Top of Mind

- **Assigned Reading:** Hunicke, LeBlanc, Zubek "MDA Framework" + Sellers "Advanced Game Design" Skim Ch.1, read Ch. 2 - *(discuss next week)*
- Bring in a playable prototype of your Tabletop Game
- FILL OUT THIS FORM so you can tell me if I'm giving you too much homework!

Advanced Game Design and Development - Graduate Level

- **Course Code:** 5310
- **Schedule:** Tuesday 5:30-7:45pm
- **Instructor:** John Bezark
- **Office Hours:** Thursday 1-5pm. Book here

Course Description:

This course offers a thorough introduction to the world of video game design and development. Using the Godot 4.5 game engine, students will learn everything from the fundamentals of how to create 2D and 3D video games for any platform, up through advanced game development concepts like systems architecture and programming patterns. Additional topics covered will include introductory 3D, animation, UX/UI, AI agent design, multiplayer and networking and iterative design processes. This course is highly project focused as students will create several different games over the semester including paper prototypes, 2D games, and 3D games. Game design is an inherently collaborative art form, and as such group work will be a major component of the class. No prior game design or development experience is necessary, but a fluency with at least one programming language is required. Even for students who are not ultimately interested in video game design, the skills of systems thinking and systems design, team collaboration and communication and iterative based design process are broadly applicable to many technical fields.

Course Objectives:

- Master the fundamentals of 2D and 3D game development using Godot 4.5.
- Apply advanced game development concepts and programming patterns.
- Develop skills in UX/UI design, multiplayer networking, and 3D
- Execute a comprehensive game project from concept to completion.

Assessment

This advanced graduate course uses **theory as a lens** for game development practice. Each assignment builds on theoretical frameworks while maintaining focus on creating sophisticated games in Godot.

Assignment	Description	Due Date	Points
Class Participation	Active engagement, peer feedback, code reviews	Ongoing	20
Tabletop Game	Systems-focused analog game with playtesting analysis	9/16	14
Lab 1: Godot Foundations	Essential 2D Godot skills and project structure	9/30	5
Lab 2: Emergence	Simple agents producing complex group behavior	10/7	5
Lab 3: Pattern Documentation	Identify and document design patterns while building	10/14	5
2D Game	Complete 2D game demonstrating clean architecture	10/28	34
Lab 4: Walking Sim	3D environmental storytelling experience	11/25	24
Capstone Presentation	Technical presentation of final project architecture	11/18	10
Capstone Game	Sophisticated game showcasing advanced concepts	12/9	33
Total Points Possible			**150**

All assignments will be graded on the due date. Students are always allowed to revise and resubmit work **up until the class before the final.**

Attendence: Students are allowed 3 unexcused absences. After this, each unexcused absence will lower your final grade by 1/2 a letter, i.e. B+ -> B or A -> A-

Course Outline

This advanced course uses **theory as a lens** for game development practice, structured around four theoretical frameworks: - **Weeks 1-3: Systems Thinking** - Understanding games as complex adaptive systems (analog focus)

- **Weeks 4-8: Design Patterns** - Building reusable, maintainable game architectures (2D development) - **Weeks 9-12: 3D Environments & Interaction** - Creating immersive, interactive spaces (3D development)
- **Weeks 13-15: Professional Practice** - Industry context and professional development

Course Philosophy

- **Theory Informs Practice**: Readings provide frameworks for approaching Godot development
- **Background Projects**: Major assignments span multiple weeks like professional development
- **Graduate Depth**: Sophisticated concepts without overwhelming separate academic work
- **Professional Preparation**: Industry-relevant skills and presentation practice throughout

Detailed Weekly Structure

Each 2-hour class is structured as four 30-minute segments: Reading Discussion, Lecture, Workshop, Q&A

PART I: SYSTEMS THINKING (Weeks 1-3)

Understanding games as complex adaptive systems - No digital tools yet

Week 1 9/2 - Introduction & MDA Framework

Assigned Reading: Hunicke, LeBlanc, Zubek "MDA Framework" + Sellers "Advanced Game Design" Skim Ch.1, read Ch. 2 - *(discuss next week)*

- Week 1 Slides
- FILL OUT THIS FORM so you can tell me if I'm giving you too much homework!

Class Structure: 1. **Welcome/Introductions** - Who are you? Why game design? 2. **Lecture:** What is a Game? MDA Framework - Mechanics, Dynamics, Aesthetics 3. **Workshop:** Tabletop game design using MDA 4. **Q&A:** Course expectations, assignment clarification

Project: Tabletop Game **(ASSIGNED)** - Design analog game demonstrating clear MDA structure - No digital tools - focus purely on game logic and player interaction

Week 2 9/9 - Systems Design & Causal Loop Analysis

Assigned Reading: Sellers "Advanced Game Design" Ch.7 (Creating Game Loops) - 12 pages *(discuss next week)*

Class Structure: 1. **Discussion:** MDA Framework and Advanced Game Design reading - applying theory to your tabletop designs - MDA Framework - Systems, Loops and Balance - Advanced Game Design Chapter 2 2. **Playtesting:** Present and playtest tabletop game prototypes with peer feedback 3. **Workshop:** Causal Loop Diagram methodology - learn S/O notation and loop identification 4. **Practice:** Create causal loop diagrams for your tabletop games with instructor guidance

Project: Causal Loop Diagram **(ASSIGNED)** - Create causal loop diagram with S/O notation and labeled feedback loops for your tabletop game - Identify at least two loops: one reinforcing (R) and one balancing (B) - Brief analysis of how loops create intended gameplay dynamics

Week 3 9/16 - Emergence & Complex Systems

Assigned Reading: Johnson "Emergence" Ch.1 (The Myth of the Ant Queen) - *(discuss next week)*

Class Structure: 1. **Discussion:** Advanced Game Design reading - game loops and system architecture 2. **Presentations:** Present tabletop games with system diagrams and MDA analysis 3. **Lecture:** Emergence, complexity from simple rules, bottom-up vs top-down design 4. **Introduction:** Overview of Godot editor (demo only, no hands-on yet)

Due: Tabletop Game + System Diagram

PART II: DESIGN PATTERNS (Weeks 4-8)

Building reusable, maintainable game architectures

Week 4 9/23 - Introduction to Godot & Patterns

Assigned Reading: Alexander "A Pattern Language" Introduction + 3 selected patterns - 15 pages *(discuss next week)*

Class Structure: 1. **Discussion:** Johnson Emergence reading - complexity from simple rules in games 2. **Lecture:** Introduction to Godot editor, nodes, scenes, basic scripting 3. **Workshop:** Hands-on Godot basics - create your first scene with movement 4. **Q&A:** Godot installation issues, basic troubleshooting

Project: Lab 1 - MDA Implementation **(ASSIGNED)** - Analyze existing game using MDA Framework - Create simple Godot prototype demonstrating clear Mechanics→Dynamics→Aesthetics chain **Project:** Lab 2 - Emergent Behavior **(ASSIGNED)** - Create simple agents in Godot that produce complex group behavior from simple rules - Demonstrate emergence principles discussed in Week 3

Week 5 9/30 - Patterns & Component Architecture

Assigned Reading: Nystrom "Game Programming Patterns" - Observer + State patterns - 10 pages *(discuss next week)*

Class Structure: 1. **Discussion:** Alexander reading - how do architectural patterns apply to code? 2. **Lecture:** Godot's signal system, node composition, scene instantiation 3. **Workshop:** Build component-based player controller, practice signals 4. **Code Review:** Share Lab 1 solutions, discuss different approaches to MDA implementation

Due: Lab 1 - MDA Implementation **Project:** Lab 3 - Pattern Documentation **(ASSIGNED)** - Document 3 design patterns you discover while building - Use Alexander format: Context, Problem, Solution **Project:** 2D Game **(ASSIGNED)** - Begin planning and early development - 4+ weeks to develop complete 2D game demonstrating clean architecture - Focus on maintainable, well-organized code using design patterns

Week 6 10/7 - Software Patterns in Practice

Assigned Reading: Martin "Clean Code" Ch.1 (Clean Code) - 8 pages *(discuss next week)*

Class Structure: 1. **Discussion:** Game Programming Patterns reading - Observer & State in your work 2. **Lecture:** State machines, strategy pattern, practical code organization 3. **Workshop:** Implement state machine for AI or player behavior 4. **Peer Review:** Lab 2 presentations - critique emergence implementations

Due: Lab 2 - Emergent Behavior

Week 7 10/14 - Clean Architecture & 2D Game Development

Assigned Reading: McConnell "Code Complete" Ch.22 (Testing) - 15 pages *(discuss next week)*

Class Structure: 1. **Discussion:** Clean Code reading - how does this apply to game development? 2. **Lecture:** 2D game architecture, organizing complex projects, asset management 3. **Workshop:** 2D game development session - implement core mechanics 4. **Individual Check-ins:** Brief meetings on 2D game progress and architecture decisions

Due: Lab 3 - Pattern Documentation

Week 8 10/21 - Integration & Polish Techniques

Assigned Reading: Norman “The Design of Everyday Things” Ch.1 (Psychopathology of Everyday Things) - 12 pages *(discuss next week)*

Class Structure: 1. **Discussion:** Code Complete testing reading - how do you test games? 2. **Lecture:** Game polish, juice, feedback systems, testing strategies for games 3. **Workshop:** 2D game development session - focus on feel and polish 4. **Progress Review:** Show current 2D game state, get feedback on architecture

---

PART III: 3D ENVIRONMENTS & INTERACTION (Weeks 9-12)

Building immersive, interactive spaces

Week 9 10/28 - 2D Game Presentations & 3D Introduction

Assigned Reading: Norman “The Design of Everyday Things” Ch.1 (Psychopathology of Everyday Things) - 12 pages *(discuss next week)*

Class Structure: 1. **Discussion:** Code Complete testing reading - how do you test games? 2. **Presentations:** 2D game presentations and peer review 3. **Lecture:** Introduction to 3D in Godot - scenes, cameras, basic movement 4. **Workshop:** Create your first 3D scene - basic navigation and interaction

Due: 2D Game Complete **Project:** Capstone Game **(ASSIGNED)** - Begin planning sophisticated final project **Project:** Lab 4 - 3D Interactive Scene **(ASSIGNED)** - 3 weeks to develop **Project:** Walking Sim **(ASSIGNED)** - Environmental storytelling experience

Week 10 11/4 - 3D Physics & Spatial Design

Assigned Reading: Reynolds “Steering Behaviors” online article - 8 pages *(discuss next week)*

Class Structure: 1. **Discussion:** Norman reading - how do physical objects suggest their use? 2. **Lecture:** 3D physics systems, rigid bodies, collision detection, environmental design 3. **Workshop:** Build physics-based interactions, experiment with materials and forces 4. **Design Session:** Plan Lab 4 & Walking Sim - what kinds of 3D spaces will you create?

Week 11 11/11 - Environmental Storytelling & AI Agents

Assigned Reading: Jenkins “Game Design as Narrative Architecture” - 15 pages *(discuss next week)*

Class Structure: 1. **Discussion:** Reynolds reading - steering behaviors in your 3D environments 2. **Lecture:** AI agents, navigation meshes, state machines vs behavior trees 3. **Workshop:** Implement basic AI agents with navigation and simple behaviors 4. **Progress Review:** Show Lab 4 & Walking Sim development, get feedback on spatial design

Week 12 11/18 - Capstone Presentations & Professional Development

Assigned Reading: Raymond "The Cathedral and the Bazaar" (excerpts) - 10 pages *(discuss next week)*

Class Structure: 1. **Discussion:** Jenkins reading - how does space tell stories in games? 2. **Presentations:** Capstone project technical presentations 3. **Lecture:** Open source development, Godot's community model, contributing to projects
4. **Workshop:** Final capstone development sprint planning based on presentation feedback

Due: Capstone Presentation

PART IV: PROFESSIONAL PRACTICE (Weeks 13-15)

Industry context and professional development

Week 13 11/25 - Advanced AI & Interactive Systems (Thanksgiving Week- maybe no class...)

Assigned Reading: Isla "Handling Complexity in the Halo 2 AI"(excerpts) - 10 pages *(discuss next week)*

Class Structure: 1. **Discussion:** Cathedral & Bazaar reading - how does open source affect game development? 2. **Lecture:** Advanced AI techniques, perception systems, believable behavior (shortened format due to holiday) 3. **Workshop:** Polish capstone projects - integrate AI agents with environmental storytelling 4. **Individual Check-ins:** Brief meetings on capstone progress and final scope

Due: Walking Sim

Week 14 12/2 - Production & Polish Practices

Assigned Reading: Schell "The Art of Game Design" Ch.27 (Playtesting) - 8 pages *(discuss next week)*

Class Structure: 1. **Discussion:** Cathedral & Bazaar reading - how does open source affect game development? 2. **Lecture:** Production practices, export settings, optimization, professional workflows 3. **Workshop:** Capstone development - polish, optimize, and prepare for release 4. **Individual Check-ins:** Final troubleshooting and scope adjustment for capstone projects

Week 15 12/9 - Final Presentations & Professional Growth

Class Structure: 1. **Discussion:** Playtesting reading - how do you know when a game is "done"? 2. **Final Presentations:** Capstone game presentations with live gameplay demos 3. **Retrospective:** Course reflection using theoretical frameworks as analysis tools 4. **Professional Development:** Portfolio building, continuing education, career paths in game development

Due: Capstone Game + Post-mortem - Completed game demonstrating technical and design growth - Written retrospective analyzing your development process using course frameworks

Necessary Materials:

- Godot 4.5 : An open source 2D, 3D and XR game engine.
- Github Desktop: an industry standard version control system

Tools Used

- Godot 4.5 : An open source 2D, 3D and XR game engine.
- Blender : an open source 3D modeling and animation program.
- Github Desktop: an industry standard version control system

Godot

- Source Code
- Docs
- Demo Projects
- Forums
- Godot Discord
- **TUTORIALS**

Git & Github

- Dan Shiffman's Github for Poets

Instructor Details

John Bezark (he/they) is a Brooklyn based Game Designer, Creative Technologist, Immersive Experience Designer and Video Artist. They graduated from NYU's ITP with an MPS in Interactive Telecommunications and from Fordham

University with a BA in Theatrical Directing. John has made everything from Solar Powered Websites to Immersive Game-shows about Systems Thinking.

Nowadays they like to use the Godot Game Engine to create one of a kind immersive and interactive experiences, games and performance art.

When not creating weird things, John also really likes to work in education and he's passionate about teaching others how to design immersive spaces, how to make games and how to see the world through the lens of Systems Thinking. **Email: jbezark@fordham.edu**

Office Hours: Thursday 1-5pm. Book here