visa maintenance, scholarships, and financial aid—it is the student’s responsibility to earn that grade by working consistently throughout the semester. Grades will not be changed based on student need, nor will extra credit opportunities be provided to an individual student without being made available to the entire class.

Grading Rubric:

The following breakdown will be used for determining the final course grade:

|  |  |
| --- | --- |
| Assignment | Percent of Total Grade |
| In Class Milestones | 20% |
| Programming Assignments | 50% |
| Group Project | 15% |
| Individual Project | 15% |

I expect to use the following grading scale at the end of the semester. You should not expect a curve to be applied; but I reserve the right to use one.

|  |  |
| --- | --- |
| Score | Grade |
| 93 – 100 | A |
| 90 – 92 | A- |
| 88 – 89 | B+ |
| 83 – 87 | B |
| 80 – 82 | B- |
| 78 – 79 | C+ |
| 73 – 77 | C |
| 70 – 72 | C- |
| 60 – 69 | D |
| <60 | F |

BlackBoard:

You will submit your assignments via BlackBoard. Click the title of assignment (blackboard -> assignment -> <Title of Assignment>), to go to the submission page. You will know your score on an assignment, project or test via BlackBoard. BlackBoard represents only the raw scores. Not normalized or curved grades. Extra credit will be notified via e-mail.

Scores:

Your scores will be available via BlackBoard and the solutions (if applicable) via Piazza.

Special Accommodations

Students who have documented disabilities should contact me privately to discuss any specific needs and to agree upon a reasonable accommodation. Please see the campus Disability Resource Center for information on the kinds of accommodations available through the University. Students MUST notify me at least a week before exams if they are requesting additional time so that we can find a proper room.

Schedule and Announcements

The schedule will be maintained on the course website, and any updates to it will be announced on Piazza.

A rough schedule is outlined below (it is subject to change).

**Week 1 – Introduction**

**-** Course overview

- Unity Overview

- Creating a new Unity Project

- Layouts, Panes and Tabs

- Scene

- Hierarchy

- Project

- Inspector

- Game Objects

- Selecting

- Moving

- Rotating

- Scaling

- Snapping

- Parenting

- Character Controller

- Adding scripts

***Reading:***

***-*** Blackman: Chapters 2, 3

***Assignment:***

***-*** Write a simple pong-like game in Unity.

**Week 2 – Navigation & Terrain**

**-** Navigation and Functionality

- Cursor Control

- Terrain Generation

- . Add terrain to the project

- . Sculpt terrain using a height map

- . Sculpt terrain using Unity tools

- . Download and import the terrain toolkit

- . Utilize the terrain toolkit

- . Add textures to a terrain

- . Add foliage to a terrain

- . Add water to a terrain

- . Add ambient details to a terrain

- . Navigate and refine our world

***Reading:***

***-*** Blackman: Chapters 4, 5, 6

***Assignment:***

***-*** Write a simple running & jumping -like game in Unity with your own terrain.

**Week 3 – Assets, Actions & State**

**-** Importing 3D Models

- Importing Audio

- Importing Textures

- Importing Scripts

- Asset Packages

- Unity Audio Basics

- Building and Playing Game Music

- Imported Assets

- . Add game objects to our project

- . Use meshes in our project

- . Apply textures to meshes

- . Work with cameras in Unity

- . Utilize layers to organize objects

- . Illuminate your world with lights

- . Create and use prefabs

- Action Objects

- Managing State

***Reading:***

***-*** Blackman: Chapters 7, 8, 9

***Assignment:***

***-*** Extend running & jumping -like game in Unity with interactions with imported assets to a “temple run” like game

- Requirements document for Group Project

**Week 4 – Physics and Special Effects**

**-** Physics

- Rigid Bodies and Colliders

- Physics Material

- Cloth

- Rag Doll

- Special Effects

***Reading:***

***-*** Blackman: Chapter 11

***Assignment:***

***-*** Add special effects to “temple run” game.

**Week 5 – Menus, Inventory, Dialogue & Levels**

**-** Menus

- Building a Title Screen

- Building the Main Menu

- Inventory

- Dialogue

- Levels

***Reading:***

***-*** Blackman: Chapters 12, 13, 14, 15

***Assignment:***

***-*** Add menus, inventory, dialogue & levels to “temple run” game.

**Week 6 – Space Ace – Putting it all together**

**-** Review basic Unity by build a shooter game called “Space Ace”

***Assignment:***

***-*** Extend “Space Ace” code

- Requirements document for Individual Project

**Week 7 – Game Artificial Intelligence**

**-** Representing state

- State search

- Probability

- Examples of AI for common genres: Pong game, Match 3 (Tile-matching), Shooter (Asteroids, Defender, Space Invaders), Maze game, Fighting game (Street Fighter, Tekken), Platformer (Donkey Kong, Sonic, Mario, Temple Run), Puzzle (Tetris, Sudoku)

***Reading:***

***-*** Ertel: Chapter 6

***Assignment:***

***-*** Add artificial intelligence to “temple run” or “space ace” game.

**Week 8 – CgFX and Shaders for Unity**

**-** CgFX

- Unity Shaders

***Reading:***

***-*** Cg Toolkit <https://developer.nvidia.com/cg-toolkit>

- Unity Shaders <http://docs.unity3d.com/Documentation/Manual/Shaders.html>

***Assignment:***

***-*** Create a custom shader in Unity using CgFX.

**Week 9 – Network Programming**

**-** Unity Network API

- .NET Sockets

- Implementing Master Server semantics

- Hosting a game and registering with the LAN

- Joining a LAN-Hosted game

- Issues with NAT and Multi-Player Network Gaming

- Issues with LAN and Multi-Player Network Gaming

***Reading:***

***-*** Unity Networking Tutorial <http://www.palladiumgames.net/tutorials/unity-networking-tutorial/>

***Assignment:***

***-*** Add network friend data to “temple run” or “space ace” game.

- Individual Project Milestone Due

**Week 10 – Game Server and Games with Friends**

**-** Building a game server

- Testing on the Unity Master Server

- Running your own Master Server

- Managing a Master Server

- Hosting a Game and Registering with your Master Server

***Reading:***

***-*** Unity Master Server <http://docs.unity3d.com/Documentation/Components/net-MasterServer.html>

***Assignment:***

***-*** Add Internet friend data to “temple run” or “space ace” game.

- Group Project Milestone Due

**Week 11 – Mobile Programming**

**-** Mobile Programming

- . Design for mobile devices

- . Set up the Mobile Development Environment

- . Use a phone’s accelerometer for game input

- . Use a phone’s multi-touch screen for game input

***Reading:***

***-*** Chu: Chapter 4

***Assignment:***

***-*** Convert “temple run” or “space ace” game to mobile device.

**Week 12 – Game Analytics**

**-** High level overview of Game Analytics.

- Research Paper Due

***Reading:***

***-*** Seif El-Nasr et al.: Part I

***Assignment:***

***-*** Research Paper Due

**Exam Week – Final Projects Due (**Individual & Group Project**)**