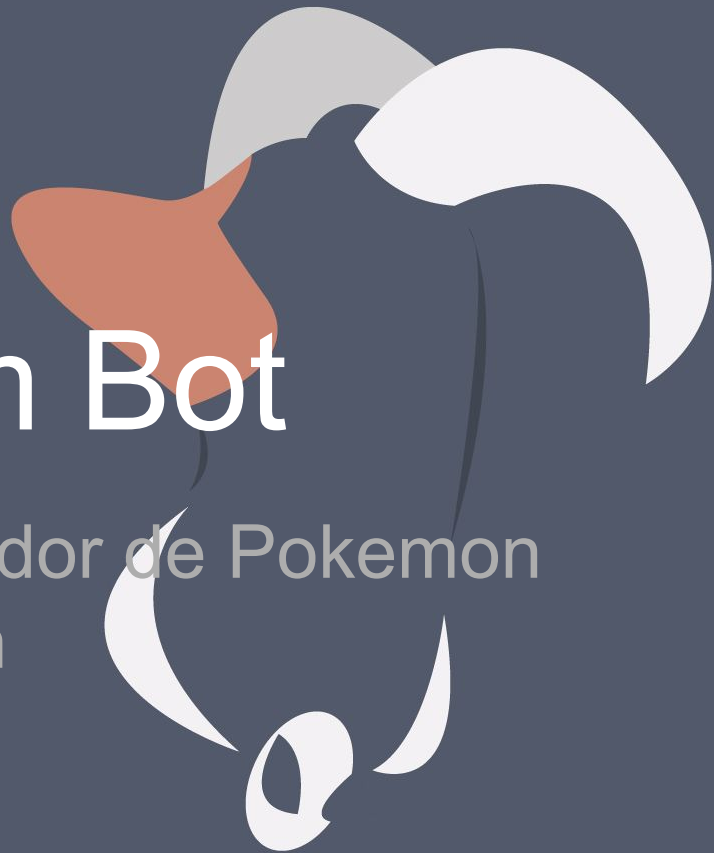


Houndoom Bot

Um agente inteligente jogador de Pokemon
Showdown



Arquitetura

- Abordagem de algoritmo de “Hill Climbing” ou “Greedy”: busca a melhor solução local para o problema;
 - Vantagem: simples implementação, chega rapidamente uma solução;
 - Desvantagem: dificilmente encontra a solução ótima. Tende a se acomodar em máximas locais;
- Heurísticas foram utilizadas para decidir por usar um “MOVE” ou realizar um “SWITCH”;
 - Compara-se o potencial de dano do meu pokemon no pokemon inimigo, com o potencial de dano do inimigo no meu pokemon. Uma pontuação é atribuída cada possível confronto, envolvendo o pokemon ativo do inimigo e todos os meus pokemons(tanto os da reserva quanto o ativo), sendo que a prioridade é realizar um MOVE e não um SWITCH;

Problemas Encontrados

- Realizar muitas trocas diminui drasticamente o desempenho. Mas ao dar prioridade para a ação de MOVE, diminuindo o peso da “ameaça” inimigo no cálculo da pontuação, foi possível chegar a um desempenho muito bom, em comparação a bots de minha autoria que simplesmente atacam com o melhor MOVE e ao perder o pokemon fazem o SWITCH para um pokemon que tenha vantagem contra o pokemon atual do oponente;
- “Arena Trapping” é um problema para o bot. Tanto abilities(Shadow tag e Arena Trap) quando moves(Fire Spin) podem impedir a ação de SWITCH, mas eu não encontrei uma flag que aponte este estado de “preso”, portanto lidei “na mão” com o problema das abilities, já que estes moves que causam “trapping” não são muito usados competitivamente;

Problemas Encontrados

- O agente ainda possui a limitação de não utilizar golpes de status efetivamente. Já que a arquitetura foca em utilizar corretamente os moves ofensivos.

Considerações Finais

- Os scripts “kochance.js” e “damage.js” foram essenciais para montar o bot. E eu somente os encontrei ao “fuçar” pelos diretórios e pelos scripts do leftovers-again, na maioria das vezes enquanto buscava solucionar algum bug;