

Schedule



A porcupine wearing a backpack and a hardhat with a light looking for the loot, digital art

The schedule is broken down into core pieces: Breakdown of the week, **Weekly Learning Objectives**, Stake Assignment Opportunities, Game Engine Systems, Program Design Patterns, Data Standards, Random Access Technology, & Live Demonstrations. Everything will be nested under the weekly dates and some items are subject to change as we move through the class. I will make sure to update you all if there is a change in the schedule - as I do have some travel for work coming up that will have at least one of the classes to be used to work on assignments and/or work remote.

0.0.1 August 29 & August 31

Overall TL;DR: *The name of the game is continuous iteration, we all have different starting points and we all have different paces, but as long as you keep on iterating and keep creating/building you will be okay, and you will succeed in this class.*

Monday Class Associated with introducing me and my philosophy, grading agreements, and software installs. Install Unity Hub, VSCode, Visual Studio 2022 Community, setup Git, and GitHub. Register and sign-up for GitHub classroom. Identify what this class is and make sure those who are registered understand what they are getting into. This class doesn't come with a programming prerequisite, but it will by the Fall of 2023. That's me drawing a line in the sand! I will do everything in my power to include visual programming concepts from Unity to help directly offset this skills gap that a majority of you might have but I also will not jeopardize the class potential for the lack of University guidance at this point in the game.

Wednesday Class Associated with software account setups. Unity and GitHub. Help me confirm if the lab machines have all of the basic software installed. I will talk about Disciplined Agile Delivery and Kanban boards, we will at a high level cover version control software (VCS), and will get into how to access Unity educational material.

0.0.1 Weekly Learning Objectives

TL;DR: *In other words: overview of me and the class and I am going to be relying on you all to help me test out my teaching methods, and in return, you will learn, build confidence, and have the opportunity to get some professional advice from me down the road.*

I am approaching this class in a different way for the first time teaching it - you all are going to be my indefinite test class and I am going to be in your debt of which you can use that token (after you've completed the class) to get some professional advice from me whenever you need it - one shot/opportunity for me to go full brain to help from a structure/systems perspective: limit to 2 hours! You can also waste that token and ask my opinion on whatever you want to talk about and I will gladly give you my perspective on that, but make sure you

preference your request accordingly. This week is more about setup and understanding my approach. This week is not a 'free' week - this week is for you to absorb the approach and identify if this is right for you. Everything I'm presenting already exists across the internet - hidden in YouTube videos, software documentation, books, forums, etc. If you're taking this class to force you to learn a game engine: I suggest you not take it because you're just wasting your money.

Talk about community building, talk about non-grading agreements and introduce the larger plan for VMASC/ODU and the game design group. This class is tied to a final project, working within version control, working within a Project/Board, and we will learn Unity from Unity's material and with my guidance. I will teach important software/game design patterns and talk about how these patterns help me everyday on my projects, and will introduce you to the Gang of Four (GoF) design patterns that somehow I missed until later in life because I have strong opinions on middle tier state higher education programs and computer technology bureaucracy.

0.0.1 Stake Assignment Opportunities

- Time in class to setup your GitHub profile page.
- Time in class to connect the GitHub classroom account with your GitHub account.
- Time in class to review high level Markdown

0.0.1 Game Engine Systems

TL;DR: *Game engines are extremely complicated software that are constantly changing. It is important to learn how to use them now and to keep up with the changes that will happen over the next few years as it will drastically change*

Right out the gate: I want you to understand that a game engine isn't just a 'game engine', that it is some of the most complicated and advanced software you can interact with on the market today. I want to be very honest with you all, over the next 5-10 years game engines will be experiencing massive changes. I do not know what's going to come out the other end. As we start to have companies put more emphasis on spatial computing, game engines will start to solidify as one pathway to help, but you will also need to keep an eye on the sheer massive amount of web developers that exist. Overnight these web developers will become game engine architects as game engines go full blown web. There are no guarantees right now on a 'winner' - Unreal vs Unity vs Godot vs Lumberyard vs Minecraft vs Garrys Mod? What's important right now, is to be creating, tinkering, building, and learning within one of these major engines right now! Identifying how you're going to take that content and present it to the world (build your portfolio!) If you aren't: this class will force you too, if you still don't think that's the case, please just drop the class and take something else - save me and you the headache. Okay, now you've picked an engine, now go Create! YouTube it. Use the educational tutorials provided by the companies. Break it. Be ready to keep constantly maintaining a pace of learning and iterating over the next 5 years of your development cycle. This is vitally important to anything else you're doing. You have to maintain this consistent pace, or you will fall behind, and in this industry falling behind can alter your opportunities. I can guarantee that as long as you're not an ass-hat, you will have opportunities within this area, and if you're willing to travel and/or do freelance work early in your career, you will have a lot more options. I cannot stress enough the importance of continuously building content in 'real-time sim fuck the world' engines.

What you are seeing with SnapChat and Niantic are prime examples of what we used to think of 'you can only do that with a game engine or a real-time light engine' are now being rolled out as almost full 'engines' of themselves. What happens when Apple announces their next updated tooling platforms and/or their "engine" for spatial experiences? How it requires minimal code, and all previous iOS developers can immediately take advantage of their new 'whatever not a game engine, game engine' you're going to watch the indie world go up in flames with

an over-saturation of 'everyone is a game developer' and 'no one is a game developer' worry not, you are okay, we are in the right place, you're not alone, and now join me in what I call my day job!

These engines that build incredible experiences are always breaking, they are always "building the plane while flying it" and in almost all cases, the correct way to manage them, is to not stop using them - to hunt down the forums, to post dumb questions, to answer questions no one has answered yet, and get used to reading changelogs and 'known bugs'. The best advice I can give you up front, is one that works for all areas of projects/work, show up. Open that project at least once a week, and explore it, even if you don't put any 'new work' into it, the habit of visiting the project, opening it up, going through it, is very important to understanding how these complicated systems all manage to come together. These tools are here to help us creatives deliver insane off-the wall solutions to almost everything. Game engine is not the right term - not sure what the term is - we can come up with it in this class as a hypothetical assignment if we want - prototype engine? real-time sim engine? user experience world creator? whatcha ma call it generator? all of these are right in some form or fashion.

0.0.1 Program Design Patterns

What the hell is a program design pattern? Why should you care? Why did it take me years after my masters degree to come to find these, why aren't we all learning the program patterns best for game design? You're not a programmer - no problem - you're still going to learn them but before we go after specific, well-known program design patterns, we need to address a simple concept: formats, standards, and then data serialization practices

0.0.1 Data Standards

Markdown! High level dive into Markdown and why I enjoy it and why GitHub loves it. For more information on Markdown please see [MarkdownGuide](#)

0.0.1 Random Access Technology & Live Demonstration

I will demonstrate Stable Diffusion running on my laptop. Stable Diffusion is a 'product' that is a machine learning model that uses text to generate images in what's called a 'latent diffusion model'. [Stable Diffusion](#) is just one of a few that are out there. Others you can look up are [Dalle](#) and [MidJourney](#). If we have enough time I will take some prompts from you all and generate some pretty interesting images 



A bear wearing a backpack and a hardhat with a light looking for the loot, digital art

0.0.2 September 5 (Holiday) & September 7

Overall TL;DR: Everyone that touches software needs to use Git and needs to understand how data flows within their system - talk about data serialization and why it's useful and how it can help become a quick way to save your experience data for easy loading and unloading.

Monday Class It be a holiday! Enjoy it, be in the moment, bridge the technology medium with a natural medium and use it as an opportunity to enjoy the outside.

Wednesday Class Version control & covering Pro Git Resource Material. What is git? Why should you care? Git the world? Basic workflows. GitHub: fancy git with our friendly neighborhood 'cloud'. GitHub Classroom setup and Opening up Unity.

0.0.2 Weekly Learning Objectives

- Documentation, manuals, and where to get help with Unity and GitHub, how to use the forums and social media for help/inspiration.
- Unity Education
- Talk about data and standards around data.
 - JSON, XML, YAML, Markdown, general data standards and showcase serialization

0.0.2 Stake Assignment Opportunities

- GitHub Project Management: Required Medium Stake Assignment
- Look into GitHub additional training: Optional Low Stake Assignment
- Look into Communication training: Optional Low Stake Assignment
- Work towards data standards: Optional Medium Stake Assignment
- Introduction to a couple Game Design Program Patterns: Required Medium Stake Assignment

0.0.2 Game Engine Systems

Game Engine Help 101. For the most part, if you have a problem with Unity, someone else has had a similar problem and the community has probably identified a solution and/or an alternative to what you're trying to do. This is more about how to search and find your solutions.

- Game Engine Documentation.
- Game Engine Versioning.
- Game Engine ChangeLog.
- Game Engine Forums and Community.

0.0.2 Program Design Patterns

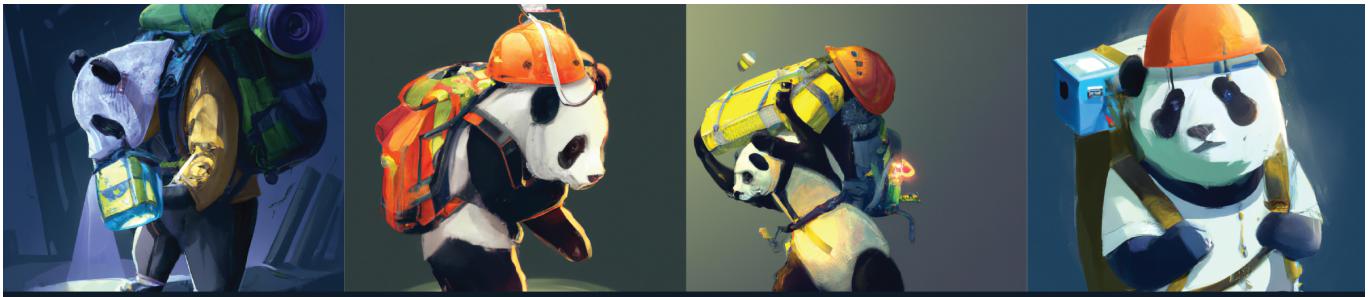
Serialization and Data - think about your software and your game as what data is needed, when is it needed, how do you move that data between your software, and who needs access to what data and how this connects to what is called 'The Singleton Pattern'. We won't get too deep on the programming pattern but it will come back later and I encourage you all to understand this specific pattern out of all of them.

0.0.2 Data Standards

JSON and Serialization. We will talk about other data standards but will focus in on JSON and when we say serialization what we mean normally and what we mean for '[Unity Serialization](#)'.

0.0.2 Random Access Technology & Live Demonstration

Will show off a simple Unity Package that I created that helps on the serialization side and on the singleton side. We will come back to this after we dive more into Unity. Time permitting I will then show you GitHub Co-pilot :hugs:



A panda wearing a backpack and a hardhat with a light looking for the loot, digital art

0.0.3 September 12 & September 14

TL;DR: You want to withdrawal? Do it before the 13th! Oh, you want to stay with us, lets really get into the Unity Interface! Let's actually begin using Unity and work through setting up a project

Unity Education Get the Setup Wizard Monday Class First portion of the class will be me explaining the updates based on the student questionnaires then it's Unity Time. Tag along and go with me while we explore Unity's interface and go all Unity 101-Introduction with going through sort of Unity Editor 101 and setting up Unity with your external tools: Edit→Preferences→External Tools and getting right into the accessibility features for the actual editor software: Edit→Preferences→Colors (accessibility issues). Go through the Unity Package Manager and the Manifest file, oh snap is this JSON... now I understand why John was such a JSON :clown_face: last week! Go into the basic windows, the toolbar, navigating within the software, hot keys, adjustments you might want to be aware of, and other misc. Unity things I've dealt with. Dive into how you can setup Unity for Git and GitLFS. Introduce Unity Templates and Unity EDU site and how you can import/export packages the old way and the new and improved way!

Wednesday Class Setting up a Unity Project with Version Control and Unity Project Templates, you get to now [make your own project template!](#) Setting up a blank Unity Project and importing assets and a sort of general project management approach to folders and structuring folders within a Unity Project. Use Wednesday to recap what we don't get to on Monday and time permitting open up some Unity Templates.

0.0.3 Weekly Learning Objectives

- Basics of Unity Editor
- High Level Package Manager Concepts
- High Level Content Delivery Services like Unity Addressables
- Custom Unity Template Creation
- Integrating Version Control and Unity

0.0.3 Stake Assignment Opportunities

- Identify a Unity beginner project that aligns with your Unity Learn/Certification Path: High Stake Assignment
- Work on setting up a GitHub Project Board: Required Medium Stake Assignment
- Introduction to a couple Game Design Program Patterns: Required Medium Stake Assignment

0.0.3 Game Engine Systems

This week will be more focused on Unity basics and the Editor - as we talk about high level software delivery and packaged content I will briefly talk about Unity Addressables - we will get much deeper into this later in the semester.

0.0.3 Program Design Patterns

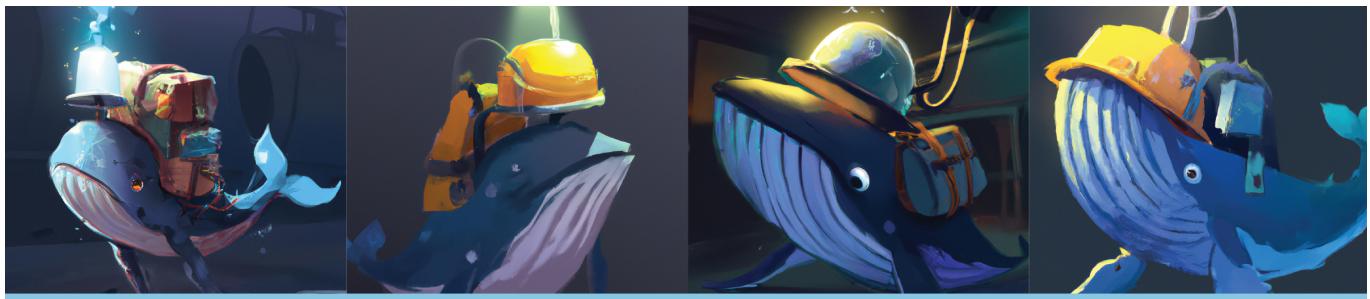
This week will be more focused on the Package Manager and as a high level software concept: 'delivering content' how Unity does this, how modern games do it, and why you can do it as well.

0.0.3 Data Standards

This week will be more a focus on how JSON and YAML are behind a lot of Unity's systems. I am not planning on introducing anything 'new' here.

0.0.3 Random Access Technology & Live Demonstration

Time permitting: this might get pushed to next week, but between this week and next I will show off a very much in development current project that uses a 1:64 scale 3D printed laser cut model, Apple ARKit, an iPad, custom trackable markers, and a Unity Application that helps explain and visualize an interesting solution to warehouse inventory using leading edge 5G architecture to help a Warehouse facility that is being upgraded for the Marine Corps.



A whale wearing a backpack and a hardhat with a light looking for the loot, digital art

0.0.4 September 19 & September 21

TL;DR: Basically doing what we did last week but more! We will keep exploring the interface, this week builds on the previous as we will look more into Unity Educational information and how it ties into Critical Thinking concepts associated with Game Design and Software Development.

Monday Class Unity Input Systems, Message Systems, and Unity Events **Wednesday Class** Unity Visual Scripting System and Input to make *something* move

0.0.4 Weekly Learning Objectives

- Furthering of Unity Editor understanding
- Core Unity Systems that let you get up and get some input into Unity and output reactions
- Get Familiar with Unity Input Manager, Unity Event Manager, and Unity Messaging Systems

0.0.4 Stake Assignment Opportunities

- At this point I should have met with each of you for a one-on-one for your Grading Agreement Check-ins:
Required Low Stake Assignment
- Identify a Unity beginner project that aligns with your Unity Learn/Certification Path: High Stake Assignment
- Work on setting up a GitHub Project Board: Required Medium Stake Assignment
- Introduction to a couple Game Design Program Patterns: Required Medium Stake Assignment
- Work towards data standards: Optional Medium Stake Assignment

0.0.4 Game Engine Systems

This week will still be focused on Unity basics and the Editor - as we get deeper into some of the first unity systems, I will go more into a game manager concept within unity and how this can connect to the delivery and packaged content we talked about last week. I will reference Unity Addressables some more and then focus in on how these tie into program design patterns

0.0.4 Program Design Patterns

I will spend more time on Singleton concepts here and show how a singleton is a great approach to managing a first sort of experience within Unity. Go into the pros/cons and how other people have used it.

0.0.4 Data Standards

Continuation of JSON, YAML, and Markdown. I am not planning on introducing anything 'new' here but we will explore in more depth the newer Unity Input Manager and what's going on behind the scenes from a data standard perspective and how you can break/manipulate that as needed at runtime.

0.0.4 Random Access Technology & Live Demonstrations

Time permitting - I might have not gotten to the previous week's demo: if I did I will introduce the updates on that and bring in a HoloLens 2 and briefly talk about Spatial Computing and why you all should be thinking about.



An octopus wearing a backpack and a hardhat with a light looking for the loot, digital art

0.0.5 September 26 & September 28 (NO PHYSICAL CLASS)

TL;DR: John is out of town for Wednesday Unity Visual Scripting is probably going to consume all of us, let's start on Monday and use Wednesday for you to work on your own via some online educational pathways via Unity Learn.

Monday Class Unity Visual Scripting System and sequencing of events by state machines **Wednesday Class NO PHYSICAL CLASS** please use the classroom if you need it, but I will not be there, use the time to work on Unity Learn.

0.0.5 Weekly Learning Objectives

- Unity Visual Scripting System
- Unity Versions and Package Manager Versions

0.0.5 Stake Assignment Opportunities

- Keep maintaining your GitHub Project Board: Required Medium Stake Assignment
- Introduction to a couple Game Design Program Patterns: Required Medium Stake Assignment
- Work towards data standards: Optional Medium Stake Assignment
- Start thinking about the bigger end project: High Stake Assignment

0.0.5 Game Engine Systems

Visual Scripting Interfaces - they are everywhere now in software as more and more companies and software are recognizing and addressing an easier way to get into custom software. Within entertainment and virtual production these tools have been around for a while - a graph based system that uses nodes and edges that takes the output from one node and processes a mathematical function on the next node and dumps out the next output.

0.0.5 Program Design Patterns

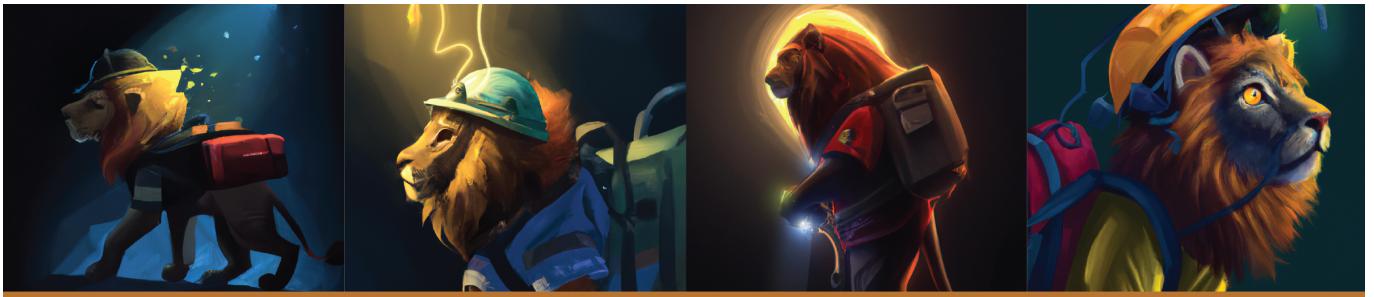
None this week as I'm out for travel and I want you all to focus in one Unity Visual Scripting and Scripting in general

0.0.5 Data Standards

I will probably introduce C# structs here as a light weight way of storing runtime data.

0.0.5 Random Access Technology & Live Demonstrations

There won't be any live demonstrations this week due to travel - I plan on bribing you all when I get back with opportunities to win conference giveaway junk that I bring back from Las Vegas 🚀



A lion wearing a backpack and a hardhat with a light looking for the loot, digital art

0.0.6 October 3 & October 5

TL;DR: Unity, feeling dangerous, bring it all together to create a blank version controlled project, rip out Lego Assets, and create some custom scripting (visual scripting or traditional) to invoke the Unity Event Management System.

Monday Class Continue to work through Visual Scripting and lets make sure we are still aligned with our Unity Learn environment - work through Raycasts and introduction to a very small piece of Unity Physics, talk about Systems Dynamics and State Machines, and continue on some Unity scripting fundamentals **Wednesday Class** Continue on the concept of a State Machine, and the rest of class is to work towards your project/high stake deliverable.

0.0.6 Weekly Learning Objectives

- Unity Raycast and Unity Physics
- Unity Visual Scripting System
- Exporting Unity Asset Packages
- Modification of existing work
- Image data standards and formats

0.0.6 Stake Assignment Opportunities

- Time to focus on the core project: High Stake Requirement
- Register for Unity Unite: High Stake Requirement

0.0.6 Game Engine Systems

- Raycasts and collision systems are one of the core systems to all game engines. We will start exploring the physics system within unity but under the use case of raycasts. Why they are awesome, how you can use them for all sorts of fun things, and why basics of linear algebra should be understood.

0.0.6 Program Design Patterns

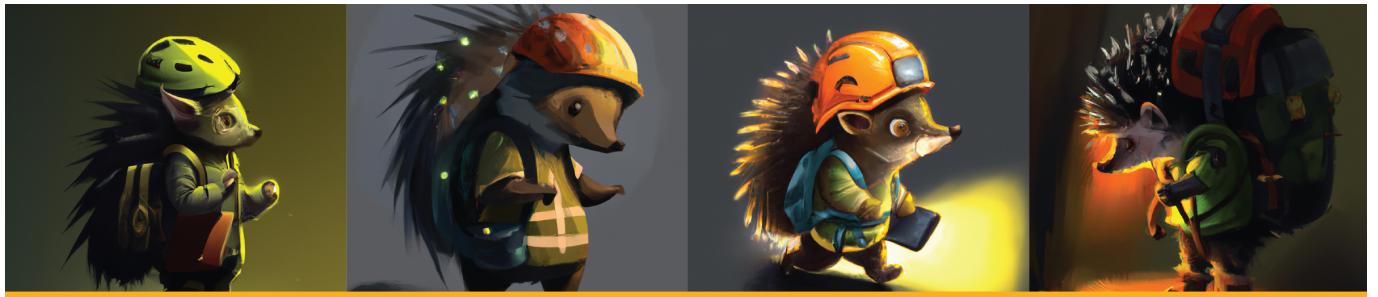
State Machines: what is state? why is it everywhere? Why should you care? Where does one state end and one begin? What is system dynamics and why it's important to look into ways you can use these patterns to conceptually think through your projects needs. In a lot of cases you don't need to be the one to program this - a lot of this stuff already exists. It's about how to use these program design patterns to talk with other individuals and/or find the solutions you need to your problem.

0.0.6 Data Standards

Image data standards and 360 image standards. Briefly talk about PNG/JPG and warping to get you to a 360 image.

0.0.6 Random Access Technology & Live Demonstrations

Will introduce GPT-3. A natural language processing tool that easy to use - paid service by OpenAI but there are a lot of other options out there. Run through some prompts and how this will change real-time story telling forever and give your characters that can talk about stuff... forever. Every walk around a game and approach an NPC only to get the same prompt? Well services like GPT-3 give us a way to give you a lot more depth on those characters.



A porcupine wearing a backpack and a hardhat with a light looking for the loot, digital art

0.0.7 October 10 && October 12

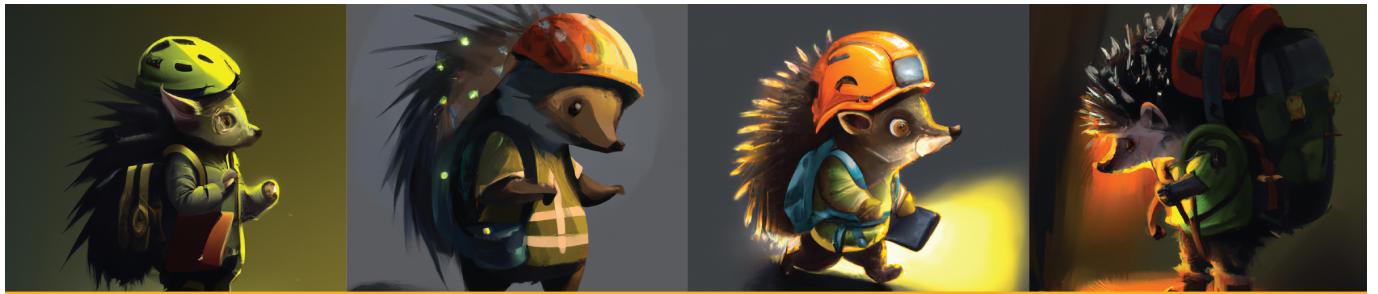
TL;DR: Enjoying the break, decisions made before the break come back to haunt you and/or you have lost all hope.

Monday Class It's fall break, but it probably doesn't feel like fall yet, so enjoy your late summer not quiet fall yet break, and whatever you do, avoid Unity at all costs! Unless you decided to put off your first high stakes deliverable, in that case, you better be ready to werk!

Wednesday Class Time in class to revisit any/all of the previous concepts and if you needed that additional time to work on your high stakes deliverable you can use the class towards that effort.

Fall Break Restructure

Currently it's 8/31/2022 and I am going to not flush out the rest of the calendar as I will be evolving the remaining weeks tied to how our progress is and how we feel as a class a this time to check in via Fallbreak. This gives me roughly a 7 week lead time to build out weekly content around the high level information broken down below. I will break it out every week in multiple chunks as we get closer to fall break. Once we hit fall break I will use that week to make any sort of pivots and we will go from there. It's a good time to do this, as we are around the midterm window and it's a good time to reflect and pivot as needed. If you have any additional questions let me know.



A porcupine wearing a backpack and a hardhat with a light looking for the loot, digital art

0.0.8 October 17 & October 19

Monday Class Back to the board: Asset Management Systems, Unity Package Manager, and what would you do differently given this information? Possible call in from Industry expert.

Wednesday Class Project Disclosure Feedback time: using the class time to schedule 1 vs 1 time with me to work through scope, expectations, and the overall project plan, while we are doing this the rest of you will be working on modifying your first high stakes deliverable to be the starting point for your nice and clean Unity Template and Unity Project.

0.0.9 October 24 & October 26

Monday Class Unity 07-Physics & Animation, going to skirt right into need to know sort of concepts, as these systems could be broken out into their own 16 week sessions. Possible call in from Industry expert.

Wednesday Class Physics & Animation Continue: Planning and lab time to test out prototypes and concepts: Address your project major system needs, are you planning on using physics? What does your input look like? What's a core game mechanic you're adopting?

0.0.10 October 31 & November 2

Monday Class Halloween Audio Systems **Wednesday Class** Partial class: Revisit basics of Unity UI and how these are nothing but glorified Event Managers and Message Systems.

0.0.11 November 7 & November 9

Monday Class VFX Systems and Particle Systems **Wednesday Class** Partial class: revisit visual scripting again



A koala wearing a backpack and a hardhat with a light looking for the loot, digital art

0.0.12 November 14 & November 16

Monday Class Unity Graphics Pipeline **Wednesday Class** Unity Build Pipeline and versioning practices, semantics across minor, major, epic, sprint, feature, and releases.

0.0.13 November 21 & November 23

Monday Class Time in lab for High Stakes Deliverable work - AMA with John! **Wednesday Class** Holiday - save a turkey - eat some veggies!

0.0.14 November 28 & November 30

Monday Class Partial class: Rapid Wheel Spin to dive deeper in one of the many systems we've covered, aka Flex time. Rest of class time for final project work. **Wednesday Class** Time in lab for final project work and 1 vs 1 updates. Schedule time to meet with me to address how far the wagon has fallen off, what have you compromised, and what is now on the backlog.

0.0.15 December 5 & December 7

Monday Class Time in lab for final project work and continue 1 vs 1

Wednesday Class Time in lab for final project work, John buys something random and gives it away, maybe there's food, maybe there's a clown who does balloons, maybe I schedule cats to hang out, who knows...

0.0.16 December 16th

Friday 12-16-2022: "SHOW ME WHAT YOU GOT"