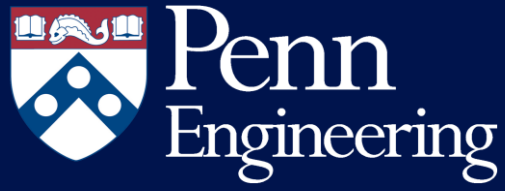
Final Projects Overview

Shehzan Mohammed

CIS 5650 – Fall 2025



Final Projects

- Large GPU programming endeavor of your choice.
- Your choice of focus - graphics, GPU computing, or both.
- 2-3x more work than the hardest project this semester.
- Expect 12-20hrs/week of work. It will be worth it.

A Good Project =

Your Interest + GPU Programming + Industry-Worthy

Final Projects

Final Projects must be some combination of:

- Fulfill **a real need in the community**
- Explore a **new API**
- Reproduce the results of **recent research**
- Add a **novel extension** to existing work
- Implement something **completely original**

Guidelines

- Teams of **two or three or four** students
  - **Team number preference: 3 > 4 > 2**
- Each team member will receive the same grade
- Work product expected to scale with team size
- Teams of 1 not allowed due to class size

Guidelines

- Use GitHub – Open Source is encouraged but not required
- Programming language, graphics and compute APIs, and target platforms are up to you
- Allowed to use existing code and libraries – Credit, identify and describe usage
  - Libraries must not be core to the problem itself

Tentative Schedule (Subject to change)

Date	Time	
October 27, 2025	11:59pm	Teams Finalized and Pitch Slots Selected
November 1, 2025	11:59pm	Final Project Pitch Documents due
November 3, 2025	5:15pm	Final Project Pitches – Part 1
November 5, 2025	5:15pm	Final Project Pitches – Part 2
November 7, 2025	11:59pm	Shadow Team Selections due
November 12, 2025	5:15pm	Milestone 1 Presentations
November 24, 2025	5:15pm	Milestone 2 Presentations
December 1, 2025	5:15pm	Milestone 3 Presentations
December 7, 2025	11:59pm	Final Projects Due (No late days)
December 8, 2025	4:00pm	Final Project Presentations Due
December 8, 2025	5:30pm	Final Project Presentations (Live)

Details will follow in Ed

Milestone Presentation Logistics TBD due to Class Size. More details will follow.

All times are Eastern Standard Time

Final Project Pitches

- Use Final Project Pitch Sign Up Sheet – **First come first serve**
  - Projects may be pitched earlier
  - You may need to do a follow up pitch on Nov 6
- Use office hours to discuss your initial ideas
  - Please do not come with too many ideas
  - Be focused, have the right topics, scope, structure, and we can answer targeted questions.
  - Any special OH information will be posted on Ed.

Grading

- The final project is worth 45% of your final grade.
- The breakdown is:
  - Milestone 1 Progress: 20%
  - Milestone 2 Progress: 20%
  - Milestone 3 Progress: 20%
  - Final Submission: 20%
  - Presentations (3 milestones + final presentation): 20%

Project Pitches – Do

- Keep within 1 page (for doc) or 4 slides - **Conciseness is best**
- Have **clear goals and outcomes**
- **Clear application of GPU programming**
- 1-2 paragraphs about **why this project matters**
- A schedule for Milestones 1-3, and Final
- APIs, Platforms etc. you are planning to use
- Include references, links to content/inspiration, picture is fine too
- Any third-party code you are planning to use

Project Pitches – Don't

- Do not surprise – run your idea by Shehzan/TAs at least once informally to get buy-in
- Do not include math equations/code etc
- Do not exceed 2 pages / 5 slides
- Don’t bring repeat ideas from previous years
- Don’t come under-prepared – be ready to defend why you think your project should be accepted

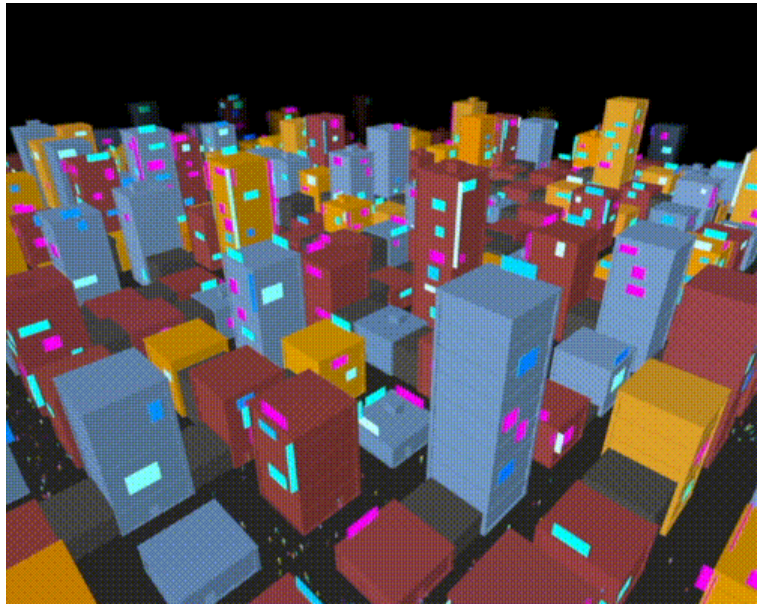
Good Final Projects from Previous Years

- These projects highlight well-implemented projects
- Look at results, presentations, documentation etc
- Don’t try to re-implement the idea
  - Putting a spin on it is fine
- Ideas from 3+ years are solved and will not be accepted without significant “new stuff”

Note: Being listed here is not the only stamp of excellence – I’ve only picked a few from each year

Good Final Projects from Previous Years - 2024

SurfelPlus: Real-time Global Illumination Based on Surfels – Zhen Ren, Ruipeng Wang, Jinxiang Wang

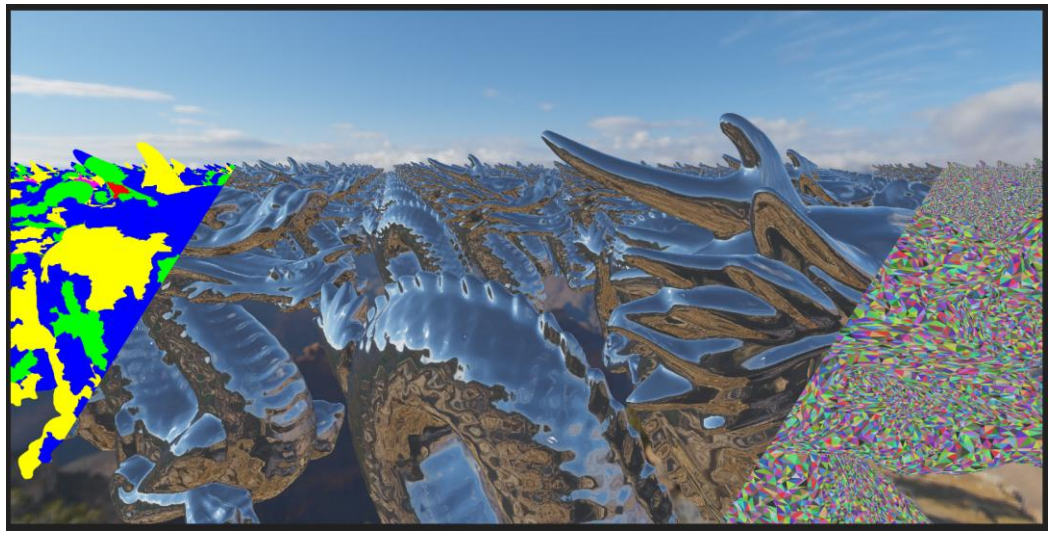


Citopia: Crowd Simulation Featuring Detailed Daily Routines of Diverse NPCs in a City Environment. – Deze Lyu, Paulina Tao, Christine Kneer

SDFGI: Real-time SDF-based Global Illumination – Michael Mason, Carlos Lopez Garces, Logan Cho

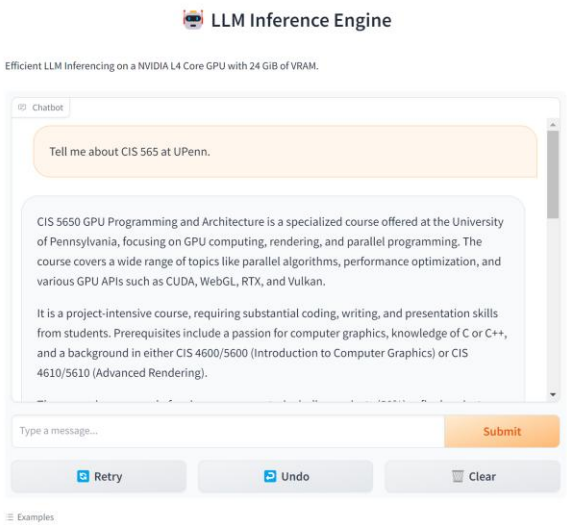
https://cis565-fall-2024.github.io/projects/

Good Final Projects from Previous Years - 2023



Vulcanite - A Vulkan Implementatioin of Nanite in Unreal Engine 5 – Han Yang, Yian Chen

Mega Minecraft - Real-time path tracing with OptiX and GPU-accelerated terrain generation with CUDA – Aditya Gupta, Helena Zhang, Alan Qiao



Efficient LLM Inferencing on a Single GPU – Xitong Zheng, Zhenzhong Tang, Yinuo (Travis) Xie

Penn Engineeri

https://cis565-fall-2023.github.io/projects/

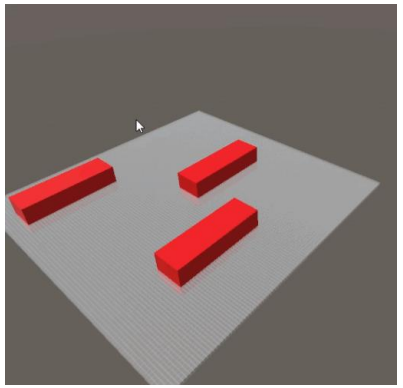
Good Final Projects from Previous Years - 2022

GPU-Accelerated Heterogeneous Volume Rendering with Null-Collisions – Nick Moon, Megan Reddy

EIDOLA - Real-time GI Path Tracer – Chang Liu, Alex Fu, Yilin Liu

ARC - An AR Procedural Generation App – Guanlin Huang, Shutong Wu, Rhuta Joshi

Realistic Butterfly Flight and Behavior Simulation with WebGPU – Shineng Tang, Jiajun Li, Haoquan Liang

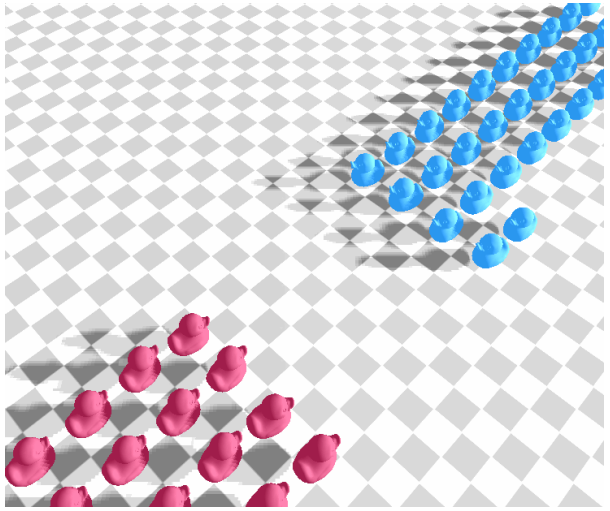


GPUVerb: Real-Time Dynamic Sound Propagation – Evan Si, Runshi Gu, Tongwei Dai

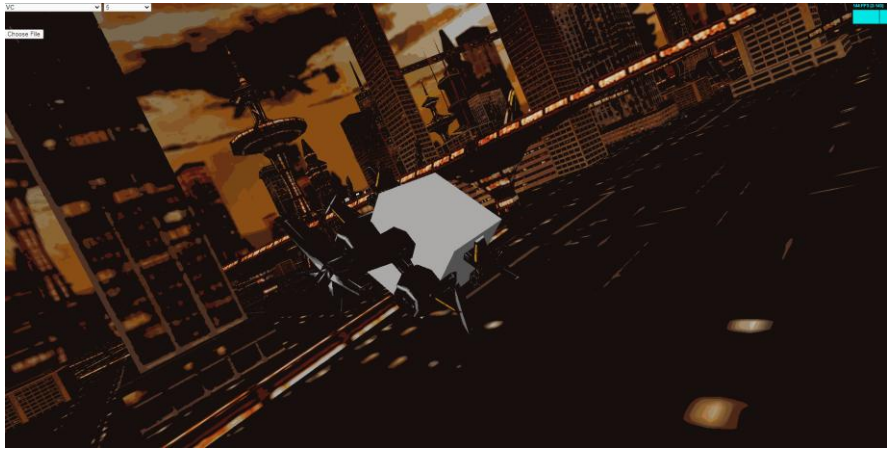
Penn Engineering

https://cis565-fall-2022.github.io/projects/

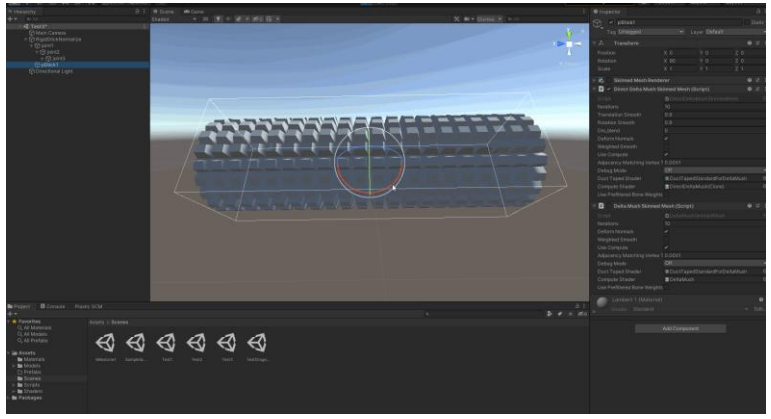
Good Final Projects from Previous Years - 2021



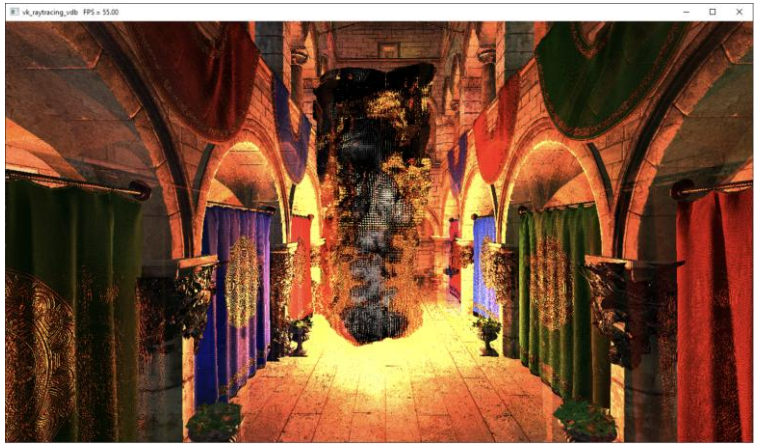
Position-Based Multi-Agent Dynamics on WebGPU
Matthew Elser, Ashley Alexander Lee, Wayne Wu



WebGPU based glTF Viewer – Jiyu Huang



Real-Time Mesh Skinning with Direct Delta Mush
Xuntong Liang, Bowen Deng, Beini Gu



Fast Volume Rendering with Spatiotemporal Reservoir Resampling
Zhihao Ruan, Shubham Sharma, Raymond Yang

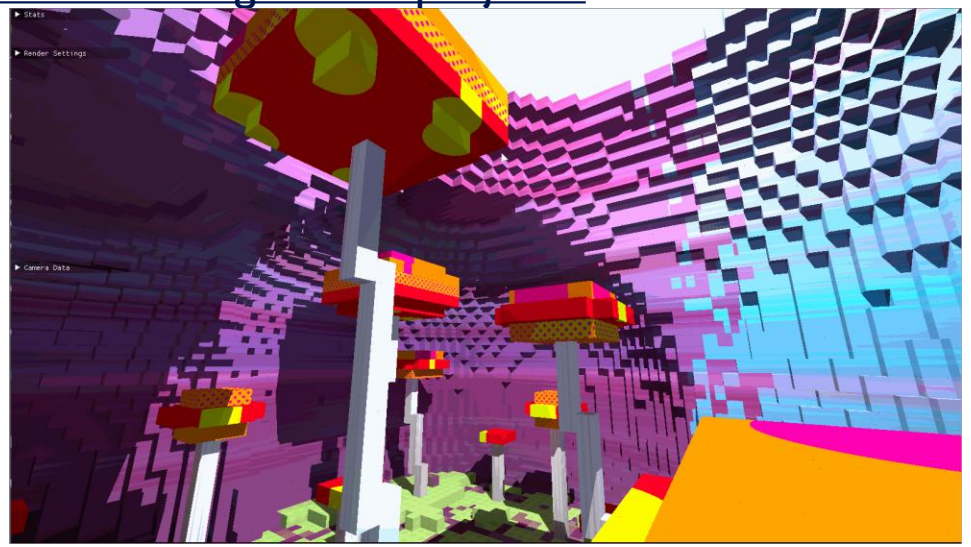
Penn Engineering

https://cis565-fall-2021.github.io/projects/

Good Final Projects from Previous Years - 2020

https://cis565-fall-2020.github.io/projects/

https://github.com/tatran5/Reservoir-Spatio-Temporal-Importance-Resampling-ReSTIR



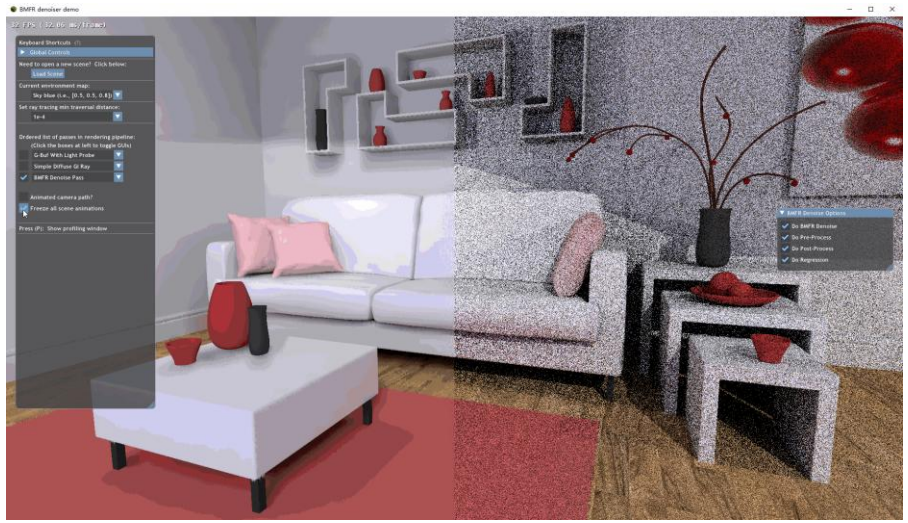
https://github.com/helenl9098/ddgi_minecraft

https://github.com/chetan-parthiban/WebGPUMPM

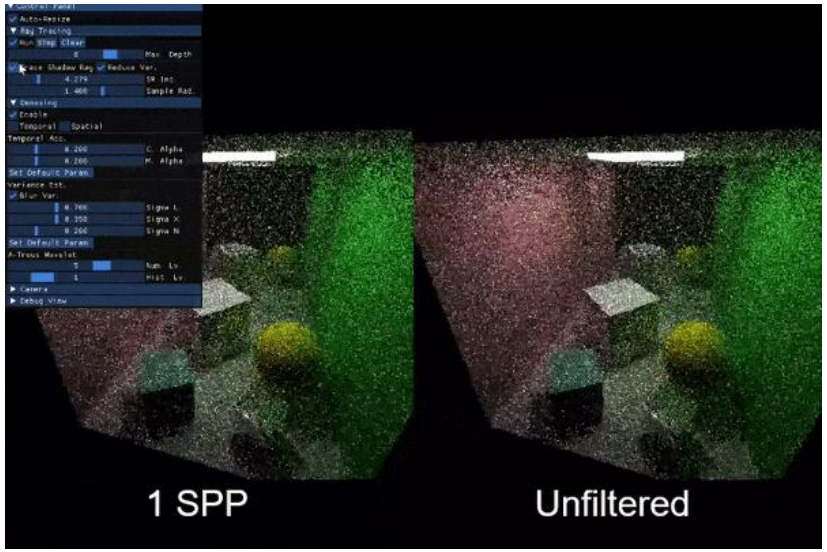
https://github.com/ZijingPeng/DXR-Hybrid-Rendering

Good Final Projects from Previous Years - 2019

http://cis565-fall-2019.github.io/studentwork.html

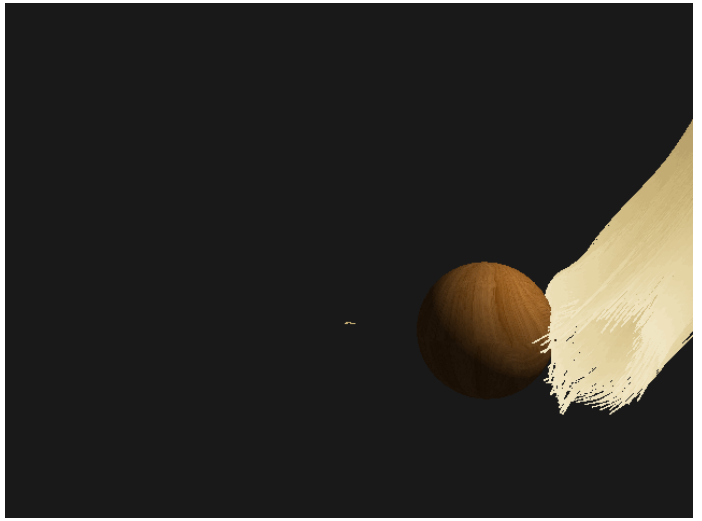


https://github.com/gztong/BMFR-DXR-Denoiser



https://github.com/ZheyuanXie/CUDA-Path-Tracer-Denoising

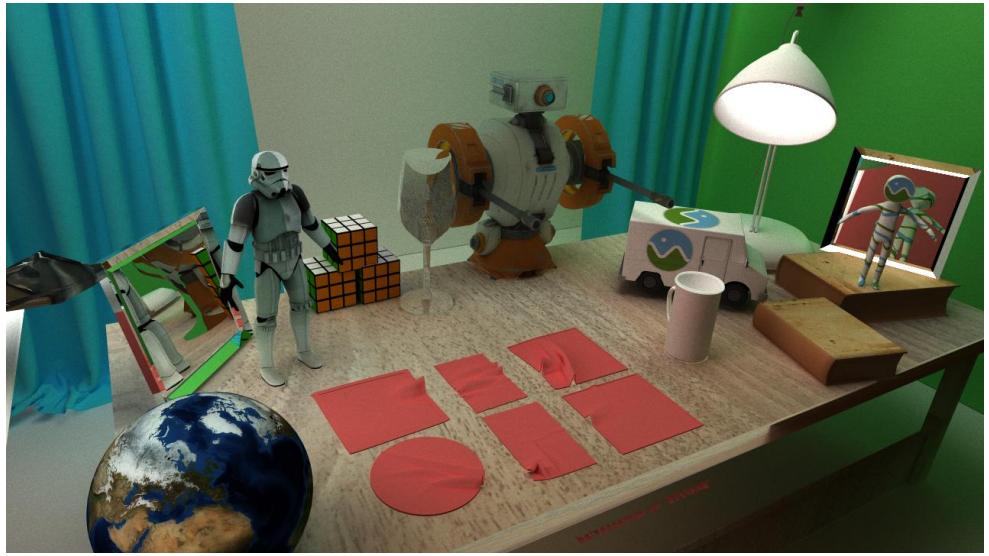
https://github.com/pnorouzi/Robot-Parallel-Motion-Planning



https://github.com/clach/Realtime-Vulkan-Hair

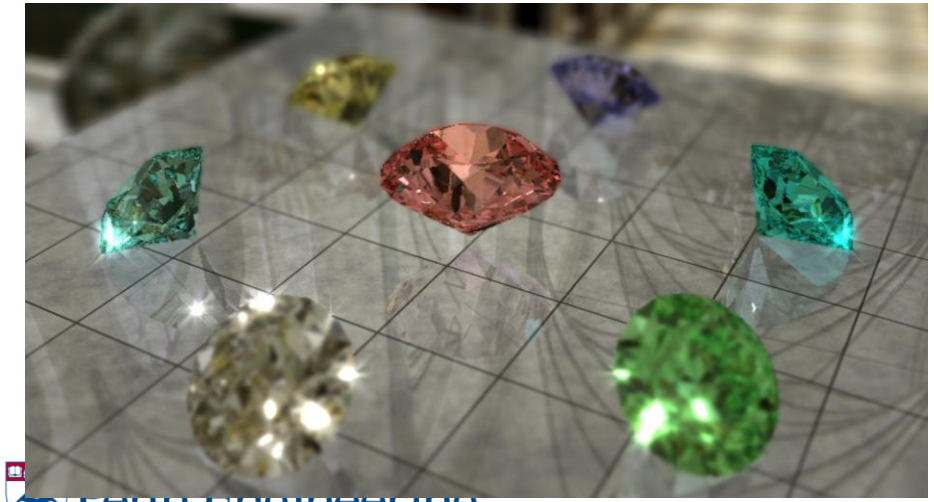
Good Final Projects from Previous Years - 2018

http://cis565-fall-2018.github.io/studentwork.html

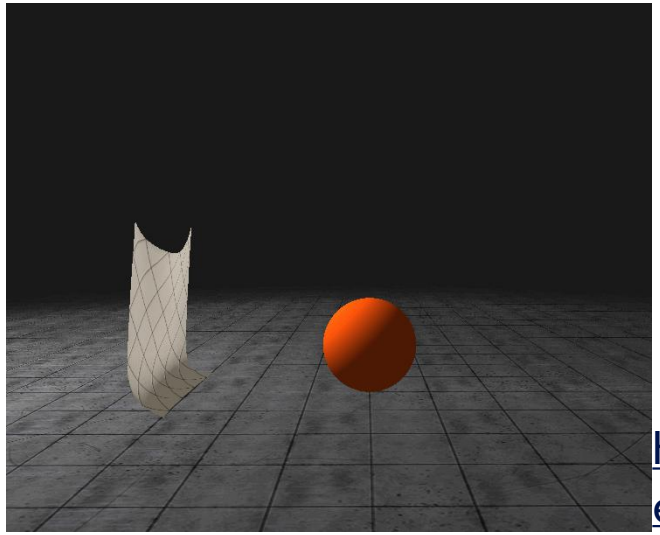


https://github.com/rtx-on/rtx-explore

https://github.com/460xlin/Vulkan_Hybrid_PBR/



https://github.com/WanruZhao/CIS565-Final-Project



https://github.com/vasumahesh1/Flamenco

Good Final Projects from Previous Years - 2017

http://cis565-fall-2017.github.io/studentwork.html

https://github.com/mmerchante/organic-mesh-growth

https://github.com/byumjin/Vulkan-VXGI-VR-FrameWork

https://github.com/WilliamKHo/Pollux-Renderer

Projects with good Pitch, Milestones, Final Presentation

- All:
  - https://github.com/tatran5/Reservoir-Spatio-Temporal-Importance-Resampling-ReSTIR
  - https://github.com/vasumahesh1/Flamenco
  - https://github.com/hanbollar/Odin
  - https://github.com/rtx-on/rtx-explore
  - https://github.com/WindyDarian/Vulkan-Forward-Plus-Renderer
- Partly (because other media wasn't part of repo):
  - https://github.com/WanruZhao/CIS565-Final-Project/blob/master/CIS565%20Final%20Project%20Pitch.pdf
  - https://github.com/Jiaww/Vulkan-Forest-Rendering-Engine/blob/master/milestones/Vulkan-%20Forest%20Rendering%20Engineer.pdf
  - https://github.com/AmanSachan1/Meteoros/blob/master/Project%20Pitch%20%26%20Milestones.pdf
  - https://github.com/AmanSachan1/Meteoros/blob/master/Project%20Pitch%20%26%20Milestones.pdf

Project Ideas

- These are only some projects ideas that I have come across
- Please think outside the box on ideas that you are interested in

Real-time Ray Tracing (DXR or VKRay)

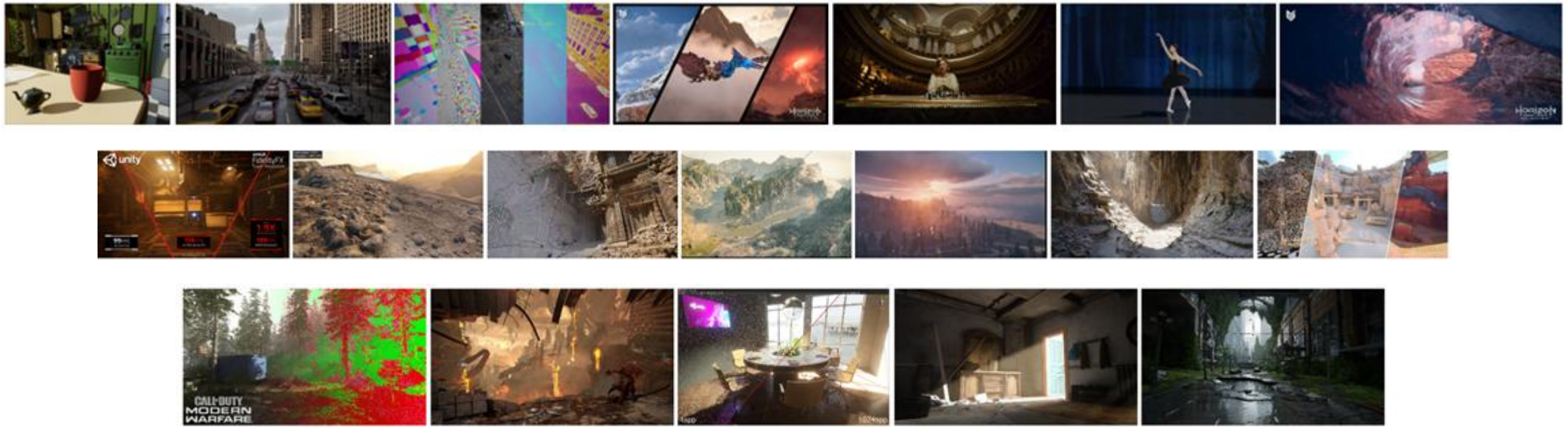
- Build a project using DirectX Ray Tracing or Vulkan NVIDIA VKRay
  - May use 2019 DXR Project as base code
- Very open ended to what you want to do
- Look up unique uses of DXR / VkRay in the industry
- Connect with Eric Haines/others at NVIDIA

NVIDIA Optix

- Use **NVIDIA Optix** for your Path Tracer
- All state-of-the-art path tracers use NVIDIA Optix
  - e.g. VRay, Redshift, Arnold, RenderMan etc.
- Also has support for real-time ray traicing
  - Eric Haines works “closely” with Optix
- May be used as future Project 3

Advances in Real-Time Rendering

- Annual SIGGRAPH Course
- Talks hosted here: https://advances.realtimerendering.com/
- More CG Conferences: https://kesen.realtimerendering.com/



Machine Learning – 3D Graphics

Examples

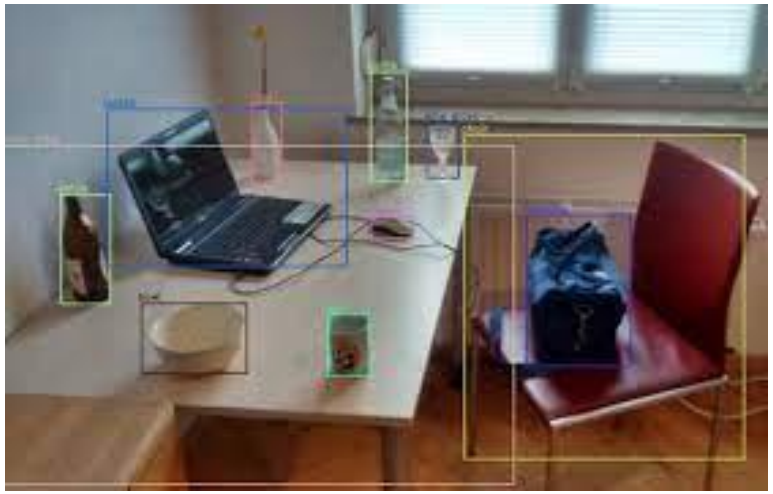
- 3D Gaussian Splatting / NeRF
- Differentiable Rendering
- Differentiable Dynamics for Physics Simulation
- Reinforcement Learning for Character Control
- 3D Mesh Classification/Segmentation
  - https://medium.com/stanford-cs224w/deep-learning-on-3d-meshes-9608a5b33c98

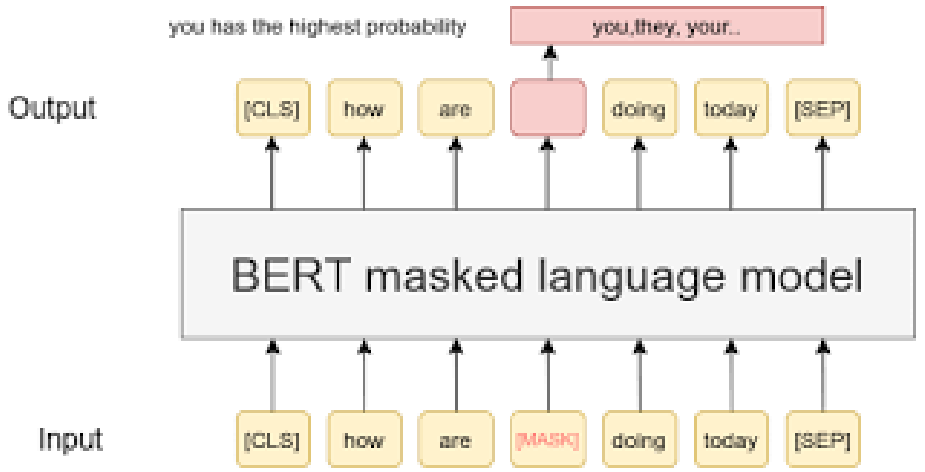
•Check out the course for inspiration:

- http://cs348i.stanford.edu/

Machine Learning – Compute

- Can also do a GPU Compute ML project!
- Examples:
  - Train custom object detection model (AlexNet)
  - Train custom speech to text model (CNN-LSTM)
  - Train a set of word vectors on raw text to use in downstream tasks (word2vec, BERT)
  - Train a generative language or dialogue model (GPT-3)
- Liam can help directly supervise any NLP projects





Vulkan, DirectX, Metal Projects

- **No OpenGL projects**
  - unless it's just to draw something simple (e.g. CUDA-GL-interop)
- Use advance topics, extensions etc. in Vulkan to do amazing projects
- The scope is wide open, try to take inspiration from previous years
  - But do not try to replicate them – Advance the capabilities
  - Will be great for future class projects

Mesh Shading

- Check out Manuel Kraemer's 2020 Guest Lecture
  - https://onedrive.live.com/view.aspx?resid=A6B78147D66DD722%2195310&authkey=!ABgWsMe8WjwYIMU
- Can use part of samples as base code
- An interesting final project will get attention from a very wide and diverse audience

WebGPU Projects

- **No WebGL projects**
  - CIS 565 has a history of amazing WebGL projects
  - Build similar projects for WebGPU (we had 2 in 2022, 3 in 2021)
  - Direct porting will not be enough – will need to add features
- Shrek Shao' Guest Lecture will be extremely helpful
- Can connect with WebGPU developers for leading technical advice on your projects
- Help Babylon.js or Three.js advance WebGPU! (they have already adopted it)
- **Why Web? Because you can demo it anywhere!**

WebGPU Projects – From Google/Chrome

- **A biome generator and renderer**
  - The goal is to display a AAA looking live editable 3D nature scene with minimal data - each part can be separate but
- Algorithmically generated terrain
- GPU based erosion
- Algorithmically generated plants, flowers, trees, rocks etc
- GPU based instance placement
- GPU based path based roads / trails

Reference:

- From Battlegrounds to Fairways: Terrain Procedural Tools in Frostbite - shows that with the right GPU work you can build beautiful environments in real time

WebGPU Projects – From Google/Chrome

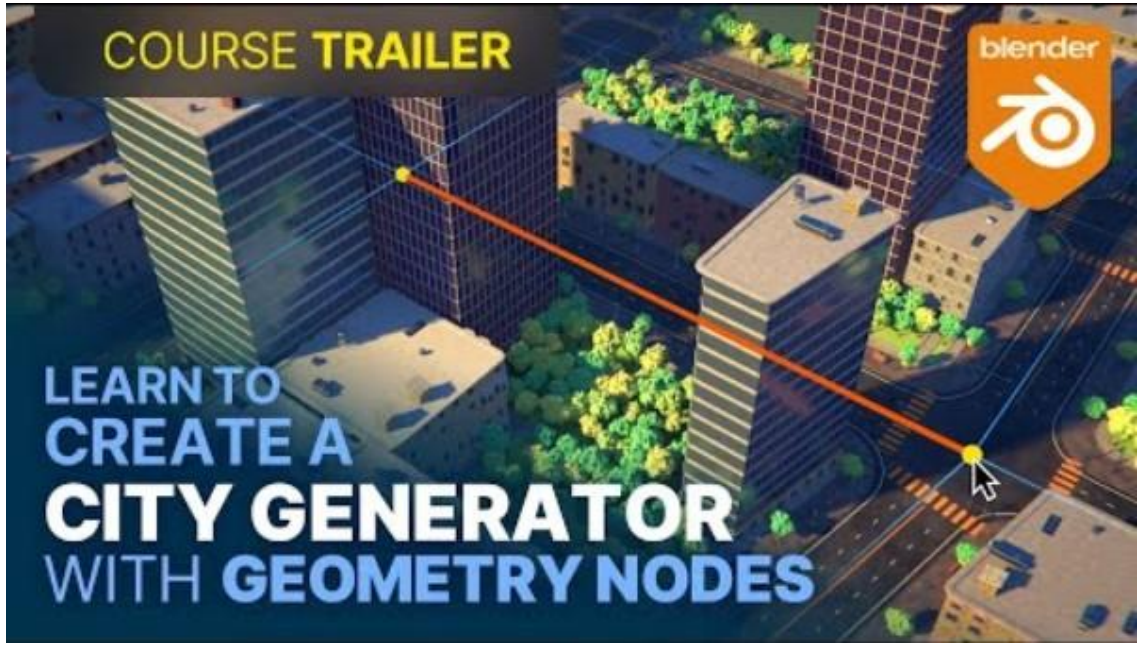
- **A GPU based node based geometry generator**
  - node based geometry generation is famous from Houdini and was relatively recently added to blender. It would be great to see a fast compute based implementation in the browser
  - Note: Could be related to previous idea since there are lots of examples of using geometry nodes for building trees/plants/rocks.

Reference:

- Geometry Nodes from Scratch

WebGPU Projects – From Google/Chrome

- **A GPU based city generator**
  - the goal here is to have AAA 3D editable city in the browser with minimal data
  - This might also be just more nodes



Reference:

- Learn to create a procedural city with Blender and Geometry Nodes

WebGPU Projects – From Google/Chrome

- **A GPU based node based texture generator**
  - The goal here is to have some small JSON file or other format that describes the nodes and can be used in other projects with some smallish library to generate textures

Reference:

- What is Substance Designer | Adobe Substance 3D

The goal of all of these are to make it easier or provide a base or at least inspiration for web based experiences that are small to download and get started. A huge issue for web based WebGPU is having to download gigs of assets. Systems like those above could provide a path to much smaller downloads. I get that they might seem like giant projects but many of them can be reduced to a core set of small features. Once you have a node editor (you could use an existing one) and you get 2-3 nodes working it goes fast to get amazing results.

Projects with NVIDIA Omniverse / USD

- Never been attempted.
- NVIDIA Omniverse is NVIDIA's metaverse platform including server, RTX renderer, and various connectors (python and C++)
- Come up with a unique idea about how you can create a great use-case with Omniverse. Can be:
  - Graphics, Rendering, Ray Tracing
  - USD related (Specifically: USDRT, Fabric)
  - AI/ML
  - Simulations
- Look into **NVIDIA Warp** for GPU acceleration:
  - https://developer.nvidia.com/warp-python

NVIDIA OMNIVERSE™

Penn Engineering

Using Game Engines

- You may build projects on leading game engines like Unity and Unreal
- Must be working with GPU or GPU APIs in a way that is adding real value to the game engines
  - Not a game building project – but a game may be a good way to show off what your focus is
- For example: https://github.com/Ninjajie/Fusion
- Most likely some sort of compute shader work.

Add Features to Godot or O3DE

- Add features to open source engines – O3DE has a lot of scope and community
- Ideas included in O3DE Guest Lecture from 2021 – or bring your own
  - Check out the 2021 guest lecture: here
- Connect with the O3DE Community for help
- Merging PR into mainline is not required

Penn Engineering

O3DE

CesiumJS / Cesium for Unreal

- Cesium for Unreal is an open-source, high-precision globe engine built in Unreal
- Wide open world (literally) and you will receive time and resources from the best 3D developers at Cesium

Idea from NVIDIA:

- How do you denoise correlated samples?
  Basically, a traditional renderer has white noise / uniform noise characteristics. This is the traditional noise in both real-time settings (and, largely, offline). However, algorithms that reuse samples (MCMC, MLT, ReSTIR) tend to create noise that is correlated, for instance blotchy artifacts that fade in and fade out or clusters of fireflies (that got reused from a single firefly).

- What kind of denoising techniques can be used to filter these sort of correlated signals? (Traditional denoising filters or learned ones.)

- For testing, it's pretty easy to create correlated signals. E.g., aggregating samples from corresponding pixels in the past 4-8 frames or reusing light samples for next-event estimation in 7x7 pixel regions.

Voxel Space (Idea from NVIDIA)

- One great use of CUDA / Compute shaders is for "one-to-many" rendering algorithm where in a single thread (or choose your terminology), many pixels are output and benefit from sharing data with each other (whereas in a traditional pixel shader is limited to one "thread" to a pixel / fragment).
  - Think Matrix Multiply, Transpose, Reduction etc
- One example of such an algorithm that can easily benefit from "one-to-many" is the VoxelSpace algorithm: https://github.com/s-macke/VoxelSpace
- If that is not challenging enough, adding a camera with iteratively more degrees of freedom presents additional challenges. As does making the output 'voxels' perfectly cubic (instead of screenspace vertical lines)

ReSTIR PT (Idea from NVIDIA)

- Simple version of either gradient-domain path tracing or ReSTIR PT, perhaps with just the reconnection shift for simplicity.

- Getting to understand Generalized Resampled Importance Sampling, which ReSTIR PT is based on, could be an excellent exercise for an advanced student and make them stand out. GRIS can also be used for integrating many other things than pixel colors—like (ir)radiance caches, for example.

Other Topics (from NVIDIA)

- Massively Parallel Collision Detection
- Try improving speed or memory without affecting the algorithm at all, or minimizing some approximation error -- there is a lot of good experience that can come from just learning how to make things fast

Other Topics

- Recent papers from NVIDIA GTC, SIGGRAPH, other Deep Learning/Machine Learning conferences
- Check out open source projects from the Academy Software Foundation

Anti-ideas

- These ideas being the core of the project is outdated.
  - These can be a part of the project for acceleration, but should not be the core
- Just Vulkan compute shader – Combining it with leading research is good
- Multi-GPU
- A very sophisticated simulation – unless it is leaps and bounds better
- No “yet-another-CUDA-fluid/smoke/cloth simulation” unless it has a unique spin
  - For example https://github.com/vasumahesh1/Flamenco implemented cloth sim in their own game engine
  - https://github.com/Ninjajie/Fusion did GPU Cloth sim in Unity and published as plugin
- Likewise, please do not propose extend-my-CUDA-path-tracer-with-these-n-effects unless it is truly unique.
- No Pure-ReSTIR (we had 4 in the last 2 years!) – May try significantly different new papers
- From NVIDIA: “ ‘*go implement a relevant paper or talk from the last five years*’ - That got you a B+. For higher work you were expected to make either a modification or an improvement and show its impact (even if it was just the same)”

Shadow Teams

- Teams must pair up as each-others **shadow teams**
  - **Shadow Team selections must be made by Nov 7, along with your weekly meeting time.**
- Shadow teams must be used for practicing presentations
- Good for getting presentation content/timing into shape
- Plan to meet once a week for 30-60 minutes
- Shadow teams may be used for bouncing technical ideas
- Ideally, your shadow team should be working on a similar idea for their final project

Shadow Teams Survey

- A shadow team survey will be made available after Nov 8
- Each student must complete their own survey after (or during) each session
- Survey will include
  - Date, time, length of the session
  - Rating your practice, idea sharing
  - Rating your shadow team's practice
  - Any interesting highlights, feedback, advice, ideas, help
- Failure to complete surveys on-time will result in penalties

FAQ

I want to do a project on a topic that has been done in the past. Is that allowed?

- Main goal of final projects is for you to work on industry relevant projects. If you are trying to do something like year(s) old project, then **you are starting behind**.
- Projects such as adding to Path Tracer or doing procedural generation, will have a **higher bar** and will only be allowed if the **additions are going to be novel**.
- Projects like Vulkan, VR, DirectX Raytracing, Machine Learning - the technology is recent and not much work has been done in class. You will be the pioneers of these.
  - Working on these will also open the possibility of the project becoming a regular project in the future!
- Simulations/physically-based animations will be allowed if you are implementing a novel technique that is vastly different from the seminal work in that field.

FAQ

I really want to work on a topic/technology, but I have limited or no experience.

- Be brave. There are a ton of online courses out there. Additionally, we'll try to connect you to any industry experts we know of.

FAQ

I want to do a DXR Project, but I do not have access to GPUs.

- Azure, AWS, and Google Cloud all have remote GPUs that you can use.
- All platforms provide student credits.
- Shehzan has tested these platforms and will be happy to help set you up.
- Please reach out ASAP.
- **Note this in your Final Project Sign Up Sheet.**

FAQ

What if I'm not doing a graphics or visual project? What should I do for final demos?

- It is completely ok to do a non-graphics project. Not all useful software on GPU is graphics related.
- You can do a lot of things for final projects, including, but not limited to, some great graphs, comparisons with industry software etc.
- Focus on optimization and performance analysis!

FAQ

I'm from CIS (or non-graphics). Can you suggest projects for me to work on?

- The core piece of advice would be to take something really amazing you have done, and then port it to the GPU.
- There are a ton of other topics you can think of as well. Machine Learning, deep learning, autonomous driving to name a few. Additionally, embedded systems with GPU such as Tegra or ARM GPUs, Apple Metal etc. are also welcome.
- It is all about doing projects using the GPU that are relevant to the industry and your field of interest.

FAQ

I want to work on a project that requires data. I don't have any.

- There are usually a lot of **open datasets** available, most of which are a simple Google search away.
- Also try talking to **other departments, professors or students** who might have such data. There is always a good chance by approaching the right people.
- If you still cannot find it, **we can help connect you to industry experts** who might have such data.
- **Start Early!!!** Don’t wait until after your pitch to check if data exists – its too late then.

FAQ

I've been working on a long-term project (research/PhD/etc). Can I do my final project on that?

- Yes, if:
  - There is no double-dipping (getting grades in 2 different courses for the same work).
  - The goals are clearly laid out.
  - The final project work is a GPU enhancement to other work.
  - If necessary, the other Faculty/Researcher also approves.
  - Other conditions that may be taken up case-by-case.

FAQ

I may miss one of the milestones because .....

- Missing milestones is **NOT** acceptable, except if you have obligations such as interviews, recruitment etc., or obtain prior permission. In that case, do the following:
  - Ensure your team presents on the day of the milestone presentations.
  - Schedule a time with a TA and your team **prior to the milestone presentations** and discuss the progress made.
  - Be transparent and communicate.

FAQ

There are odd number of teams and we don’t have a shadow team pairing

- You can pick your group of 3-teams.

Final Projects – Deliverables via GitHub

- Project Pitch Document(s)
  - PDF/Doc (1 page recommended, no more than 2) **OR** Slides (no more than 5 slides)
- 3 x Milestone Presentations
- Final Presentation
- Final code submission
- 4 x Shadow Team Surveys (not on GitHub)

A Good Project =

Your Interest + GPU Programming + Industry-Worthy

Tips

- **Scoping** is very important.
- Be realistic about the **feasibility** of the project.
- Reach out to the community.
- Have fun!

Questions

Penn Engineering

Action: Sign up for your pitch

Final Project Pitch Sign Up Sheet – First come first serve

Penn Engineering

Appendix

Note: This section will be covered during a future lecture

Penn Engineering

Milestone Presentations - Do

- Strictly stick to time limits – `N minutes` is `N x 60` seconds!
  - Default length will be 5 minutes, but may change – Will be posted on Ed Discussion
- **Show progress since last milestone**
- Videos, screenshots and demos
- Include goals for next milestone
- Know your audience
  - i.e. your fellow students, not the instructor or TAs
- Add presentation to your GitHub repo.

Milestone Presentations - Do

- Use social media – Great time to show off your work
- **Get in touch with original authors** – They really like it
  - And do this earlier than later
- See the Cesium Presenter's Guide (or your favorite company) for tips on presenting
- Be sure to present as a team; for a great example, see http://www.youtube.com/watch?v=OTCuYzAw31Y

Milestone Presentations – **Don't**

Doing any of these may result in grade penalties

- Don't exceed time limits for presentations
- Don't include code/math equations in your presentation
  - Exceptions: Something cool, good to know, or required for another part of your presentation.
  - If you need to walkthrough the code/math, don't include it

Final Presentations

- **Final presentations will be open to public audience!**
- Your audience is the world, not just fellow students
- Presentation must be much more polished than milestones
  - Do not include information about milestones
- **Maybe: 5 minutes presentation + 2 minutes live demo**
  - Subject to minor change
- Same dos and don’ts as Milestone presentation

Final Presentations

- Give an **overview of the technical approach** with high-level diagrams (1-2 slides)
- Followed by a **demo/video of the results** in representative scenarios
- Followed by a careful **performance analysis and optimization results**
  - **Clearly shows "Did X and got Y% improvements"**
- Finally, **shortcomings or future work** of the approach.
- It is OK to present this as a few slides with a demo/video for the results section, or to prepare a video for the entire presentation.
  - Then talk to the video as your presentation

Final Presentations

- Final Project submissions are due Sunday, Dec 8
- Final presentations are Monday, Dec 9 5:30pm (tentative)
- Use this time for:
  - Get a good night's sleep
  - Polish, practice and your presentation
  - Work on any demo videos etc.
- Unless you find a serious bug, don't make code changes