



# CSE 190

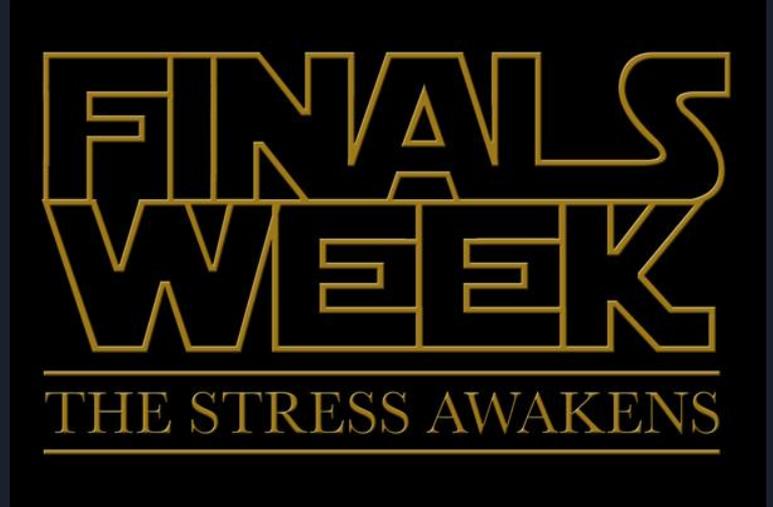
# Discussion 8

Final Project: Social VR



# Agenda

- Final Project Review
- VR Experience
- Extra Credits
- Presentations and demo
- Project examples
- Network Setup Review
  - Setup socket network
  - Other Tips
- Q&A





# Final Project Recap

- Link to the assignment: <http://ivl.calit2.net/wiki/index.php/Project4S18>
- Due Date: Tuesday on Finals Week (June 12th at 3:00pm)
  - Next Blog Due Date: Monday on Finals Week (June 11th at noon)
- Worth 25% of your grade!
- Some features that you must include:
  - Network - Dual user VR application
  - Original Model - Custom at least one model on your own
  - Collaboration - Two users need to work together on something
- Make anything you want! Your own idea!



# VR Experience

## - Usability

- Usability determines a big part of the experience!
- You shouldn't have to (overly) explain your controls!
  - Make control simple! You don't have to work on UI too much though.
  - Anyone with decent amount of VR experience should be able to easily figure it out
- Some question to ask to yourself:
  - How intuitive is my application to use?
  - Will my friends or other fellow classmate be able to figure it out?
  - Does user have any guide throughout the experience?
  - How does my controls compare to real world actions?
  - Am I showing too much information at same moment?
- Demoing the app to the grader in a right way also matters!



# VR Experience

## - Creativity, Aesthetics and More

- How original/unique is your application?
  - Try googling your ideas online or search in the Oculus Store / Steam.
  - See how other people implement the idea, and can you do something different?
  - Take advantages of the techniques used in past assignments!
- Many aspects that you can innovate:
  - Topics and application styles
  - Interaction techniques
  - Audio, etc.
- Think about color schemes, models, textures, materials, etc.
- [Resources of guidelines toward designing a good VR experience](#)



## Extra Credit

- We created a list of hackathon-style awards for teams with outstanding apps.
- They also serve as a good guideline for the features we encourage everyone to take care of in your design
- You are also encouraged to nominate yourself in your second blog and give fair justifications.

# Presentations



- Agenda for Tuesday, June 12th
  - 3pm - 4pm: Screening of videos made by each team (CSE 1242)
  - 4pm - 5pm: Group A science-fair style demos in the VR Lab (B210)
  - 5pm - 6pm: Group B science-fair style demos in the VR Lab (B210)
- Everyone needs to show up at 3pm for the video screening
- Graders will all be trying out your application during demos.
  - Be sure to practice demoing
  - Grades will not be decided on the spot
- Other students/guests may come to the demo session
- Try out people's projects from the other group as well!



# Good Presentation Tips

1. Make it short and tight!
2. Have a clear, and real, problem/target to solve/do.
3. Don't make them watch you type (or any other misleading actions).
4. Practice in the venue before doing the actual presentation
5. Have a backup machine/connection plan.
6. Never troubleshoot in real time.



# Give a good demo of your application

- A good/bad demo experience can sometimes boost/dampen user's impression when trying out your application!
- Note that your grade on the “subjective” aspects can depend your demo.
- Some other students/faculty who come to our final demos might try out your application as well!
- Some slides made by the VR Club might be helpful for you to prepare for a good pitch during the demo: ( [Link](#) )



# BLOG posts #1

You need to create a blog to report on the progress you're making on your project. You need to make at least two blog entries to get the full score. The first is due on Monday, June 4th at 12 noon, the second is due on Monday, June 11th at 12 noon.

The first blog entry needs to contain (at a minimum) the following pieces of information:

The name of your project (you need to come up with one)

The names of your team members

A short description of the project

One or more screenshots of your application in its current state



## BLOG posts #2 + video

In week 2 you need to write about the progress you made and update on any changes you made to team or team name. You also need to post another screen shot.

You are free to create the blog on any web based blog site, such as Blogger or WordPress. You should use the same blog each time and just add new blog entries. You are free to add more entries than the two required ones.

Each team also needs to make a 1-2 minute long Youtube video of their application, to show during the first hour of the grading event during finals week. We are going to create a Youtube playlist to which we are going to ask you to add your video.

Blog option:

Blogger Wix Weebly Wordpress ...



# Project Examples

- Projects made by students from last year
- They serve as inspirations but it does not mean all the projects received full credits.
- [https://docs.google.com/presentation/d/1iB5Ps-GM\\_eHtXD\\_adKBN\\_m80BPEIXbrVYAUtozPTe74k/edit?usp=sharing](https://docs.google.com/presentation/d/1iB5Ps-GM_eHtXD_adKBN_m80BPEIXbrVYAUtozPTe74k/edit?usp=sharing)



# Network Setup Review

- Read the example posted on the website:  
<https://www.codeproject.com/Articles/412511/Simple-client-server-network-using-Cplusplus-and-W>
- It goes over the basics of setting up the connection and sending packets over.



# Network Setup Review

- Messages may be split over the network.
- You may need to handle this if your packets are too large.



# Network Setup Review

- TCP vs UDP
  - In our case, local network with high bandwidth and low latency, TCP is good enough.
- Tick Rate - Ideally you can get it up to 90 ticks per second, but starting at a slower rate might help if you send too much data over the network



# Network Architecture

- To integrate it with the minimal example, it needs to create a “ClientNetwork” to connect to a “ServerNetwork” created and running in a different program/thread.



# Network Architecture

- You don't need to create a perfect networked game engine, there are a lot corners you can cut to make it easier on you.



## Network Architecture - Packets

- Create a serializable struct (no pointers) so you can have the server fill it out, and have the client just copy over the memory received over the network. (and vice versa)
- Makes it easier to parse each packet type.



# Network Architecture - World State

- Ideally, you would have a scene graph that would be updated over the network with deltas.
- But if you design it so the client doesn't care about the structure of the scene graph, you can get away with sending over an array of world space matrices with model ids.



# Network Architecture - Input

- Collect all the inputs into a single “snapshot” every frame, so the server can parse it easier.
- If the server runs at a different tick rate, you will need extra logic to update the input state.



QUESTIONS?

Also Feel Free to ask on Piazza!