Your 1st, 2nd, 3rd, and Nth Game Engine

Sean Middleditch

DigiPen Game Engine Architecture Club

Sean Middleditch
http://seanmiddleditch.com/

Who I Am

- DigiPen RTIS Senior
- Subsonic, SONAR, Core
- Too many years experience
- Total architecture nerd

Sean Middleditch
http://seanmiddleditch.com/

This Lecture

- Starting from scratch
- Getting a game done
- Consistent strategy to success in games

Sean Middleditch
http://seanmiddleditch.com/

What is a Game Engine?

- "*A game engine is a software system designed for the creation and development of video games.*" – Wikipedia

- I don't like this definition
- A game engine is the framework underneath any game
  - Rarely is it something "designed for game***s***"

Sean Middleditch
http://seanmiddleditch.com/

What is Software Architecture?

- "*The software architecture discipline is centered on the idea of reducing complexity through abstraction and separation of concerns*." – Wikipedia

Sean Middleditch
http://seanmiddleditch.com/

What is Software Architecture?

- "*The software architecture discipline is centered on the idea of reducing complexity through abstraction and separation of concerns*." – Wikipedia

- Fancy way of saying:

"Make it as simple as possible... but no simpler."

Sean Middleditch
http://seanmiddleditch.com/

What is Software Architecture?

- "*The software architecture discipline is centered on the idea of reducing complexity through abstraction and separation of concerns*." – Wikipedia

- It is ***not*** about adding tons of cool features or complex algorithms.

Sean Middleditch
http://seanmiddleditch.com/

Write Games, Not Engines

http://scientificninja.com/blog/write-games-not-engines

- "*You can sit down and write a game without writing a pre-written engine, and in fact this is very often the better approach, regardless of why you want to write an engine.*" – Josh Petrie

- Everyone should read this

Sean Middleditch
http://seanmiddleditch.com/

Look For The Bare Necessities

- You don't need to use all the cool tricks we talk about
  - Components aren't necessary
  - Metadata isn't necessary
  - Scripting isn't necessary
  - Editors aren't necessary
  - Threading isn't necessary
  - Art pipelines aren't necessary

Sean Middleditch
http://seanmiddleditch.com/

The Simple Bare Necessities

- You need a game
  - Window with graphics
  - Collect input
  - Run game logic
  - Menus, intros, help, win/lose screen

Sean Middleditch
http://seanmiddleditch.com/

Keep It Simple, Superdev

- You have a game idea
  - Write only what you need for that game idea
- Be lazy
  - Write the least code you can get away with
- Be smart
  - Write the most useful code first

Sean Middleditch
http://seanmiddleditch.com/

Getting Started

- You need a game
  - Window with graphics
  - Collect input
  - Run game logic
  - Menus, intros, help, win/lose screen
- Create a window
- Draw a triangle/quad

Sean Middleditch
http://seanmiddleditch.com/

Getting Started

- You need a game
  - Window with graphics
  - Collect input
  - Run game logic
  - Menus, intros, help, win/lose screen
- Read Win32 message loop input messages
- Move the tri/quad around with arrow keys

Sean Middleditch
http://seanmiddleditch.com/

Getting Started

- You need a game
  - Window with graphics
  - Collect input
  - Run game logic
  - Menus, intros, help, win/lose screen
- Detect when “player” hits another tri/quad
- Make another tri/quad chase player

Sean Middleditch
http://seanmiddleditch.com/

Getting Started

- You need a game
  - Window with graphics
  - Collect input
  - Run game logic
  - Menus, intros, help, win/lose screen
- Necessary TCR stuff
- Just get it working for now

Sean Middleditch
http://seanmiddleditch.com/

Status Check

- A game has been born
- Yes, it's horrible, ugly, and probably not too fun
- But it *works*
- Foundation to build on

Sean Middleditch
http://seanmiddleditch.com/

Building Up

- Identify tools that make life easier
  - Move repeated code to functions/objects
  - Abstract complex operations into simple ones
- Refactor
  - The initial code is probably “bad”
  - Clean it up if (and only if) it's being a problem

Sean Middleditch
http://seanmiddleditch.com/

Case Study

- Proof of concept that you can play with
- I made a game in roughly 8 hours
- The source is available
- The game is playable and kinda fun
- Here's how I did it

Sean Middleditch
http://seanmiddleditch.com/

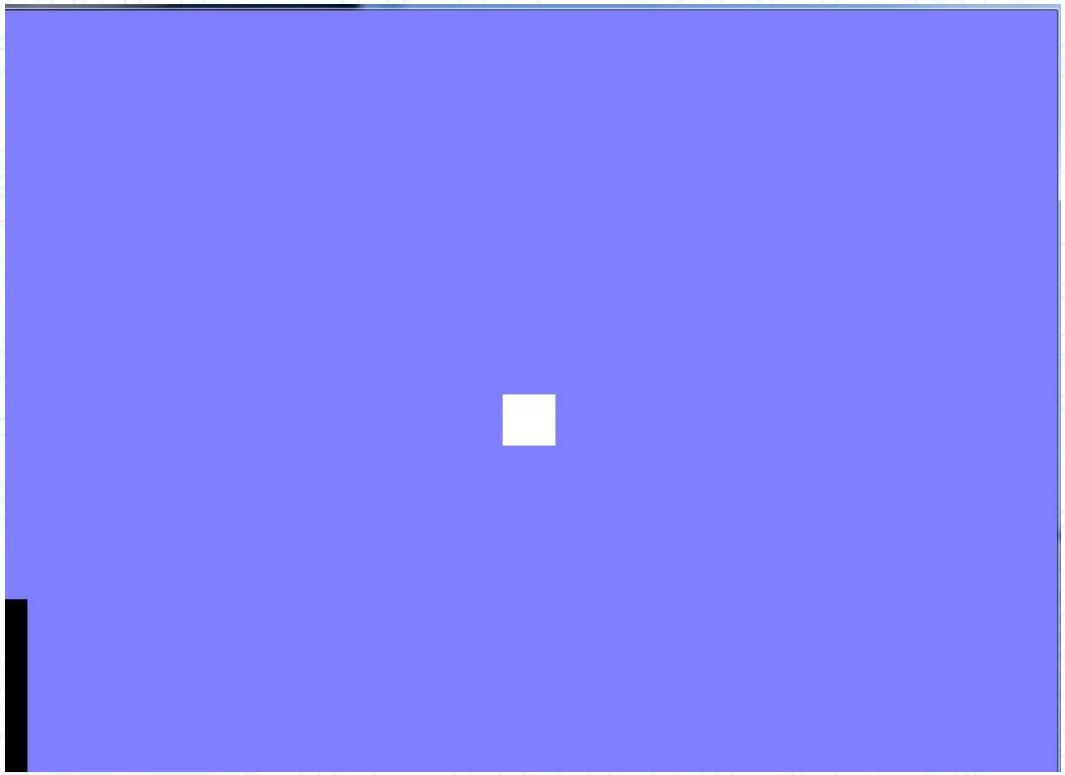
Bad-ass Bubble Battle Boid Bots Bonanza

- 1 – 4 players
- PvP and PvE
- Shield for defense and offense
- Flocking enemies (which can turn into projectiles)
- Walls, pits, survival
- Totally bad-ass (in a quiet, soundless kind of way)
- This is my Nth game ~~engine~~

Your 1st, 2nd, 3rd,
and Nth Game Engine

Sean Middleditch
http://seanmiddleditch.com/

Hour 1



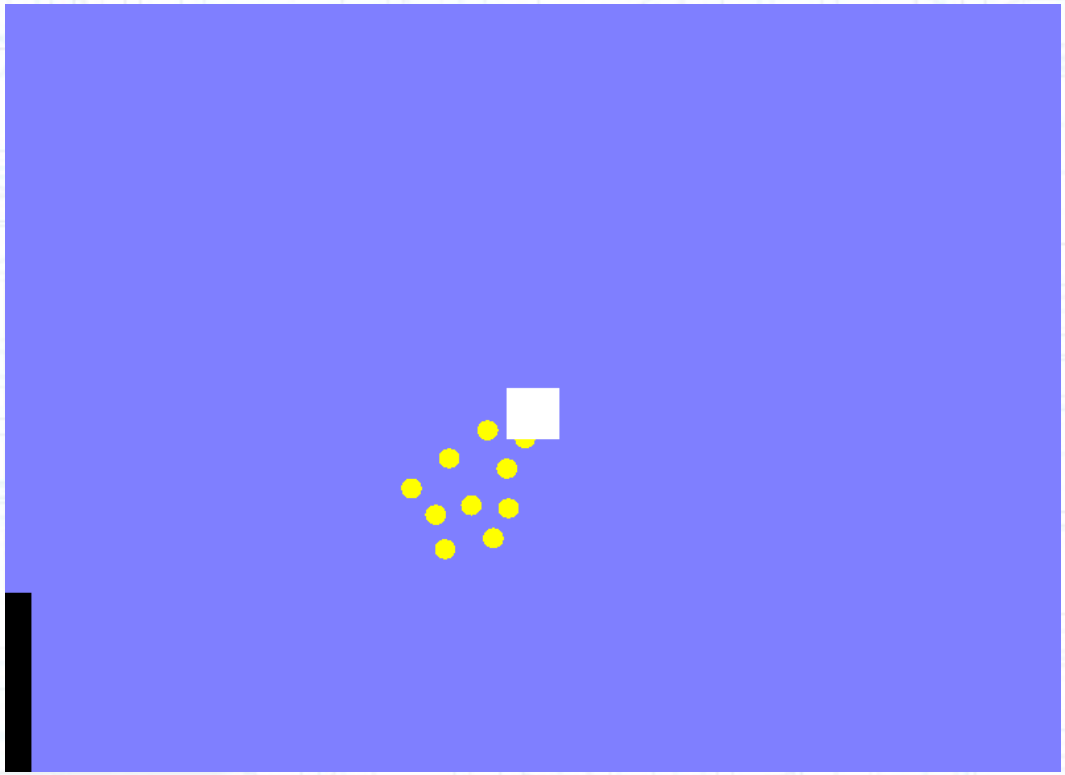
Sean Middleditch
http://seanmiddleditch.com/

Hour 1

- Visual Studio setup and boilerplate work
- OpenGL + GLUT main window
- Very simple inheritance-based GameObject
- Wall object (white square) and Pit object (black rectangle)
- Close window on Escape key

Sean Middleditch
http://seanmiddleditch.com/

Hour 2



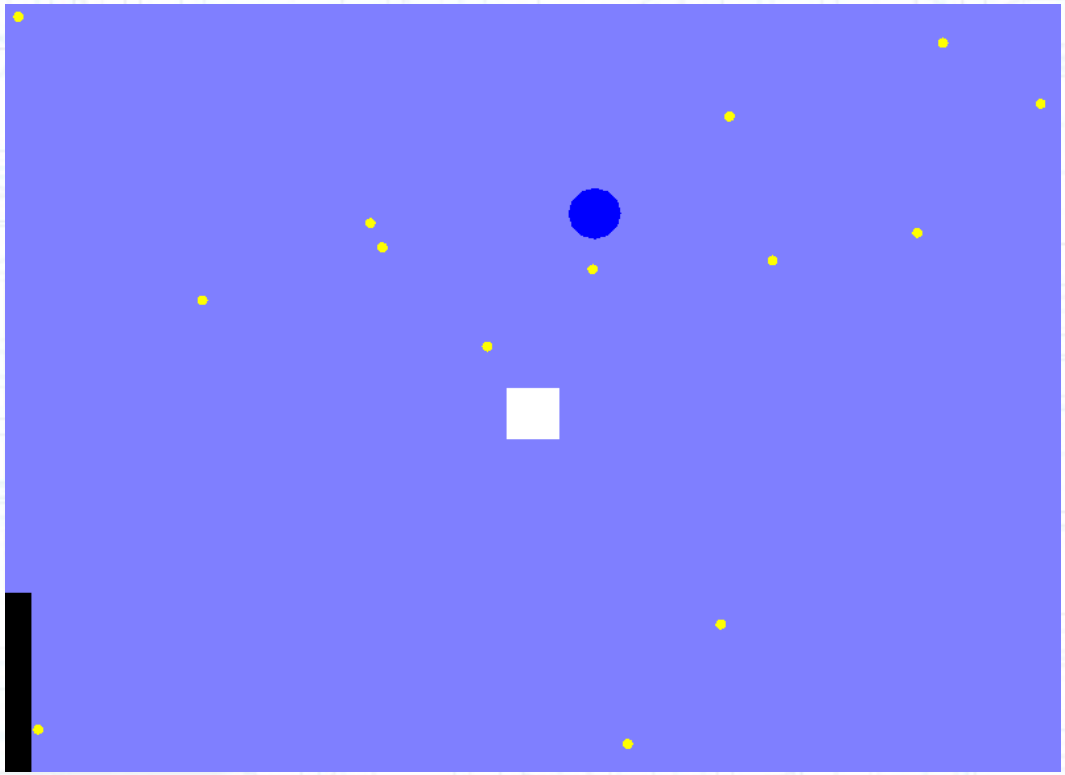
Sean Middleditch
http://seanmiddleditch.com/

Hour 2

- Added Boid object
- Implemented simple (buggy) flocking
- Added Vec2 math library, a few other support functions

Sean Middleditch
http://seanmiddleditch.com/

Hour 3



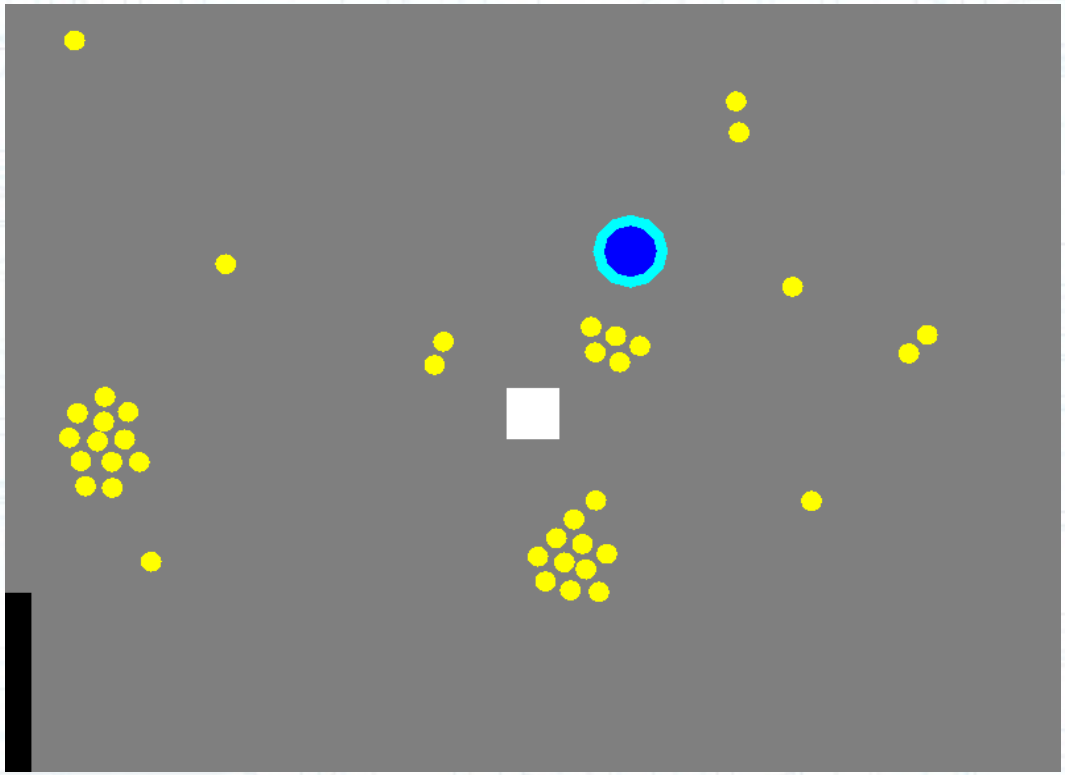
Sean Middleditch
http://seanmiddleditch.com/

Hour 3

- Added Player object
- Super simple Xinput handling for gamepad
- Introduced a severe bug with flocking (oops)

Sean Middleditch
http://seanmiddleditch.com/

Hour 4



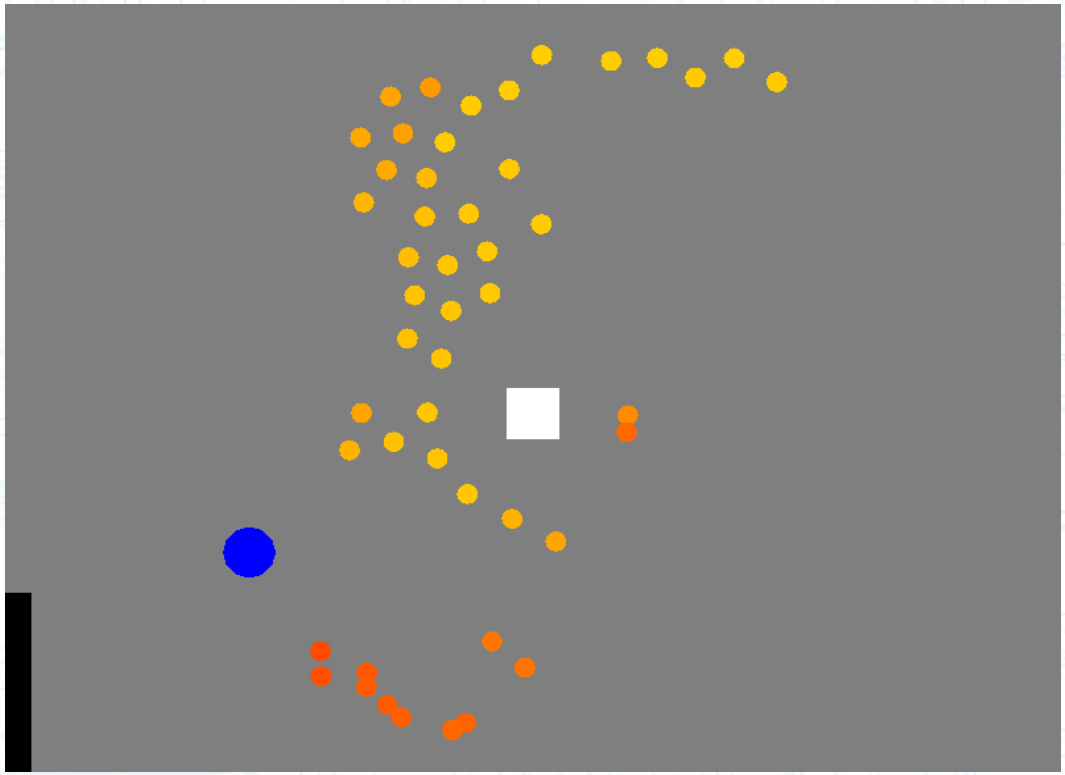
Sean Middleditch
http://seanmiddleditch.com/

Hour 4

- Fixed flocking and made algorithm work nicely
- Added first attempt at player shield mechanic
- Some (broken) physics is implemented

Sean Middleditch
http://seanmiddleditch.com/

Hour 5



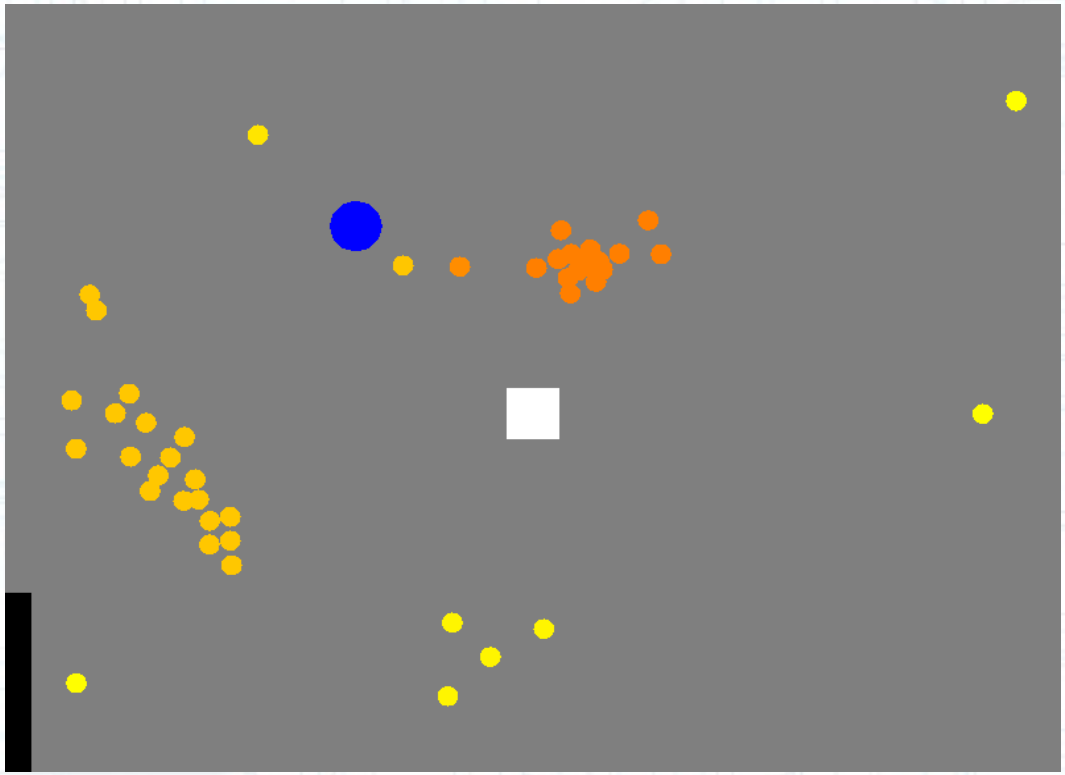
Sean Middleditch
http://seanmiddleditch.com/

Hour 5

- Implemented Boid anger
  - Get redder over time
  - Chase player when angry enough
- Graphics finalized, no more iteration there

Sean Middleditch
http://seanmiddleditch.com/

Hour 6



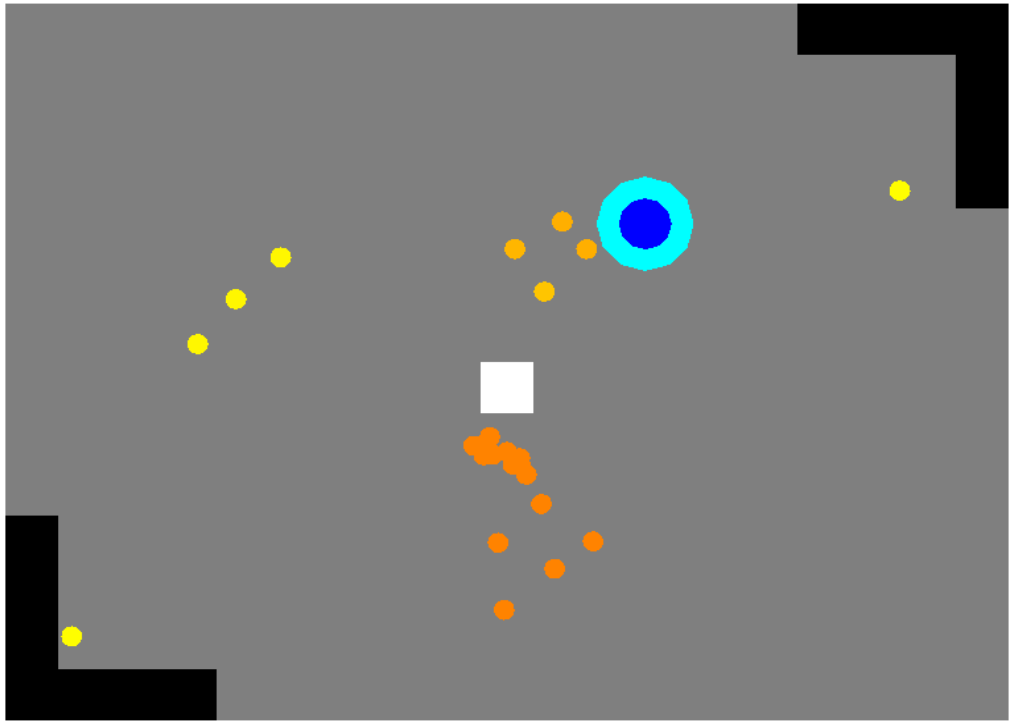
Sean Middleditch
http://seanmiddleditch.com/

Hour 6

- Tons of physics work
- Shield can bounce you off walls and bounce Boids
- By far the “hardest” part of the game for me

Sean Middleditch
http://seanmiddleditch.com/

Hour 7



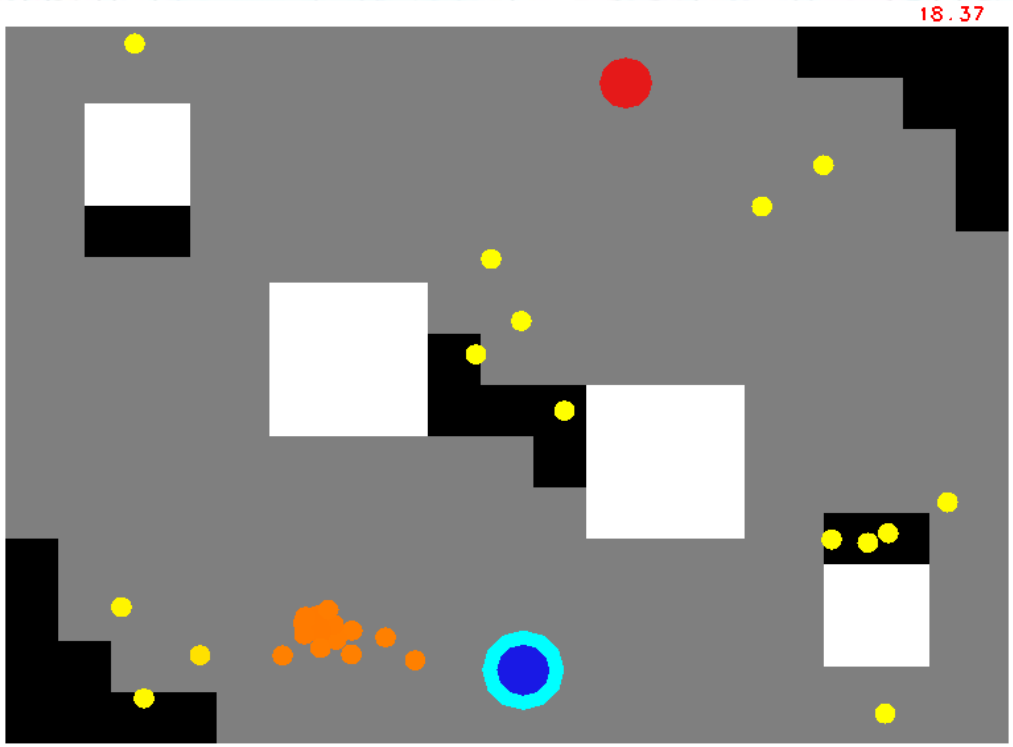
Sean Middleditch
http://seanmiddleditch.com/

Hour 7

- Bigger pits (you still can't fall in them)
- Boids spawn over time from pits
- … more physics fixes and tweaks
- … more Boid behavior tweaking
- Implemented 2+ players

Sean Middleditch
http://seanmiddleditch.com/

Hour 8



Sean Middleditch
http://seanmiddleditch.com/

Hour 8

- Scaled boxes for walls
- Pits kill you if you aren't moving fast enough
- Ridiculously simple (and bad) menus
- Other tweaks and fixes

Sean Middleditch
http://seanmiddleditch.com/

8 Hour Game Summary

- Still has some bugs and needs more tweaks to be "awesome fun"
- In pretty good shape for only 8 hours
- Tiny, simple code will make it easy to iterate further
- Imagine what a whole team of four DigiPen students with a whole weekend could do!

- http://github.com/seanmiddleditch/BubbleBattleBoids

Sean Middleditch
http://seanmiddleditch.com/

Demo Time!

Visual Studio... *Engage!*

Sean Middleditch
http://seanmiddleditch.com/

Bubble Battle Boids Post-Mortem

- Less time to write than my last game object library
  - More fun to write, too
- Most bugs were me failing at simple math
- Still having fun hacking away at it
  - Will post more GitHub updates eventually
- Lots of gameplay bugs and needs tweaks, but the core game idea seems solid and entertaining
  - A great prototype if nothing else

Sean Middleditch
http://seanmiddleditch.com/

The Next Step

- There can only be a “next” step after taking the first
- Think about game features first
  - Power-ups? Sound?
  - Random levels?
- The engine grows only when I need it to
  - More object types → components
  - More intricate and reliable physics
- Not even thinking about complexity it'll never need

Sean Middleditch
http://seanmiddleditch.com/

DigiPen Game 200/300/400

- Aside from the menus/TCR issues, this game will almost certainly pass Game 200
- Juniors would need better graphics and less physics bugs, but otherwise... same deal
- Seniors need to concentrate even more heavily and visual pizazz (and not at all on internals that potential employers can't see)

Sean Middleditch
http://seanmiddleditch.com/

But... Architecture is Fun!

- Some of us (me) really love coding
  - We want to solve interesting problems
  - Learn new techniques
  - Do awesome stuff with code
- Learning advanced techniques is important
  - Gain proficiency and expand code toolbox
- Refactor your game *after* it's working and playable
  - Rest of the semester to ~~goof off~~ experiment

Sean Middleditch
http://seanmiddleditch.com/

Exceptional Games

- Some games need some particular tech to work
- Nitronic Rush's first milestone
  - 3D car driving around on a flat plane
  - Quick and dirty proof of concept
  - Up and running first, crazy-awesome after

Sean Middleditch
http://seanmiddleditch.com/

AAA Titles

- Big AAA titles are quick and dirty, too
- They still prototype games first
  - Use “old” engine or licensed engine
  - UDK, CryEngine, Unity, Source, etc.
- Saves time by having tons of complex features
- Opposite situation to DigiPen students
  - Save time by not even thinking about complex features

Sean Middleditch
http://seanmiddleditch.com/

AAA Engines

- Poorly kept secret:
  - There is no perfect universal engine
  - They all suck
- They all grew organically out of games they were hacked together for
  - Just like your engines could (and should) be

Sean Middleditch
http://seanmiddleditch.com/

Summary

- Make games, not engines
- Write what you need and nothing more
- Quick and dirty first, pretty and cool after
- Prototypes can still be fun (and pass your game class)
- Engines come from games, not the other way around

Sean Middleditch
http://seanmiddleditch.com/

Thank You

Questions? Comments? Praise and Adoration?