Capstone Project Ideas

go/davis-capstone-projects

note: none of these project ideas have been tested, so it's unknown how well they'll work as capstone projects

1. Create a load-balanced, autoscaled, autohealed system of VMs. We'd need to think of some workload for which load balancing would be needed. One possibility might be FEA simulations. Another might be distributed ray tracing. Someone from the art department might be able to make use of a render farm. This would be an introduction to load balancing. *(It needs to be somewhat unpredictable load, data source? 10s of VMs.)*
2. Create an e-commerce website that can accept credit card payments *(making sth. secure enough, stripe, paypal, braintree, google wallet. Can be applied to social good, e.g., for local artists, animal shelters, recruit supplies and customers together)*.
3. Write a simple massively (or not so massively) multiplayer mobile game (possibly taking inspiration from netrek or pokemon go) *(shared state among many players hooked up with db, latency challenge, more broadly, can clone any existing video games. 50% of video game work is graphics)*
4. Use bigquery to sort through large data sets and route the results into advanced information visualization *(might be tricky to get access to a sufficiently large and interesting data set, though)* Bigquery has dataset for users to use. Suitable for text. Huge logs of events to visualize it.
5. Implement face recognition software such that could be used in face unlock, possibly using ML *(use their own photos?)*
6. Create a collaborative drawing app that can be accessed from web and mobile (a simple jamboard).*(Travis' favorite. Multiple clients with shared state. Latency (within 2s))*
7. [one-pager here] Create a shared cloud development environment based on CodeMirror. Bonus points if it can deploy to cloud functions or AWS lambda without needing to be downloaded to the local machine. *(low income neighborhood no real good ways to learn programming, being able to develop without installing anything. Infinitely scalable. A huge number of ways to be specialized.)*
8. remote control for your remote control: create a miniaturized device that can be attached to your TV remote that will make a sound when you press a button from across the room. *(challenge: make it small. Probably not a good fit for CS, but Xin will pass to ECE and MAE.)*
9. solar-powered, software-controlled wood burning kit. You would robotically move a magnifying glass along a path specified by a simple drawing program (which you would write). The wood would be positioned at the magnifying glass' focal length such that it would burn along the path. This would share much of the control concepts from DIY 3d printing rigs, with the added complexity that you'd need to find a way to keep it (or a mirror above it) pointed at the sun. (good embedded system project, hard end. MAE-CS joint)
10. Distributed rendering with deep compositing. Treat opacity as a per-pixel linear-piecewise function of depth for each channel (R,G,B). Render multiple scenes with transparency and combine them afterwards. Create a simple animation with one of the transparent parts moving over time, but keeping the rest fixed. (Done before as a bug-report, Kunfu Panda could not get blurred images. In graphics, ; maybe start with ray tracing? Only suitable for someone really into graphics. For someone in that category, will be the right scale.)