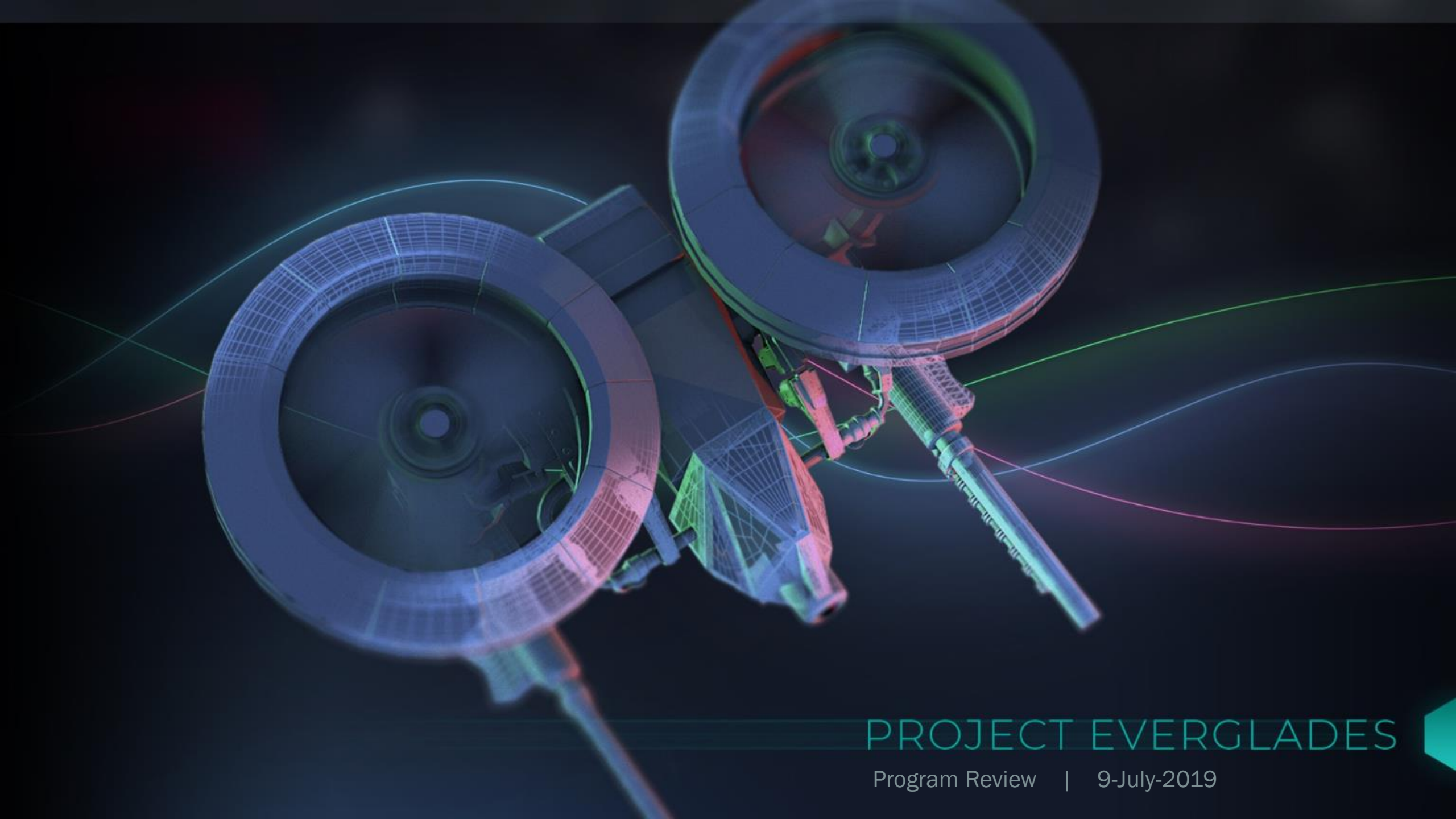


PROJECT EVERGLADES

Program Review | 9-July-2019

Design Challenge

The goal of Project Everglades is to create a competitive AI platform that engages viewers, provides an intuitive interface for AI creators, and reinforces elements of the Lockheed Martin brand, including innovation and cooperation.

- Initial 6 week development period
- $250,000 budget

Our Vision

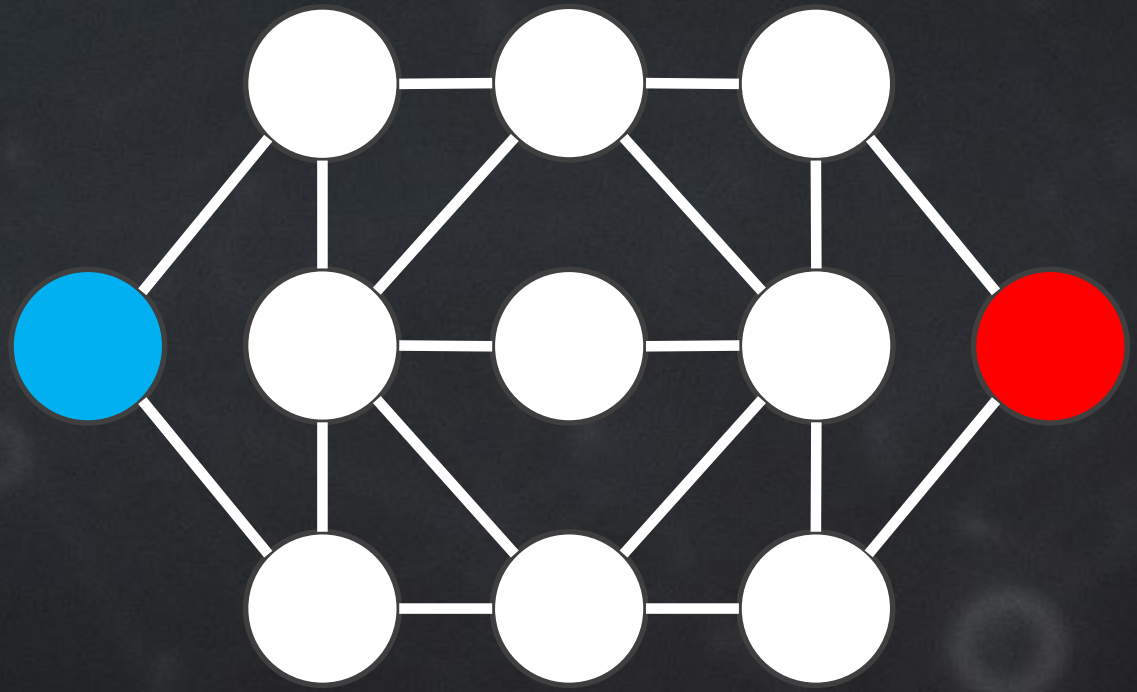
To produce an AI/ML platform with enough depth and engaging content to attract gamers, AI developers and academic users, it is essential to develop a product that is:

- Simple – easy to get started.
- Strategic – each “move” must be significant.
- Expandable – complexity can be added over time and the framework can expand to meet growing user sophistication.

Game Design

A brief overview of the game

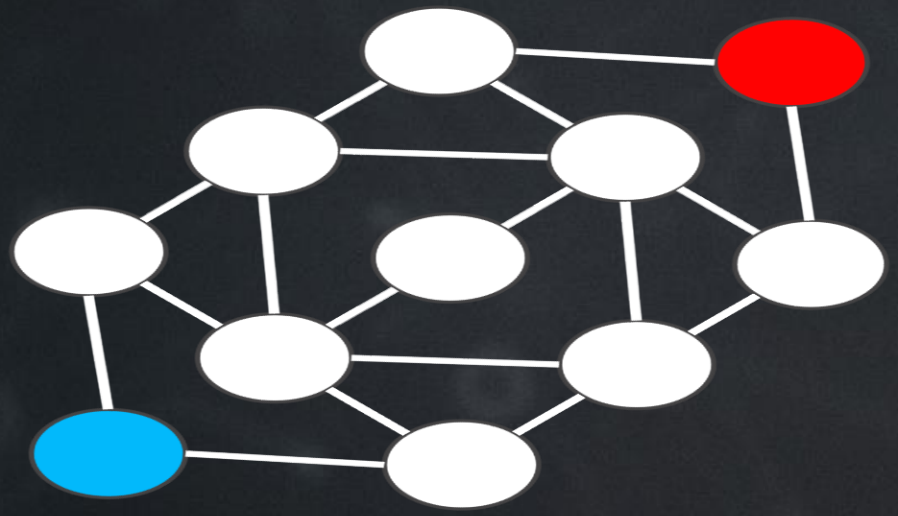
Game Board



Blue Team’s base is on the left side of map. Red Team’s is on the right.

The objective is to capture the opposing team’s base or eliminate them completely.

Visual Presentation



To enhance visual appeal and capture viewer interest, an Everglades playback session can be rendered out in a high-fidelity 3d environment complete with intense combat and “spectacular deaths.”

However, the game remains a turn-based strategic game “under the hood”.

Drone Class Stats

Drone Class	Movement	Armor	Damage	Capture Speed
 STRIKER	+50%	-50%	+50%	*base*
 CONTROLLER	*base*	*base*	*base*	+50%
 TANK	*base*	+50%	*base*	*base*

Player Forces

Each player begins with a pool of drones that can be composed into forces to serve the strategic need. When two opposing forces enter a territory, combat occurs and the winner claims or retains the territory.

There are three different classes of drones that comprise a player's force:

- The Striker is an offensive unit that moves 50% faster than other units and does 50% more damage, but has 50% less armor.
- The Controller is a specialist unit that captures territory 50% faster but has no other advantage or disadvantage.
- The Tank is a defensive unit that has 50% more armor.

Drone class bonuses can be easily modified by editing the game rules file using a text editor.

Capturing Territory

Between each of the starting player bases are nine neutral territories that can be captured and occupied through combat.

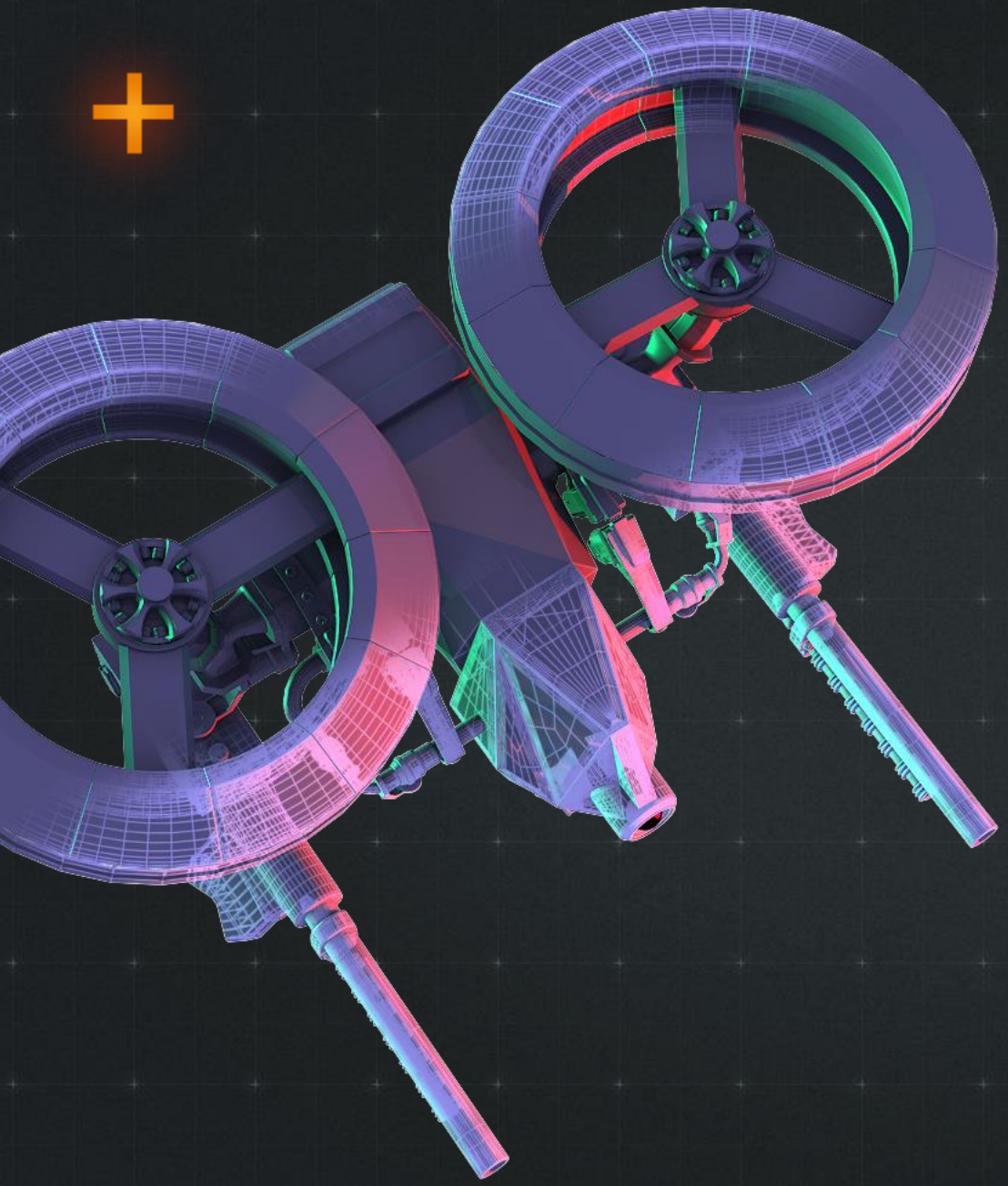
- Some territories provide player benefits such as improved defense or scouting range if held.
- Because of FOG OF WAR, players cannot see what is happening in a territory unless they have captured it or have active units in that area.
- It takes a small amount of time to capture a neutral territory, and a longer amount of time to recapture a territory from the enemy, so defending territory is important.

Territory Bonuses

The map has special territories that provide additional tactical value to whichever player possesses them:

- Fortresses grant an additional defense bonus for any allied units in the area during combat.
- Watchtowers extend a player's scouting range to adjacent territories even if those territories are under enemy control.

Territory bonuses and their placement can be easily modified by editing the map data files using a text editor.



Future Development

The sky is the limit for future Everglades features and development. Some key elements we believe would enhance the game's depth and player engagement include:

- Additional unit classes and effects
- Different environments, including procedurally generated terrain
- Additional game modes
- Environmental effects (wind, rain, underwater etc.)
- Multiple teams – both cooperative and competitive
- "Spectacular deaths" and visual effects

Spectacular Deaths

One of the key features that we would like to explore in future phases of the project is adding addition details to each conflict to make them more interesting than they need to be for the AI, but ideal for an human audience. Details such as :

- Slow motion / Time dilation.
- FX Hangtime.
- Explosion build-up & chain explosions
- Environmental damage & scarring

In addition to effects and particles, elements that are affected by physics also add to impact of explosion, missile strike or hail of bullets. The more kinetic reactions are often more engaging when chained together or creating a compound reaction:

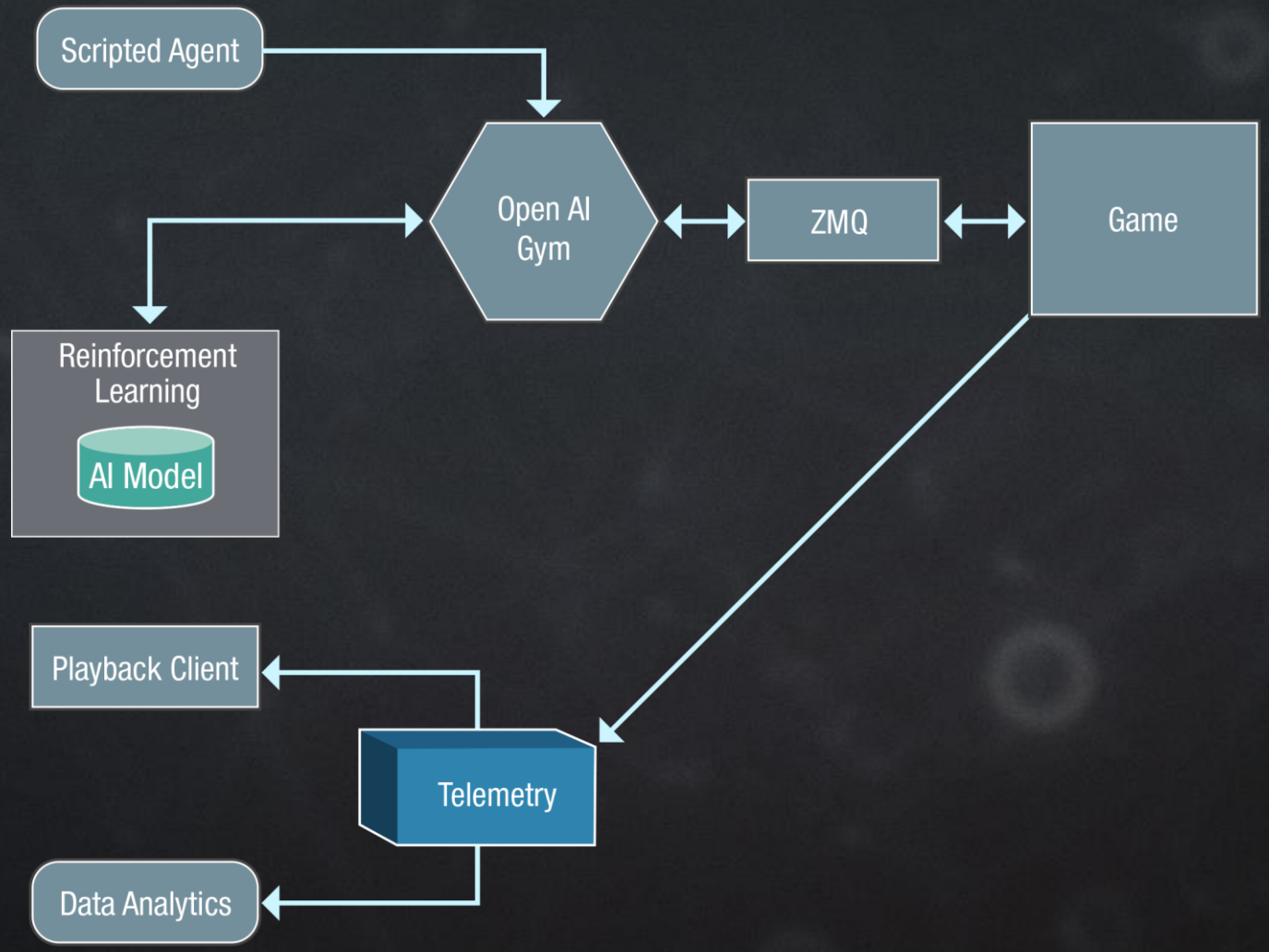
- Drone parts breaking and scattering
- Accurate physics models on drone parts interacting with each other and the world.
- Compound explosions and chain reactions

Spectacular Deaths

AI: Development & Training

How the Everglades ecosystem works for developers

Developing AI for Project Everglades



How to: Train an AI

Training an AI takes time. This is typically accomplished by scripting a process to play the game repeatedly. This generates telemetry data that a reinforcement learning AI can utilize to improve. It can also be viewed in the Playback Client, and/or analyzed with data science tools.

First a participant would create a python script:

- The script would detail a state machine allowing the user to understand the environment.
- The script would be placed in to a docker file and loaded in the OpenAI Gym.
- This script is the backbone for the behaviors that are refined during Reinforcement Learning.

Through reinforcement learning, the AI behavior is improved and refined:

- Each iteration produces telemetry and a “data brain” model.
- Over hundreds of playthroughs the data brain improves – either through updates to the core python script, or based on reinforcement updates.
- To compete or be judged, the participant would submit both their python script and their highest scoring data brain.

How to: Compete with an AI

When the participant feels their AI is ready for competition, they can submit their work to a Proctor and the Proctor will decide how to evaluate the work:

- Each participant submits a docker file containing their AI agents
- The Proctor performs some checks to ensure everything is correct and legal
- The Proctor runs one or more Everglades simulations, pitting AI agents against one another and generating a result.

Note that the Proctor is a person or team of people, however this process could be automated, with a web-based server that automatically processes entries to create and adjudicate tournaments, leagues and other competition formats.

Technology

A peek under the hood

Phase One Deliverables

For the first Project Everglades phase deliverable, we developed:

- A headless server that could run multiple iterations of the game, generating text-based telemetry that could be used for both visual playback and reinforcement learning.
- (5) archetype AI scripts that demonstrate the various features of the game, which also serve as an example of how the APIs work.
- A high-fidelity, visually appealing playback engine (based on the industry-leading Unreal 4 Engine) to demonstrate the reading, interpolation and display of the raw telemetry data.

Phase One Server Features

The Everglades server is designed to run headless (for large scale ML training) or as part of an easy-to-use front-end (for novice AI developers) and includes the following features:

- A Docker-based image that runs on Windows or Linux operating systems
- A ZeroMQ text-based messaging queue with cross-language support that allows development of AI in any language that the server would process and run the game
- Efficient and low resource utilization

Phase One Client Features

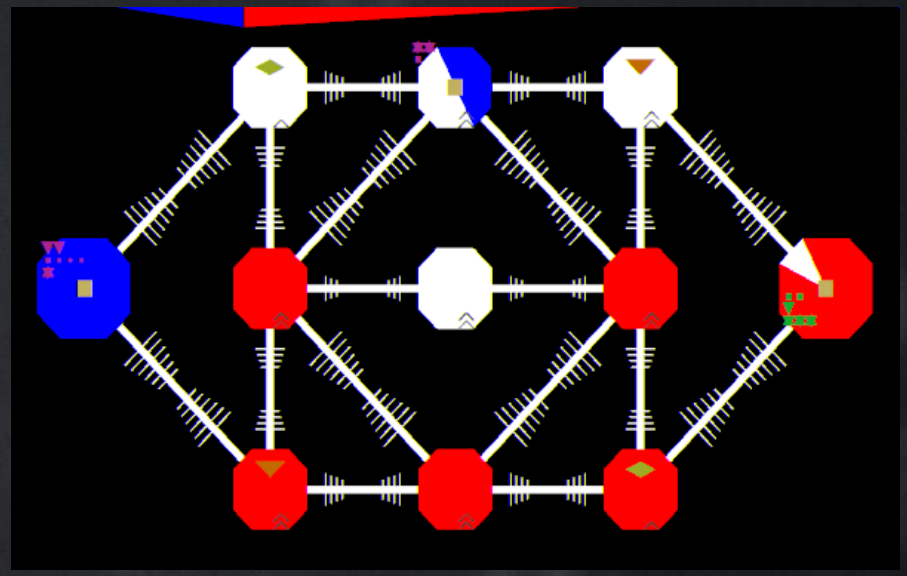
The Phase One client utilizes Unreal Engine 4 to deliver a high-fidelity “E-Sports” style playback and presentation suitable for streaming on popular services such as Twitch.

- Playback includes a “smart” cinematic camera that follows action around the map.
- Detailed Heads Up Display (HUD) in the style of popular Esports so the audience understands what’s happening.
- Telemetry playback takes place on sample “desert mesa” for visual interest; however, nearly any visual style or setting can be implemented / utilized.

Phase Two Plans – August

Everglades Phase Two, based on Lockheed review and feedback, include the following:

- General Server improvements – further optimizations and improved stochasticity
- Example reinforcement learning AI based on raw text telemetry.
- Example machine vision-based learning AI using “off the shelf” tools.
- OpenAI gym environment.
- OpenGL simplified game render for machine vision based reinforcement learning.



OpenGL render (AI view)

Phase Three and Beyond

Lockheed Martin and Cubic | Austin have had some initial discussion about a potential third phase of Project Everglades might look like. Suggested features for Phase Three would include:

- Improve reinforcement learning with replays.
- Adjustable sensor properties.
- Parallelization – playing multiple games at once.
- Audio FX and music

The longer term roadmap could include additional systems and features such as:

- Supporting multiple game clients (Unity etc.)
- Options for Real-Time Rendering
- Human Player capability
- Integrated streaming and video recording tools (Twitch, YouTube etc.)
- Web-based Proctor / Agent sharing
- Online tournaments and leagues

Thank you