Game Development: Project Studio (OART-UT 1612)

Robert Yang <ry14@nyu.edu> // office hours by appointment
721 Broadway, Room 944

Session I, May 28 – July 5: Monday & Wednesday, 12:30 – 4:30pm
Session II, July 8 – August 15: Tuesday & Thursday, 12:30 – 4:30pm

DESCRIPTION
What do you need to know to make a video game? *Everything.*

To that end, all students will learn "everything" -- a basic understanding of programming, visual art, and prototyping practices -- before specializing. We will emphasize technique, yes, but we will also look at theory and philosophy to help guide our practice and develop our games thoughtfully.

The philosophy of the course is learning through doing, and the majority of student work time will be spent in actual design and production, which will be structured and guided by the instructor. This production time will be supplemented by in-class exercises, readings and discussion, and talks from visiting game developers. The session will culminate in a final group project, a whole entire video game. **You are also expected to work, a lot, outside of class.**

(recommended pre-reqs: Games 101, Thinking About Games)

YOU WILL NEED:

- A laptop, and/or USB drive to save your work if you intend to use lab computers.
- (if personal laptop): **Unity3D**, free indie version. ("Pro" is okay, but not necessary at all.)
- (if personal laptop): **Autodesk Maya** (Autodesk offers free 3 year licenses to students)
- (if personal laptop): **Adobe Photoshop** (Adobe recently made its entire CS2 suite free.)
- **SourceTree**, a free and fantastic Git client. You can use a different Git client if you prefer.

WE WILL READ: (I'll provide PDF excerpts, but you can buy these books if you like.)

- **10PRINT,** by Nick Montfort, Ian Bogost, et al. *The philosophy of code and expression.*
- **Game Feel,** by Steve Swink. *The art and science of game input and perception.*

LEARNING GOALS:

- Unity editor interface and common workflows.
- Iterative prototyping processes and troubleshooting, isolating bugs and problems.
- C# syntax, input and control structures (if / else / for / while), basic code patterns.
- Conceptualizing 3D space / raycasting / basic vector math.
- Maya, basic polygonal modeling / texturing / animation.
- Git and source control practices.
- (Coders:) simple NPC AI state machine design, simple flocking / steering.
- (Artists:) texturing in Photoshop / basic considerations, painting normal / specular maps.
- (Level / world design:) planning envs, greyboxing, modular building, architecture thinking.

CONCEPTUAL GOALS:

Familiarity with Heidegger / phenomenology
Understanding of code and 3D as unique discourses.

CLASS WEBSITE:
Session I, May 28 – July 5: http://github.com/radiatoryang/gamedev_may2013
Session II, July 8 – August 15: http://github.com/radiatoryang/gamedev_july2013

TO TURN-IN HOMEWORK:
Click "wiki" on the Github repo page, and edit the HW page. Upload and link TWO (2) things:

1) A link to a **Unity webplayer .HTML**
2) A link to **your project folder** hosted on a public Git repository.

Late individual work is accepted, but you will be penalized a grade level and receive no feedback.

CLASS SCHEDULE (subject to change)

WEEK 1	Introduction to Unity C#: thinking in 3D space Maya: basic workflow, modeling. Introduction to Git + SourceTree. **[HW] Play with terrain.** **Make a model in Maya.**	Maya: texturing workflow, UVs. C#: Vector math, inputs, moving stuff. Git: review and workflow. **[HW] Play with lights / camera /sound.** **Read 10PRINT, Ch 1 + 2.** **Make a short music video.**
WEEK 2	Maya texturing review Vector math review Unity: Colliders, triggers, physics. **[HW] Rube Goldberg machine.** **UV a model in Maya.**	Maya pipeline review, Vector math review Unity: raycasts / spherecasts / oh my. Unity C#: character controllers, 2 ways. **[HW] Read 10PRINT, Ch 3 + 4 + 5.** **Finish your controllers.**
WEEK 3	Maya / vector math / raycast review. **>>> MIDTERM -- the Ludum Dare.** Movie: "Being in the World." "What do prototypes prototype?" "Shi*ty first drafts" **[HW] Keep working on your game.**	Maya / vector math / raycast review. **>>> MIDTERM -- ... the Ludum Dare.** Open-ended review / check-in. Last minute sprint, cutting room floor. **[HW] Publish your game, get press or $?** **Read Game Feel, Ch. 1 + 8 + 9**
WEEK 4	Thinking with systems. C#: working with data structures. **>>> BEGIN FINAL PROJECT <<<** Collab: Post-it storming + "say yes." **[HW] 3 group pitches, moodboards.**	Maya: animation / bone-weighting. Prototype iterations / check-in / workday. Collab: Git + Task lists + Agile dev theory. **[HW] Work on a second iteration.** **Read Game Feel, Ch. 12-16**
WEEK 5	SPECIALIZATION DAY Check-in / open workshop / workday Advanced labs: flocking NPCs, normal / spec maps, world building. **[HW] Systems-complete ALPHA.**	**MILESTONE: ALPHA.** Workshop. Student choice, what do you need to know Arcade + Guests: indie game developers. **[HW] Implementation-complete BETA.** **Write Feel Analysis of your game.**
WEEK 6	**MILESTONE: BETA** Open workshop / check-in. Arcade + Guests: game journalists. **[HW] Fix bugs! Go for the GOLD!**	**MILESTONE: GOLD** Last minute sprint, cutting room floor. Final presentations. Last hour: public Arcade, friends / family.

ASSESSMENT

Students will be given grades based on a 100-point scale. Each assignment will be graded on a point scale, and these points will be added up to determine the final grade, according to the following:

98-100 A+
92-97 A
90-91 A-
88-89 B+
82-87 B
etc.

The following are the components of the grade:

Attendance & participation	30
Homework	15
Midterm	10
Alpha milestone	10
Beta milestone	15
Gold milestone	20
TOTAL	= 100

- Late individual work (the first 2 weeks) is accepted with penalties / no feedback.
- Late group work is NOT accepted.

Attendance & Participation

The attendance and participation portion of your grade is based on the following:

- Your attendance in class and tardiness
- Participation in group discussions and critiques
- Peer grades and participation in writing group evaluations

Private peer grades

You'll give a grade to each member of your group. You can add a short explanation if you like, and you must add some explanation when giving a grade of C or below.

A = Fully participated and contributed ideas - hard worker and great teammate
B = Generally was present during the process - no complaints
C = Attended some meetings, but could have contributed more
D = Was absent from most or all meetings, or counter-productive in some way
F = Completely absent from the process

Group evaluations

Students will also write an evaluation of each team member at each milestone. These evaluations will be sent to all group members and to the instructor. They must include:

a) Two positive observations. Particular skills, behaviors, decisions, or other ways in which a member made a positive contribution. Each observation should be written in a few sentences.

b) Two areas for improvement. At least two observations that point out how the team member can change their working style, collaborative approach, or other aspects of their behavior to improve the project and the team dynamic.

Tips for Working Successfully in a Group

From the Building Virtual Worlds class at Carnegie Melon's ETC Program

Meet people properly. It all starts with the introduction. Then, exchange contact information, and make sure you know how to pronounce everyone's names. Exchange phone #s, and find out what hours are acceptable to call during.

Find things you have in common. You can almost always find something in common with another person, and starting from that baseline, it's much easier to then address issues where you have differences. This is why cities like professional sports teams, which are socially galvanizing forces that cut across boundaries of race and wealth. If nothing else, you probably have in common things like the weather.

Make meeting conditions good. Have a large surface to write on, make sure the room is quiet and warm enough, and that there aren't lots of distractions. Make sure no one is hungry, cold, or tired. Meet over a meal if you can; food softens a meeting. That's why they "do lunch" in Hollywood.

Let everyone talk. Even if you think what they're saying is stupid. Cutting someone off is rude, and not worth whatever small time gain you might make. Don't finish someone's sentences for him or her; they can do it for themselves. And remember: talking louder or faster doesn't make your idea any better. Check your egos at the door. When you discuss ideas, immediately label them and write them down. The labels should be descriptive of the idea, not the originator: "the troll bridge story," not "Jane's story."

Praise each other. Find something nice to say, even if it's a stretch. Even the worst of ideas has a silver lining inside it, if you just look hard enough. Focus on the good, praise it, and then raise any objections or concerns you have about the rest of it.

Put it in writing. Always write down who is responsible for what, by when. Be concrete. Arrange meetings by email, and establish accountability. Never assume that someone's roommate will deliver a phone message. Also, remember that "politics is when you have more than 2 people" – with that in mind, always CC (carbon copy) any piece of email within the group, or to me, to all members of the group. This rule should never be violated; don't try to guess what your group mates might or might not want to hear about.

Be open and honest. Talk with your group members if there's a problem, and talk with me if you think you need help. The whole point of this course is that it's tough to work across cultures. If we all go into it knowing that's an issue, we should be comfortable discussing problems when they arise -- after all, that's what this course is really about. Be forgiving when people make mistakes, but don't be afraid to raise the issues when they come up.

Avoid conflict at all costs. When stress occurs and tempers flare, take a short break. Clear your heads, apologize, and take another stab at it. Apologize for upsetting your peers, even if you think someone else was primarily at fault; the goal is to work together, not start a legal battle over whose transgressions were worse. It takes two to have an argument, so be the peacemaker.

Phrase alternatives as questions. Instead of "I think we should do A, not B," try "What if we did A, instead of B?" That allows people to offer comments, rather than defend one choice.