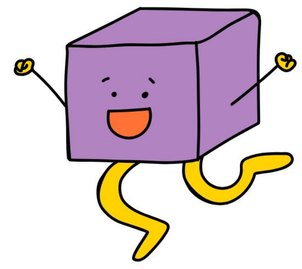
Connexus: Think Outside the Blocks

Team B4: Alexis Duong (aduong), Nicole Feng (nvfeng), Wen Hui Leng (wleng)
18-500 Capstone Design, Spring 2025
Electrical and Computer Engineering Department
Carnegie Mellon University

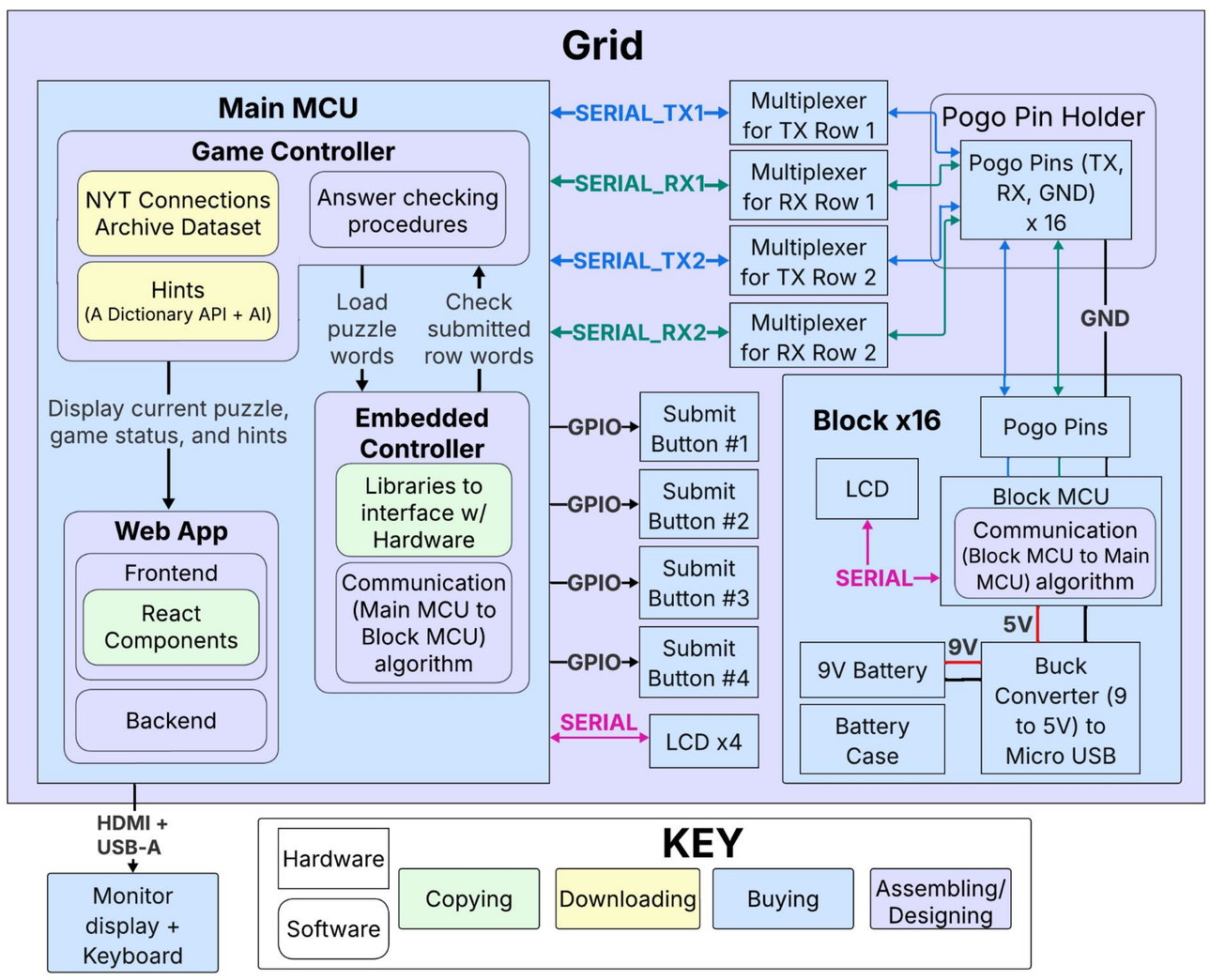


Product Pitch

Connexus is a physical and digital reimagining of the New York Times Connections game, designed to be more **accessible** and **engaging**—especially for English Language Learners and novice players. While the original game relies heavily on cultural knowledge and vocabulary nuance, Connexus uses **programmable word cubes** for tactile play and pairs them with a **virtual companion** that offers hints and guided support. This rendition of the game reduces frustration, cultivates critical thinking, and transforms the game into an interactive learning experience. Connexus successfully met all defined use-case requirements, including hardware constraints, fast and accurate game flow, and virtual companion reliability. User testing also confirmed improved game completion and high satisfaction among players.

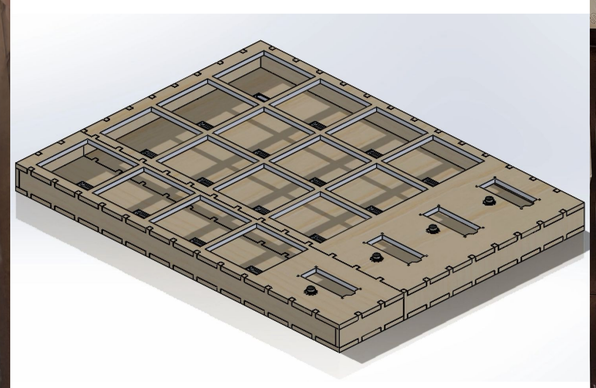
System Architecture

Connexus consists of two subsystems: the **grid** and the **blocks**. The grid contains the main microcontroller, which runs the web app, manages game flow, and coordinates communication with each block. Each block includes its own microcontroller and LCD to display words and execute a communication algorithm that syncs with the grid. Serial multiplexing and GPIO inputs enable efficient data flow, while a virtual companion assists users through an integrated web interface.



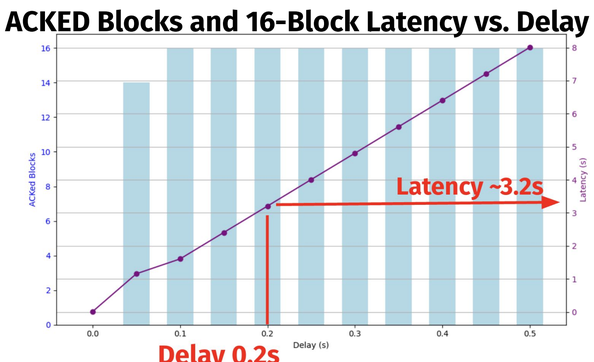
System Description

The grid contains a **Raspberry Pi 5** that runs the webapp that was made with **React, Fast API, Socket.io, Merriam-Webster API,** and **Gemini**. Users can look at hints (and super hints) and get feedback through sound effects from the webapp as they play the game. It also coordinates indexing through each grid position with the multiplexer to establish a **UART** link with the block. The embedded controller will send queries to the block such as send word/color and get word, and there is an algorithm to receive ACKs from each block. Each row has a submit button and LCD to display the category after a successful submission. Each block has a **Raspberry Pi Pico** that parses and fulfills queries, a **9V rechargeable battery** with an accessible micro-C port and on/off switch, a power adapter that directly plugs into the Pico, an LCD to display the word and change color according to submission and **magnetic pogo pins** to establish an electrical connection to the grid.



System Evaluation

We tested each subsystem focusing on latencies and meeting all the physical specs. We looked into how delays between attempts and baud rate affected our upload latency as well as looked at hint retrievals. Gemini was added to cover holes in Merriam-Webster API. Also, we ensured block in all 16 positions.



Single Word Retrieval Latency

Merriam-Webster	0.11 sec
Gemini Flash 2.0	1.5 sec → ~15x slower!

[MW] Average "good" game latency = **1.81 sec**
[MW + Gemini] Average "bad" game latency = **4.28 sec**
Average game latency = (1-(33/614))*1.81+(33/614)*4.28 = **1.94 sec**
We will meet our 3.2 sec latency requirement ~95% of the time

Use-Case /Design Requirements:

Metric	Target	Actual
Puzzle Upload	< 4s/block + 100% accuracy	3.23 s + 97.5%
Hint Retrieval	< 3.2 s/block + 95% rate	1.94s + 100%
Size/Weight	Block: < 4" x 4" x 4" + < 400 g Grid: < 2' x 2' x 2.8"	Block: 3.3" x 3.3" x 3.3" + 181g Grid: 1.27" x 1.8" x 2.5"
UART	< 400 ms/word + 100% ✓	200.3 ms + 100%
Answer Checking	< 4s/block + 100% accuracy	1.99s (❌: 0.94s, ✅3.04s) + 97.5%

Conclusions & Additional Information

Team B4 worked hard in not only achieving our MVP of 8 blocks but also surpassing our goals by fully implementing the game with all 16 blocks. Alignment was our greatest technical challenge and we did the best we could with the grid frames to guide the block in and 3D printed holders. There is still so much to expand upon in terms of the virtual companion and additional enhancements to the grid and block itself. Connexus made us think outside the blocks and we hope you have as much fun as we did implementing this.

A special thank you to Professor Tamal Mukherjee and Ankit Khandelwal for helping us get here!

https://course.ece.cmu.edu/~ece500/projects/s25-teamb4/

Electrical & Computer ENGINEERING
Carnegie Mellon