CSE 125 Survival

Please hold tomatoes til the end.
About me
About me

Nobody cares, let's continue
About me

Andrew Huynh
1st year PhD student
Machine Learning/Comp. Vis.
a5huynh@cs.ucsd.edu

Ultra Capture Siege Dungeon

Originally a tower defense game using a multitouch device

I did: Bug Hunter + Graphics + Level Design + Game Logic
Where to start?

It all starts with an idea. So pick a good one.

Be creative, go crazy!

Make sure everyone is comfortable with the ideas.
Where to start?

It all starts with an idea. So pick a good one.

Be creative, go crazy!

Make sure everyone is comfortable with the ideas.
Where to start?

It all starts with an idea. So pick a good one.

Be creative, go crazy!

Make sure everyone is comfortable with the ideas.
Where to start?

It all starts with an idea. So pick a good one.

Be creative, go crazy!

Make sure everyone is comfortable with the
But really

Think about gameplay

Think about presentability

Think about ease of development

Don’t expect everything to go your way =[ 
Teamwork

Split up tasks evenly so no one is idle

Have someone whose main job is:

- logic
- graphics
- input
- sound
- animation
- GUI
- network
Teamwork

Split up tasks evenly so no one is idle

Have someone whose main job is:

logic  graphics  input  sound

animation  GUI  network

COMMUNICATION
Teamwork

Have hard deadlines for each major part

And of course, actually play your game!

Some tips to improve team dynamics

hang out with your team
Teamwork

Also talk with other teams!

They’ll probably run into similar problems!

Common themes in games

Usually

This year
Teamwork

Also talk with other teams!

They’ll probably run into similar problems!

Common themes in games

Usually

This year
Teamwork

Also talk with other teams!

They’ll probably run into similar problems!

Common themes in games

Usually

This year
Some ideas
Some ideas
Some ideas

Mutated fruits rebel against human overlords
Some ideas

Mutated fruits rebel against human overlords
Some ideas

Mutated fruits rebel against human overlords

Professor Voelker fights back zombie hoards of grad students with research papers
Dev time!
More fun than a barrel full of monkeys
DirectX vs OpenGL

DirectX

More than just graphics (Direct3D)

Restricted to Microsoft platforms

Learning curve (left-hand coordinate system, matrix multiplication order, texture addressing order)

More samples, more supported media formats

Flexible shader management system (may be easier to create cool effects)
DirectX vs OpenGL

OpenGL

Likely more familiar to you, and cross-platform

May have less support from other teammates

My recommendation
DirectX vs OpenGL

OpenGL

Likely more familiar to you, and cross-platform

May have less support from other teammates

My recommendation

Assembly. If machines use it, it must be good.
DirectX vs OpenGL

OpenGL

Likely more familiar to you, and cross-platform

May have less support from other teammates

My recommendation

Assembly. If machines use it, it must be good.

Blazing fast speeds. No overhead.
Particle Effects

Particle effects are useful almost everywhere

- Fire for torches
- Smoke
- Explosions
- Confetti

Really makes your game stand out. Seems more polished.
Particle Effects

Particle effects are useful almost everywhere

- Fire for torches
- Smoke
- Explosions
- Confetti

Really makes your game stand out. Seems more polished.

We used particle effects to create a sparkly unicorn
Audio?

Already tons of libraries out there. DirectSound, SDL, OpenAL, FMOD

Our team used FMOD and was able to have 3D sounds (fades in as you get closer, fades out as you move away)

Lots of good, free, generic sound effects on the web.
Audio?

Already tons of libraries out there. DirectSound, SDL, OpenAL, FMOD

Our team used FMOD and was able to have 3D sounds (fades in as you get closer, fades out as you move away)

Lots of good, free, generic sound effects on the web.

I'm also available to do voiceovers.
Iterative Development

Start off with the basics and work your way up!

- Render a simple primitive
- Add a controllable camera
- Load your own models
- Play music, etc, etc

This method works with animation, graphics, networks, everything! Keyword: Baby steps
Memory Management

Memory leaks are a big deal

Can easily run out of memory and crash game

Try and code the deallocation in parallel with allocation
  - Destructors for classes for example

Alternatively you can wing it and hope that in 15 minutes you don’t run out of memory

Alternative alternatively, make restarting the game... part of the game!
Asset Management

Try and store constants for tweakable values in a separate file. You don’t have to recompile just to change how fast something is moving.

XML is a nice way to do this. TinyXML is super easy to use.

Protip: Have a hotkey to reload config files on the fly!
Code Tips

Please comment and use descriptive names.

Place comments around hacky areas or places that need work.

Tabs or spaces? Where to put curly braces?
Drives people like me (aka, lunatics) crazy, so keep that in mind.

Minimize compilation dependencies! You don’t want to sit around all day waiting for your code to compile.
Code Tips

Use your code repo! Commit often! Code, models, everything!

Use svn, git, hg, whatever. Just use something

svn revert is a godsend.

I usually like to commit after every “big” change

- Things like adding a camera system
- Successfully loading a model
- Individual bug fixes

Always remember to commit ( if needed ) before you leave as well
General Dev Tips

Compiler warnings are there for a reason.

Debug console/printfs anything to quickly understand what and where a crash happened

Google

Your best buddy

Sometimes I’m helpful too
Fin

Questions?

Comments?

Criticisms?

Taunts?