CS4344

Networked and Mobile Games
Ooi Wei Tsang
AS6 #0514
ooiwt@comp.nus.edu.sg
Monday
1200 - 1335-ish
What is CS4344 about?
What are the ingredients to building a computer game?
Game Rules
Art Work
UI and UX
Rendering and Animation
Artificial Intelligence
Physics
OS and Networking
Expected Weekly Workload

- Homework: 6 hours
- Lecture: 2 hours
- Preparation: 2 hours
Reading Materials
Algorithms and networking for computer games / Jouni Smed, Harri Hakonen.

Smed, Jouni.


Show Available

- Wiley Online Library E-books

AVAILABLE - CL Books - QA76.76 Com.Sm 2006 -

More Details

Descrip. xx, 264 p. : ill. ; 25 cm.

Bibliog. Includes bibliographical references (p. [247]-254) and index.

Note Online version restricted to NUS staff and students only through NUSNET. Also available in online version.

Subject Computer games -- Programming.

Computer algorithms.

Add Author Hakonen, Harri.

Add Corp Aut Wiley InterScience (Online service)

Add Title Algorithms and networking for computer games [electronic resource]

ISBN 0470018127 (cloth : alk. paper)
9780470018125
9780470029756 (Online)

Community Tags

Add a Tag
Research papers
Online articles
Take notes in class
Interact with me
Welcome to CS4344 (Sem 1 2012/13)

AUG 6TH  Posted by Ooi Wei Tsang  in Uncategorized  Edit

Thank you for your interest in taking this module.

You are looking at the main website for CS4344, which we will use throughout this class for discussion, announcement, and dissemination of course materials.

You may find old posts from the previous offering of CS4344 on this blog. Please feel free to look around. Some of the class materials this year will be updated, so not all will be relevant, but it should give you a good sense of what this class is about.

TEMCO KOEI Career Talk!

OCT 5TH  Posted by Ooi Wei Tsang  in Uncategorized  Edit

Venue: LT19
Date: 6 Oct 2010 (Wednesday)
Time: 6:30pm

http://blog.nus.edu.sg/cs4344/
Assumed Background
able to understand/evaluate/create **concurrent** programs
able to understand/evaluate/create networked programs
able to understand how HTTP, TCP, and UDP protocols work
able to use **Wireshark** to collect and analyze packet traces
able to pick up new programming languages and API
able to work effectively in a software engineering team
know how to use an source code version control system (we will use \texttt{git} and github)
also useful: passion about building games and some knowledge about how to build a good game
At the end of semester, you should be able to
understand the difficulties in building large-scale online multiplayer games and the current solutions
apply some of the solutions to build multiplayer games
analyze the trade offs made in different solutions and evaluate a given solution
collaborative coding

teamwork

presentation skills

HTML5/Javascript
General Structure of a Game
Example: minesweeper
Networked Games
Games that involve more than one host communicating over the network
Collect Events

Simulate Game

Render

Wait

Game States

events

states
Collect Events

Simulate Games

Game States

events

states

Wait

Render

Wait
Turn-Based Networked Games

Players take turn to make decision about their moves in the game
The All-New Yahoo! Chess

You talked, we listened! The All-New Yahoo! Chess has received a facelift based on your feedback, now with brighter colors, clearer game pieces, and easier game play options.

Looking for the standard Chess?
Our standard version of Yahoo! Chess is still available and includes tournaments, ladders, leagues, and more! Play Now!

what do you think?

We want to hear what you think about the All-New Yahoo! Chess. What do you like? What don't you like? What features would you like to see included in future upgrades to the game? We're counting on you to help us make Yahoo! Chess the best Chess on the Internet!
Collect Events

Simulate Games

Collect Events

Wait

Render

Game States

Wait

Render

Wait

Wait

events

states
events → Collect Events → Simulate Games → Game States → Update State → Render → Wait

Wait → Collect Events → Simulate Games → Game States → Update State → Render → Wait

delta → Game States

states → Game States
Centralized Architecture

smart server, dumb client
(game simulated in server only)
Centralized Architecture

dumb server, smart client
(game simulated at clients only)
Point-to-Point Architecture
Consistency Model
Consistency Model

P1
Game States

=  

P2
Game States

=  

Server
Game States
Turn-based Games with Server
Consistency Model

P1
Game States = P2
Game States = Server

(at the time when players need to make decision or see the results)
Turn-based Games with Server

P1

P2

server
Turn-based, Point-to-Point Games
Real-Time Networked Games

Players asynchronously make decision about their moves in the game.
Real-time Multi-player Games

I challenge you to the battlegrounds!

I challenge you to the battlegrounds!

MMORPG
First Person Shooter

A diagram showing the process of simulating games. The server is at the center, with arrows pointing to and from the game states. The process includes:

- Collect Events
- Simulate Games
- Game States
- Collect Events
- Wait
- Update State
- Render
- Wait
- Game States
- Game States
- Game States
- Game States
- Game States
- Game States
- Game States
Real-Time, Server-based Games

P1

server

P2
Real-Time, Point-to-Point Games

P1

P2
Issues for Designing Networked Games
Consistent

(players have the same view of the game state)
Responsive

(reacts to player’s action quickly)
Scalable

(supports many players)
Efficient

(minimal amount of traffic, CPU cycles, memory)
Fair

(no disadvantage to any players)
Cheat Proof

(can avoid/prevent cheating)
Project
to develop an awesome browser/mobile-based real-time multiplayer game using HTML5/Javascript in a team of 3-4
you can design your own game
(with simple 2D graphics)
Milestones

Week 3: proposal
Week 7: checkpoint 1
Week 10: checkpoint 2
Week 13: presentation
https://vimeo.com/79655960/
Assignment 1
what does game traffic look like?