

# GPU Architecture

Winter 2021

GAM 576

**instructor:** Ed Keenan  
**email:** [ekeenan2@cdm.depaul.edu](mailto:ekeenan2@cdm.depaul.edu)  
**office hours:** REMOTE: Tuesdays 9:15-11:15pm Synchronous Zoom Meeting or by appointment  
**office:** CDM 830  
**phone:** (312)362-6747  
**website:** [piazza.com/depaul/winter2021/gam576](https://piazza.com/depaul/winter2021/gam576) (Preferred communication)  
**lecture:** REMOTE: Tuesdays 6:00-9:00pm Synchronous Zoom Meeting  
**Desired to Learn (D2L):** [d2l.depaul.edu](https://d2l.depaul.edu) (Grades, Viewing lectures, Announcements)  
**Version Control:** perforce: **140.192.39.61:1666**  
**All Communication:** Piazza for all communication, reply within 24 hrs during school week

## Description:

Real-time application GPU development. Topics include GPU pipeline, data parallelization, hierarchy data flow on GPU, asset compression, non-graphics data processing, graphics rendering and data conversion. Design and implementation of parallel GPU applications across multiple GPU cores. Exploration of different parallelization techniques for large (non-graphical) computationally heavy systems. Developing and debugging blind data flow on GPU hardware. Students will design, develop and implement a real-time GPU application (i.e. Skinned Animation rendering engine) that integrates large converted data assets using 100s of parallel GPU cores.

## Prerequisites:

- GAM 575 Game Programming Engine II

## Learning Goals:

- Students will be able to design and implement a large complex software system across CPU and GPU systems.
- Students will understand GPU graphics architecture including the Vertex, Fragment, and Compute shaders.
- Students will explore parallelization techniques to distribute computation across multiple GPU cores.
- Student will implement computational heavy (non-graphical) operations on 100s of GPU cores.
- Students will implement off-line data conversion for GPU parallelized data assets with different compression techniques.
- Students will develop a run-time graphics engine that supports 2D, 3D, and Keyframe animation.

## Grading

### 90% - Milestones

- (30%) Hybrid Engine { 2D, 3D animation & 3D static models}
- (30%) GPU Skinned Model Rendering
- (30%) GPU/CPU Animation Compression
- Needed for all:
  - Conversion tool, compression and runtime implementation for skinned models with keyframed animation data

### 10% - Individual Research project

- Project proposed by student and approved by instructor

## Textbooks and printed resources

Additional course material will be many supplied through class notes, handouts or online links.

- 2 Required Books
  - Game Engine Architecture, 2nd edition, Gregory, A.K. Peters Ltd., 2015
    - ISBN: 978-1466560017
  - OpenGL® SuperBible: Comprehensive Tutorial and Reference, 6th edition, Wright, Lipchak, Sellers & Haemel, Addison-Wesley Prof./Pearson, 2014.
    - ISBN: 978-0321902948
- Various GPU and Shading cookbooks
  - To be researched and determined
  - Online material

### **Additional Material**

- Will be provided by the instructor
- Lectures, links, SDKs and other corresponding material

## Software

- **Microsoft Visual Studio 2019 Enterprise Edition (not Community)**
  - [MSDNAA Depaul – Visual Studio 2019 Enterprise](#)
    - C++ and C# install (future classes)
  - Any other variants are not used in this class
  - Students are responsible keeping their development tools working
- **Perforce Server**
  - Download and configuration instructions will be provided in class
  - **Perforce – Helix Visual Client (p4v)**
    - <https://www.perforce.com/downloads/helix-visual-client-p4v>
    - Server address: **perforce.dpu.depaul.edu:1666**

## Topics will include:

- ***Tentative Lecture Topics:***
  - 2D/3D hybrid engine {simultaneously all the below}
    - 2D rendering
    - 2D Fonts
    - 3D model
    - 3D animation
  - Skinning Models
    - extraction of data
    - reducing the number of influence
    - runtime layout
    - compression
  - Advanced Keyframe Animation
    - compression
    - data layout
    - CPU/GPU issues
    - runtime formats
  - Scene Graph
    - agnostic tree layout for heterogeneous data types
    - models, skeletons, animations, cameras
    - Spacial layout
    - Quad/Octal trees, Feudal, Binary Spacial Partitioning
    - Bounding volumes
  - Object Management
    - Layout / organization
    - reflection / communication
    - Self describing
    - Protocol buffers
  - Cameras
    - Movement - paths
    - Auto-targeting
    - Spring (dynamic follow)

## Required Programming Assignments – 90%

Three required major milestones, each milestone builds on the previous milestone. There are mini checkpoints to make sure the student is on convergence path between milestones.

- **Milestone 1:** Hybrid game engine 30%
  - 2D Game Engine with Sprites
    - Font systems
  - 3D Engine with 2 models being displayed
  - 3D Animation with support 2 or more skeletons on the same screen
    - Swappable animations on each skeleton
  - No Memory Leaks!!
- **Milestone 2:** Model Skinning - 30%
  - Write exporter tool to extract Skinned model data
    - Reduce the number of influences from N to 4 per vertex
    - Create a runtime format
  - Create the corresponding GPU code for the skinned data
    - Rework the game Engine to load the skinned data
    - Create the corresponding GPU systems to have the skinned data automated
- **Milestone 3:** Advanced Animation - 30%
  - Write a an animation extractor tool that exports compressed keyframe animation
    - Export animation data for a FBX file
    - Compressed the data using an offline tool
    - Convert the runtime to process the compressed data
      - Combination of CPU / GPU code
    - Replay animations using GPU code
    - Add blending between 2 or more separate animations

## Individual Research Assignment – 10%

Individual Research project. There are mini checkpoints to make sure the student is on convergence path between milestones.

- **Indv. Milestone 1:** Student's choice
  - Student picks a topic that requires research and software experimentation
    - Will look at current industry/academic practices
    - Write a summary / present summary of research material
  - Create a prototype of technology or show a demo of the technology
  - Project needs to be approved by professor

## Perforce Submissions

- Everyone is expected to submit several submissions to perforce a week.
  - Minimum of **five** significant (real) submissions on **three** separate days.
  - To promote incremental development and prevent last day rush.
  - Grade deduction will occur if not followed
- The biggest reason students get into trouble with software design:
  - Not starting the project early
  - Not working on the material frequently enough
  - Taking too large of a bite(byte) of the design
- Both are minimized with this Perforce RULE
- Even my simplest programs take 10-20 submissions.
  - For these project assignments, my average is 40-400 submissions, so five will be no problem.
- Detailed perforce changelist comments are expected

## Piazza Discussion forum

- Previous classes have highly participated in Piazza and did quite well
- Statistics show: students who participate more and help other students do better!
  - The correlation is ridiculous!
    - Poor understanding / poor participation.
    - Great understanding / Great participation
  - As you master the material, help others learn!
    - Want to be a Master programmer so master it!
- Everyone is expected and encouraged to participate on the Piazza discussion forum. All class-related discussion here this term.
- Everyone is expected to keep up with the material on Piazza and are responsible for its content. Critical class updates and directions will be presented there.
  - Not participating or reading the material on Piazza is **NOT** an **Excuse**.
- All correspondence that is not personal in nature should be vectored through Piazza
  - Sensitive material, use Piazza private note, not email.
- The quicker you begin asking questions on Piazza (rather than via emails), the quicker you will benefit from the collective knowledge of your classmates and instructors. I encourage you to ask questions when you are struggling to understand a concept.
- Keep the forum professional and positive, help each other out.
  - Karma really pays off here.
  - Help each other whenever you can.
    - There will be a time when you will need help from the class (trust me).

NOTE: Do **NOT** post until you have watched the entire lecture **FIRST** (in class or online)

This will prevent frustration on all sides (members asking or answering questions)

## Collaborating together on programming assignments

- You are encouraged to work together
  - Use the Piazza forums heavy
  - Even share your material with others in the common directory
    - Obviously not the answers
- Everyone is 100% responsible for the work they do.
  - If you get help with a section of code,
  - Please refactor the code the **snot out of it**
    - Comment and understand that material
    - Transform the code to **make it yours**
  - Be able to answer **any** question regarding the code you commit
- System for Detecting Software Plagiarism
  - We will be using MOSS - Measure of Software Similarity (Stanford University)
    - Indicates possible code infringements (plagiarism)
    - MOSS - will detect the similarity independent of naming convention, indentation style or formatting, it compares abstract syntax tree of your code.
  - I will pursue any plagiarism/integrity violations aggressively, arguing for full expulsion from the university for the offenders.
    - Don't put me or you in this scenario
    - Any integrity violations will result in a failing grade for the course
- If you gain significant support / help from another student or website
  - Fully disclose the support / help you had in a Readme.txt file submitted with your assignments.
    - Disclosing the help, is **not permission** for copying the code.
    - Only there to clarify and acknowledge help you were given from a fellow student.
- Modifying any Unit Test or Project setting to alter the outcome results is also an **Academic Integrity Violation**
- If you are stuck and find yourself even tempted to plagiarize
  - Ask for help!!!!
    - Use on Piazza -> Visit during offices hours, make an appointment
    - **Don't ever compromise your integrity!**
- Material was uniquely created for this Class.
  - By the process of tuition, you "paid" for the contents and material of this class.
    - Do not share this **copyrighted** material in any form
    - It is design for your personal use, while enrolled in the Class.
  - Do **NOT** post any content or revealing material to any external website or forum outside of this class.
    - The Class Piazza forum is provided for this service, ask questions there, not on the internet (i.e. StackOverflow and other software forums)
- After you leave this class
  - You are expressly **FORBIDDEN** to provide or share the content with others.

- Academic Integrity Violations can still be applied to students who provide material support to other students even after completion of the class.
- Just follow the golden rule:
  - *"I have neither given, nor received, nor have I tolerated others' use of unauthorized aid."*

## Miscellaneous

- **Late Policies**
  - Due dates and times are verified by the submission record on the Perforce Server
    - No extensions are allowed
  - All assignments need to compile without warnings
    - Failure to compile "as-is" results in a 0 for the grade
- **Memory Leaking**
  - For assignments that have memory tracking enabled
    - If an assignment is determined that its leaking memory
      - A deduction of 20% is applied to the grade of that assignment
  - Leaking status is provided during development
- **Crashing**
  - Assignments are expected to work for a set duration  
(long enough to demo all the features)
    - A grade of 0 is given to any project that throws an exception, ends unexpectedly, crashes or hangs (not proceeding forward).
    - Crash – program locking up or quitting unexpectedly
- **Integrity Violation**
  - Any form of integrity violation will receive an "F" letter grade for the course, no exceptions
  - All material submitted is from this current offering of class, any material from the outside is considered a violation

## Tentative Schedule:

Week	Lecture	Assign	Due
1	Engine Review 2D Sprites Topics / Layout	Project Selection Engine Clean Up (Hybrid Engine)	
2	Font system Export Skinning		
3	Skinning GPU coding GPU and data implementation design	(Skinning GPU)	<b>(MS1 – Hybrid)</b>
4	Animation Compression Skinning cleanup		
5	GPU architecture / flow Data Driven GPU		
6	GPU animation Blending Animation Intro Compute Shaders	(Animation GPU)	<b>(MS2 – Skinning)</b>
7	Compute Shaders applied GPU data layout CPU/GPU dependency		
8	Object Management Protocol Buffers Presentation techniques		
9	Design issues with GPU development General purpose GPU acceleration		Feedback Review
10	Open Topics Presentations PART A		Presentations begin
11	Presentations PART B		<b>(MS3 – Animation GPU)</b> Student write up

University Dates (Drop, Withdrawal, Audit, Exam)

- <https://academics.depaul.edu/calendar/Pages/default.aspx>



## Course Policies

### Changes to Syllabus

This syllabus is subject to change as necessary during the quarter. If a change occurs, it will be thoroughly addressed during class, posted under Announcements in D2L and sent via email.

### Online Course Evaluations

Evaluations are a way for students to provide valuable feedback regarding their instructor and the course. Detailed feedback will enable the instructor to continuously tailor teaching methods and course content to meet the learning goals of the course and the academic needs of the students. They are a requirement of the course and are key to continue to provide you with the highest quality of teaching. The evaluations are anonymous; the instructor and administration do not track who entered what responses. A program is used to check if the student completed the evaluations, but the evaluation is completely separate from the student's identity. Since 100% participation is our goal, students are sent periodic reminders over three weeks. Students do not receive reminders once they complete the evaluation. Please see <https://resources.depaul.edu/teaching-commons/teaching/Pages/online-teaching-evaluations.aspx> for additional information.

### Academic Integrity and Plagiarism

This course will be subject to the university's academic integrity policy. More information can be found at <https://resources.depaul.edu/teaching-commons/teaching/academic-integrity/Pages/default.aspx>.

### Academic Policies

All students are required to manage their class schedules each term in accordance with the deadlines for enrolling and withdrawing as indicated in the [University Academic Calendar](#). Information on enrollment, withdrawal, grading and incompletes can be found at: <http://www.cdm.depaul.edu/Current%20Students/Pages/PoliciesandProcedures.aspx>

### Incomplete Grades

An incomplete grade is a special, temporary grade that may be assigned by an instructor when unforeseeable circumstances prevent a student from completing course requirements by the end of the term and when otherwise the student had a record of satisfactory progress in the course. All incomplete requests must be approved by the instructor of the course and a CDM Associate Dean. Only exceptions cases will receive such approval. Information about the Incomplete Grades policy can be found at <http://www.cdm.depaul.edu/Current%20Students/Pages/Grading-Policies.aspx>

## Students with Disabilities

Students seeking disability-related accommodations are required to register with DePaul's Center for Students with Disabilities (CSD) enabling them to access accommodations and support services to assist with their success. There are two office locations:

- Loop Campus (312) 362-8002
- Lincoln Park Campus (773) 325-1677
- Email: [csd@depaul.edu](mailto:csd@depaul.edu)

Students who register with the Center for Students with Disabilities are also invited to contact Dr. Gregory Moorhead, Director of the Center, privately to discuss how he may assist in facilitating the accommodations to be used in a course. This is best done early in the term. The conversation will remain confidential to the extent possible.

Please see <https://offices.depaul.edu/student-affairs/about/departments/Pages/csd.aspx> for Services and Contact Information.

## Online office hours

Faculty should be accessible to students using Zoom, Skype or other similar platforms for the duration of the office hours. Faculty must be accessible on the designated platform for the duration of the office hours.