

INTRO TO GAME DEVELOPMENT

GAMES-UT 120 FALL 2018

Monday & Wednesday 9:30AM-12:15PM, 2 Metrotech Center 8 FL, Rm. 825

| | | |
|------------|----------------|--------------------|
| Professor: | Aaron Freedman | aefreedman@nyu.edu |
| TA: | Carol Mertz | carol@nyu.edu |

Course Description

Introduction to Game Development is a practical course that introduces students to the methods, tools, and principles to the development of digital games. Students will work alone to create two prototypes and use these prototypes to create a version of a polished game. This is a hands-on, lab-based course, and the focus is on learning-by-doing rather than on reading and discussion.

Prerequisites

Games 180 - Intro to Game Programming

Credits Allocated

4 Credits

Course Objectives

By the end of the course you should be able to:

1. Apply multiple technical skills used in digital game development, including digital content creation, sound design, user experience design, and interaction design and programming.
2. Learn to implement game assets and code in Unity, an established digital game engine.
3. Identify major principles of game design and learn techniques for crafting gameplay experiences.
4. Critique video game prototypes and provide meaningful feedback.
5. Present your work to a group and articulate its strengths and weaknesses.
6. Develop a personal creative process and be able to turn an idea into a playable prototype.

Course Format

Introduction to Game Development is a project-based course. Each week consists of a lecture that contains a critical feedback session and also cover various technical skills, and a lab during

which students will work with assistance from the course TA. Students are expected to present their work for feedback in every lecture class.

Assignments

The assignments in this class cover basic implementation of simple games in three major video game genres. For each project, you will have a choice of example games and must implement a version of that game in Unity. The goal for the first phase of each project is to recreate the original game as faithfully as possible - or improve on it - while not being worse in any dimension. After your clone is submitted, you will have a week to modify the project creatively. You are graded both on the accuracy of the clone, and the creativity of the additional week's elaboration.

Reference versions of the games are available online at this URL
<http://gcserver.magnet.nyu.edu/~bennett/emu/intro.html>

Assignment 1: Action Game

Clone (recreate) of one of the following two games:

- *Arcade Volleyball*, Commodore 64, Rhett Anderson, 1988
- *Freeway*, Atari 2600, Activision, 1981

You will have four weeks to complete your technical implementation of the game, and an additional week to add creative touches and personalize it.

Due: In class, Week 6

Assignment 2: Puzzle Game

Clone of one of the following two games:

- *Sokoban*, Gameboy, Pony Canyon, 1982
- *Blockade*, Arcade, Sega, 1976

You will have three weeks to complete your technical implementation of the game, and an additional week to add creative touches and personalize it.

Due: In class, Week 10

Assignment 3: Narrative Game

Choose one of the following games:

- *Kimmy*, PC, Star Maid Games, 2017
- *LostMemoriesDotNet*, PC/Web, Star Maid Games, 2017

You will have two weeks to recreate a portion of the game and an additional two weeks to complete the audiovisual polish.

Due: In class, Week 14

Assignment 4: 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:

- 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/>
- Game Design Workshop, Tracey Fullerton - Chapter 9: Playtesting, pp. 248-276

Assignment Grading

Each assignment will be graded according to the following rubrics, with additional points for the in-class presentation of the assignment. Each game has a set of technical requirements that must be implemented in the final version of the game in order for the assignment to receive full marks.

Action Prototype

Phase 1: Choose one of these two games, and faithfully recreate it, at minimum implementing these features:

| | |
|--|--|
| <p>Arcade Volleyball</p> <ul style="list-style-type: none"> • 2-Player • Scene Management — Game Reset • Sprites — Player, Ball, Net • Prefabs — Ball, Player • Mechanics <ul style="list-style-type: none"> ◦ Rigidbody2D ball ◦ Player movement and interaction with ball • State <ul style="list-style-type: none"> ◦ Counting number of touches | <p>Freeway</p> <ul style="list-style-type: none"> • 2-Player • Scene Management — Game Reset • Sprites — Cars, Player, Road • Prefabs — Cars, Players • Mechanics <ul style="list-style-type: none"> ◦ Moving cars ◦ Car and player collision ◦ Player input and movement • State <ul style="list-style-type: none"> ◦ Score |
|--|--|

| | |
|---|--|
| <ul style="list-style-type: none"> ○ Ball hitting net ○ Ball hitting ground ○ Score ● Audio — Simple sound effects | <ul style="list-style-type: none"> ○ Detecting win state ○ Detecting fail state ● Audio — Simple sound effects |
|---|--|

Phase 2: Spend a week creatively elaborating the cloned game. You will be graded on originality, creativity, and functionality.

Grid Prototype

Phase 1: Choose one of these two games, and faithfully recreate it, at minimum implementing these features:

| | |
|---|---|
| <p>Sokoban</p> <ul style="list-style-type: none"> ● Scene Management <ul style="list-style-type: none"> ○ Multiple scenes/levels ○ Game reset <ul style="list-style-type: none"> ■ At will ■ After game over ● Mechanics: <ul style="list-style-type: none"> ○ Discrete motion on a grid ○ Player interaction with blocks ● Prefabs — Player, Blocks, Goal ● Sprites: <ul style="list-style-type: none"> ○ Player ○ Blocks ○ Walls and Environment ● State: <ul style="list-style-type: none"> ○ Detect win state ● Audio — Simple sound effects | <p>Blockade</p> <ul style="list-style-type: none"> ● Scene Management <ul style="list-style-type: none"> ○ Multiple scenes/levels ○ Game reset <ul style="list-style-type: none"> ■ At will ■ After game over ● Mechanics: <ul style="list-style-type: none"> ○ Discrete motion on a grid ○ Player is a “snake” that occupies multiple positions ○ Player who runs into other player loses ● Prefabs — Player, Environment ● Sprites: <ul style="list-style-type: none"> ○ Player ○ Walls and Environment ● State: <ul style="list-style-type: none"> ○ Detect win state ○ Detect fail state ● Audio — Simple sound effects |
|---|---|

Phase 2: Spend a week creatively elaborating the cloned game. You will be graded on originality, creativity, and functionality.

Narrative Prototype

Phase 1: Reconstruct a portion of a reference narrative game in Unity using the Ink plugin. The reference source will be provided to you and you will have two weeks to implement the game.

Phase 2: Spend two weeks elaborating on your narrative game by adding audiovisual elements and optionally adding or modifying the narrative content.

Your final prototype will require the following features:

- **Scene Management**
 - Multiple scenes/levels
 - Title screen (scene)
 - Menu screen (scene)
 - Game reset
 - At will
 - After game over
- **Mechanics:**
 - Navigation through interactive narrative
 - Content branches and responds to player input
- **Sprites/Textures:**
 - Characters
 - Items referenced in dialogue
- UI
- Three different “environments”
- **State:**
 - Detect game end state
 - Track position in dialogue
 - Persistent state through multiple sessions
- **Audio**
 - Simple sound effects
 - Ambient sound
- **Animation** — Basic sprite changes
- **Length of Experience:** 2-3 minutes
- **Plugins** — Ink

General Design Considerations & Glossary

The following list of items comprise the general criteria upon which your work will be judged and are arranged in an approximate order of priority.

- **Functionality:** The game is free of critical bugs and free of performance issues. The game is functional on supported platforms.
- **Scope:** You constrained your tasks so that it you were able to accomplish all of your work by the end of the project timeline.
- **Progress:** You improved on your prototype over weekly iterations to create a game that is better than its preceding versions.
- **Usability & Accessibility:** The game clearly communicates the rules and mechanics of the game to the player. The game is playable within an acceptable range of use-cases.
- **Feel:** Your prototype engages the player and uses techniques from class to create your intended gameplay experience.
- **Creativity:** You attempted to solve design problems with imagination and personal style. Your game a distinct audiovisual style. The aesthetics help to accomplish your design goals.

- **Coherence:** The visual elements establish a coherent style. The audio creates a cohesive sonic environment.
- **Polish:** The game shows attention to detail. This includes but is not limited to elimination of audiovisual flaws, confusing user experience, and a general sense of being “finished”.

Weekly Critique and Participation

Every week before lecture you will present the current state of your prototype to the class and to the instructors and you will receive critique based on this presentation. Critiques are part of your final grade and each week is graded on a Pass/Fail basis.

Passing the weekly critique will involve accomplishing the following tasks before the class period:

- Successfully upload an updated and playable build to your project's Itch.io page
- Push your project changes to your GitHub repository
- Articulate your current Backlog
- List which tasks you completed during the previous week
- List which tasks you intend to complete in the coming week
- Successfully implement specific tasks related to the week's lecture topic and demo

Please note that if you miss a critique due to an excused absence, you are required to make-up your presentation by emailing the instructor and TA your answers to the above requirements *before* the class lab period.

In the case where you fail to send this update before lab, or miss your presentation due to an unexcused absence or excessive tardiness you will automatically fail your critique for that week.

Critique Participation

You are also expected to participate in your classmates' critique sessions. During critiques all laptops and other devices must be put away. You are not required to provide feedback or ask questions during critiques, but you are encouraged to do so. Failing to participate in critique sessions will be cause for you to lose a portion of your Participation grade at the instructor's discretion.

Assignment Submission Requirements

In-class presentation

Students are required to present their completed assignment during class using a published WebGL build hosted on their Itch.io project page. Presentations are conducted in a fashion similar to the weekly critiques.

Online submission

In addition to the oral presentation of the game to the class, you must also submit each completed assignment to NYU Classes under the appropriate assignment listing as a .zip file including the following items

- A copy of your project repository **from GitHub** (download a clean copy from your repo)
- A standalone executable build (Windows or Mac)
- 3 screenshots of the game
- A text file with
 - A description of the game
 - Instructions for how to play
 - A statement of self-evaluation on the game's merit
 - A URL to the Itch.io project page
 - A URL to the repository on GitHub

Late policy

Projects submitted after the due date will receive a grade of **zero**.

Final Grade Calculation

You will receive a final grade based on a 100-point scale. Each assignment will be graded on a point scale, and these points will be summed to determine your final grade according to the following scale:

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

Your final grade will be comprised of the following scores:

| | |
|--------------------------------|-------------------------|
| Participation | 20 |
| In-Class presentations+backlog | 30 (Pass/Fail per week) |
| Action Prototype | 10 |
| Puzzle Prototype | 14 |
| Narrative | 16 |
| Game Jam | 10 (Pass/Fail) |
| TOTAL | 100 |

Attendance Policy

Attending and arriving on time to all class sessions is required and expected. This includes all labs, recitations, and critiques. If you intend to miss a class due to illness or unavoidable personal circumstances, **you must notify your professor before class via email** for the absence to be excused. Please be aware that you are still required to turn in work on time, even if your absence is excused.

Unexcused absences and being late to class will lower your final grade. Excessive tardiness will also be counted as absences.

3+ unexcused absences lowers grade

2 tardies = 1 absence

15+ min late = 1 tardy

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.

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

Please adhere to the following guidelines:

- You are expected to write your own code, within reason
- Copying or mimicking code from course materials, official documentation, or tutorials *is permitted*
- Attribute your sources
- Using code from Unity's standard assets is permitted unless specified otherwise
- You may not use content from another class in your assignments for this class
- You may not use any assets downloaded from the internet, Unity Asset Store, or some other source without written permission
- Make sure to check licenses for any code you use
 - Just because it's online doesn't mean you can use it
 - You may or may not have fair-use protections

Official Tisch policies on academic integrity are available here:

<http://tisch.nyu.edu/student-affairs/important-resources/tisch-policies-and-handbooks>

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

Subject to change

Week 1 – Intro to Unity and Rider

Tutorials and in-class exercises

- Unity topics
 - Creating new Unity Projects & Unity Hub
 - Introduction to the Editor interface, important windows and preferences
 - Unity workflow
 - Components
 - Scripting
 - Scene & Game View, play mode
- Intro to Rider & Scripting
- How to make a WebGL build

Homework:

1. Create a new Unity project and import the 2D UFO Tutorial
2. Go through the tutorial
3. Make your own level and make some improvement to the game
4. Create an Itch.io project page
5. Upload a WebGL Unity build of your version of the 2D UFO Tutorial

Week 2 – How to Clone a Game, Intro to GitHub & Project Management

Weekly Critique: Itch.io build check-in

Tutorials and in-class exercises:

- Review UFO tutorial
- Unity topics
 - Input
 - Physics, colliders, and triggers
 - Scripting interactions
 - Prefabs
 - Instantiation
- Intro to Project Management and GitHub
 - Developing a backlog
 - Time boxing and project management

Homework:

1. Pick a game to clone for your Action Prototype
2. Create a new Unity project
3. Create a new Repository in the class Organization on GitHub
4. Develop and begin working on your backlog using your repo's Issue Tracker
5. Upload the first iteration of your prototype to Itch.io (put link in repo readme)
6. Have SourceTree installed

Week 3 – Intro to Version Control and More Unity Scripting

Tutorials and in-class exercises:

- Intro to Distributed Version Control Systems (DVCS) and Git
 - What is DVCS
 - Repositories
 - Remote
 - Local
 - Git workflow
 - Branch
 - Commit
 - Pull/Merge
 - Push
- Intro to SourceTree

Homework:

1. Initialize a git repository in your clone's project folder
2. Add your GitHub repository from last week as a *remote* on your repository
3. Push your project to GitHub
4. Update your build on Itch.io
5. Install Photoshop

From now on all repeated tasks required for your weekly check-in will not be explicitly listed as homework (updating your build, updating your backlog, using version control).

Week 4 – Sprite Workflow

Tutorials and in-class exercises:

- Photoshop 101
 - Creating new images
 - Layers
 - Editing tools
 - Pencil vs Brush
 - Marquee and Selection
 - Exporting
- Importing art assets to Unity
- Implementing Sprites in Unity

Homework:

1. Create and add sprites to your prototype
2. Install Audacity

Week 5 – Implementing Sounds

Tutorials and in-class exercises:

- Review of building executables
 - Build settings
 - Platforms
- Sound playback in Unity
 - Components
 - AudioSource
 - AudioListener
 - Assets
 - AudioClip
 - File types
 - Compression
 - Importing
 - AudioMixer
 - Spatial audio

- Looping
- Scripting
 - Playback
 - Referencing Clips and PlayOneshot
- Creating sound assets

Homework:

1. Implement at least one sound in your prototype

Week 6 – Action Games Due; How to Clone a Puzzle Game

Tutorials and in-class exercises:

- Action games presentations
- Breaking down a puzzle game
 - Core elements
 - Levels
 - State

Homework:

1. Create a new repository and Unity Project
2. Create a backlog of tasks for cloning your game of choice

Week 7 – Data Structures, Scene Management

Tutorials and in-class exercises:

- Data Structures
 - Arrays
 - Lists
 - Graphs
- Scene Management
 - Loading and unloading scenes during runtime
 - Multi-scene workflow
 - Build settings
 - Persistent data

Week 8 – Tuning

Tutorials and in-class exercises:

- Discuss the importance of, and methods for, tuning the gameplay systems in your prototype
- Techniques for tuning a Unity prototype
 - Presets
 - ScriptableObjects

Homework:

1. Implement a ScriptableObject and use it to tune your game

Week 9 – Playtesting Methods and Data Collection

Tutorials and in-class exercises:

- Methods for testing games and documenting tests
- Setting up a digital game for ease of testing and collection of data

Homework:

1. Implement at least three analytic data points in your game
2. Output session data to a text file
3. Conduct a playtest with another person with data collection enabled

Reading: Game Design Workshop, Tracey Fullerton - playtesting chapter

Week 10 – Puzzle Games Due; Intro to Narrative Games

Puzzle game presentations

Tutorials and in-class exercises:

- Introduction to text-based narrative game implementations
 - Linear vs Branching
 - “Procedural” narrative
 - Presentation styles
- Plugins and Unity
 - Asset Store
 - Package Manager
- Ink, Inky, and Inclecate

Homework:

1. Create a new GitHub repo and Itch.io project page
2. Choose a narrative to clone and implement 1-2 minutes of content using Inky
3. Familiarize yourself with the Ink plugin for Unity using their example project
4. Develop your backlog in your GitHub project
5. Push your .ink file to your GitHub repo

Week 11 – Canvas UI and Parsing Data

Tutorials and in-class exercises:

- Unity Canvas UI system
 - Essential components (Canvas, Event System, Canvas Scaler)

- Fitting and Sizing options
- Arranging UI elements
- Organizing the Canvas
- Interactive Canvas elements
 - Buttons
 - Panels
 - Groups
- Data parsing
 - Data formats (text, JSON, XML, CSV, etc.)
 - Data encoding
 - Understanding APIs
 - Storing and referencing data
 - Read/Write and modification at runtime
 - Creating data manager classes

Homework:

1. Implement a simple “title screen” scene
2. Implement a “menu” scene
3. Navigate between your main game scene and your title scene using the menu

Week 12 – “Animation”

Tutorials and in-class exercises:

- Technique for replacing assets at runtime
- Creating visuals that “animate”

Homework:

1. Animate at least one element of your narrative game

Week 13 – Game Feel and “Juice”

Tutorials and in-class exercises:

- What is game feel?
- Group analysis of a game in terms of feel
- Discuss principles of game feel.
- Interactive game feel exercise Group analysis of two ‘juicy’ games.
- Understand the function of ‘juice’ and how it is added to a game.

Homework:

1. Juice-up your narrative game

Week 14 – Narrative Games Due; Version Control for Teams

Presentation of narrative games

Tutorials and in-class exercises:

- Version control
 - Branching
 - Merging
 - Actually being responsible about your commit practices

Week 15 – Game Jam

Present final game jam games in class. Critique session.

