Website

Zach Simons
Josh Cobian
Alan Ekstrand

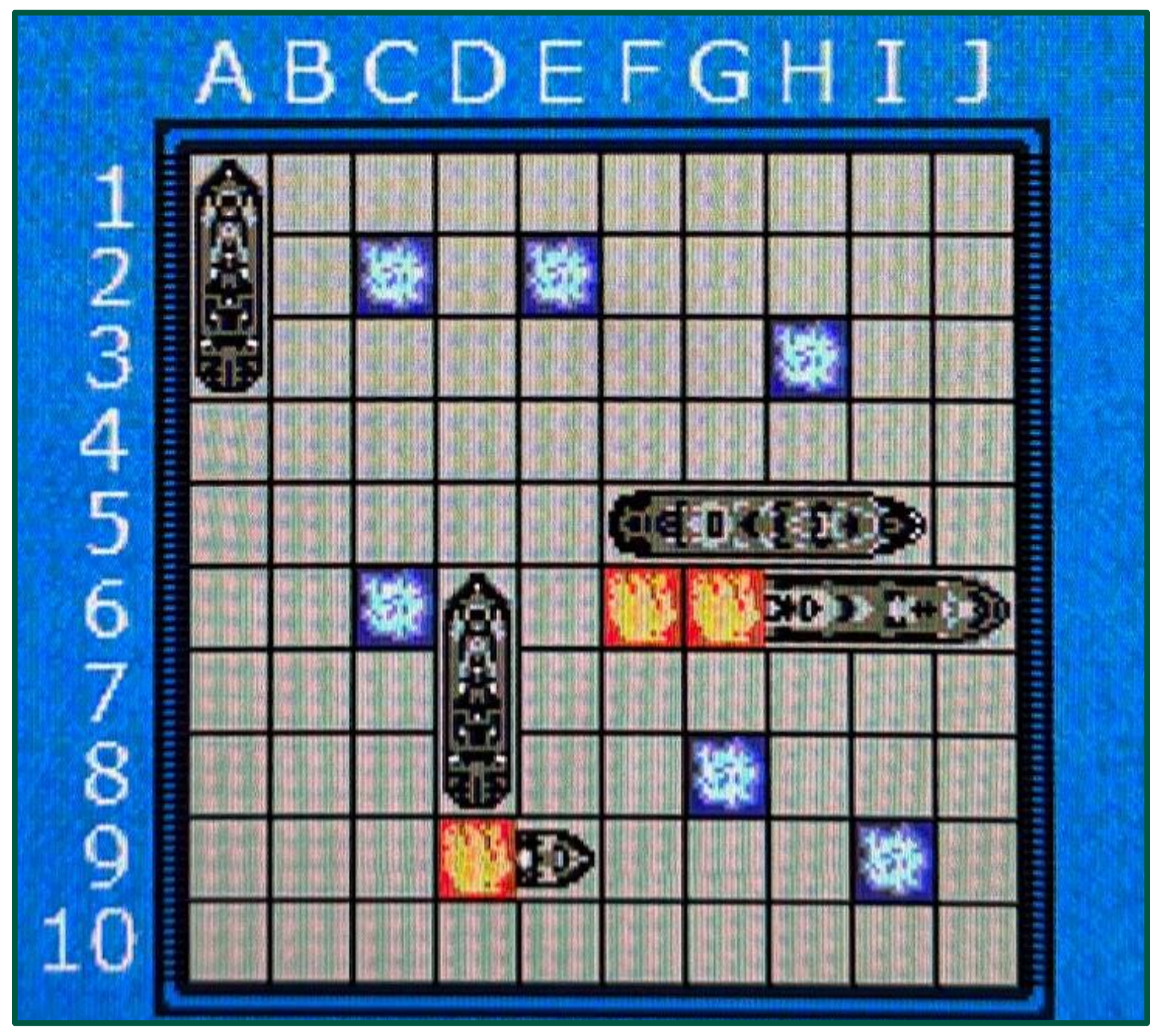
ECE 554

Battleship

Jacob Edmundson
Jaime Campos

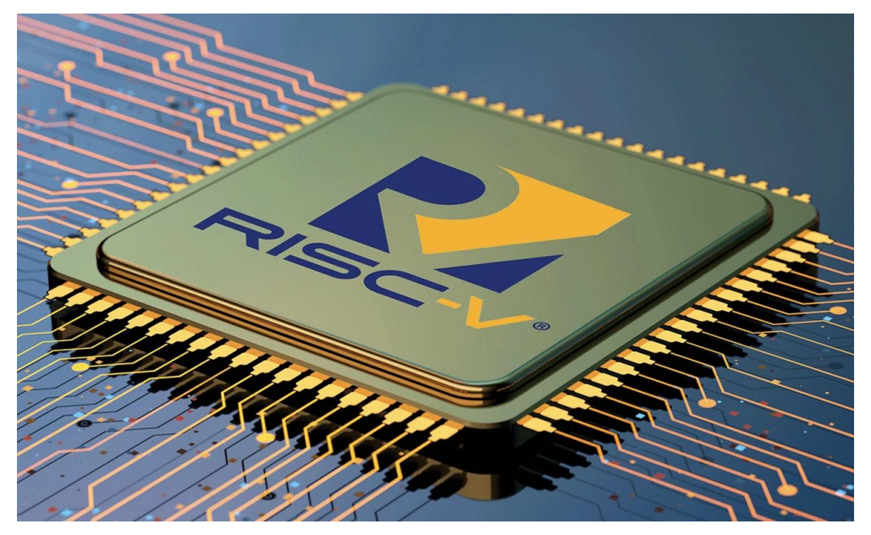
Overview

We built the classic board game Battleship on an FPGA! Grab a friend and challenge them on the high seas. Defeat them in a game of wits and deceit to ensure your victory!



Processor

RV32I ISA

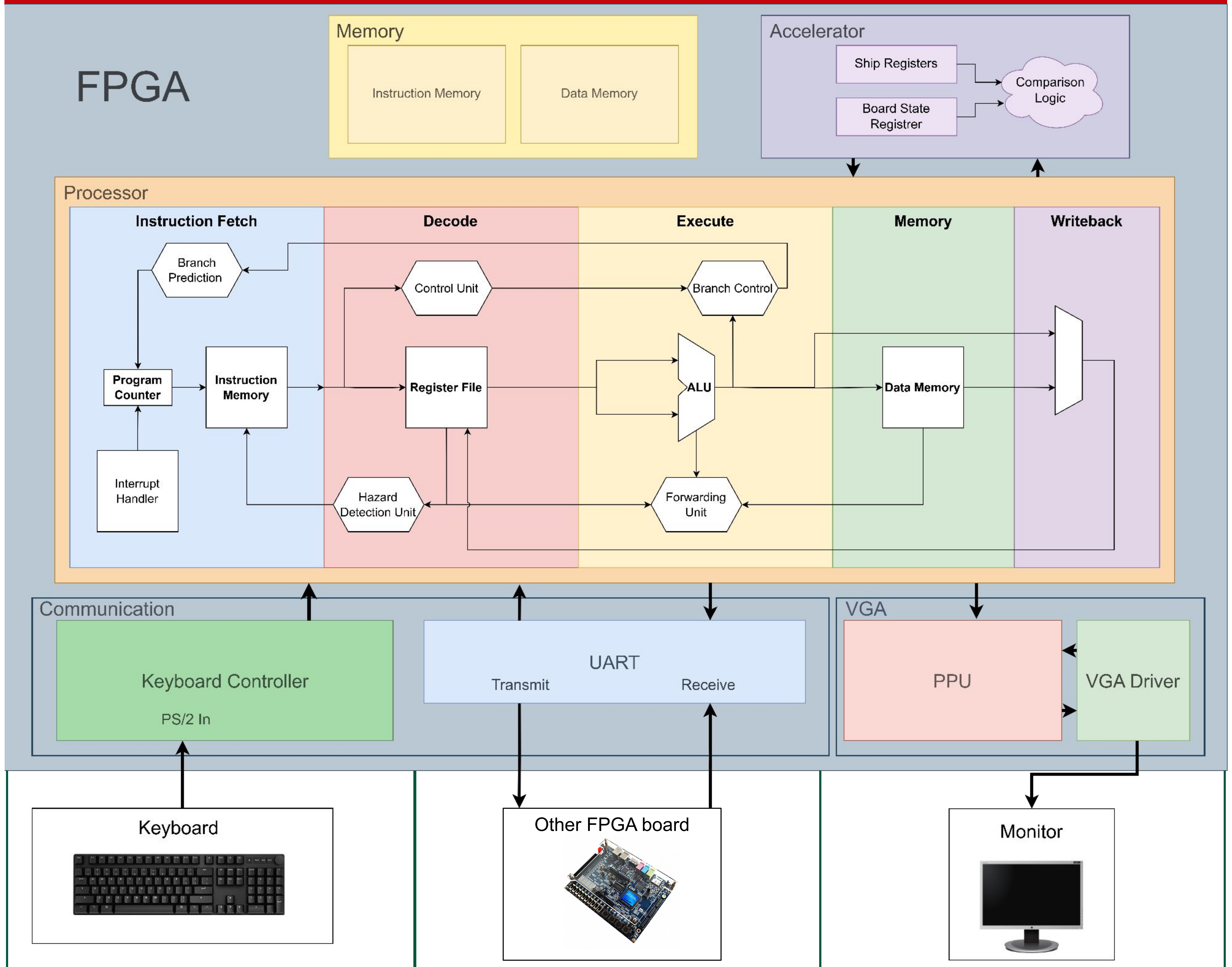


- Added custom instructions for communication and interrupts.

Hardware Optimizations

- 5 Stage Pipeline
- Branch prediction
- Pipeline forwarding
- Accelerator for increased prediction algorithm accuracy

System Block Diagram



Player Input

- When a key is pressed, data is sent triggering an interrupt in the processor
- Depending on the key pressed, a different byte is sent

Communication Between FPGAs

- GPIO pins on the FPGA are used to send data over UART, triggering an interrupt in the processor
- Used to synchronize gamestate between FPGAs

Picture Processing Unit (PPU)

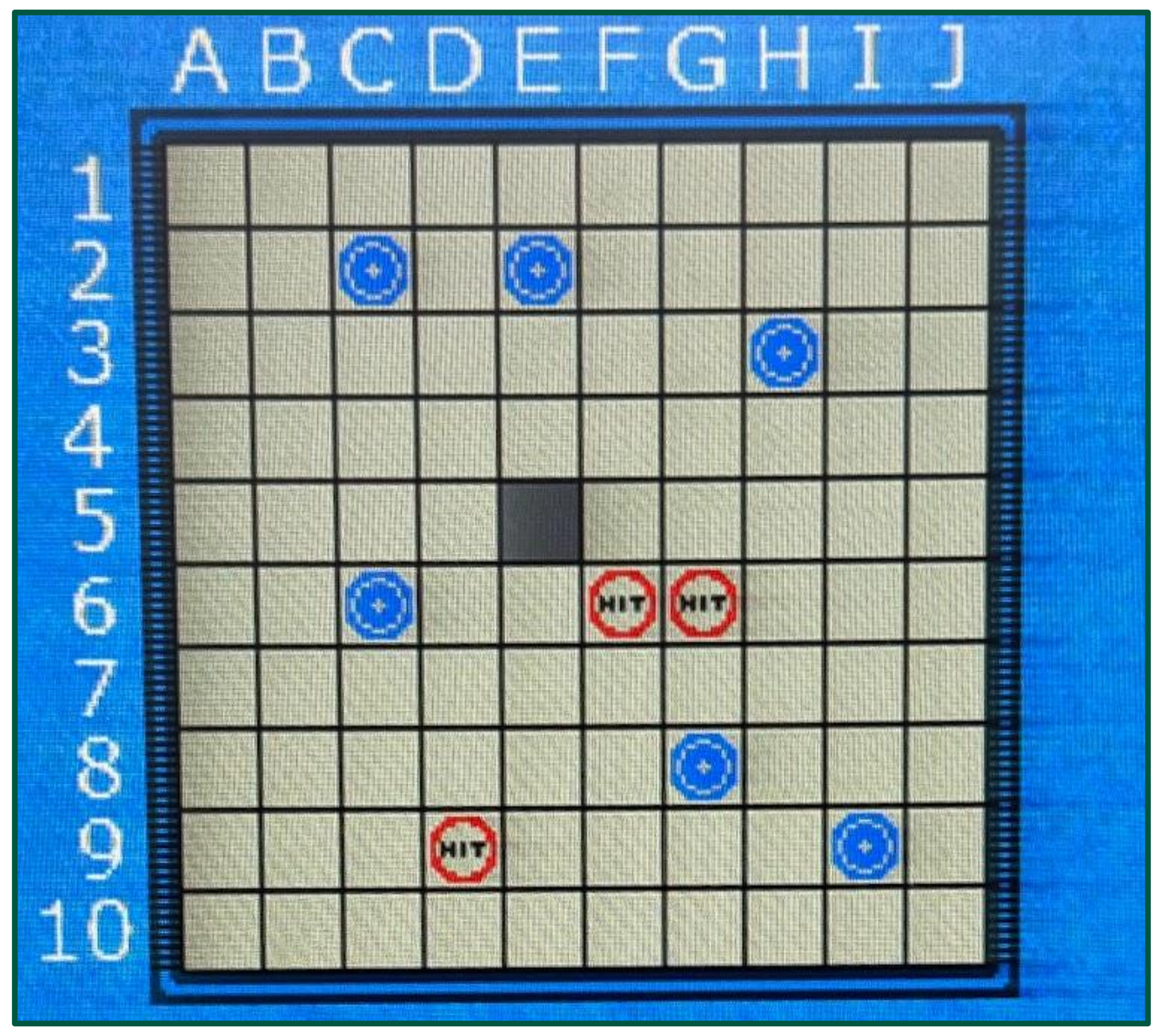
- Uses 8-bit color for game assets and 16-bits for background
- Receives data from the processor to update the screen
- Stores sprite data in ROMs

I/O

How to Play

Press the arrow keys to choose where to hit and press enter on your turn to fire. Make sure to follow your First Officer's aiming suggestions marked in **green**. To win sink all the enemy ships before your own are sunk.

Good Luck Captain!



Software

- Wrote **custom assembler** script in Python to assemble C code (compiled with RISC-V toolchain for GCC with our custom instructions added) into machine code
- Implemented a **Monte Carlo prediction algorithm** to determine suggested move and leveraged accelerator to improve performance