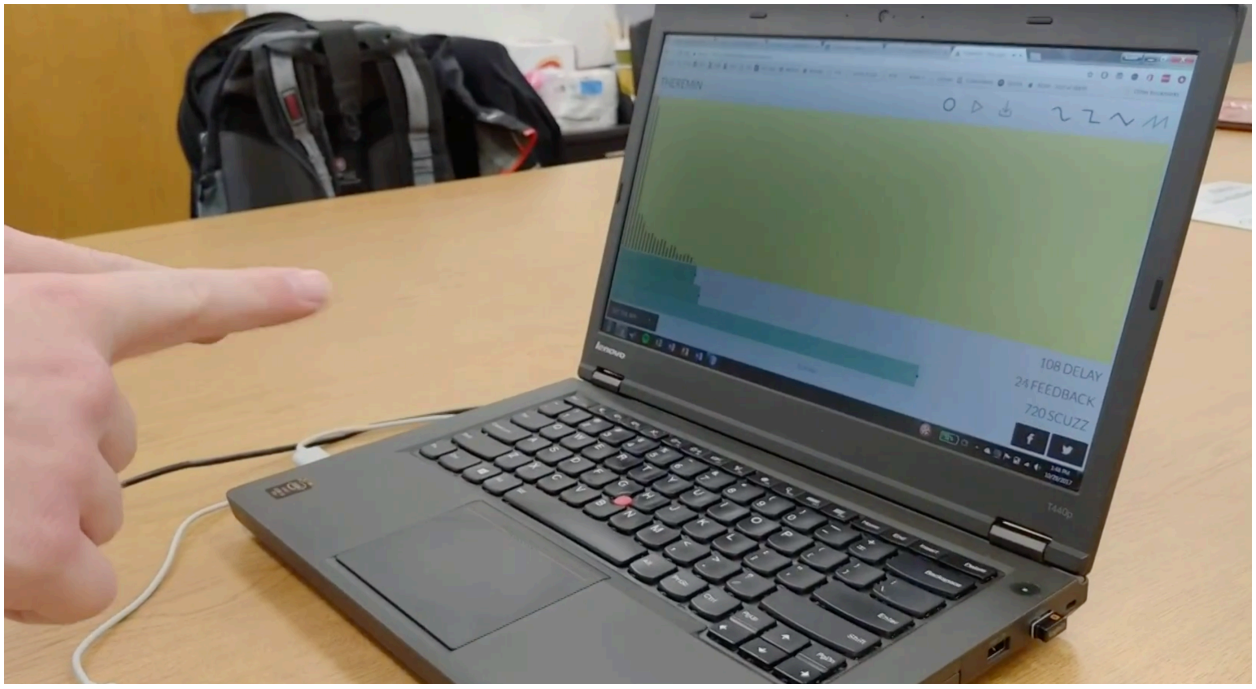
DESIGN FOR EXPRESSION / CREATIVITY

- **Group size:** Teams of 4
- **Prototype Demo Day:** Nov 1, 2022

‼ Before you begin ‼ Read this entire document and consider how you might want to split up work. Learn from your last design sprint. Consider collaboratively writing your Design Doc as you go. Don't forget about my expectations in your design document, along with the rubric I'll use to grade you. **You do not need a demo video in this assignment.**



Demo from previous HCI class... mapping sounds to hand input (theramin!)

The Prompt

The goal of this design project is **EXPRESSION** and/or **CREATIVITY** in **3D User Space**. How can you build an interaction that *feels* more expressive than the traditional mouse + keyboard?

- **You will use browser-based pose estimation libraries** to infer hand and/or body positions
- **You have complete freedom to define the output.** While I demonstrated **p5js** in class as a library that lends itself well to expression, you are free to use any other library or system... as long as it can be run in the browser! You could also choose to have no visual output - controlling sound instead?
  For example: https://www.youtube.com/watch?time_continue=19&v=CC9RBx1hKwc
- **You may design a game if and only if** using hands / body offers *clear* benefits over mouse/keyboard input and is creatively incorporated into the game's input. Remember: the design goal is to enhance creativity and expression, not to simply map the controls from an existing game to clumsy gestures.

This is intentionally open-ended! Build something that you find interesting!

Additional Readings to help inform your design:

- 3D User Interfaces by Doug Bowman
- Criticism about in-air interfaces by Antti Oulasvirta

Team Suggestions

Read through this document and assign some roles. You can collaborate however you'd like, but at the very least I recommend

- Determine meeting times and deadlines (that are well before my deadline)
- Set up a Glitch account that you can collaborate on.
- Separate coding tasks
  - Someone who writes code that focuses on the output (visuals / audio that happen on the screen)
  - Someone who writes code that focuses on the input (capturing gestures / 3D user input)

You could also consider...

- Have someone who is responsible for organizing the design evidence
- Have someone who is responsible for the final design document

In either case, your design goal is simply to maximize the enjoyment that people have with your app.

Note for keeping design evidence: Think about how you want to document your design process. For example, given the physical nature of 3D UIs, you may need to rely more on videos and pictures than your previous design activities.

Ideation

> *In traditional UIs, we usually try to design without regard for the display or the input device (i.e., display- and device-independence) ... But in 3D UIs, what works on one display or with one device very rarely works exactly the same way on different systems.*
> - Doug Bowman, 3D User Interfaces

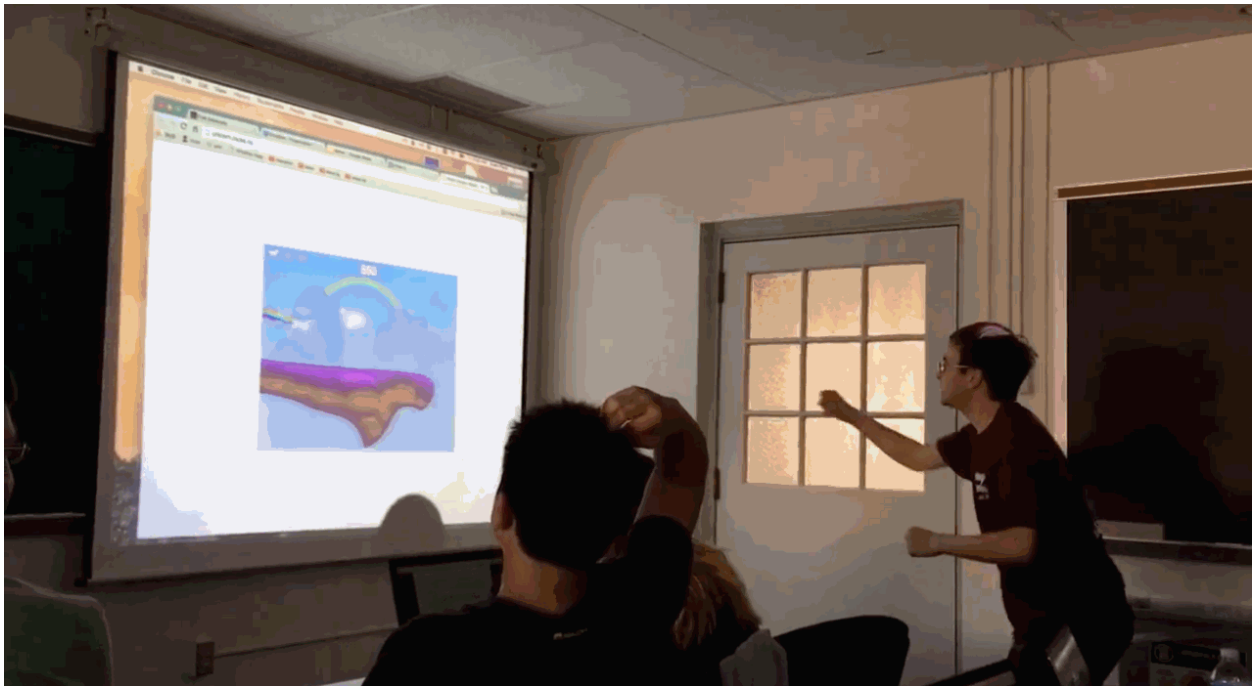
The goal of expression is different than an efficient or usable interface. Your brainstorming shouldn't consist of *just* writing down sketches and trying to intellectually determine how it fosters expression of creativity. Do it! Try the gestures yourselves! **Expression isn't just embedded in the application, but in the interaction design... are some motions more engaging than others, even if they might be less accurate?** You might even want to try ideating with the wizard-of-oz approaches described in the next section.

Consider your constraints: In our previous assignment, I had you ideate without the burden or limitations of technical constraints. However, in this case, your hardware (monitor size, for example) is likely to have a significant impact on interaction. In addition, the pose estimation may surprise you in the ways that it is accurate and/or inaccurate.

I would recommend that you get a rough sense of your tech's capabilities right from the start - perhaps assigning a person or two on your team to begin playing pose estimation in p5js (or via teachable machine models) while the rest of your group brainstorms. Check out the resources I provide in a later section

---

Formative Testing



Before you build it, Wizard-of-Oz your idea with other students. To do this, use the following process:

- Define a set of actions for input
- Show and/or explain those actions to interact with your experience.
- **IN PERSON:** The *participant* should be looking at the application monitor, but someone in YOUR group (the *controller*) should have access to the keyboard/mouse.
    - The *controller* should ONLY look at your participant, NOT the screen.
    - When the participant performs an action, the *controller* hits the corresponding button (or moves the mouse). While you may not be quite as responsive as a computer, it will let other people experiment with your design.

For an example, see the in-class prototyping we did in HCI. In this video, a student in the bottom left corner (only occasionally in the screen) is hitting buttons while the participant jumps and punches to play Robot Unicorn Attack.

Building It

You are not *required* to use p5js and ml5, but they are what I am suggesting given that they will help you more rapidly prototype. My suggestions:

- Given how easy it is to make interactive visuals + sound, use **p5js** for your output
- Useml5 + teachable machine models for input
    - Use posenet/handpose/etc coordinates for continuous input
    - Use teachable machine models for discrete input
- Set up your code on Glitch (for collaboration)
- You might want to consider coding the output first - initially program it to respond to dummy input (keyboard presses or mouse movement, for example). This will allow you to independently and simultaneously prototype the pose input before coding it (two people could even work on these components in separate code bases!).

Resources for p5js:

- P5js reference: https://p5js.org/reference/
- P5js tutorial videos: https://thecodingtrain.com/tracks/lang/p5-js/topic/all
- Basic p5js template on glitch: https://glitch.com/edit/#!/p5js-template-fa22

Basic p5js examples using OOP:

- P5js circle OOP example: https://editor.p5js.org/evanpeck/sketches/wEUvA1QqL
- P5js bouncing balls OOP example: https://editor.p5js.org/evanpeck/sketches/O7MjzPFxb

Resources for posenet:

- ml5 website: https://ml5js.org/
- Handpose p5js demo in Glitch: https://glitch.com/edit/#!/p5j-ml5-template
- BASIC p5js DEMO: https://editor.p5js.org/evanpeck/sketches/SJmcoWcpm
- Silly HCI Game: https://editor.p5js.org/evanpeck/sketches/IiFCWka3W

Very helpful code + tutorial:

- Posenet + p5js video tutorials: https://thecodingtrain.com/tracks/ml5js-beginners-guide/ml5/7-posenet/1-finding-key points

Other important resources:

- Teachable machine (for model creation): https://teachablemachine.withgoogle.com/

Testing It

On our class demo day, you will have the opportunity to test your creation with real users. **This is also part of the design process**. While you likely won't have time to make significant changes after this point, think back to our user testing lecture and carefully consider how you want to determine what changes you should make if you had another week or month. I would expect these reflections to be in the design document.

Deliverables

- **Prototype Demo Day (Tuesday, November 1st):** You should have an accessible link (likely on Glitch) handy and ready for you to present in class on the day of your demo.
  - You will post your link on Discord during class
  - Your classmates will explore critique your work (using the *I like, I wish, What if* framework)