Computer Systems from the Ground Up

Winter 2025

Have you ever wondered ...

- How a computer represents data?
- What operations a computer understands?
- How a program executes?
- What happens when a user types on keyboard?
- How text and drawing appears on a display?

- How things *really* work inside this wondrous box?

These questions and more to be answered by studying **computer systems!**

Learning goals

1) Understand how computers represent data, execute programs, and control peripherals

2) Master your tools

Understanding is empowering!

RISC-V processor and memory architecture

Peripherals: GPIO, timers, UART, …

Assembly language and machine code

Low-level representation of information / bits

From assembly language to C

Function calls and stack frames

Serial communication and strings

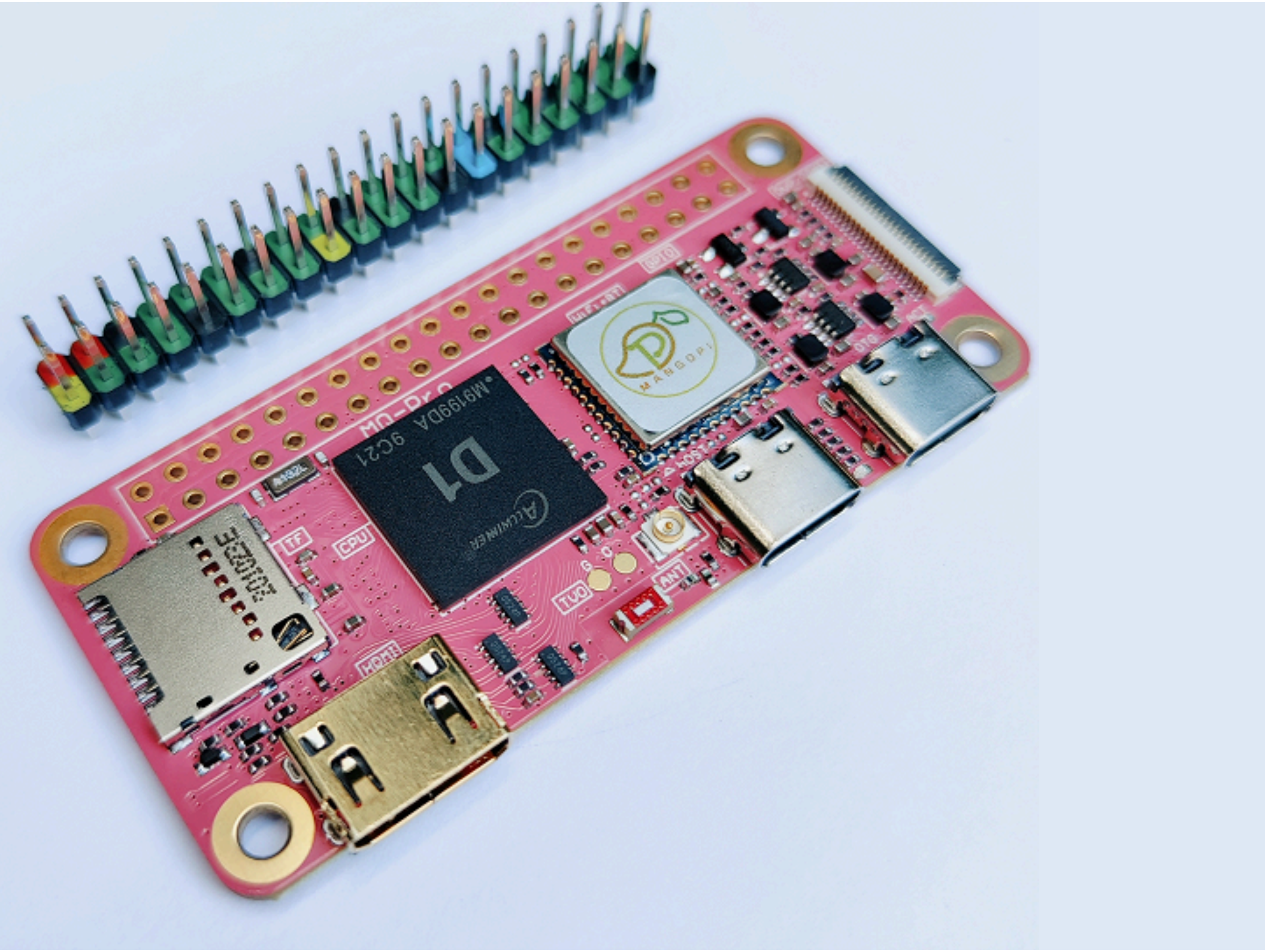
Modules and libraries: Building and linking

Memory management: Memory map & heap

Processor control, interrupts

Master your tools

UNIX command line: bash, cd, ls, …
Text editor: vim, emacs, sublime, …
Programming languages: C, assembly
Compiler: gcc
Assembler: as
Linker/loader: ld
binutils: nm, objcopy, objdump, …
make
git and github.com
documentation: markdown

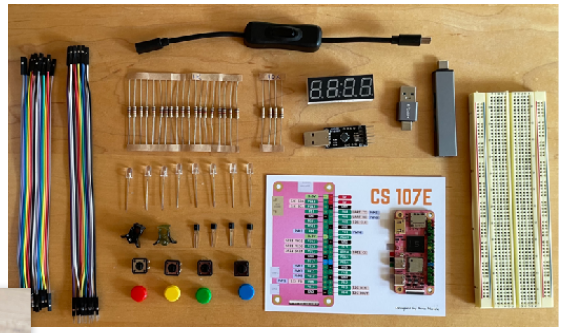

MANGOPI
D1
TF
CPU
HDMI
TVO
ANT
OTG

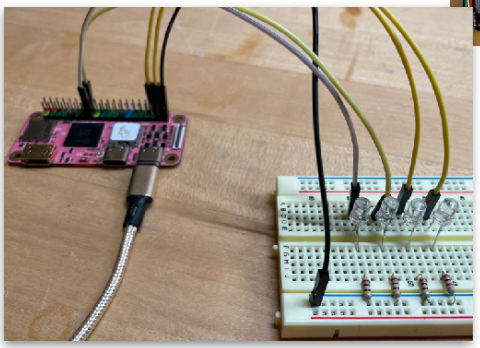
Course Syllabus

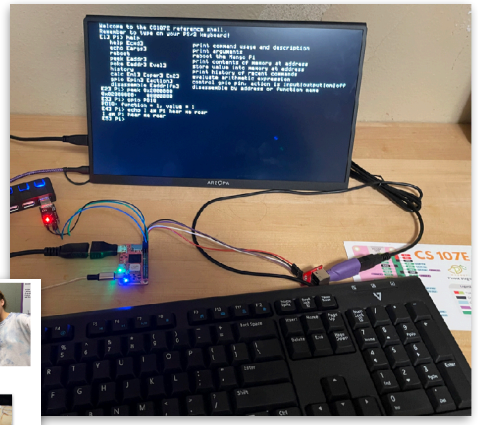
§1 Bare Metal Programming
- RISC-V architecture and assembly language
- C functions and pointers
- Serial communication
- Linking and loading
- Memory allocation

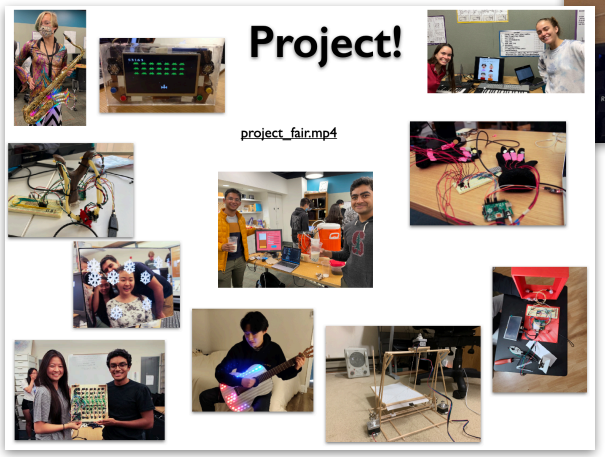
§2 Build a Personal Computer
- Keyboard
- Graphics
- Interrupts

§3 Create Your Own Project
- Sensors
- Performance









Schedule

First ~8 weeks follow **weekly cadence**, each week has focus topic

- Pair of coordinated lectures on **Fri** and **Mon 10:30am-12pm**
- Lab session Tues/Wed evening **6:30-8:30pm**
- Assignment out after Wed lab, due before next Tue lab

Last 2 weeks are **project**-focus

- Lecture on special topics
- Labs used as project work sessions
- Projects due at end of finals week
- Demo day is **Fri March 21st 9-11am**

Lectures

Attendance is **necessary**

Content is unique to our course, no textbook
The readings/slides not a standalone resource
Lectures not recorded

In-person attendance allows you to participate, ask questions, keep on schedule, stay connected

Labs

Set of guided exercises, follow up on lecture content
- Work in groups
- Complete exercises and check in with staff
- Leave lab ready to start assignment!

Lab participation is mandatory

Philosophy: lab is hands-on, collaborative, supported, **fun**!

Assignments

Weekly assignments that build on each other

This is where the learning really happens!

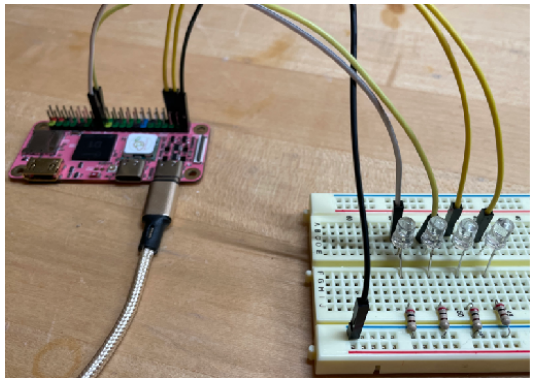
Each assignment has

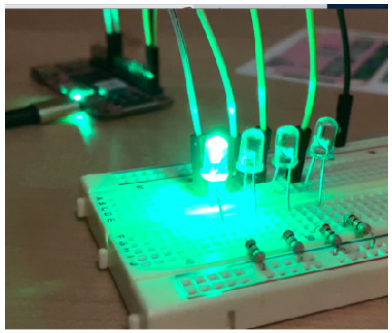
- **Basic** requirement (tight spec, guided steps)
- Optional **extension** (opportunity for exploration/ creativity)

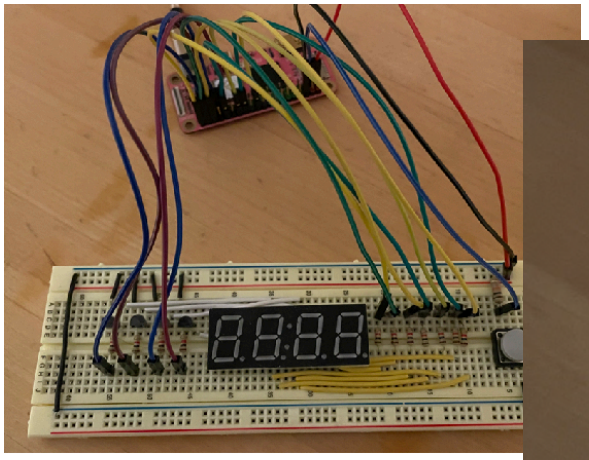
Opportunity to revise/resubmit to correct missed basic requirements

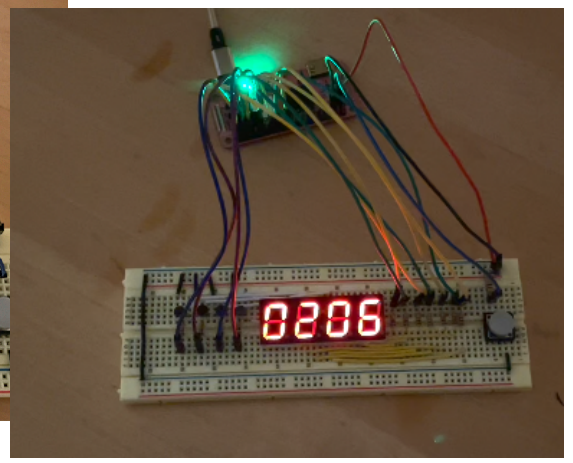
End goal is complete working system of your own

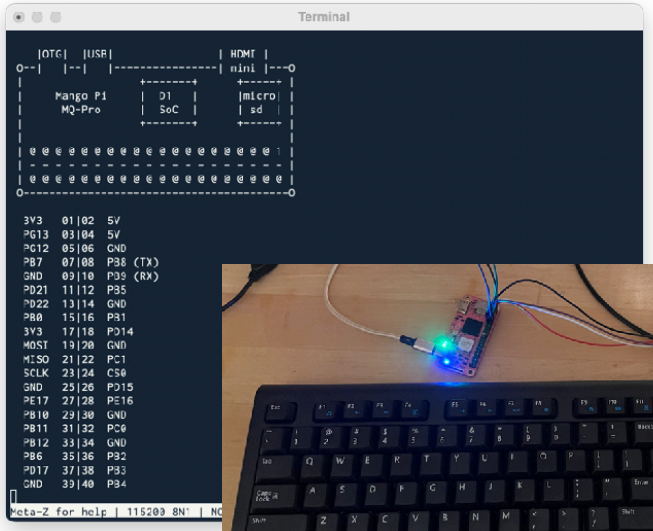
Your complete system

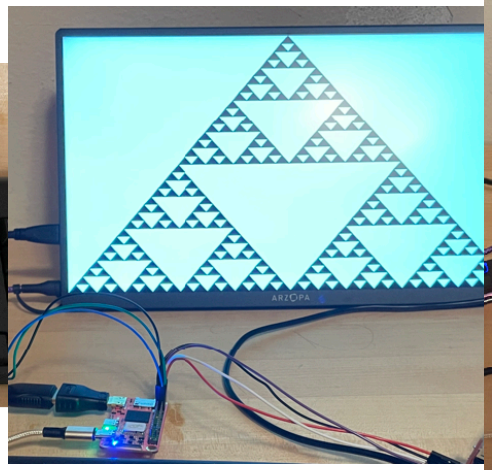


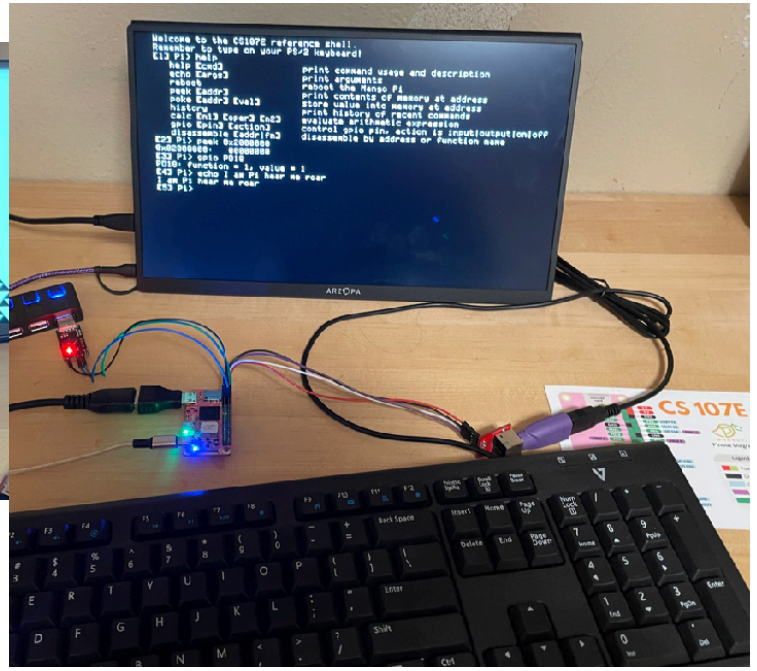






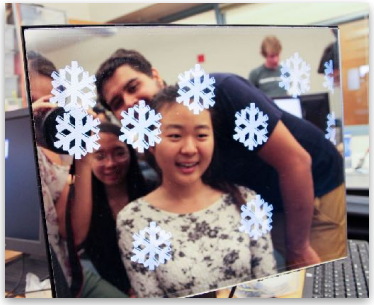




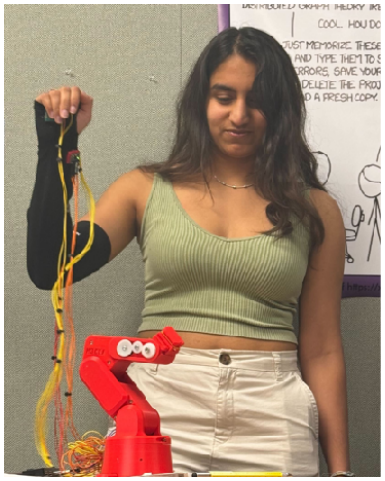


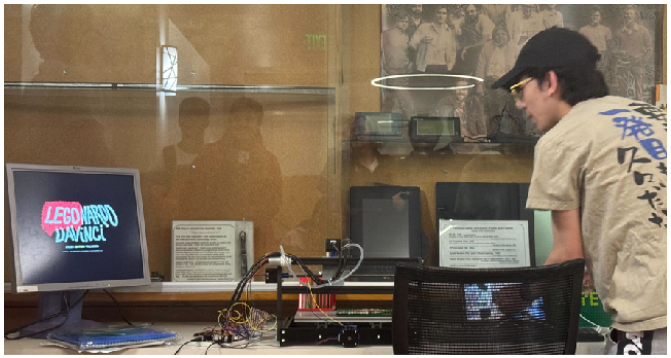
Nearly every instruction is code you wrote yourself!

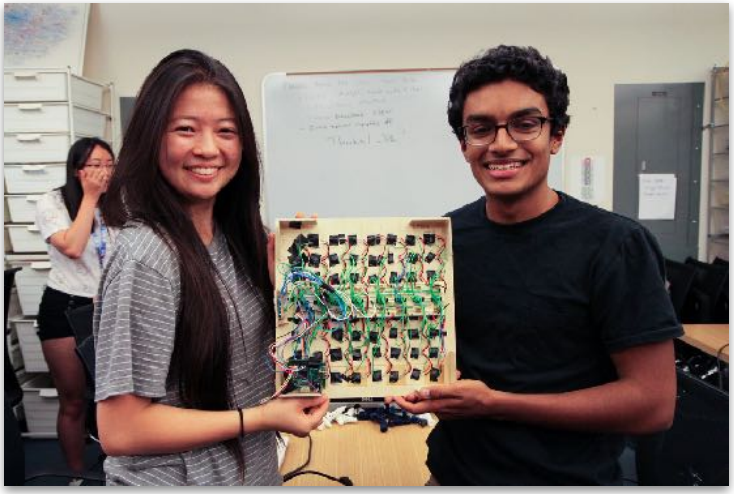
Project!

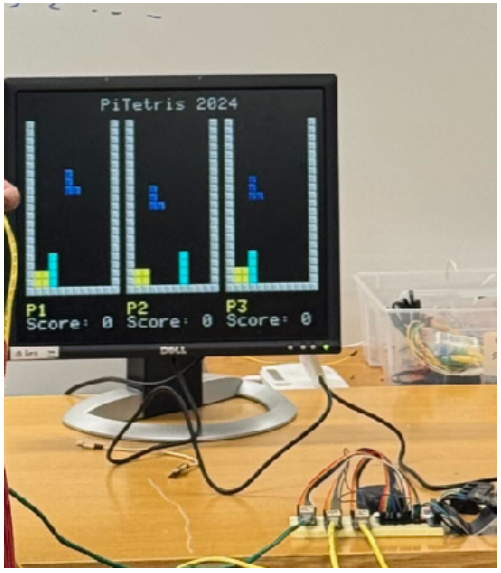


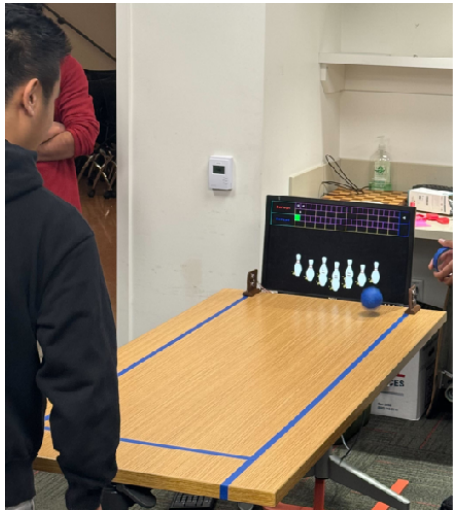
project_fair.mp4











Markers for success

Required: CS106B, C++, debugging

Also bring your
- Curiosity
- Perseverance
- Motivation

Value and appreciate small learning community!

How to thrive
- Consistency, follow through
- Leverage our resources, support, feedback
- Be **curious**. Learn by **doing**. Ask for and offer **help**.

Enrollment process

We can enroll up to 40 students

Lightweight questionnaire for you to share your interest and fit (linked at https://cs107e.stanford.edu)

- Questionnaire open Dec 6 through Dec 20
  - *Must submit questionnaire to be considered, don't forget!*

- Decision email over winter break, ask for your response
  - *Please only accept if your commitment is firm*