Dogfight AIsteroids

Final Project Proposal

Arye Mindell

CMSI 5998: AI Game Development

Project Plan Category: Research

Detailed Project Description

Overview. *Dogfight AIsteroids* extends the core movement and asteroid-physics loop of Atari's 1979 *Asteroids* into a modern space-dogfighting prototype featuring intelligent, adaptive opponents. Player controls, rigid-body asteroid dynamics, and player navigation are already implemented. This proposal focuses on transforming the sandbox into a research test-bed for adversarial AI by (i) enriching combat mechanics with health and new weapons and (ii) developing both rule-based and reinforcement-learning (RL) enemy pilots.

Scope Alignment. The project satisfies the *Research* category by investigating two quantitative questions: (1) Can deep-RL produce a convincing AI fighter in a continuous, physics-heavy arena? (2) Does behavioral cloning (BC) from baseline agents or human traces accelerate RL convergence and improve final performance? Deliverables include empirical comparisons against a deterministic "Dummy" AI baseline, player feedback from play-tests, and reusable Unity assets.

Gameplay Mock-ups

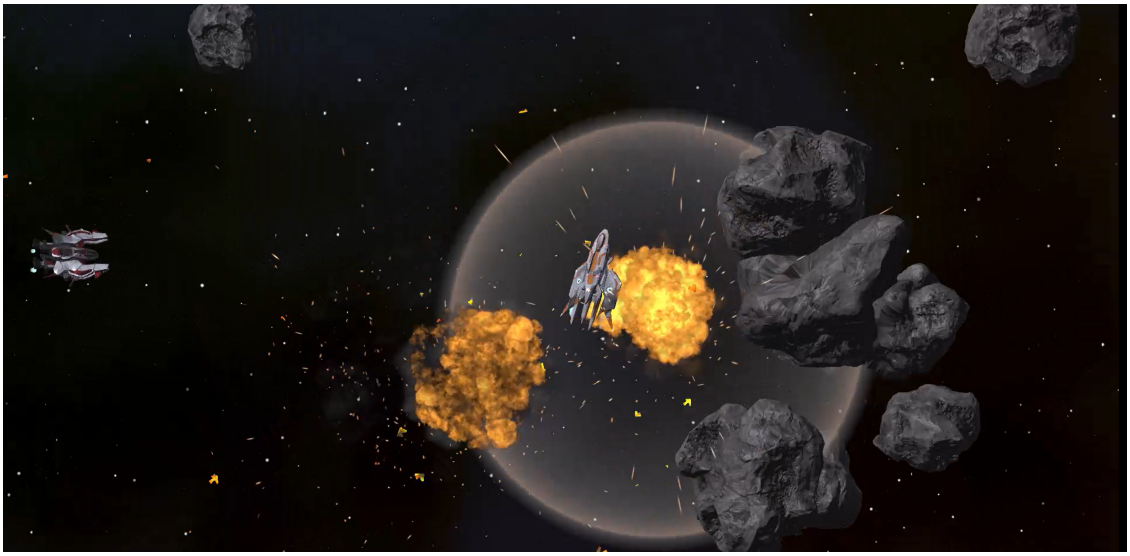

Figure 1: Player (right) evading RL drone (left) amid debris.

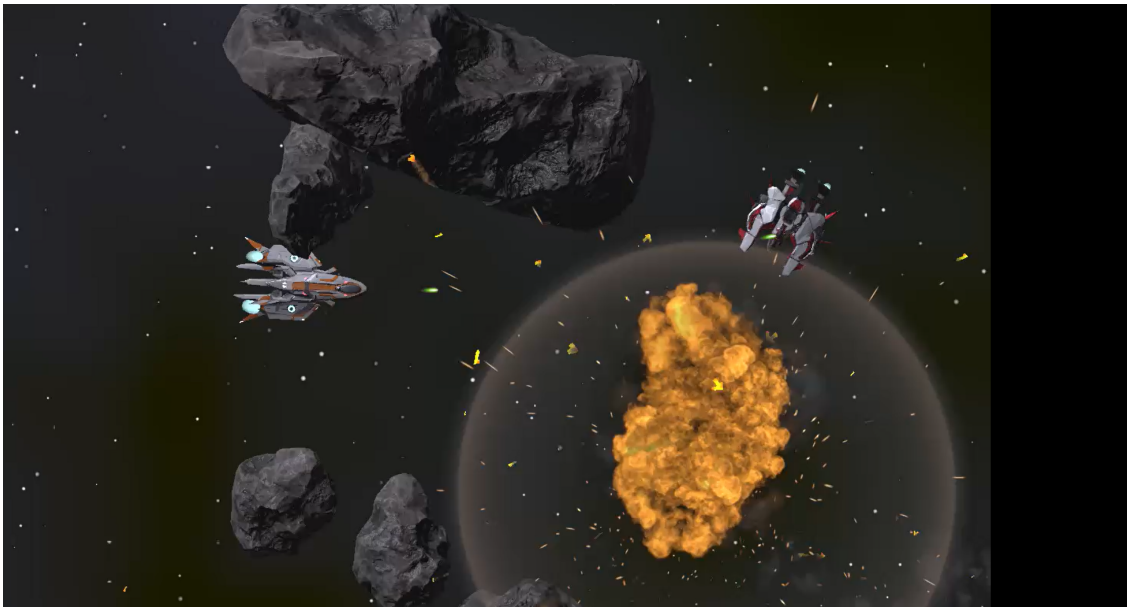

Figure 2: Player (left) attempting to lock on to RL agent (right)

Demo Video

Watch the full gameplay demo

Big-Ticket Features

F1: **New Weapon Types** – Implement new weapon types to promote varied gameplay (e.g. a missile for big asteroids, concussion charge to deter chasing)

F2: **Health & Damage System** – Add hull integrity, shield depletion, death states, and UI indicators for both player and AI ships.

F3: **'Dummy' Enemy system** – Develop a simple behavior-graph based agent to act as a baseline with which to compare RL Enemies, and to test BC

F4: **RL Enemy Pilot** – Train an agent with Proximal Policy Optimization (PPO) to pursue, evade, and engage the player while avoiding asteroids. Experiment with traditional PPO and with added BC pre-training

Play-Testing

P1: **Basic Feature Validation**
All implemented features from F1-F3 are validated using PlayMode tests in Unity's NUnit test framework.

P2: **Combat Balance Test**
Five players engage the Dummy AI across difficulty tiers to validate damage values, TTK, and clarity of health UI. Metrics: mean survival time, subjective fairness (1–5).

P3: **AI Believability Study**
Same players face RL and Dummy AIs in random order; record win-rate, engagement time, and post-match questionnaire on perceived intelligence and fun. Statistical tests compare RL vs. baseline.

Research Questions

R1: Does PPO with BC pre-training achieve a significantly higher player kill-rate than a rule-based Dummy within 1M environment steps?

R2: Does BC from *human* traces outperform BC from *Dummy* traces in terms of convergence speed (episodes to reach $> 60\%$ win-rate)?

Methodology

Phase 1 – Core Combat Foundations

- Design & unit-test the health/shield pipeline.
- Integrate the asteroid-blaster with VFX, SFX, and damage falloff.
- Build a deterministic *Dummy AI* via NavMesh and Behavior Graphs (patrol, pursue, evade, fire).

Phase 2 – Learning Pipeline

- Collect datasets: Dummy–Dummy self-play; Human–Dummy duels.
- Behaviorally clone fight logs to pre-train PPO.
- Conduct curriculum training in a bounded arena, gradually relaxing constraints (e.g. asteroid density, projectile speed). Repeat with and without BC pre-training.

Phase 3 – Evaluation & Iteration

- Measure win-rate, time-to-kill, and *believability* (Likert survey) of RL and Dummy opponents vs. Players and each other.
- Tune reward shaping and retrain as needed.