

# CSSE 220

Project Workday

Checkout *EventBasedGame* project from SVN

# Objectives

- Learn how to avoid common project errors
- Learn how to setup a timer to create events
- Live-code examples of these
- Practice interaction between game elements
- Think about applying this material to project
  - 2<sup>nd</sup> hour work time for project
  - project adviser will visit

# Concurrent Modification Exceptions

- Happens when you try to change something that is being used
- Example:
  - Removing something from a list while the list is itself being iterated through (drawing/updating)
- Why?

# Threads

- Computers can run more than one thread of execution at the same time
  - Even single core processor can simulate this by timesharing (more about this in future courses)
- Main starts every Java program
- Graphics start a *separate* thread
- Multiple threads can create very painful problems
  - Can be hard to debug (race conditions)
  - Example include ConcurrentModificationExceptions

# Timer

- An alternative to a thread that waits/sleeps
- Creates events periodically
- Allows main thread to finish
- Designed to work in same thread as graphics
- Superior approach for ArcadeGame project

# Live-coding

# Reminders

1. Use your UML design when you code!
  - That's why you did it!
2. ...but DON'T let it dictate the order that you code
  - Instead always do the next thing you can TEST.
  - Start with a Level then a Hero and whatever you need to make them display.
  - There is ZERO benefit to creating empty Monster classes at this point!

Work time

*Be sure everyone is getting a chance to drive.*

# **TEAM PROJECT**