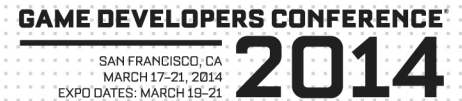# Extracurricular Videogame Development Clubs
How They Complement the Classroom Experience

Chris DeLeon
Developer & Instructor, HobbyGameDev.com

What I'm Talking About

- How clubs differ from the classroom
- Regarding faculty involvement
- Practical tips to grow more gamedev clubs
- Notes from other student-led clubs

Clubs established by my process

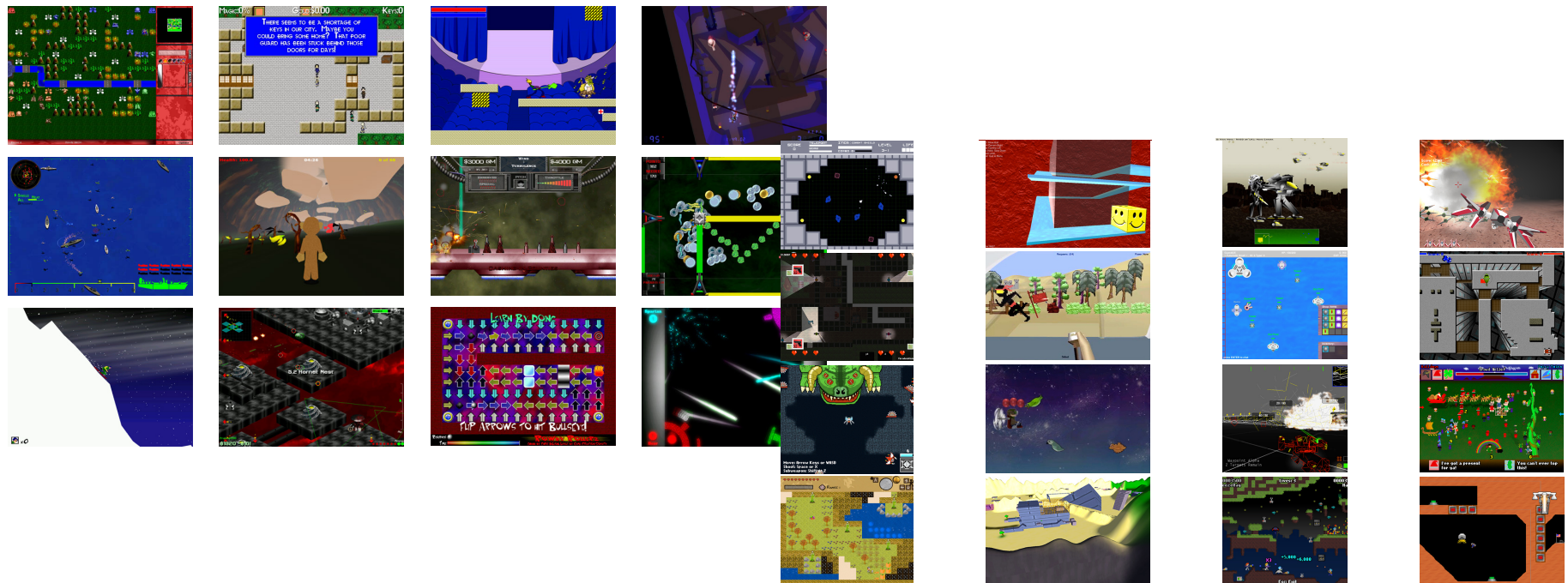- Carnegie Mellon, Game Creation Society (2004)
- Georgia Tech, VGDev (2010)

Both groups: ~30-50 developers/semester
Each completing 5-8 games/semester since
Both clubs are *mostly undergraduates*
Very few members from ETC or DM program

The process -> over 100 games

Addl. reference clubs (interviews)

1. **Ithaca College**: Game Developers Club
2. **Southern Alabama**: Video Game Development Club
3. **Bloomfield**: Game Dev Club
4. **Georgia State**: Game Development Club
5. **Champlain**: Extracurricular Game Developers
6. **Princeton**: Game Developers Group
7. **Tennessee State**: Game Programming Club
8. **SCAD**: Game Development Network
9. **Carnegie Mellon**: Game Creation Society (2009-2013)

We all want the same things

- We want students happy
- We want students creating
- We want students learning
- We want students networking
- We want students on a good track

But we do it very different ways

Classes (generally, not all!):	Clubs (esp. CMU/GT):

• Often artistic, conceptual
• Focus on process
• Failing is literally failing
• Compel full participation
    • People need the hours
• Teaching to specialize
• Challenges of siloing
• Disdain for industry (!?)
	• Mostly conventional genres
	• Focus on finished result
	• Safety and freedom to fail
	• Variable involvement ('varsity')
	    • People can quit, be fired
	• Encourage skill branching out
	• Can be fully cross-disciplinary
	• Emulate, learn from industry

On faculty - including other clubs

- Often a faculty sponsor, mostly on-paper
- "Hands off" (4X) meetings and games
  - Please don't try to absorb a club into your lab!
- "I don't know if any faculty/staff know we exist…" / "…we get nearly no support from our faculty." / "Basically not at all."
- Some helped with room/event scheduling, funds, school web hosting, speakers, or club visibility

https://xkcd.com/616/

Clubs are a great *outlet*!

- Don't want projectile action in class?
- Don't want tired sci-fi/fantasy in class?
- Many of your students want to try these.
- We let them see it's harder than it looks
- In many cases it helps them move on

http://apassportaffair.com/2013/11/01/travelish-things-i-found-on-the-internets/

Our process

4 weekly meeting activities

- Meeting 1: Intro to process, values talk
- Meetings 2-6: Game pitches (10-15 min.)
- Meetings 3-14: Project updates
- Meeting 15: Final/everyone playtesting

  One week after, the games go on our site.

New Game Pitch Sheet

This sheet is for estimation purposes, and is not expected to be a perfect prediction.
We'll adapt! This gives us material to talk about in pitch meeting, as part of our process.

Videogame Title: _____

Number of Players: ____

(Single)   Alternating   Split-Screen   Same Screen   Networked

Approximate Project Complexity:

Retro	NES	Genesis	Playstation	X-Box/PS2	Web/Casual
1975-85	80's	Early 90's	Late 90's	Early 2000's	2000's

Game Genre(s) or Roughly Similar Game(s)

_____

Names of Current Team Members: *fine for now if project lead only*

_____ _____ _____ _____
_____ _____ _____ _____

Main Goal(s) Each Week   *lines intentionally limited in space - please keep tasks high level*
At start: _____

___ ___: _____
___ ___: _____
___ ___: _____
___ ___: _____
___ ___: _____
___ ___: _____
___ ___: _____
___ ___: _____
___ ___: _____
Polish: Determine catch up or clean up goals for this week as it approaches
___ ___: _____

By pitch date I'll have code compiled and running on our target platform in the form of (please circle one):
Game is in-development   Gameplay demo (lacking art/levels)   Prototype   Image moved by input   None

If everyone else had to leave the team, I, as project leader, can and will finish it (in some form): Y N

Values Talk Main Ideas

- Setting realistic goals
  - Build up by "Decades Progression"
  - Some genres are more asset-efficient
- Be more craftsperson than artist, build momentum
- Project Lead should be able to finish alone, fill in
- Small teams are fastier, easier, and more agile
- "Design doc" as one page and mock-up screenshot
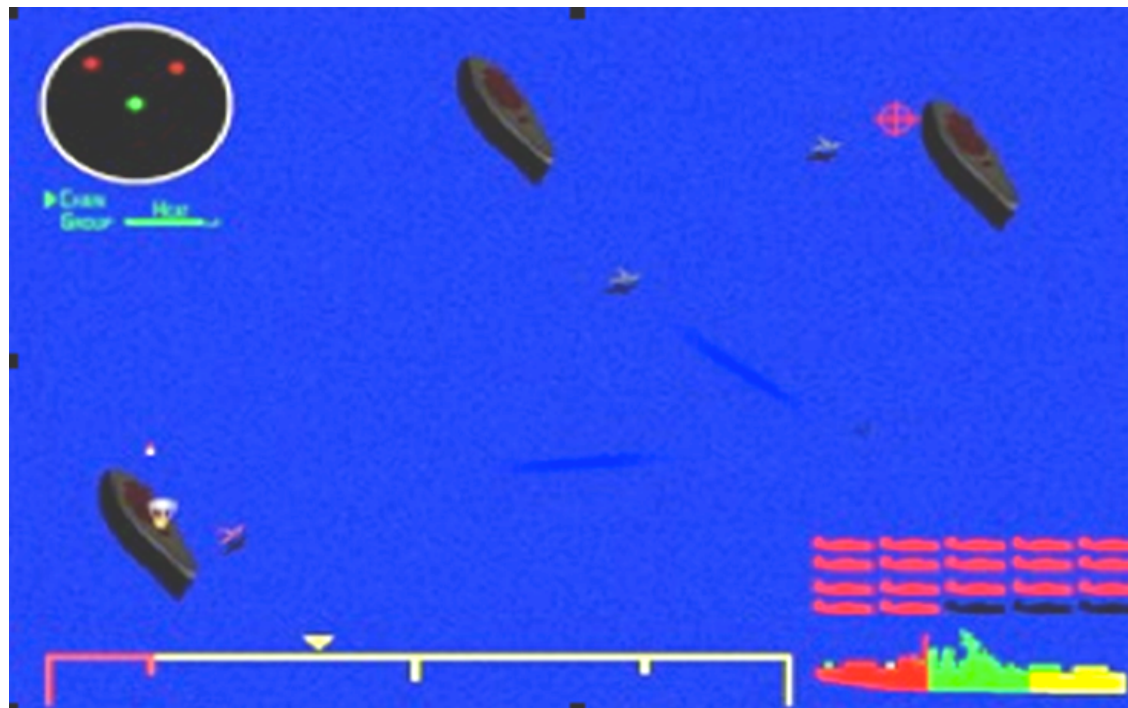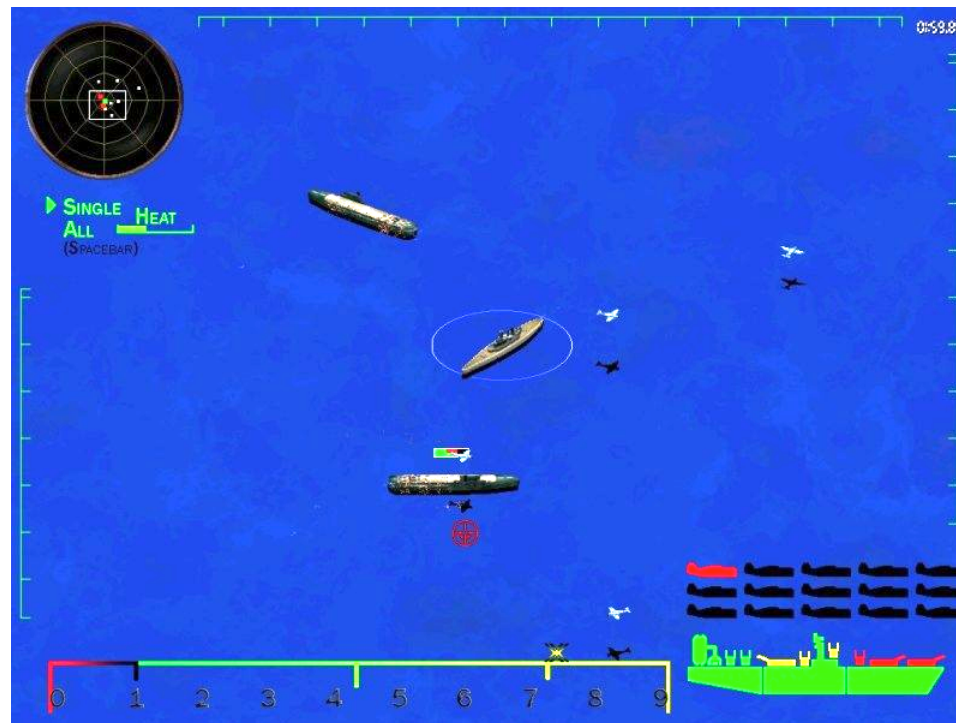- Finishing matters, fit deadline. "Demo is the game"

Saturn Storm: Mock-Up

Saturn Storm Screenshot

Battleship 88: Mock-Up

Battleship 88 Screenshot

Battleship 88's Entire Design Doc

Battleship 88: Iron Hero – An arcade game birthed from the concept of the unfinished *Flagship Carrier*

"Storyline"

-Storyline: WWII, and ships are destroying each other. There really isn't much else to it. We don't want anything to take away from the action element – no screens of text, no intro cut scenes, and no lame excuses for why the player is the only ship surrounded by hordes of enemy vessels. We also won't attempt to answer questions like, "If Germany used subs, and Japan used carriers, then how can we be fighting both?" We're fighting both because it'll make for a damned good time.

Features

-Planes fly in from off-radar and from off-screen enemy carriers, to machine gun strafe and torpedo bomb the player's ship. Destroy planes before they escape for reload to earn kill marks – every twenty plane-kill marks, US air support flies in to shoot down enemy planes and help bomb nearby enemy ships.
-Player character is an arcade-simulation hybrid of a battleship. Think of the MechWarrior control scheme – throttle instead of forward/backward, weapons systems can shift modes, and so on.
-Camera loosely follows ship, but is locked at ¾ isometric overhead
-Long range radar, so the player never wonders where to go next
-Weapons on ship: AA flak for anti-plane, big cannons for anti-ship/sub
-Submarine: periscope & tower shows shortly before and after firing. Otherwise it acts more or less as any enemy ship would. It shows up on radar at all times, and leaves a minor (but visible) wake on the surface when moving. Only ways to kill it: US Airstrike torpedos, aiming too close to send shells underwater, or shooting it while it surfaces for attack.
-Water: blue swirls, constantly in drift, with a layered effect
-Minimal Menu: Start game, adjust options (difficulty, volume, controls), view high scores, exit
-Missions done as "random" skirmish –parameters determine # of enemy ships and planes, but the locations are spawned differently each time
-Enemy ships: aircraft carriers, battleships, small destroyers, submarines. Ships will all use a different model, except for the enemy battleship, which will be a retextured version of the player.
-Enemy Planes: machine gun only (3 pass), torpedo bomber (2 passes), kamikaze (direct flight), and a couple that mix the functions of each (to keep gameplay less predictable). Planes will be distinguishable primarily by their coloration.
-Enemy AI heads straight for player at a distance, switches to circle strafing & firing when close enough, and fights to death. Nothing fancy. Aircraft carriers do their best to stay just outside of the player's firing range, but they move considerably slower than the player.
-Game ending screen for victory or defeat (fancy renderings, a handful of images)
-No allied ships
-Right mouse button: fire machine guns. Left mouse button: cannon fire
-Location based damage of ship. Underwater–hull damage, in front (maneuverability impaired), middle (ship closer to destruction), or rear (speed impaired), and above water damages weapons systems and/or cooling unit.
-Red light beacons float away from sunken enemy ships, and when the player gets close enough it causes a US plane to fly in and drop a crate into the ocean. Crates are powerups for armor plating, temporary speed/weapon boosts, and point bonuses.
-Points are relevant to high scores, and one of the mission parameters might be to earn X-amount of points in a limited time frame. More than likely, they will be kept for high scores use only.

Art needed:

-Texture map for Battleship
-Ship/Sub models + Textures
-Plane model + Textures
-Bomb models + Textures

-Powerups with parachutes
-Light beacons
-Interface art
-Victory & Failure screens (done last)
-Main Menu Render (done last)
-Water done parametrically, no models etc. needed
-Explosion & smoke will be done parametrically

SFX Needed:

1-Ship explosion (taking damage, system malfunction)
2-Major explosion (ship annihilated)
3-Ship sinking
4-Plane taking damage (from AA flak)
5-Plane torpedo bomb release
6-Plane machine gun firing
7-Ship Flak firing
8-Ship cannon fire
9-Airstrike fly in (Bonus sound, to alert the player that s/he's destroyed 20 enemy planes and has earned temporary air support)
10-Ship at 4 or 5 different throttle speeds, from idling to full power
11-Barely burning ship
12-Badly burning ship
13-Burning plane falling from the sky
14-Underwater torpedo firing (submarine)
15-Powerup acquisition (armor, points) - "pick up" sound
16-Splash from a cannon shell hitting the water
17-Splash from a plane hitting the water
18-Whatever else our SFX guru feels the game needs

Schedule:

Week 1 (by March 18): Weapons function, game has full particle effects and weapons behaviors, enemy AI is fundamentally out of the way (planes strafe and fly offscreen for ammo, ships avoid collisions, both attack based on how much the player has aggravated them), full interface functionality, 1 plane type, location based damage implemented. Not implemented at all: powerups, beacons, US planes, any enemies besides battleship & torpedo plane, high scores, menus.
Week 2 (by March 25): Submarines and carriers added, plus a kamikaze plane.
Week 3 (by April 1): Powerups added, gameplay balanced/tweaked, scores system added, all other planes and ships implemented.
Week 4 (by April 8 - UC Friday): Testing and polishing (menus, credits) completed; Single player game is effectively done, but kept inside for touch ups and testing.
Week 5 (by April 15 - Spring Carnival): Game finished, playtested, and released.

The Warm Up Semester
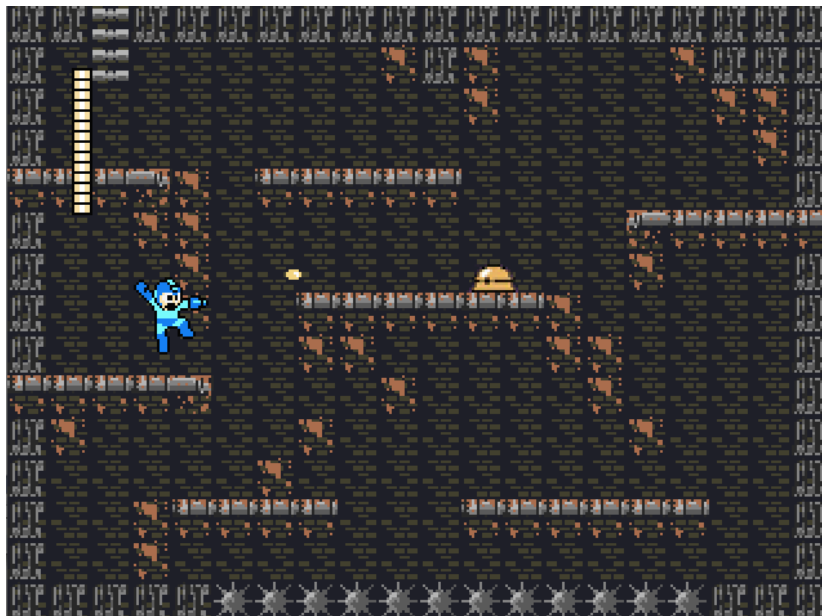
Patterns found with other clubs

- Don't let it turn into a game playing club
  - We avoid game nights, pizza at meetings…
- Keep teams small, games one semester
- Unity, Flixel, Game Maker… but be flexible
- Game Jams are handy, but *very* different
- Didn't start as democracy, grew into one
- Mostly CS majors, even for art/audio/lead

Questions? Other perspectives?

- E-mail me at HobbyGameDev@gmail.com
- Tweet me at @HobbyGameDev
- Or browse related writing and videos at
  HobbyGameDev.com
- Students have questions?
  No question is 'too noob!'

(first GCS logo, from 2004)