Native Console Multiplayer Game Development
CSCI 499 (4 units)

Objective
Design and develop **multiplayer** 3D games in native C++ running on multiple consoles (hereinafter referred to as *Consoles*) including **XBox 360, Playstation 3, IPad, Playstation Vita**.

This course can serve for any of the four MS in Computer Science (Game Development) concentration areas as well as an elective for the BS in Computer Science (Games) degree program.

Instructors
Artem Kovalovs
Game Programmer, Naughty Dog Inc.

Mike Zyda
Director, GamePipe Lab, USC

Time
Lecture: Thursday 7:00-9:50p
Lab: Wednesday 5:00-7:00p
Office Hours: Saturday 10:00-12:00p

Course Structure and Description
Students will work individually and in teams towards creating a multiplayer *Console* game implementing all necessary core engine systems to support it.

Students will be given PrimEngine framework as a starting platform - a minimalistic codebase with support for rigid and skinned mesh rendering, networking layer, and level editing. Importantly the framework runs on all *Consoles* making sure that students don’t start completely in “the dark,” as *Console* game development can be hard and scary.

The curriculum will start with short introduction to PrimeEngine and move on to multiplayer game development gradually introducing *Consoles* one at a time. In addition to *Console* development, and multiplayer development, lectures will include topics on fundamental game development areas like linear algebra, 3d math, graphics, physics. The multiplayer networking subsystems will be implemented by students following TRIBES networking model.
The timeline below shows the introduction of Consoles, lecture topics and student task implementation. Note the class incrementally adds complexity while continuing to work on previously introduced areas:
In the process of developing the demos students will have to expand the given framework to support visual and gameplay requirements of their demos. For this reason, students are required to do readings from Jason Gregory’s book “Game Engine Architecture” to learn about internal systems of modern game engines. Students are also going to be assigned readings from various websites on network programming and multiplayer game programming.

This class is a unique opportunity to gain experience in real-world multiplayer Console game development. Since no other school in the world provides curriculum of developing in native C++ on listed real-life development hardware, the students will be primary candidates for job opportunities in games industry and in software development in general.

Skills acquired by taking the class:

- Unique
  - Console game development
  - Multiplayer development with clients on different hardware
- Rare
  - Low level engine development
  - High level engine development to accommodate new game design
    - Level editor tools
    - Profiling tools
  - TRIBES networking model implementation including prediction/correction
- Common
  - C++
  - 3D Math
  - Socket Programming

Students will use USC asset database for 3D assets for their demos as well as interact with other schools within and outside of USC.

The course will be divided into two main sections:

1. Weeks 1-2: Learning framework, creating a simple gameplay demo based on concepts learned from tutorials. (Individual)
2. Weeks 3-End: Students work in teams on the multiplayer Console game demo. Weeks 3-End are divided into subsections where students will group in different teams implementing core engine subsystems required by their games, thus working towards a bigger goal.
   2.1 Weeks 3-6: Creating a demo with networking working on XBox 360, using TRIBES event manager and TCP
   2.2 Weeks 7-10: Creating a demo with networking between XBox 360 and IPad. Objects are created/destroyed using TRIBES ghost manager, reliable events with UDP.
2.3 Weeks 11-End: Work towards final demo running on Consoles with matchmaking and object movement prediction/correction

Textbooks Required


Prerequisite

Significant participation in a prior game development effort. Permission of instructor.

Lecture/Lab 3 hr/week

Student Developed & Utilized Materials

1. Perforce for source & asset control
2. Google docs for project maintenance, task and bug tracking
3. Students will be provided website space if needed for demo, docs, etc.

Grading

- Project 1: 5%
- Quizzes: 10%
- Linear Algebra / 3D Math Test: 15%
- Project 2: 20%
- Project 3: 20%
- Project 4: 30%

Statement 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 TA) 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. Website and contact information for DSP:
http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.html, (213) 740-0776 (Phone), (213) 740-6948 (TDD only), (213) 740-8216 (FAX) ability@usc.edu.

Statement on 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, (www.usc.edu/scampus or http://scampus.usc.edu) contains the University Student Conduct Code (see University Governance, Section 11.00), while the recommended sanctions are located in Appendix A.
Emergency Preparedness/Course Continuity in a Crisis
In case of a declared emergency if travel to campus is not feasible, USC executive leadership will announce an electronic way for instructors to teach students in their residence halls or homes using a combination of Blackboard, teleconferencing, and other technologies.
CS-499 Native Console Multiplayer Game Development Course Content

Week 1: Introduction and Course Overview. Linear Algebra, 3D Math, Physics
- Introduction, course overview
- Source control (Perforce) overview
- Trigonometry, Linear Algebra, 3D Math, Newtonian Physics
- Introduction to PrimeEngine framework
- Individual project 1 assigned (same for all students)
Read: Jason Gregory, Ch. 3 Fundamentals of Software Engineering for Games, Ch. 4.1-4.4 3D Math for Games

Week 2: Learning the Framework. Graphics overview
- Working with the PrimeEngine framework
- Using Maya as level editor
- Binding custom data to Maya objects
- Export and import of level objects
- Overview of python asset exporter
- Overview of graphics pipeline
Read: Jason Gregory, Ch. 7 The Game Loop and Real-Time Simulations

Week 3: Developing for Xbox360. Introduction to Sockets. Network stream interfaces
HW Due: Project 1
- Building code for Xbox360 and deploying the code to console
- Deploying assets to console
- SDK resources, where to look up code examples
- Introduction to sockets, Beej’s Networking Tutorials
- PrimeEngine network streams
- Students group into teams of 3-4 engineers and start working on their final game demo design
- Project 2 assigned
Read: Beej’s Networking Tutorial

Week 4: TRIBES Connection Manager. IPad development
- Students finish development schedule
- Maintaining connections
- UDP
- Introduction to XCode and IPad development
Read: Continue Beej’s Networking Tutorial, TRIBES Networking Model

Week 5: TRIBES Event Manager. Connecting XBox 360 to IPad
- Endianness differences
- Connecting devices to same server
- Overview of event manager
Read: TRIBES Networking Model (event manager)

Week 6: TRIBES Event Manager cont.
HW Due: Project 2
- Sending different events
- Event debugging/logging
- Network profiling
- Transmission records
- Project 3 assigned

**Read:** TRIBES Networking Model (event manager)

**Week 7: TRIBES Event Manager UDP, Introduction to Playstation 3**
- Playstation 3 examples, SDK
- Programming for Playstation 3
- Adding unreliable events over unreliable UDP

**Read:** TRIBES Networking Model (event manager), Playstation 3 Docs

**Week 8: Reliable UDP**
- Implementation of reliable events over unreliable UDP
- Sliding window protocols
- Artificial packet drops
- Packet send/receive throttling
- Acknowledgement packets
- Profiling of network with packet drops

**Read:** TRIBES Networking Model (event manager)

**Week 9: TRIBES Ghost manager**
- Maintaining state of game object
- Game object creation/destruction
- Late join object state/creation

**Read:** TRIBES Networking Model (ghost manager)

**Week 10: TRIBES Move Manager, Playstation Vita**
- Networking of game object movement through Move Manager
- Compression of data
- Transmission of animation data
- Playstation Vita examples, SDK

**Read:** TRIBES Networking Model (Move manager)

**Week 11: TRIBES Move Manager: Object Interpolation**

**HW Due: Project 3**
- Adding over-time interpolation of objects for smoothing
- Dealing with latency, packet drops for smooth object movement
- Playstation Vita programming
- Project 4 assigned

**Read:** TRIBES Networking Model (Move manager)

**Week 12: TRIBES Move Manager: Prediction/Correction**
- Adding prediction to object movement
- Adding correction to object movement
Read: TRIBES Networking Model (Move manager)

Week 13: Online: Game Server Browser
- Implementing server that maintains active server list
- Game statistics
- Registering games with server browser
- Dummy clients (bots) for testing server load

Week 14: Online: Matchmaking
- Automatically connect clients to servers
- Add dummy client connection/disconnection for matchmaking testing

Week 15: Final Game Presentations
HW Due: Project 4 Beta
- Projects presented on Consoles
- Projects reviewed for scheduling last fixes

Classes End – Finals Week: Project Gold Disc Presentation – Date and Time TBA
- Final presentations of projects with applied fixes