

# CSE 125 Boot Camp

Or: How I Learned to Stop Worrying and Love The Lab

# About Me

- Game Developer since ~~2010~~ *forever*
- Founder and President of VGDC
  - [gamedev.ucsd.edu](http://gamedev.ucsd.edu) (shameless self-promotion ftw)

- Favorite game: 

[alex.ferbrache@gmail.com](mailto:alex.ferbrache@gmail.com)

619.602.3606 (call me maybe?)



I look like this



What the



#@%&

did I get myself in to?

- One of the most work-intensive undergraduate CSE courses
- More work than a genius AI can accomplish alone in 10 weeks
  - You're human - don't even try.
- B220 is beautiful, and you'll soon learn to see it
  - This could be considered Stockholm Syndrome
- No starter code, no hand-holding, Final Destination

# Please Tell Me There's a Bright side

There's a bright side! You get to make...

VIDEO GAMES!!!! 1

# ...Still Waiting for the Bright Side

Okay, real talk. CSE 125 gives you:

- Creative and academic freedom
- Real-world experience
- Marketable skills
- Job opportunities

And most importantly...

**UNLIMITED BRAGGING RIGHTS**

# Okay, You Convinced Me. Where Do I Start?

- It all starts with a good idea
  - Pick something fun!
  - Make sure everyone is in love like with the idea
  - Keep your expectations reasonable
- Think about **gameplay**  
**presentation**  
**ease of development**



# There's no 'I' in Teamwork (but there is 'work')

- Two folks on each of the following:
  - Graphics
  - Networking
  - Gameplay
- Pair programming is a thing. A great thing
- Split up your tasks so everyone is busy

# Management. Management Never Changes

- Elect a group leader from the get-go. This person is now the game's producer in addition to their other roles.
  - Producers keep the schedule and crack the whip
- Have someone do full-time design work for the first week
  - Get ideas hashed out and get a feel for the gameplay
  - Put together a design document above and beyond the required documentation



# Schedules. Schedules Usually Change

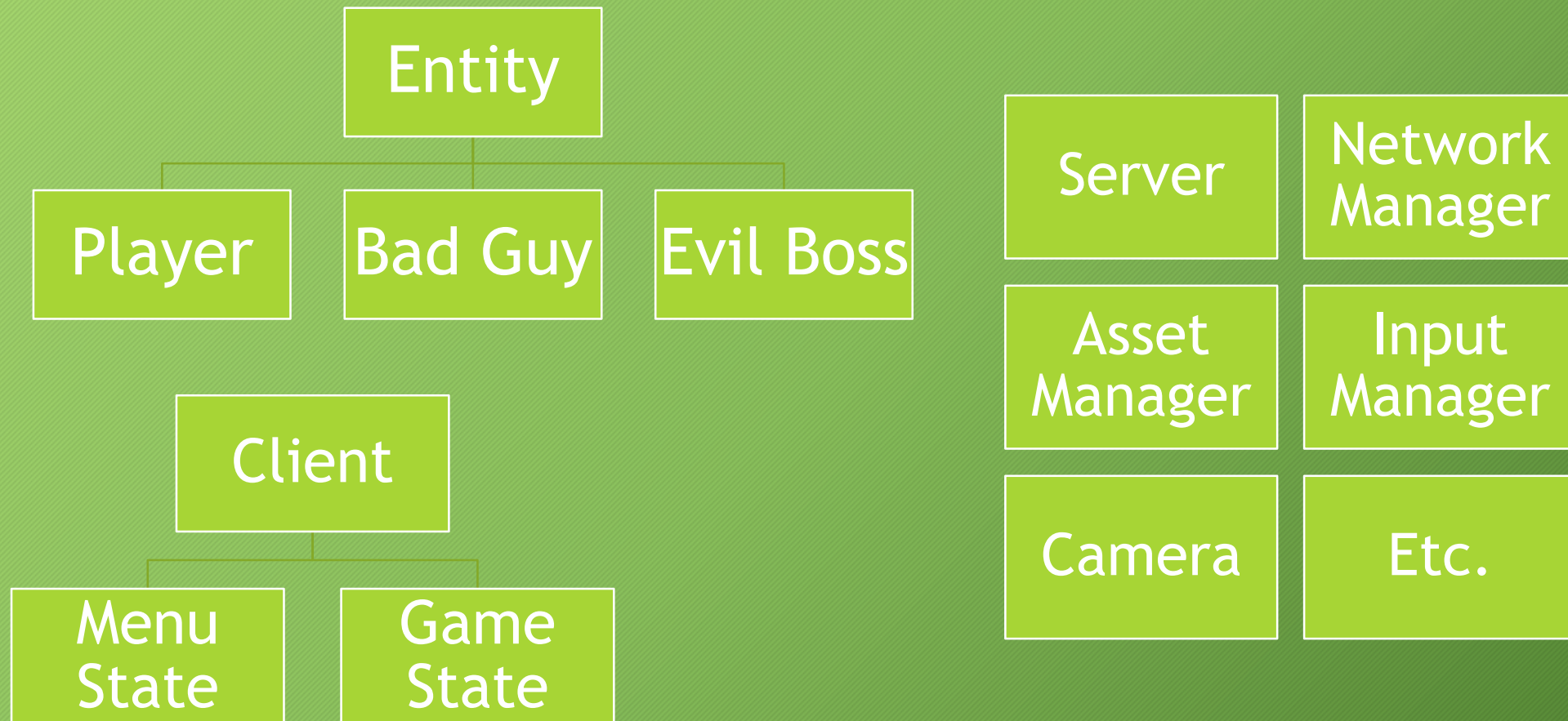
- Have a working prototype done AS SOON AS POSSIBLE
  - Play your game, make sure it's fun
  - More time to polish graphics and add features
- Get packets flying fast
  - Once it's done, the networking component won't need much maintenance
  - Network folks can move on to other tasks
- Take things week-by-week (iterative development)

# An Absurdly Simple Game

Don't know the first thing about programming games? Here you go:

```
while Geoff.isCool() { // Always
    update(); // Update the state of the game
    draw(); // Draw the new state to the screen
}
```

# Something A Bit More Complex



# How Do I Graphics?

## DirectX

- Windows-only
  - C++ or C#
- Game Industry Standard
- More samples & support
- Better shaders = better shininess
- More than just a graphics library

## OpenGL

- Cross-platform
- Wrappers for almost any language
  - Even Fortran. Yes seriously.
- You probably remember it from 167
- Becoming more common in mobile
- Linux & Mac fanbois will love you

# I Have Further Graphics Questions

Ask this guy



# What does “Out Of Memory Exception” Mean?

- Memory leaks are a **big** deal
  - Could run out of memory and crash during the demo
- Front-load your memory allocation, and don't allocate/deallocate during the game.
- Keep instantiated objects in a stack, pull them off when needed and put them back when you're done
- AVOID THE **new** KEYWORD LIKE THE PLAGUE

# Convenience is Convenient

- Recompiling your project to change how fast something moves is an **absurd waste of time**
  - Spend some time writing a system to read values from a config file
  - While you're at it, put in a 'mute' flag
- Debugging a real-time networked program with breakpoints or print statements is really hard
  - Do yourself a favor and code up an in-game debug console

# For Lack of a Better Title: Audio

- Lots of audio libraries for the picking:
  - DirectSound
  - SDL
  - OpenAL
  - FMOD
- Free sound effects on the web



# Git 'er Done!

- Use source control!
  - We have SVN repos for you on pisa
  - Bitbucket provides free private Git and Mercurial hosting
    - Github does too if you register a student account
- Git is an amazing tool (smarter merging than SVN)
  - *Really* bad for large binary assets (like, I don't know, models and textures?)
  - Git for code, Dropbox for models and textures

# This Slide is for Artists

- I'll hold a two-part crash-course in 3D Studio Max in the next two weeks
  - Part 1 - Modeling
  - Part 2 - Texturing
- Art Resources:
  - [ConceptArt.org](http://ConceptArt.org)
  - [3DBuzz.com](http://3DBuzz.com)
  - [3DTotal.com](http://3DTotal.com)

# Bullet Points That You May Find Helpful

- Use vertex buffers for speed
- Use a particle system
- Pass by reference
- Functional programming limits side-effects
- Draw attention to hacky things for future fixin' fun
- F12 key in VS takes you to definition
- Commit often
- Use a CMS for your website
- Code warning-free

# The Journey Has Only Begun

Get help from Geoff, myself, Ryan, or other teams

GOOD LUCK!!!!