

# Real-Time Software Development II

Fall 2021

GAM 575

**instructor:** Ed Keenan  
**email:** [ekeen2@cdm.depaul.edu](mailto:ekeen2@cdm.depaul.edu)  
**office hours:** Remote: Tues 9-10:30 pm Synchronous Zoom Meeting or by appointment  
**office:** CDM 830  
**phone:** (312)362-6747  
**website:** [piazza.com/depaul/fall2021/gam575se585](https://piazza.com/depaul/fall2021/gam575se585) (Preferred communication)  
**lecture:** REMOTE: Tuesday 5:45-9:00pm Synchronous Zoom Meeting  
**Desired to Learn (D2L):** [d2l.depaul.edu](https://d2l.depaul.edu) (Grades, Viewing lectures, Announcements)  
**Version Control:** perforce: *perforce.dpu.depaul.edu:1666*  
**All Communication:** Piazza for all communication, reply within 24 hrs during school week

## Description:

Real-time software development with asset conversion. Topics include offline data conversion, asset packing, data driven application behavior, hierarchy base scene management, profiling and debugging large opaque data, runtime formats for low-level drivers, and advanced visualization techniques. Design and implementation of hierarchy based data systems with dynamic real-time modifications. Exploration of run-time control of behavior through real-time data driven messaging. Implementation of the complete application process: tool creation, asset condition, responsive loading and data driven runtime behavior. Students will design, develop and implement a real-time application (i.e. Animation engine) that integrates large real-time converted data assets with real-time constraints.

## Prerequisites:

- GAM 475 Real-Time Software Development I

## Learning Goals:

- Students will develop runtime data format for blind abstract data.
- Students will implement a stand-alone command-line converter that uses 3<sup>rd</sup> party SDK to convert runtime assets.
- Student will architect large and complex real-time data driven engine to render and control dynamic assets.
- Students will explore techniques to control, debug, and manage large blind data sets use in applications.
- Students will understand timing metrics in applications eliminating modulus and rounding errors through abstract data type representation of time.
- Students will organize and process data in a hierarchy based scene management structure.

## Grading

### Milestone 1 50% - Total

- Systems
  - 10 % PA1 – Archiver (Google Protocol Buffers)
  - 10% PA2 – Math
- Model Viewer
  - 10% PA3 - Converter for models
  - 20% PA4 - Model Viewer

### Milestone 2

- Animation Engine
  - 10% PA5 - Converter for animation and bones
  - 15% PA6 - Skeletal Animation Engine
  - 25% PA7 - Playback engine

## 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
- Recommended:
  - **Windows System Programming** (4th Edition), 2010, Johnson Hart,
    - ISBN: 978-0321657749
- Assumed you already have (please buy it if you don't have a copy):
  - ***The C++ Programming Language***- Bjarne Stroustrup

### **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:

- **Main Lectures:**
  - 2<sup>nd</sup> pass on Graphics engine
    - Run time formats
    - Texture manager
    - Camera Culling
    - Bounding volumes
  - GLTF
    - Model conversion
    - Animation extraction
    - Skeleton
    - Skinning
  - Transformations
    - Interpolation - Linear, slerp, blerp
    - Hierarchy relative vs flat transformations
    - Quaternions
  - Animation
    - Key frame vs motion capture
    - Animation Controllers
    - Skeletons
    - Blending / Mixing
    - Morphing
    - Move the animation by game control
    - Puppet-ting
  - Skinning - Theory
    - Rigid body
    - EA technique
    - Midway Technique
  - Object
    - Cloning
    - Replication
    - Scene Graph
- **Secondary Lectures (if time permits)**
  - Level of Detail
  - Multiple rendering targets
  - Input / Events trigger

- Sound System
- UI
- Threading

## Programming Assignments – 100%

2 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:** Model Converter and more... - 50% (due Week 6)
  - Write a generic file Archiver Google protocol buffer
    - Takes loose binary files, adds headers and formatting info to create chunks
    - Packages the chunks together into a single binary package
    - Extracts chunks on demand from the package
  - Write a GLTF converter to convert to run-time file format.
    - Place data into the file archiver
    - VBO with VAO format
    - Convert 3 FBX models to Game Engine runtime format from a batch/script file.
    - Models need to be
      - In FBX format
      - Exceeding 200 polygons each
      - Contain textures / Models need to be lit in game engine
  - Model viewer
    - Load models from the archiver
    - Rotate camera 360 around the viewer
    - Zoom
    - Add ground plane
    - Bounding volume culling and viewing
    - Demo the game engine to display the 3 supplied models + 3 others
  - Quaternion library
    - Write and integrate the quaternions into the math library
    - Validate against unit tests
- **Milestone 2:** Animation Engine 50% (due Week 11)
  - GLTF animation exporter
    - Write a converter to extract Animation data (Skeleton and animation) from an GLTF file
    - Convert several animations to a run-time file format
  - Animation engine
    - Demo the game engine to display the several different animations simultaneously
    - Be able to dynamically interpolate the play back of the animation
    - Each animation needs at least 5 or more bones

- At least 20 or more key frames
      - Each keyframe containing rotation and translation
  - Playback engine
    - Animation should be able to:
      - Play forward / backwards
      - Loop
      - Faster or slower playback rate
    - Transition to different animations
  - Documentation of the design
    - PDF 4-5 pages showing the data layout, math, stages of the animation engine

## 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 Performer 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	Syllabus, Class Overview Archiver (Protobuf)	PA1 - Archiver <i>Protobuf</i>	
2	Quaternions Keenan-Mods Quaternions Quaternions Interface	PA2 - Math Quaternions	PA1 – Archiver <i>Protobuf</i>
3	GLTF Converter Level of Details Better General shader	PA3 - GLTF converter <i>Models</i>	PA2 – Math
4	Converter Hands on demo Model Viewer Bounding Volume Culling	PA4 - Model Viewer	PA3 – GLTF Converter <i>Models</i>
5	Keyframe Animation Starting Animation development demo Motion Capture		PA4 – Model Viewer
6	Timer (Discrete time for animations) Animation System	Animation Engine start	Milestone 1 PA1 - PA4
7	Keyframe animation Animation converter Design demos	PA5 - GLTF converter <i>Anim data</i> <i>Skeleton</i>	
8	Hierarchy Animation Skeletons Design demos	PA6 - Animation System	PA5 – GLTF converter <i>Models</i>
9	Skinning Pimpl Architecture design	PA7 - Playback Instance	PA6 - Animation System
10	Animation System issues Loading / unloading assess		PA7 -Playback
11			Milestone 2 PA5-PA7

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.

### COVID-19 Health and Safety Precautions

Keeping our DePaul community safe is of utmost importance in the pandemic. Students, faculty and staff are expected to (1) wear a mask as required at all times while indoors on campus; (2) refrain from eating and drinking in classrooms; (3) keep current with their COVID-19 vaccinations or exemptions; (4) stay home if sick; (5) participate in any required COVID-19 testing; (6) complete the online Health and Safety Guidelines for Returning to Campus training; and (7) abide by the City of Chicago Emergency Travel Advisory. By doing these things, we are Taking Care of DePaul, Together. The recommendations may change as local, state, and federal guidelines evolve. Students who do not abide by the mask requirement may be subject to the student conduct process and will be referred to the Dean of Students Office. Students who have a medical reason for not complying with any requirements should register with DePaul's Center for Student with Disabilities (CSD).

### Respect for Diversity and Inclusion at DePaul University as aligned with our Vincentian Values

At DePaul, our mission calls us to explore “what must be done” in order to respect the inherent dignity and identity of each human person. We value diversity because it is part of our history, our traditions and our future. We see diversity as an asset and a strength that adds to the richness of classroom learning. In my course, I strive to include diverse authors, perspectives and teaching pedagogies. I also encourage open dialogue and spaces for students to express their unique identities and perspectives. I am open to having difficult conversations and I will strive to create an inclusive classroom that values all perspectives. If at any time, the classroom experience does not live up to this expectation, please feel free to contact me via email or during office hours.

### 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. All students are expected to abide by the University's Academic Integrity Policy which prohibits cheating and other misconduct in student coursework. Publicly sharing or posting online any prior or current materials from this course (including exam questions or answers), is considered to be providing unauthorized assistance prohibited by the policy. Both students who share/post and students who access or use such materials are considered to be cheating under the Policy and will be subject to sanctions for violations of Academic Integrity.

More information can be found at <https://resources.depaul.edu/teaching-commons/teaching/academic-integrity/Pages/default.aspx>.

### *Posting work on online sites, such as Hero*

All students are expected to abide by the University's Academic Integrity Policy which prohibits cheating and other misconduct in student coursework. Publicly sharing or posting online any prior or current materials from this course (including exam questions or answers), is considered to be providing unauthorized assistance prohibited by the policy. Both students who share/post and students who access or use such materials are considered to be cheating under the Policy and will be subject to sanctions for violations of Academic Integrity.

## 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>

## Preferred Name & Gender Pronouns

Professional courtesy and sensitivity are especially important with respect to individuals and topics dealing with differences of race, culture, religion, politics, sexual orientation, gender, gender variance, and nationalities. I will gladly honor your request to address you by an alternate name or gender pronoun. Please advise me of this preference early in the quarter so that I may make appropriate changes to my records. Please also note that students may choose to identify within the University community with a preferred first name that differs from their legal name and may also update their gender. The preferred first name will appear in University related systems and documents except where the use of the legal name is necessitated or required by University business or legal need. For more

information and instructions on how to do so, please see the Student Preferred Name and Gender Policy at <http://policies.depaul.edu/policy/policy.aspx?pid=332>

### **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.

### **Faculty Resources Available from the Dean of Students Office**

The online classroom [https://offices.depaul.edu/student-affairs/resources/faculty-staff/faculty-questions/Documents/Faculty\\_Resources\\_Online\\_Classroom.pdf](https://offices.depaul.edu/student-affairs/resources/faculty-staff/faculty-questions/Documents/Faculty_Resources_Online_Classroom.pdf)

### **Syllabus Resources Available from Teaching Commons**

<https://resources.depaul.edu/teaching-commons/teaching-guides/course-design/Pages/syllabuses.aspx#samples>