

Intermediate Game Development

FALL 2020

NEW YORK UNIVERSITY, TISCH SCHOOL OF THE ARTS, GAMES-UT 121-003

TuTh 2:30-5:15pm Blended / Online

INSTRUCTOR: Robert Yang <ry14@nyu.edu>, office hours: by appointment, email or DM me

COURSE DESCRIPTION:

This course reflects the various skills and disciplines that are brought together in modern game development: game design, programming, asset creation, and critical analysis. Classroom lectures and lab time will all be used to bring these different educational vectors together into a coherent whole; the workshop will be organized around a single, long-term, hands-on, game creation project.

PREREQUISITES:

Intro to Game Development GAMES-UT 120

COURSE STRUCTURE / FORMAT

This course meets twice a week. The lecture meeting is for instructional demos, discussion of readings, and critiques of student work. *The lab meetings are primarily self-directed work time for students to work on projects and homework.* Students are expected to spend at least 6 hours a week, outside of class meetings, on their homework.

COURSE OBJECTIVES / LEARNING GOALS:

By the end of this course, the student will be able to:

1. **Demonstrate design / dev skills via actual implementation of code and assets.**
(Learn C# basics and practice implementing 2D game art and sounds in Unity, understand game design as part of the work of implementation)
2. **Practice collaborative workflows and version control in a modern game engine.**
(Practice GitHub, Git branching, basic pull requests, and project management)
3. **Practice vital “soft” dev workflows to solve problems creatively.**
(Prototyping, iteration, debugging, playtesting, tuning, giving feedback, critique)
4. **Deconstruct and analyze technical implementations in other games.**
(Final project is to reverse engineer an existing game)

MATERIALS / TOOLS:

You will need the following:

- A desktop or laptop computer... no tablets, Surface is OK
- A mouse (any cheap USB mouse is fine)
- Unity 2019.4.x LTS (free personal edition) <https://unity3d.com/get-unity/update>
 - **Don't use Unity 2020.1+ for this class. It's too unstable.**
- Visual Studio Code (“VS Code”) (free) <https://code.visualstudio.com/>
- Adobe Photoshop CC (free Adobe CC license via NYU)

READINGS:

All course readings will be provided. However, here are some recommended books / texts:

10PRINT, by Nick Montfort, et al.

Spelunky (book) by Derek Yu

CLASS WEBSITE:

Code at: github.com/radiatoryang/fall2020_gamedev ... Discord for everything else.

ASSIGNMENTS:

- Weekly development exercises / rapid prototypes
- Midterm project: individually, **make a short autobiographical game**
- Final project: in groups, **reverse engineer and clone a 2D game of your choice**.
 - groups must study / deconstruct the game, devise scope, and plan sprints
 - we will hold you to your planned scope, so plan wisely
 - each group member: 1 page of docs + images showing personal contribution
- Final deliverable:
 - **game webpage** (text description, screenshots, video, game build)
 - **post-mortem / personal documentation of contributions** (1 page)

SCHEDULE: (subject to change, **check Discord for updated schedule**)

NOTE: listed homework is due for the week after

(e.g. Week 1's homework is due on Week 2 lecture)

Week 01: INTRO TO UNITY

- **Introductions**
- **Class discussion: what is game development?**
- **Review course syllabus**
- **Live dev lecture: intro to Unity**
 - setup a new Unity 2D project, intro to Unity interface and project folder
 - build a simple "self-portrait" scene with 2D primitives in Scene tab
 - export to WebGL, itch.io upload process

Homework due next week:

- **recreate a familiar public place... in Unity with 200+ total 2D GameObjects**
 - install Unity 2019.4 LTS + WebGL build target + VS Code
 - adapt the place like a playable 2D game world (top-down? sidescroller?)
 - no input, gameplay, or images allowed; use only simple colored primitives
 - PRO-TIP: to reach the 200+ quota faster, Duplicate game objects!
 - export to WebGL, upload to itch.io, and post the link to Discord

Week 02: 2D SPRITES AND BASIC KEYBOARD INPUT

- **Critique recreations**
- **Live dev lecture: intro to Unity C# and importing sprites**

- Hello World, code simple Text UI “press SPACE to get points” game in C#
- Simple keyboard GetKey movement prototype in C#
 - Enum and states
- Import a spritesheet, modify sprite slicing and pivot
- Art pass the simple movement prototype, sprite sorting
- **Thinking about flatgames**

Homework due next week:

- **make a “flat game” (2D exploration game with no collision)**
 - tweak basic player and camera movement
 - use 5+ images that aren’t readymade game sprites; modify in Photoshop
 - make 1 simple frame-based looping animation using C# and Time.time
 - add ambient sound / music
- **commit and push your project with GitHub + GitKraken** (see handout in lab)
- **“vector math” handout**

Week 03: VECTOR MATH AND 2D PHYSICS

- **Debrief vector math handout, intro to vectors**
- **Debrief solo Git version control**
- **Critique and playtest flat games homework**
- **Live dev lecture: intro to Unity 2D physics engine**
 - Recap how Gamemaker AABB overlap collisions worked
 - Collider, rigidbody, mass, gravity, drag, friction (PhysicsMaterial2D)
 - Hinge joints, spring joints, slider joints, distance joints
 - OnTriggerEnter2D / OnTriggerStay2D / OnTriggerExit2D
 - Constant Force, visualize physics vectors / forces with C# and Debug.DrawRay

Homework due next week:

- **make a Rube Goldberg machine (complicated series of physics interactions)**
 - 5+ “phases”, must run for at least 1 minute
 - don’t begin until player presses [SPACE]
 - use triggers to activate 3+ different camera positions / angles and an end finale
 - use 2+ composite 2D colliders and 2+ joints
 - commit and push to a GitHub repo
- **Bennett’s physics concepts GDC talk**

Week 04: CODING WITH 2D PHYSICS

- **Debrief physics engines concepts**
- **Critique and playtest rube goldberg homework**
- **Live dev lecture: coding 2D characters with rigidbody collision, two-ways**
 - Way 1: start with simple small Peggle-type game

- prototype and tune Rigidbody2D projectile (+ PhysicsMaterial2D on walls)
- Way 2: start with simple small sidescroller platformer test level
- prototype and tune Rigidbody2D “human” with simple gravity and jump
- OnTriggerStay2D below for grounded check, PlatformEffector2D on platforms

Homework due next week:

- **make a treasure hunt quest (2D exploration game with clues and an ending)**
 - Implement some kind of game manager singleton to track score / progress
 - sidescroller or a top-down game with 2 min of gameplay and placeholder sprites
 - make simple Camera or complex chase Camera with dead-zone
 - place 3+ triggers to activate buttons, doors, hints, clues, or ending
 - commit and push to a GitHub repo

Week 05: RAYCASTING + BEGIN MIDTERM

- **Critique and playtest treasure hunt homework**
- **Class discussion:** what is raycasting? possible uses of raycasting in games?
- **Live dev lecture: raycasting and layers, three ways**
 - Raycasting as a grounded check (jump only when standing on ground)
 - Raycasting as a line of sight check (roomba that wanders in a simple maze)
 - bonus: implement with Overlap / Circlecast / Boxcast / sweep?
 - Raycasting underneath the mouse cursor (like a button)
 - also discuss tags and layers
- **Review “How to do a game jam”**

Homework due next week:

- **Creatively elaborate on treasure hunt prototype into a midterm project prototype**
 - design and prototype a new mechanic / interaction
 - basic sprint planning and player stories
 - begin prototyping a new level
 - commit and push to GitHub

Week 06: 2D SPRITE ANIMATORS + ITERATE ON MIDTERM

- **Critique and playtest midterms**
- **Class discussion:** basic animation concepts (frames vs curves, keyframes, looping)
- **Live dev lecture: Unity 2D sprite animator workflow**
 - Animating with frames via pre-made spritesheet and Animation tab
 - Animating with joints via pre-made puppet character rig
 - Using Mecanim Animator Controller / state machine + C# to change animations

Homework due next week:

- **Iterate on midterm based on feedback**

- start art passing your midterm
- be ready to playtest something that is playable from start to finish
- commit and push updates to GitHub

Week 07: SOUNDS IN UNITY

- **Critique and playtest midterms**
- **Class discussion:** What is audio / waveform? Different uses of sounds in games?
- **Live dev lecture: Unity audio pipeline and workflow for a simple music toy**
 - Review sound resources (self recorded, synthesizer like BFXR, FreeSound, etc)
 - Importing a sound into Unity, different sound file formats
 - Play(), PlayOneShot(), stopping, and looping a sound with C#
 - Drum machine loop, PlayOneShot cymbals, 8 key piano via AudioSource pitch
 - “Sound Manager” singleton, GetComponent<> with multiple AudioSources

Homework due next week:

- **Finish and submit midterm**
 - add at least 1 background sound, 1 player sound, and 1 world/object sound
 - bonus: mix a sound in Adobe Audition
 - upload a WebGL build and release it publicly, send it to friends / family?
 - commit and push updates to GitHub
 - submit 1 bug report for another student’s game (repro steps, expected behavior)

Week 08: PROCEDURAL GENERATION

- **Critique / debrief midterms**
- **Class discussion:** What is procedural generation? Applications in games?
- **Live dev lecture: intro to procedural generation**
 - Simple instantiation at random position (planting grass in an area)
 - Instantiation in a 2D grid with nested for() loops (10PRINT port)
 - Instantiate into a List with while() (100 fish swim randomly, manager can control)

Homework due next week:

- **Maze generator proc gen tech demo** inspired by Vlambeer’s Nuclear Throne
 - fork the lab repo via GitHub, commit and push to your fork
 - FloorMakers drop breadcrumb trail of floor sprites and spawn other FloorMakers
 - art pass and tune values to personalize it; what does your tech demo generate?

Week 09: BEGIN FINAL PROJECT

- **Critique proc gen tech demos**
- **Class discussion: What is prototyping? What does a prototype do?**
 - Read in-class “What Do Prototypes Prototype” and discuss typology

- **Form final project groups and begin work on final project**
 - Start a GitHub repo; add collaborators; everyone starts a personal branch
 - Which game are you reverse engineering? Go to Game Library, play it together
 - What is the project scope? What is important to recreate, and what can be cut?

Homework due next week:

- **Each group member should research and prototype a core game system**
 - What are the core systems of your source game? Why?
 - Do NOT start on art assets yet. Only placeholder assets allowed!
- **Commit and push to your personal branch on the group Git repo**

Week 10: GROUP COLLABORATION

- **Final project check-in**
- **Class discussion: how to collaborate as a group**
 - standups, communication, blocking
 - practice building a task list / task board together, prioritize tasks / time estimates
 - unpacking “Scrum” / “Agile” buzzwords
 - using GitHub’s “Projects” task board / Trello ... planning handout?
- **Live dev lecture: Git version control as a group**
 - different branching models, feature branches
 - merging and resolving merge conflicts (resolve using mine / theirs vs manual diff)
 - Pull requests are special moderated merges into master branch via GitHub

Homework due next week:

- **Merge and synchronize final project files, start integrating features together**
- **Link to your personal pull request**

Week 11: UNITY UI

- **Final project check-in**
- **Git version control questions / review**
- **Class discussion: what makes a good game UI?**
- **Live dev lecture: Unity UI**
 - Text, Anchors, Panels, Buttons, Sliders
 - Game tab resolution / aspect ratio, Canvas Scaler
 - EventSystems, declaring and invoking UnityEvents
 - Masks, ScrollRects, Layout Groups (automatically resizing text box)

Homework due next week:

- **Prepare for a playtest next week**
- **Link to your personal pull request**

Week 12: COMPLEX 2D SPRITES

- **Final project playtest**
- **Class discussion: reverse engineering 2D graphics in other games**
- **Live dev lecture: more complex 2D solutions**
 - Simple particle systems
 - Pixel Perfect package, texture filtering, consistent texel resolution
 - Tilemaps, Tilemap Colliders, Unity 2D-Extras (Rule Tiles)
 - Spriteshape splines
 - SpriteAnimator rigger

Homework due next week:

- **Iterate on the final project**
- **Link to your personal pull request**

(THANKSGIVING HOLIDAY)

Week 13: GAME FEEL AND COROUTINES

- **Final project check-in**
- **Class discussion: “Juice it or lose it”**
 - What is tweening, what do tweening curves represent, applications in games
 - Show tweening packages (DoTween(), ZestKit, etc)
- **Live dev lecture: game feel with coroutines**
 - Basic use of coroutines and yield
 - Simple linear position lerp
 - Implement tweening curves with AnimationCurve.Evaluate(time)
 - When NOT to use a coroutine (sequences), when is a timing variable better?

Homework due next week:

- **Prepare for a playtest next week**
- **Link to your personal pull request**

Week 14: FILE I/O

- **Final project playtest**
- **Class discussion: file input / output, serialization (json?), applications for games?**
- **Live dev lecture: using file I/O for save game**
 - Make very simple game (e.g. press SPACE to get points, Vesper5?)
 - OnEnable, deserialize a savegame file (also, timestamp? etc)
 - OnDisable, serialize and write the savegame file
 - using ScriptableObject for game data

Homework due next week:

- **Upload final project “ALPHA” deliverable (due at the end of Week 14 lab)**
 - Itch.io web page with WebGL build, short text instructions
- **Playtest every other project and post 1 QA bug report for each**
 - your computer info (OS, graphics), repro steps, expected behavior / suggestion

Week 15: FINAL CLASS

- **Final project playtest**
- **Class debrief: what did we learn?**

Homework due next week (one week after last class meeting):

- **Upload final project “PATCH” deliverables**
 - Group: web page with WebGL build, 3 screenshots, 1 paragraph description
 - optional but recommended: gameplay video, Windows and MacOS builds
 - Individually: project post-mortem (1 page) that details 2 things that went well and 2 things that could be improved, and 1 paragraph about your contributions
- **No final exam**

ASSESSMENT

Student projects will be graded on demonstrated process and documentation, as well as the playability, readability, clarity of construction, and polish. You will be graded on a point scale, and these points will be added up to determine the final grade, according to the following:

94-100 A

90-93 A-

87-89 B+

84-86 B, etc.

The following are the components of the grade:

Participation 25% Homework 25% Midterm 20% Final Alpha 20% Final Patch 10%

ATTENDANCE

Attending and arriving on time to all class sessions is required and expected. This includes all labs, recitations, and critiques. 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 excused. Unexcused absences and being late to class 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 unexcused absence. Arriving more than 15 minutes late to class will also count as an unexcused absence.

STATEMENT OF ACADEMIC INTEGRITY / PRINCIPLE

The core of the educational experience at the Tisch School of the Arts is the creation of original academic and artistic work by students for the critical review of faculty members. It is therefore of the utmost importance that students at all times provide their instructors with an accurate sense of their current abilities and knowledge in order to receive appropriate constructive criticism and advice. Any attempt to evade that essential, transparent transaction between instructor and student through plagiarism or cheating is educationally self-defeating and a grave violation of Tisch School of the Arts community standards. 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. For all the details on plagiarism, please refer to page 10 of the Tisch School of the Arts, Policies and Procedures Handbook, which can be found online at:

<http://students.tisch.nyu.edu/page/home.html>

HEALTH AND SAFETY

Your health and safety are a priority at NYU. If you experience any health or mental health issues, we encourage you to utilize the support services of the 24/7 NYU Wellness Exchange 212-443-9999.

ACCESSIBILITY

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.

TITLE IX

Tisch School of the Arts is dedicated to providing its students with a learning environment that is rigorous, respectful, supportive and nurturing so that they can engage in the free exchange of ideas and commit themselves fully to the study of their discipline. To that end, Tisch is committed to enforcing University policies prohibiting all forms of sexual misconduct as well as discrimination on the basis of sex and gender. Detailed information regarding these policies and the resources that are available to students through the Title IX office can be found by using the following link:

<https://www.nyu.edu/about/policies-guidelines-compliance/equal-opportunity/title9.html>