SwarmPursuit: Hardware-Accelerated Multi-Agent Pursuit-Evasion on FPGA

Yen Chung Lo (yl5972), Junhao Qu (jq2434), Kevin Liu (kl3755), Boxiong Li(bl3155)

Spring 2026

The core of this project is a hardware-accelerated multi-agent pursuit-evasion "game" on the DE1-SoC FPGA. Two AI-controlled pursuers attempt to capture an evader in a grid world with obstacles, displayed in real time on VGA. All agents make decisions and move simultaneously in true hardware parallelism. The system offers two play options: a fully autonomous mode where all three agents are AI-controlled, and a human-vs-AI mode where the player controls the evader via a USB game controller.

The FPGA justification is straightforward: the two pursuers must coordinate their moves by jointly evaluating all candidate move-pair combinations (5×5 = 25) in parallel every decision step. Each evaluation involves distance computation, flanking-angle calculation, and escape-corridor counting. Running 25 evaluations simultaneously in hardware enables real-time coordinated planning that sequential software cannot match at the same decision rate. This mirrors the real-world problem of multi-robot coordination, where FPGAs and custom hardware are increasingly used for low-latency parallel decision-making in warehouse robotics, autonomous vehicle negotiation, and drone swarms.

Play Options

Option A — AI vs. AI: All three agents are hardware-controlled. The two pursuers run the coordinated pursuit algorithm; the evader runs a complementary escape algorithm that maximizes distance from pursuers while seeking open corridors. The user watches the chase unfold, and can reshape the obstacle layout in real time to observe how the agents adapt.

Option B — Human vs. AI: The player controls the evader with the game controller, attempting to survive as long as possible against the two AI pursuers. The AI coordination quality becomes directly apparent as the pursuers attempt flanking maneuvers to corner the human player.

Hardware/Software Split

Hardware - We will handle the grid world storage, agent state registers, movement engine with collision detection, the parallel pursuit evaluation engine, the AI evader logic, and VGA tile-based rendering of the grid and agents.

Software - We will handle USB game controller reading, forwarding evader commands to hardware, obstacle/map editing, mode switching, HUD overlay rendering and the Linux device driver for the Avalon bus interface.

Pursuit Algorithm

At each decision step, the hardware evaluates all 25 joint move combinations for the two pursuers. For each combination, it computes a score: $\text{Score} = -(d1 + d2) + \lambda \cdot \theta\_sep - \mu \cdot escape\_corridors$, where d1, d2 are Manhattan distances to the evader, θ_sep is the angular separation between pursuers, and escape_corridors counts open directions available to the evader. The combination with the highest score is selected. All 25 evaluations are purely combinational and complete in one cycle. The weights λ and μ will be tuned during the design phase, starting from a desktop SDL2 prototype.

Major Tasks

- Design decisions for resolution, Avalon register layout, evaluation weights, and VGA tile mapping
- Desktop SDL2 prototype to validate the pursuit/evasion algorithms before hardware implementation
- Core hardware: grid memory, agent registers, movement engine, collision detection, VGA display
- Pursuit AI module: 25 parallel move-pair evaluators with score comparison and selection
- Evader AI module: escape-maximizing evaluation for fully autonomous mode
- Linux device driver and Avalon bus interface
- USB game controller integration via libusb for human evader control and obstacle editing
- Verilator testbench for movement engine, pursuit evaluators, and collision logic