Machine Learning Applications for AI in Game Development

Evan Fridge & Marc Aubanel

LSU Center for Computation & Technology
Introduction

• Unreal Engine
• Neural Networks
• Data Sets
1. Create a simple C++ neural network library outside of Unreal Engine for testing purposes.

2. Find usable data sets and build predictive, memory-efficient neural network models.

3. Integrate the feed-forward component into Unreal Engine and import satisfactory models.

4. Build a data pipeline within Unreal Engine for the networks to access needed parameters.

5. Add in-game behavior responding to NN output.
Data Sets

- Forest Fire
- Pet Cats
- NCSU Campus Carnivores
- Mangabey Monkeys in Uganda
- Behavior Profiles
Methods

- **Object Fields**
  - Controlling the flow of data
- **Basic Protocol**
  - Encapsulating the NN process
- **Behavior Generation**
  - Enabling AI expression
Neural Network Systems

- **Movement Predictors**
  - Considers time, location, and temperature

- **Fire Spread**
  - Considers weather & ground conditions

- **Behavior Predictor**
  - Selects an abstract behavior based on time and location
  - Eight in total
Conclusions & Future Work

- Offers a context-sensitive alternative to random number generation
- Content handles new data systems organically
- Abstract nature of the basic protocol allows for complex layers of AI
Thanks to David Miller for help with implementing our initial neural networks.

This work was supported by the National Science Foundation (NSF) award #ACI-1560410.

A special thanks to Epic Games for creating Unreal Engine.

In-Game Assets provided by Epic Games, Alexander Dracott, and Protofactor Inc.

References & Acknowledgements


