

Jordan Taylor, Alexander Zacarias, Carson Brown, Jeff Hernandez, Josalyn Abbott,
Justine Talia Gacho, Phillip Cook, Jesus Martinez, Jeff Wright, Aryan Prasain, Dang Khang Lam

Executive Summary

Our project aimed to construct a claw machine and the infrastructure necessary to control the machine over the network via a remote web application.

Owners of such a project would be able to expose their claw machines to a much wider audience, while end users of this project will gain the luxury of playing a claw machine without ever needing to be physically near one.

Background

Traditional arcade claw machines rely on mechanical components and simple analog circuitry. While these games are a nostalgic fixture in entertainment venues, they offer limited interactivity and lack integration with modern digital systems. As a result, they fail to reflect the growing relevance of Internet of Things (IoT) technologies in contemporary engineering and computer science education.

This modernized claw machine serves as a compact, modular learning tool for students exploring IoT systems. It supports experimentation with cloud connectivity, sensor integration, and software development. Instructors can use the system to demonstrate real-world applications of networking protocols, control algorithms, and system integration.

Experimental Setup

The IoT Claw Machine features remote claw operation using the designed application with on-screen controls through web socket protocols. The user has a multi-camera stream to view the cabinet from multiple angles to capture their prize. This feature uses OBS for smooth video delivery and allows users to switch between views. The application offers a single sign-on by having users authenticate with their Google account, providing an extra layer of data security and accessibility. Our token simulation system integrates with the database and user account information to verify token availability and automatically deduct tokens per play session.

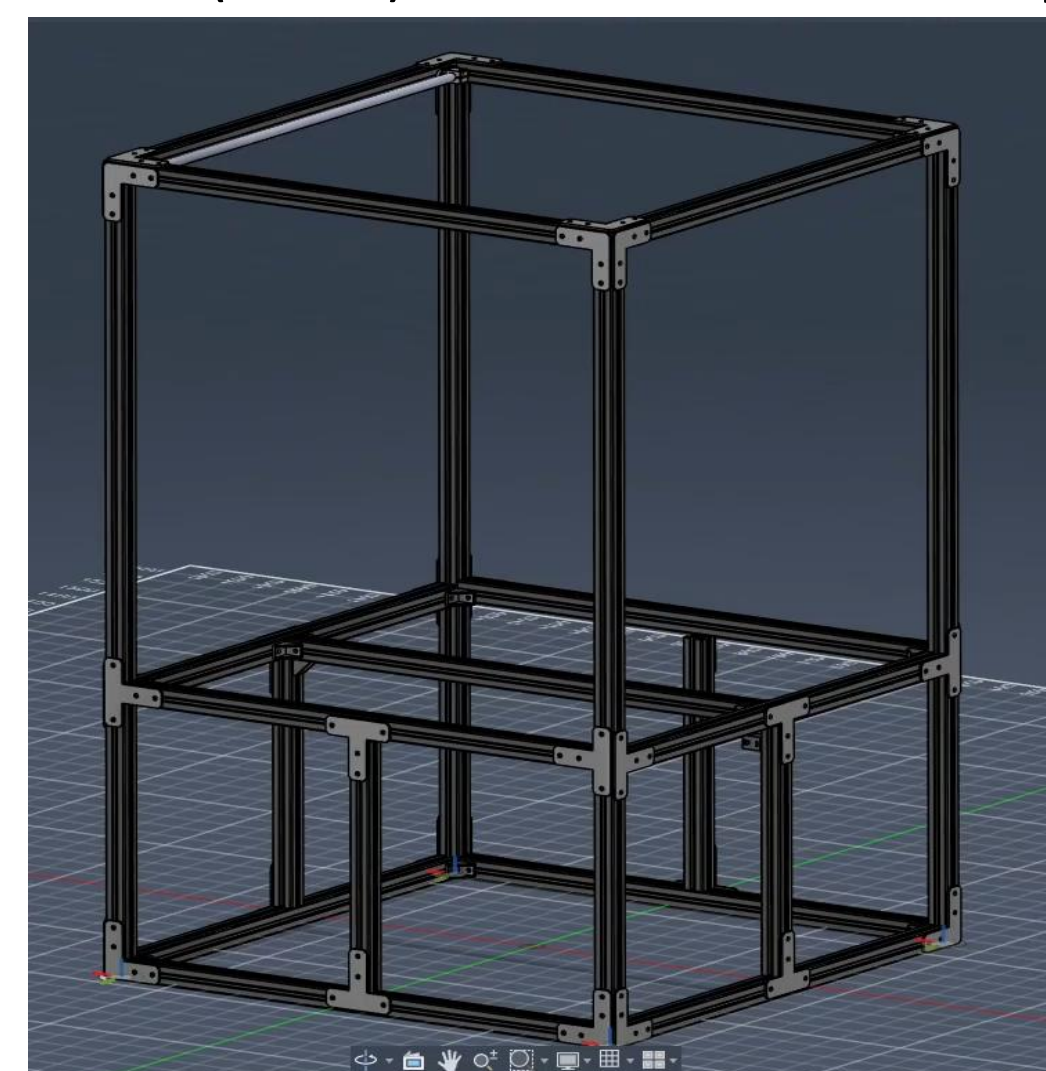
Initial analysis and constraints:

- Live video feed delays: The video feed may have latency, making it difficult for users to play the arcade machine through our application. The video stream also needs to be fast enough to synchronize with the latency of the claw machine controls to give the appearance of live control. Using the UTA networks, the stream must be compatible with the API in our application.
- Communication from backend server to the application and cabinet: Potential delays or integration issues may hinder prompt and effective communication among the three subsystems. The claw machine is network dependent. The claw machine communication will depend on the specifications of the selected microcontroller, motor drivers.

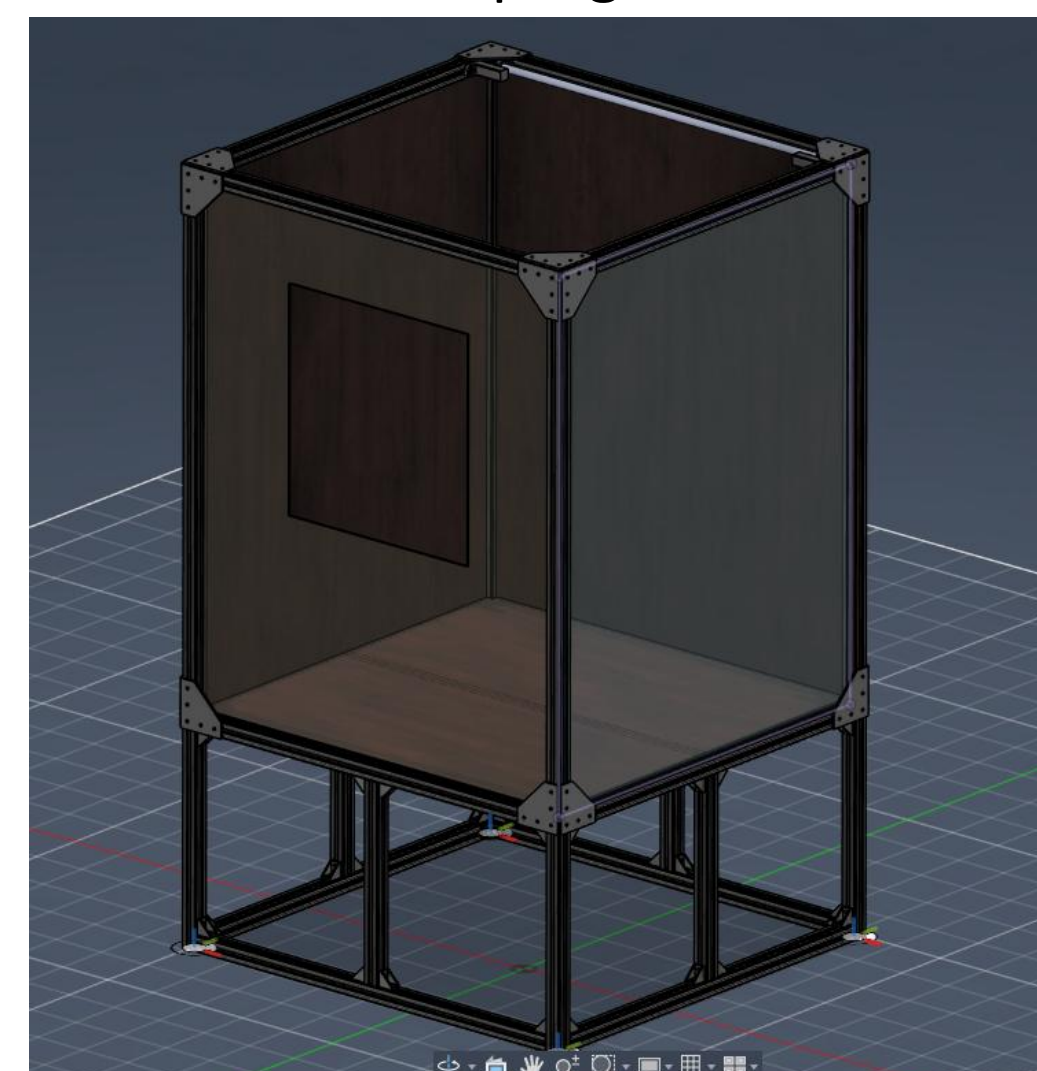
Experimental Test Plan

Three sub-teams were made to work on each subsystem – hardware, frontend, and backend. Each sub-team created and tested their section before full integration.

- The cabinet was designed using the computer-aided design software, Fusion, to build prototypes, discuss measurements, and find faults in design early. The cabinet features motors, a gantry arm, sensors, LEDs, and camera modules attached to the aluminum extrusion frame. The cabinet includes an emergency stop button to cut power during handling, along with wheels on the base for easier transportation.
- The application is written in React Native for the user interface design, user controls, token management, and live video display.
- The backend server is written in Go and is deployed on Google Cloud to handle real-time communication, user authentication, and video streaming. Our SQL database stores user information and sign-in in data from the application.
- Latency tests were conducted to determine the better communication protocol for cabinet input. WebRTC vs WebSocket: the tests concluded a WebSocket as the ideal choice for our project. The communication protocol has reasonable latency for our project and implementation of WebSocket's fits within the time constraints; however, WebRTC (WHIP) was used for video as part of the OBS Studio program.



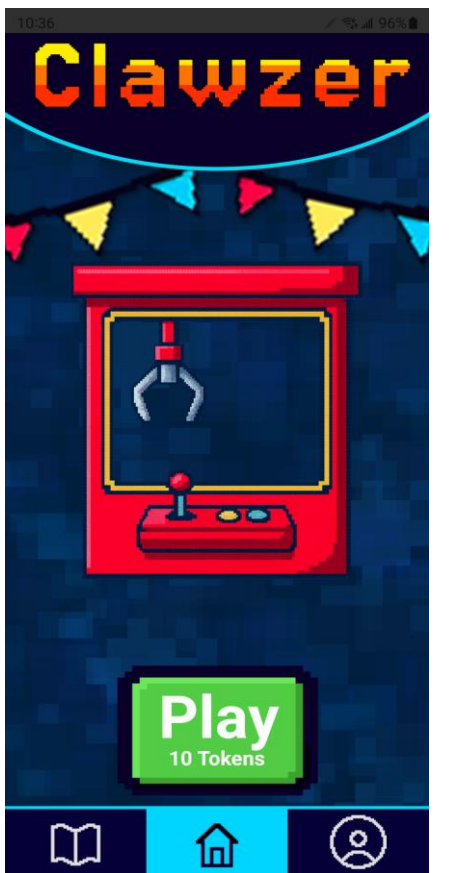
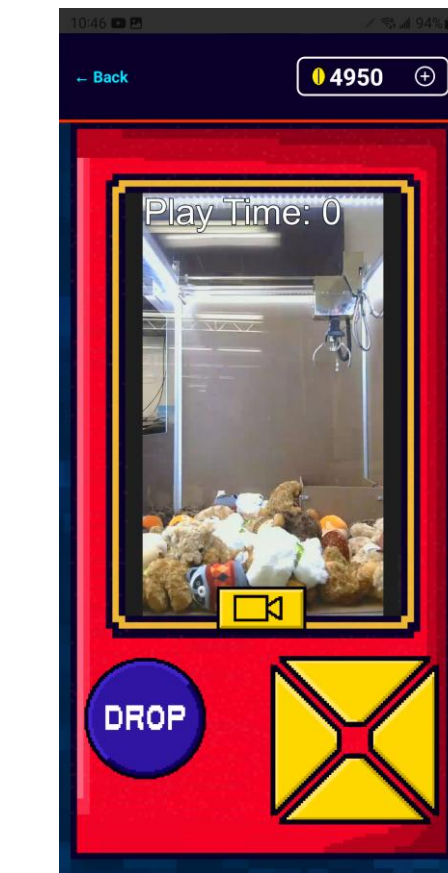
First Frame Draft



Final Frame Design

Experimental Results

- The completed application includes key screens such as the login page, arcade play page, and token purchase page. The play page features a cross-shaped control button, a cabinet view toggle, and a drop button for gameplay interaction.
- The application is user-friendly and designed to support future expansion to additional arcade machines. It also handles multiple users at a machine through a queuing mechanism
- The cabinet is structurally robust and designed to meet safety regulations. Its frame fits through standard UTA doors, while full-swivel locking wheels improve mobility and transport.
- Communication between the cabinet and application has minimal delay, achieving the design goal of live control for the users.



Conclusions

- The IoT Claw Machine successfully met its key requirements and educational goals. The project proves a functional claw machine cabinet controlled remotely by communicating with our application through standard network protocols.
- The system is built with scalability in mind, allowing more cabinets and higher traffic through application-level expansion.
- We would like to thank our sponsor, client, and lab manager for the support throughout the project.

References

1. ClawCraziness. We're making a 24/7 online claw machine app!, 2019.
2. Cameron Coward. Build your own claw machine that doesn't cheat, 2025.
3. MakingStuff. Making a claw machine, 2015.
4. aiortc library documentation. <https://aiortc.readthedocs.io/en/latest/index.html>. Referenced for possible software architecture use.
5. Ansi/ria robotic safety standards. <https://www.robotics.org/>. Includes: ANSI/RIA R15.06-2012.
6. Association for computing machinery (acm). <https://www.acm.org/>. Referenced for software engineering and computing ethics standards.
7. Ieee standards association. <https://standards.ieee.org/>. Includes: IEEE 802.11, IoT Protocol Standards, IEEE/IEC design standards.
8. International electrotechnical commission (iec). <https://www.iec.ch/>. Includes: IEC 60204-1,
9. IEC 61499.
10. International organization for standardization (iso). <https://www.iso.org/standards.html>. Includes: ISO 9241-210, ISO 9421-112, ISO 9001, ISO 27001, ISO 27018, ISO 28219.
11. Internet engineering task force (ietf). <https://www.ietf.org/>. Includes: RFC 6749 (OAuth 2.0), TLS 1.3.
12. National fire protection association (nfpa) - national electrical code (nec). <https://www.nfpa.org/>. Includes: NFPA 70 (NEC).
13. Open web application security project (owasp). <https://owasp.org/>. Includes: OWASP Application Security Verification Standard (ASVS).
14. Underwriters laboratories (ul). <https://www.ul.com/>. Includes: UL 22, UL 62368-1.
15. Web content accessibility guidelines (wcag) - w3c. <https://www.w3.org/WAI/standards-guidelines/wcag/>. Includes: WCAG 2.1 AA