NEW YORK UNIVERSITY — TISCH SCHOOL OF THE ARTS

INTRODUCTION TO GAME DEVELOPMENT

GAMES 121-001

FALL 2017

COURSE SYLLABUS

Professors: Aaron Freedman (aefreedman@nyu.edu)
Technical Instructors: Qiu Sun (qs399@nyu.edu)

Course Description

Introduction to Game Development is a practical course that introduces students to the methods, tools and principles used in developing digital games. Over the course of the semester, students will work alone to create a two digital prototypes or 'sketches', before building on them to produce a final polished game, using the lessons learned in the earlier prototypes. This is a hands-on, primarily lab-based course, and so the focus is on learning-by-doing rather than on reading and discussion.

Course Objectives

At the completion of this course, the student will be able to:

1. Practice the fundamentally integrated technical processes of digital game development, by rolling together elements of visual art and design, sound design, systems design, interaction design and code.
2. Learn to implement game assets and code in an established digital game engine (Unity).
3. Identify major principles of implementation-level game design, and learn the 'tricks of the trade' that serve to engage a user and provide subconscious-level enjoyment of a game.
4. Analyze and articulate strengths and weaknesses in the student's and classmates' work.
5. Present their work to a group, highlighting its functionality and strengths.
6. Develop a personal creative process, allowing the student to translate ideas into the form of a digital game.

Course Format

Introduction to Game Development is primarily a project-based course. Weekly meetings consist of a lecture class, which will contain skill-building exercises and critical feedback sessions, and a lab, in which students can work with assistance and hands-on technical instruction from the instructor and the TA. Students are expected to present their work for feedback in every lecture class.

Assignments And Readings

The assignments in this class work toward the development and completion of a small game in Unity. After an initial ideation exercise, each assignment focuses on one particular part of the game: the gameplay, the audiovisual aspects, and finally finishing it.

Assignment 1: Gameplay-focus Prototype
The first assignment is to create a playable prototype with abstract or placeholder aesthetic elements (visuals and sound), focused entirely on generating a pleasurable or meaningful interaction experience.
Due: In class, Week 6

Assignment 2: Audiovisual Prototype
The second assignment requires students to modify their gameplay prototype to add visual and sonic elements to it, not only to make the experience more aesthetically coherent and pleasurable, but to better convey the rules and state of the game to the player. Gameplay should not change significantly for this prototype, and the iteration and experimentation should focus entirely on audiovisual characteristics.
Due: In class, Week 9

Assignment 3: Final polished game
This will mean: finishing the audiovisual aspects, debugging and balancing the gameplay, building a complete set of levels or environments, and completing the menu or UI. In this phase, students focus on aspects like a) how game mechanics are taught to the player, b) how the game feels to play, c) responding to results from playtesting, and d) removing placeholder art and eliminating bugs.
Due: In class, Week 13

End of Term Game Jam
The final week of the semester is devoted to a brief ‘game jam’, where students will use the skills they developed over the course of the year to prototype a new game, this time in small teams.
Due: In class, Week 15

Readings

As this is a practical course there is no set text. However, we will be discussing short web-based readings and recorded lectures, including the following:

- Michael Brough, How To Do A Game Jam. http://mightyvision.blogspot.co.uk/2013/04/how-to-do-game-jam.html
- Chris De Leon: Game Programming fundamentals http://www.hobbygamedev.com/articles/vol5/game-programming-fundamentals/
- Petri Purho and Martin Jonasson: Juice it or Lose it (video). http://www.youtube.com/watch?v=Fy0aCDmgnxg
- Jonathan Blow: Push and Convey (audio + slides): http://braid-game.com/news/2008/02/another-lecture-this-time-from-denmark/

Prerequisites

Games 180 - Intro to Game Programming

Credits Allocated

4 Credits

Grading

Game assignments must be presented in class AND submitted via NYU Classes to be graded complete.

Since each phase of the class is divided by focus, the grading on each project will be different.

Gameplay prototypes will be graded on:

- **Functionality**. Has the student made a playable game that runs with no obvious bugs or game-breaking flaws?
- **Feel**. Has the student created a strong sense of engagement in the player and harnessed established rules-of-thumb to make a game that feels smooth and seamless?
- **Creativity**. Does the project show innovation and uniqueness? Does it show a creative imagination that does not solve the given design problem in an ordinary way?
- **Scope**. Did the student constrain the limits of the project in such a way that it can be considered ‘finished’?

Audiovisual prototypes will be graded on:

- **Communication**. Does the audiovisual design convey the rules of play and the state of the game to the player?
- **Creativity**. Does the game have an innovative and distinct audiovisual style? Do the aesthetics mesh with the gameplay and support it?

- **Consistency**. Do the visual elements combine to establish a consistent visual style? Do the sound effects fit together into a cohesive sonic mix?

Final projects will be graded on:

- **Polish**. Does the game show attention to detail in eliminating audiovisual flaws, areas of confusion for the player, and a general sense of being 'finished'.
- **Usability**. Has the student communicated the rules and procedures of the game clearly to the player. Is it painless and enjoyable to begin the game and play it?
- **Functionality**. Is the game bug-free and free of performance issues?
- **Progress**. These final games should build and improve on the gameplay prototype and the audiovisual prototype, to create a game that is better than the preceding versions. (Note: this project is NOT graded on whether or not a game is 'fun' to play)

In addition, students are graded on participation in class critique sessions, and given pass/fail grades for updating their development backlog every week, and for participating in the end-of-term gamejam. Finally, if a student maintains a weekly *public* development log, (on Tumblr for example), they can earn up to 5% extra credit.

Assignment Requirements

In addition to the oral presentation of the game to the class, each completed assignment must include the following, available on your Itch.io page for your prototype:

- A playable HTML5 version that runs in the browser on your Itch.io page
- A working executable version of your game, for Windows or Mac.
- A zip file containing your Unity project folder (including your source code & all assets).
    - Making a "Release" on your GitHub repo and linking to the release is sufficient
    - You may also download a copy of your repository and upload that zip file to your Itch page as a downloadable file. (Don't upload your working copy.)
- 3 x screenshots
- A one-page document detailing your game description and play instructions.
    - You may use the project description field for this
- A statement of self-evaluation on the game's merit.

Grade calculation

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:

92-100	A
90-91	A-
88-89	B+
82-87	B

etc.

The following are the components of the grade:

Participation		20
In-Class presentations+backlog		30 (Pass/Fail per week)
Gameplay Prototype		10
Audiovisual Prototype	10	
Final project		20
Game Jam		10 (Pass/Fail)
Public Devlog		5 (Extra Credit)
TOTAL	**100**	

Attendance: Attendance and arriving on time to all class sessions is required and expected, too many unexcused absences will lower your final grade. Three unexcused absences lower your final grade by a letter. Each subsequent unexcused absence will lower another letter grade. Two tardies will count as one absence. Arriving more than 15 minutes late will also count as an absence. If you will be missing a class due to illness, or unavoidable personal circumstances, you must notify your professor in advance via email for the absence to be eligible to be excused.

Late policy: projects submitted after the due date will be graded **zero** unless permission to submit late is sought from the professor in advance.

For your game projects to be graded complete, they **must** be submitted to NYU Classes under the appropriate assignment listing.

Statement Of Academic Integrity:

Plagiarism is presenting someone else's work as though it were your own. More specifically, plagiarism is to present as your own: A sequence of words quoted without quotation marks from another writer or a paraphrased passage from another writer's work or facts, ideas or images composed by someone else.

Importantly however, Intro to Game Development is not a course in programming, so while students are expected to produce original and unique gameplay mechanics, art and sound, they may borrow code liberally from their classmates or from online sources.

Accessibility

Academic accommodations are available for students with documented disabilities. Please contact the Moses Center for Students with Disabilities at 212 998-4980 for further information.

Health & Wellness

Your health and safety are a priority at NYU. If you experience any health or mental health issues during this course, we encourage you to utilize the support services of the 24/7 NYU Wellness Exchange 212-443-9999. Also, all students who may require an academic accommodation due to a qualified disability, physical or mental, please register with the Moses Center 212-998-4980. Please let your instructor know if you need help connecting to these resources.

Schedule

Week 1 – Recap of Unity, and Thinking About Your Idea

A reintroduction to working in Unity and C# and project ideation

Introductions.
A recap of Unity and C#.
Thinking about scope.
Ideation.
Beginning work on a gameplay prototype; developing and discussing the concepts.

Tutorial and in-class exercise: Unity scene organization and workflow. The input manager. The camera. Live coding a small game together as a class.

Tutorial 2: Developing a backlog. Time boxing and project management.

In-class exercise: Discuss ideas for a game with a classmate.

Homework: Finalize task list for a gameplay prototype, ready to begin sprint next week.
Install GitKraken.
Make a GitHub account.

Reading: Michael Brough, How To Do A Game Jam.
http://mightyvision.blogspot.co.uk/2013/04/how-to-do-game-jam.html

Week 2 – Present Gameplay Ideas, and Start Coding!

Presenting your task list to the class, and beginning work on your concept

Present your game task list to the class for feedback.

Learn how to use asset/source control as a structured backup and for collaboration.

Tutorial: Asset control: create a repository, make local commits, push to the remote and revert changes. Set up a Unity project for use with asset control software.

Tutorial: Further recap of Unity core features: UI buttons, Physics, Raycasts (mouse), Gizmos

Homework: Begin your project backlog.

Week 3 – Overview of Game Feel

Develop an Understanding of Game Feel

Develop ability to analyse and talk about games in terms of their feel.

In-class exercise: Game Feel: group analysis of a game in terms of feel, discuss principles of game feel.

In-class exercise: Interactive game feel exercise

Homework: Continue work on gameplay prototype and update project backlog.

Week 4 – Playtesting Methods

Understanding the importance of playtesting & how to collect data

Tutorial: methods for testing games and documenting tests. Setting up a digital game for ease of testing and collection of data.

Homework: Continue work on gameplay prototype and update project backlog.
Documented playtest of current game build.

Reading: Game Design Workshop, Tracey Fullerton - playtesting chapter

Week 5 – Tuning

Finishing work on gameplay prototype

Discuss the importance of, and methods for, tuning the gameplay systems in your prototype. Discussion of technical methods for making a project tunable in Unity.

Tutorial: interactive tunable project: take a broken game and improve it using exposed variables.

Homework: Install Adobe Photoshop

Week 6 – Intro to Photoshop

Present final gameplay prototype & begin audiovisual prototype

Presentation of final gameplay prototypes
Understand the basics of using Photoshop to prepare visual assets for Unity, and of the Unity 2D sprite rendering system.

Discussion of the development of a visual style for your gameplay prototype.

Tutorial: Basics of photoshop for Unity's sprite system. Produce a simple sprite and import into Unity.

Homework: Install Adobe Audition

Week 7 – Intro to Sound Design

Audiovisual prototype continued

Understand the basics of sound design, and how sound is used to convey information and mood in a game. Import and play sounds in Unity.

Tutorial: Basics of sound design. Recording sounds, and adapting them from third-party sources. Understand how sounds are 'mixed' by Unity's audio engine. Trigger a sound from an 'AudioSource' component.

In-class exercise: Add a sound effect to your game.

Homework: Bring completed audiovisual prototype to class for presentation and feedback.

Week 8 – Animations and Particle Effects Unity

Finishing audiovisual prototype

Understanding the technical and artistic underpinnings of animations, using Photoshop and in Unity. Learn to use Unity's particle system for a variety of effects.

Tutorial: Animations in Photoshop and Unity. Produce a simple 'straight ahead' animation in Photoshop and a keyframe animation in Unity

Tutorial: Particle systems.

Week 9 – Source control

Present audiovisual prototype

Tutorial. Refactoring code. Debugging.

In-class exercise: debugging and fixing a broken game.

Week 10 – Begin polish project

Teaching the player

Discuss methods for making games more usable, focusing on the features that enable a player to understand how to play the game, and how to understand the game's state. Discussion of tutorials and information conveyance in games. Understand some approaches to teaching players how to play a game. Discuss pros and cons of each method.

Tutorial: analysis of the ways information about game state and systems are conveyed to the player in a game.

Tutorial 2: production documentation and project planning part 2. Develop a gold-list for your game and stick to it.

In-class exercise: Scoping exercise - cut features from your game until it falls within scope.

Week 11 – Audiovisual feedback - 'Juice'

Continue Work on Final Game

Discuss the set of audiovisual polish techniques known as 'juice', that are used to increase player engagement and convey information about a game's state.

Tutorial: Group analysis of two ‘juicy’ games. Understand the function of ‘juice’ and how it is added to a game.

Homework: Continue work on polished game.

In-class exercise: Juice exercise - In-class modification of a simple game by adding feedback effects to it.

Week 12 – Beta test

Continue Work on Final Game

In-class beta test of final projects.

In-class exercise: Problem-solving exercise for addressing last-minute issues

Homework: Bring final game project to class for presentation, in a compiled ‘player build’ format.

Week 13 – Present polished games

Game jam starts

Final polished games are presented in class.
End of semester game jam starts.

Tutorial: Improving a pre-built game project solely through the modification of tunable variables, comparison in class.

Homework: Finish game jam games for final presentations.

Week 14 – Final class

Jam Presentations

Present final game jam games in class. Critique session.