ALMOST DONE

SEMESTER REVIEW (FRIDAY)

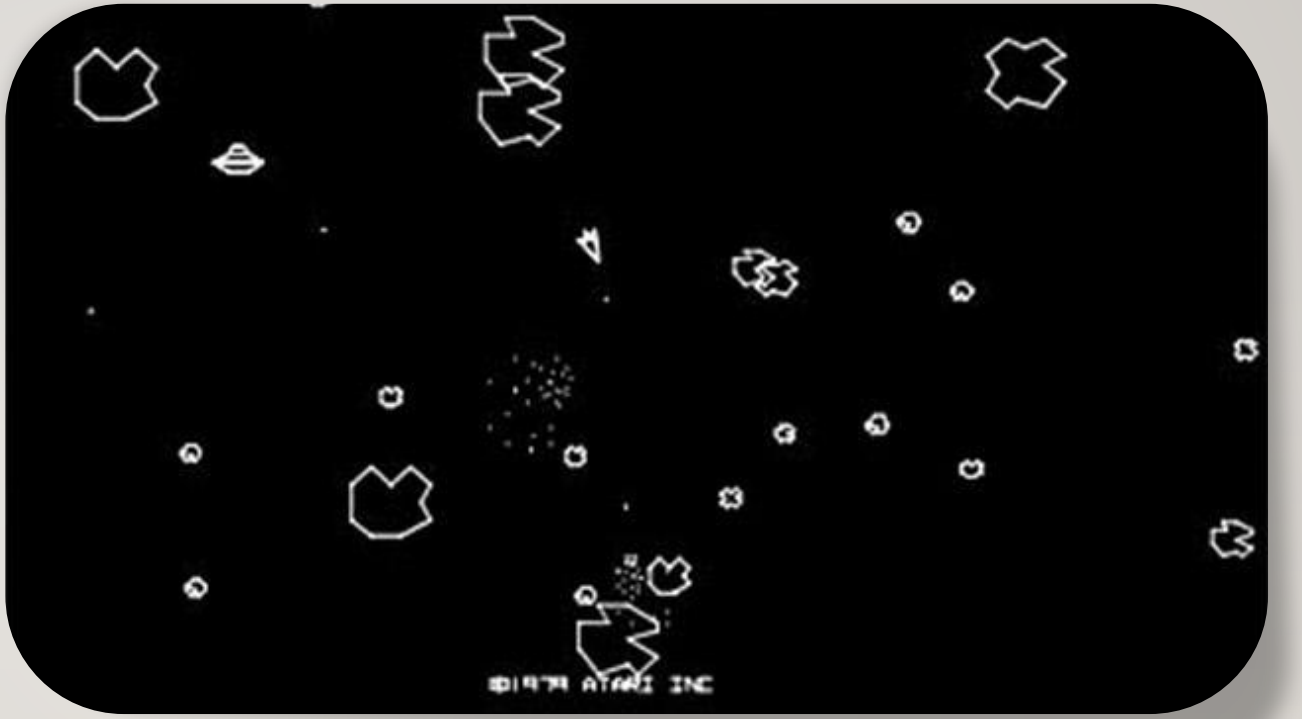
- Security stuff
- DFDs, attack trees
- Encryption
- Optimistic replication
- State replication
- SSL
- Berkeley API
- HTTP
  - Example HTTP usage

CS 261 TOPICS

- Game networking
- Real-time networked physics
- Running servers
- Optimize network code/procedures
- AWS
- Real-world debugging and interesting fixes
- More recent examples (Overwatch)
- MMO architecture
- Large-scale web and database architecture
- Game security

SEMESTER PROJECT: MASSIVELY MULTIPLAYER ASTEROIDS

- Login system (server + database)
- Inventory/catalog system (server + db)
- Cloud development
- State replication (dumb-client and optimistic)
- Networked physics
- Reliable UDP
- Prioritization and selective replication



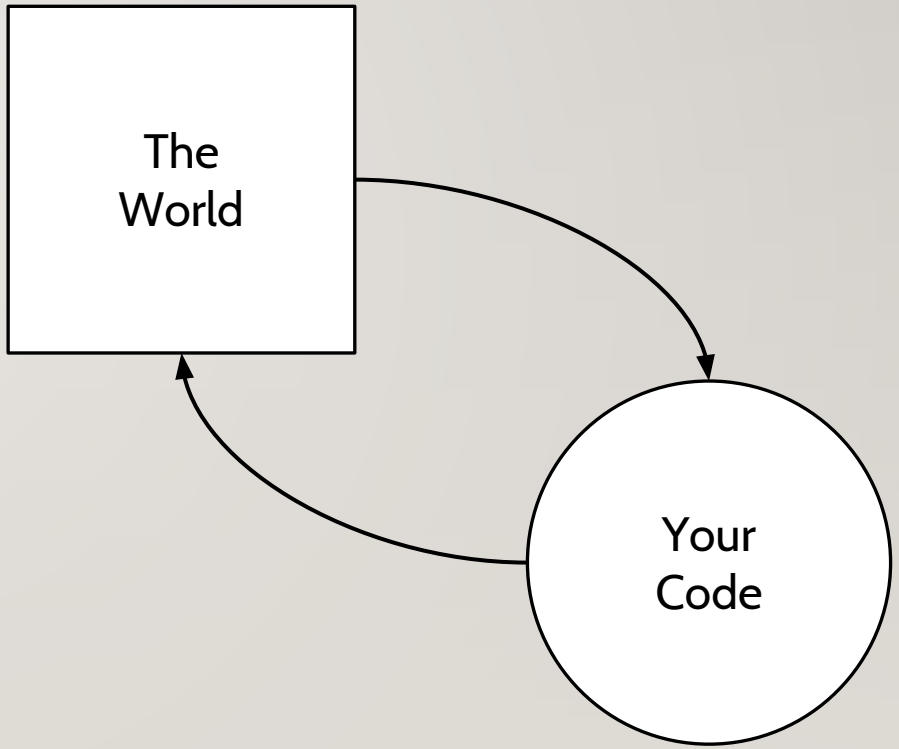
QUICK HITS

- Run a server at home
- How BitTorrent works
- How TOR works
- How Folding@Home works
- How satellite connections work

NETWORK PROGRAMMING = PROGRAMMING + DUST MANAGEMENT

DUST MANAGEMENT: DANGER

- Create the simplest possible DFD
- STRIDE x Elements
  - Can this message be *spoofed*?
  - What happens if it's *tampered* with?
  - Do I care about *repudiation*?
  - Does this message contain *secrets*?
  - What prevents a *DoS* attack?
  - Am I guarding against *elevation*?



DUST MANAGEMENT: UNCERTAINTY

- Each node of your system has an incomplete view of the total state.
- When will you prioritize *consistency*, and when *availability*?
- How do you know the state hasn't changed?
- How do you know other nodes are aware of your changes?
- How do you know whether your state is complete?
- How do you know other nodes exist and are running?

DUST MANAGEMENT: SCALE

- Two facets of scale:
  - State: Your system may process too much data for a single node to handle
  - Users: Your system may be serving too many users for a single node to handle
- How will you distribute your state among the nodes?
  - Issues of danger (STRIDE), uncertainty (CAP), and time (efficiency)
- How will you allocate users among the nodes?
- How will you handle unbounded data?
- How will you handle surprise surges in usage?

DUST MANAGEMENT: TIME

- Two facets of time:
  - Tasks take time to complete, and the system needs to be responsive and productive while waiting
  - Tasks conclude at unpredictable moments, and the system needs to receive the new data without disrupting its operation
- How can you minimize waiting?
- How can you give the system productive work while waiting?
- What happens when the system receives an update in the middle of some other task?

HOW CAN I RUN A SERVER AT HOME?

- Danger?
- Uncertainty?
- Scale?
- Time?

HOW DOES BITTORRENT WORK?

- Danger?
- Uncertainty?
- Scale?
- Time?

HOW DOES FOLDING@HOME WORK?

- Danger?
- Uncertainty?
- Scale?
- Time?

HOW DOES TOR (THE ONION ROUTER) WORK?

- Danger?
- Uncertainty?
- Scale?
- Time?

HOW DOES SATELLITE INTERNET WORK?

- Danger?
- Uncertainty?
- Scale?
- Time?