

# Explorando Combos em Jogos de Luta Através de Reinforcement Learning

**Rodrigo Peixe Oliveira**

Ciência da Computação  
UNIFESP  
São José dos Campos, Brasil  
rodrigo.peixe@unifesp.br

**Rennam Victor Cabral de Faria**

Ciência da Computação  
UNIFESP  
São José dos Campos, Brasil  
rennam.faria@unifesp.br

# Overview

## 1. Introdução

- Jogos de Luta
- Combos
- Objetivo

## 2. Metodologia

- Ambiente
- Domínio
- Experimentos
- Pré-Processamento
- Processamento
- Pós-Processamento

## 3. Resultados

## 4. Conclusão

# Introdução

# Jogos de Luta

- Gênero de jogo competitivo
- Objetivo: derrotar oponente
- Costumam ser complexos



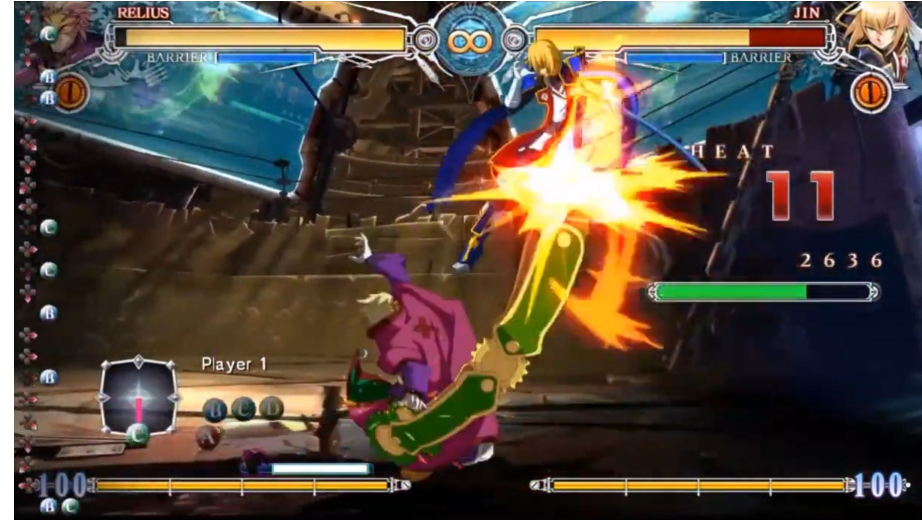
# Jogos de Luta

- Torneios com prêmios milionários que reúnem centenas de milhares de pessoas por ano
- Jogos buscam ser justos e divertidos tanto para jogadores quanto para espectadores



# Combos

- Sequências de ataques em que o oponente não pode responder (ordinariamente)
- Recompensa jogadores por encontrarem aberturas e punirem seus oponentes
- Essenciais para o balanceamento do jogo, porém difíceis de testar



# Objetivo

- Utilizar aprendizado por reforço para tentar encontrar o combo mais longo possível em um jogo de luta
- Comparar diferentes algoritmos quanto a eficiência e eficácia
  - A2C
  - PPO
  - GA

# Trabalhos Relacionados

- G. Zuin, Y. Macedo, L. Chaimowicz e G. Pappa, “**Discovering combos in fighting games with evolutionary algorithms**”, 2016
  - Inspiração para este trabalho
  - Utilizou algoritmos genéticos
  - Jogo criado para propósitos da pesquisa
- G. Zuin e Y. Macedo, “**Attempting to discover infinite combos in fighting games using Hidden Markov Models**”, 2015
  - Trabalho anterior ao de cima
  - Utilizou Modelos Ocultos de Markov
- Outras pesquisas em jogos de luta costumam focar em utilizar IA para tentar imitar o jogador ou vencer o jogo



# **Metodologia**

# Ambiente

- Jogo utilizado: Street Fighter Alpha 3 (GBA)
  - Presença de combos infinitos
  - Emuladores disponíveis
  - Leve de rodar
- Biblioteca: Stable Retro, Stable Baselines3
  - Integram o código com o emulador e permitem acessar a memória do jogo



# Domínio

- Iniciar com o oponente contra a parede
  - Mais rotas de combo
- Iniciar no modo Variable Combo (ou Custom Combo)
- Impedir que o botão de provocação seja selecionado
- Ryu VS Ken: personagens relativamente genéricos



# Experimentos

- Objetivo: maximizar a duração do combo (medido pelo número de *frames* em que o oponente não consegue agir)
- Reinforcement Learning
  - Recompensas: 1 por frame de combo
  - Condição de fim: combo termina (opponente pode agir novamente) ou limite de tempo (600 *frames*)
  - A cada certo número de passos, rodar 10 episódios de validação de forma estocástica
- Genetic Algorithm (G. Zuin *et al*)
  - Implementação completa no artigo (2016)
  - Fitness dada pela equação:

$$F = \sum_{i=1}^n f(i), \quad f(i) = \begin{cases} \frac{\text{ComboSize}}{(1+\text{LeftZeroes})^2} & \text{if right of longest} \\ \frac{\text{ComboSize}}{(1+\text{RightZeroes})^2} & \text{if left of longest} \\ \text{ComboSize} & \text{if the longest} \end{cases}$$

# Pré-Processamento

- Transformações na imagem para acelerar processamento
  - Redução do tamanho
  - Escala de cinza
- Memória de ações
  - Ajuda a aprender sequências de ações
- Frame Stacking
  - Ajuda a perceber movimento
- Limite de tempo
- Filtro de ações
  - Impedir botões inúteis

# Processamento

- A2C (Advanced Actor Critic)
  - Utiliza diversos trabalhadores para processar e eliminar a necessidade de um replay de buffer
- PPO (Proximal Policy Optimization)
  - Descida em Gradiente
  - Trabalhadores Múltiplos + Região de confiança
- GA (Algoritmo Genético)
  - Evolução biológica
  - Implementação completa no trabalho de Zuin *et al.*

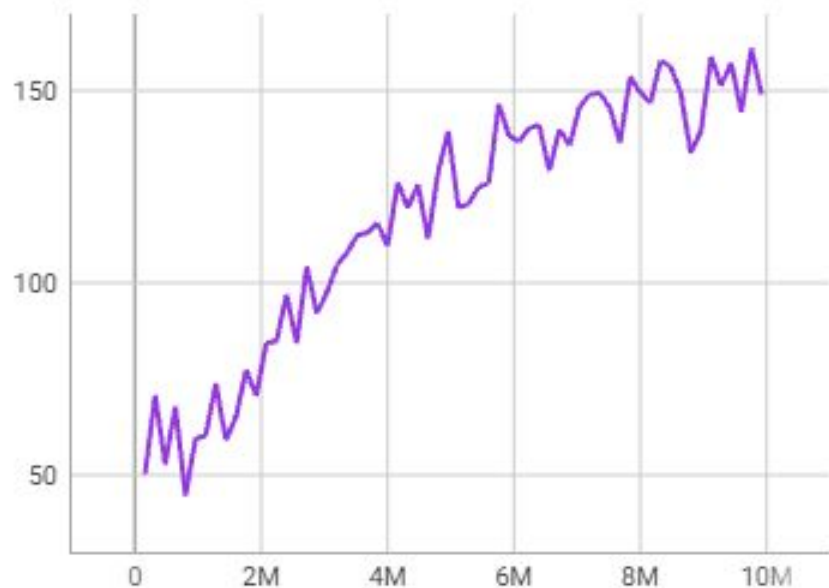
# Pós-Processamento

- Salvar resultado do combo mais longo
- Biblioteca: TensorBoard
- Analise
  - Média de recompensa por Períodos (Mean Reward)
  - Comportamentos da Função de Perda (Loss Function)

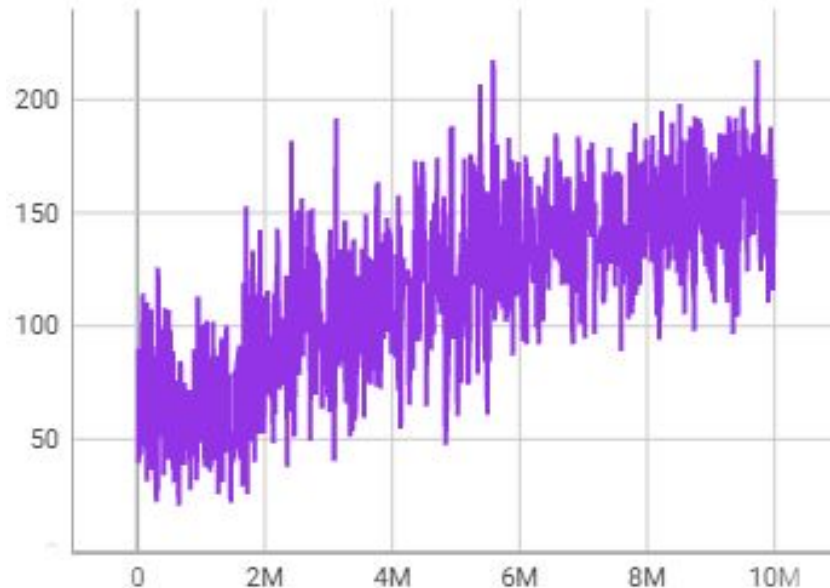
# Resultados



# A2C

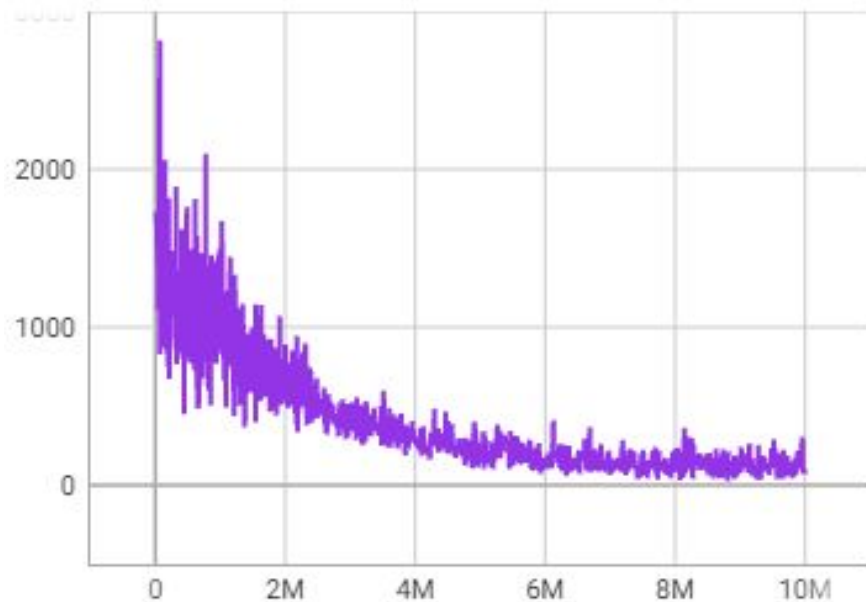


Recompensa média durante treino

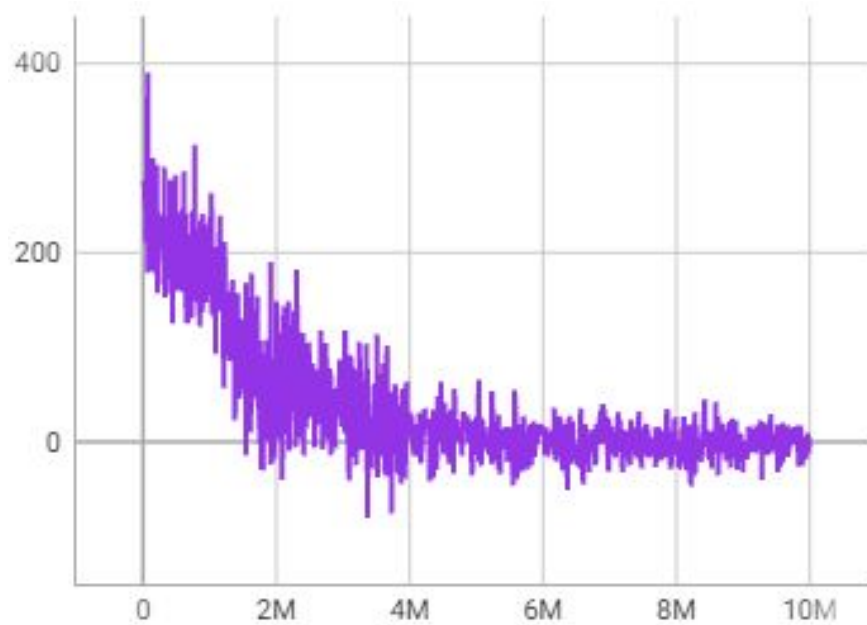


Recompensa média durante validação

# A2C

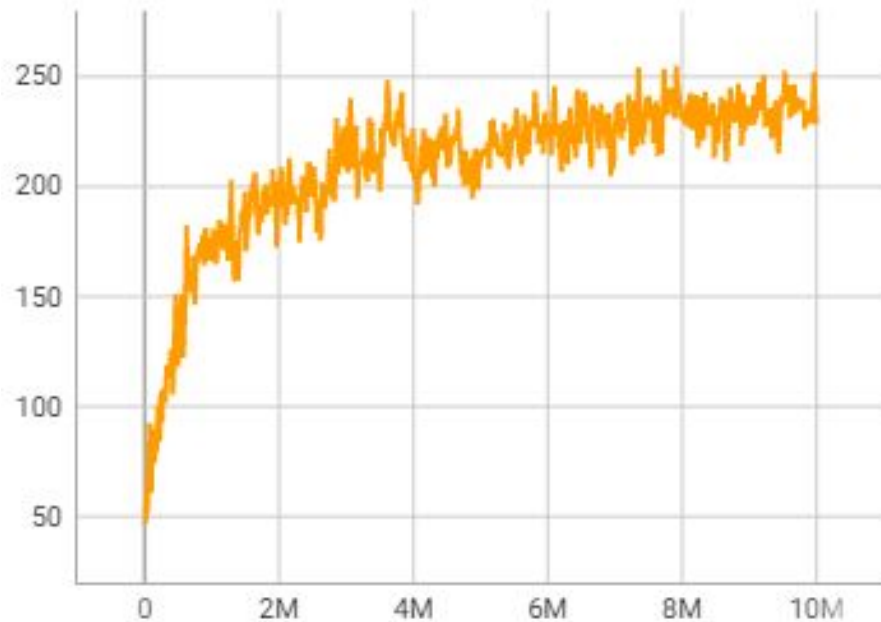


Value Loss

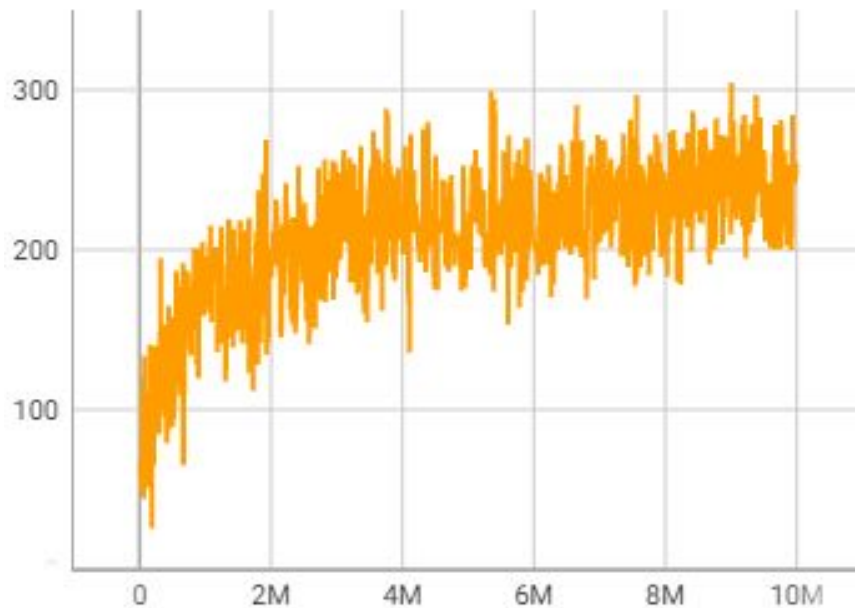


Policy Loss

# PPO

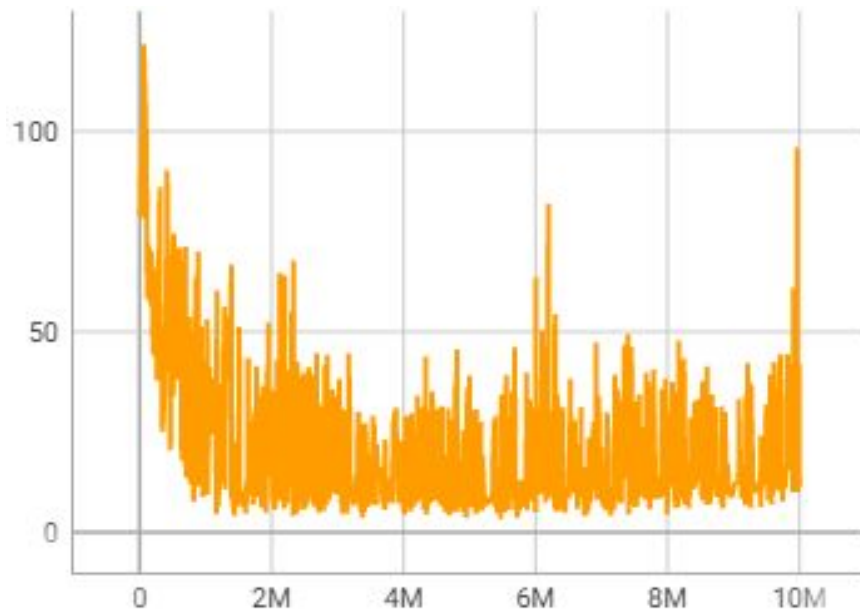


Recompensa média durante treino

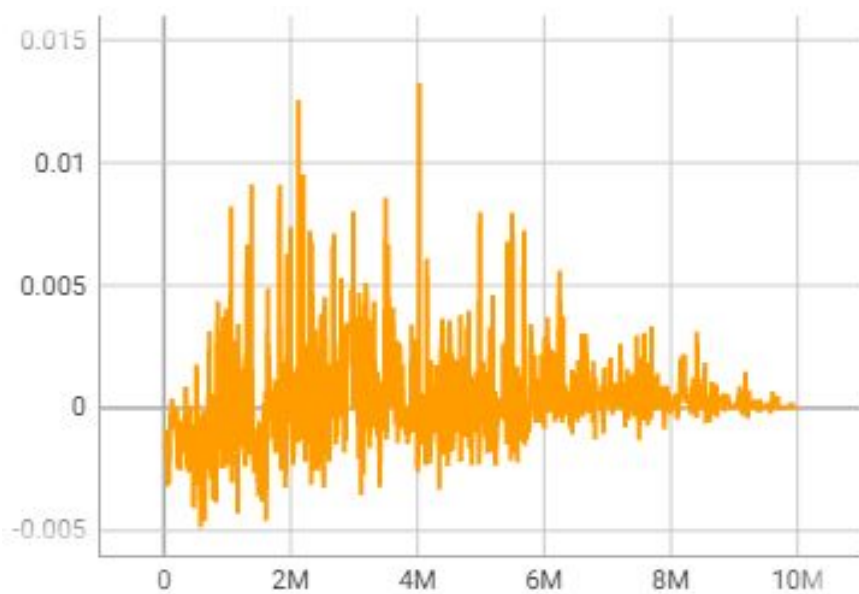


Recompensa média durante validação

# PPO

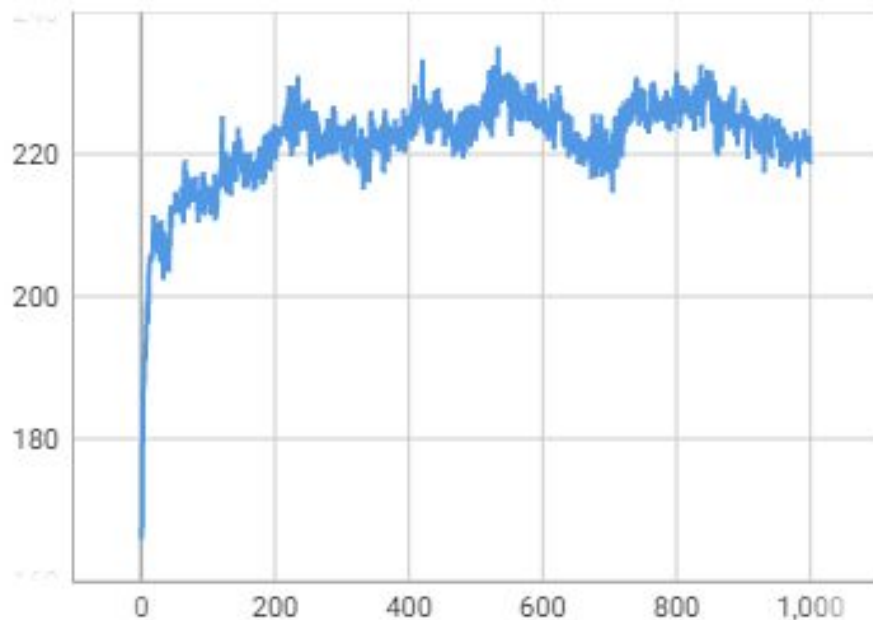


Value Loss

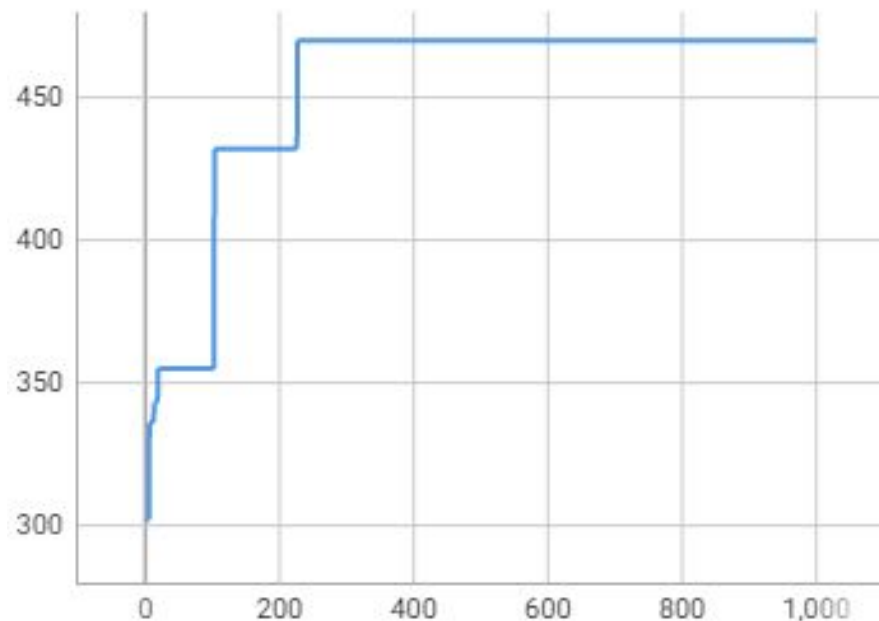


Policy Loss

# GA

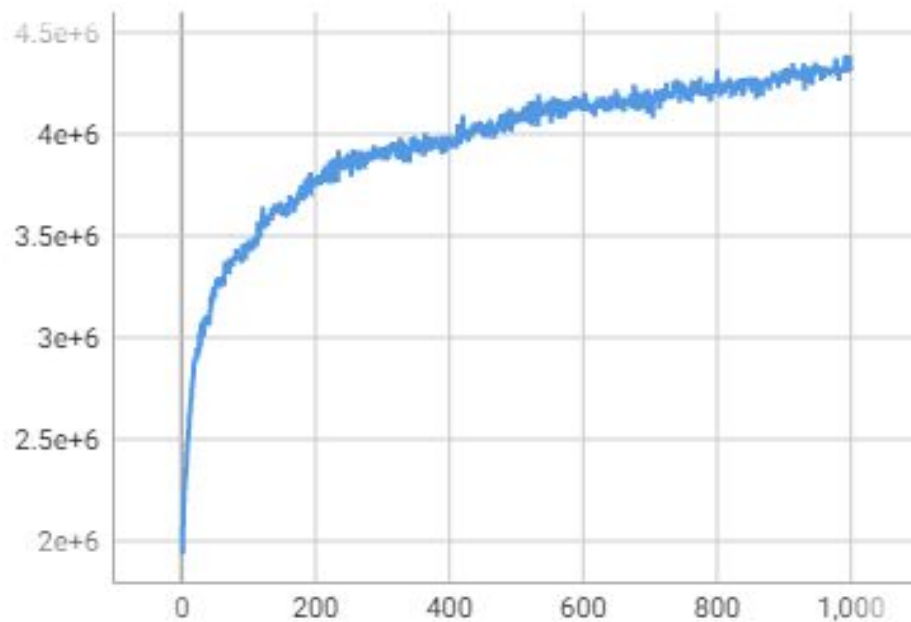


Duração de combo média por geração

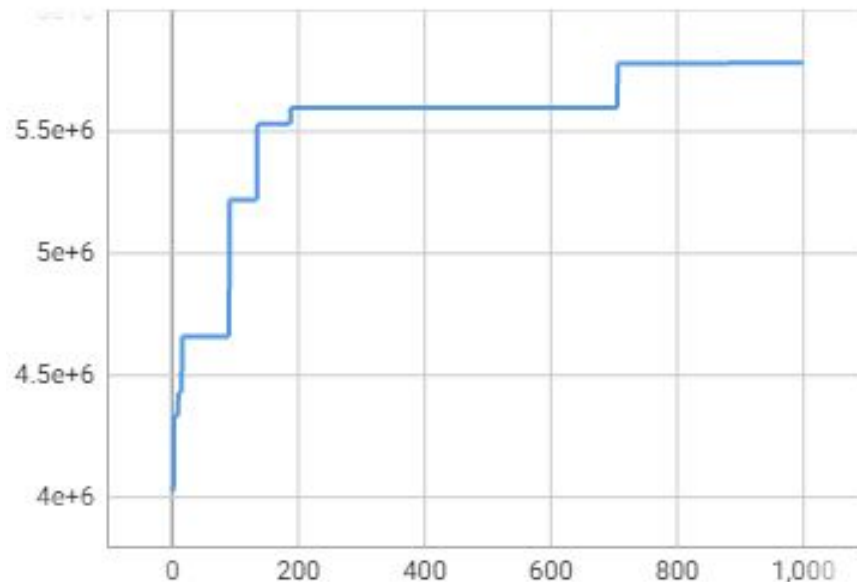


Duração de combo máxima por geração

# GA



Fitness média por geração



Fitness máxima por geração

# Resultados e Discussão

	Duração de combo máximo (frames)		
	<i>A2C</i>	<i>PPO</i>	<i>GA</i>
Média	147.35	249.22	218.56
Maior	284.00	511.00	470.00

Número de passos ( <i>timesteps</i> )	Tempo de Execução (h)	
	<i>A2C</i>	<i>PPO</i>
2M	01:00	01:36
4M	02:12	03:24
6M	03:36	05:18
8M	05:00	07:12
10M	06:30	09:12

Geração	Tempo de Execução (h)
200	00:42
400	01:18
600	01:54
800	02:36
1000	03:12

# Melhores Combos



<https://www.youtube.com/watch?v=RttH05aGbQA>



# Conclusão

- Embora não tenham sido encontrados combos infinitos, foram descobertos combos impressionantes
- GA teve uma melhora inicial muito rápida, mas logo se estagnou
  - Possibilidade de usar para uma primeira análise durante balanceamento
- Algoritmo PPO de RL encontrou o combo mais longo, mas apresentou o maior tempo de execução
  - Possibilidade de usar para uma análise mais aprofundada
- Possibilidade de explorar outros algoritmos de RL e otimizar parâmetros

# Perguntas

# Explorando Combos em Jogos de Luta Através de Reinforcement Learning

**Rodrigo Peixe Oliveira**

Ciência da Computação  
UNIFESP  
São José dos Campos, Brasil  
rodrigo.peixe@unifesp.br

**Rennam Victor Cabral de Faria**

Ciência da Computação  
UNIFESP  
São José dos Campos, Brasil  
rennam.faria@unifesp.br