**I.A – Pokemon Showdown**

Inteligência Artificial para Jogos

Aluno: Gabriel Nunes da Silva

Prof. Msc. João Ricardo Bittencourt

Data de entrega: 14/10/2016

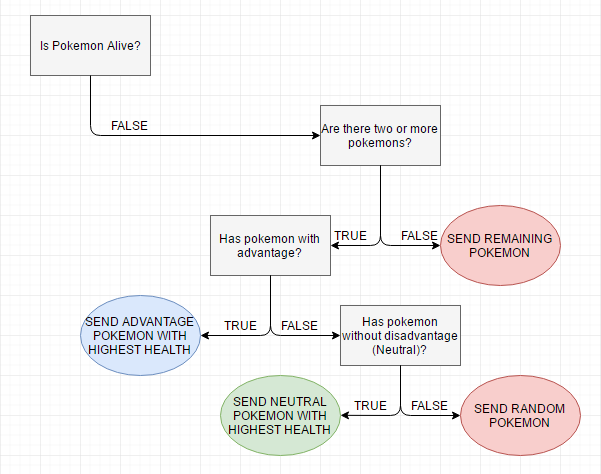
1. **Objetivo**

Ao começar a criação da Inteligência artificial para o jogo Pokemon Showdown, meu objetivo foi criar uma I.A agressiva, tentando causar o máximo de dano enquanto minimiza o dano levado. Para isso, a I.A utiliza principalmente do Typechart para detecta situações de vantagem e desvantagem para o jogador.

1. **Lógica**

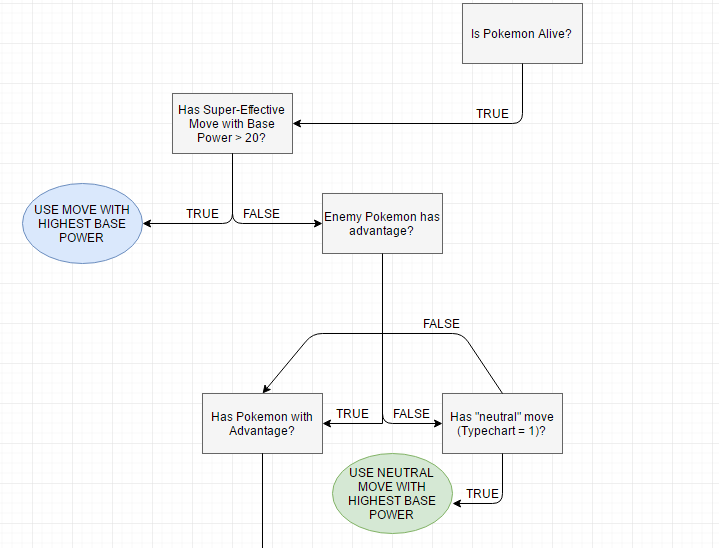
Como joguei pokemon somente casualmente, não possuía grande conhecimento sobre estratégias ou metagame, então criei a I.A com base no meu modo de jogar: utilizar um Pokémon que possua vantagem contra o inimigo, e utilizar o ataque mais forte. Primeiramente, começo detectando se o Pokémon está vivo ou não.

Caso o Pokémon atual esteja morto e a I.A seja forçada a trocar de Pokémon, o agente irá, primeiramente, detectar quantos pokemons possui de reserva. Caso possua apenas um, ele irá jogar este Pokémon, pois está sem opções. Caso possua mais de um pokemon, o agente irá utilizar do Typechart para detectar se algum Pokémon do jogador possui vantagem contra o inimigo. Se possuir, ele irá utilizar o Pokémon com maior HP entre aqueles que possuem vantagem. Se nenhum Pokémon do jogador possuir vantagem contra o inimigo, ele irá escolher, entre os pokémons que não possuírem desvantagem (ou seja, Typechart retorna 1), aquele que possuir maior HP. Caso possua somente pokemons que tenham desvantagem contra o inimigo, ele irá jogar um Pokemon aleatório.



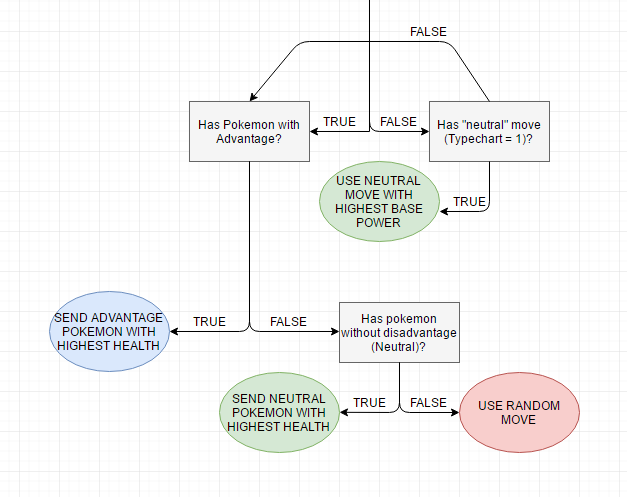
*Figura 01 – Decisões feitas caso o jogador seja forçado a trocar de Pokémon*

Caso o Pokémon esteja vivo, ele irá primeiramente checar se possui algum ataque que seja super-efetivo contra o Pokémon inimigo. Se possuir um ataque, ele irá utilizar o ataque que possuir mais Base Power. Se nenhum ataque for super-efetivo, o agente utiliza o Typechart para testar se o inimigo os tipos do Pokémon inimigo podem ser super-efetivos contra o Pokémon atual do agente. Se não estiver em desvantagem, o agente irá testar seus movimentos, escolhendo os movimentos que causem dano normal ao inimigo, sem sofrer desvantagem, e escolhendo o mais poderoso (com maior Base Power) para utilizar. Caso os tipos do inimigo possuam vantagem sobre o Pokémon do agente, ou se todos os movimentos do jogador forem fracos contra o inimigo, ele irá procurar em sua reserva se possui pokemons com vantagem.



*Figura 02 – Decisões feitas caso o Pokémon do agente esteja vivo, cuja primeira decisão é detectar se possui um movimento super-efetivo*

O Agente detecta quais pokemons tem vantagem contra o inimigo, ele irá escolher aquele que possui maior HP para jogar no campo. Caso nenhum Pokémon seja efeito, irá escolher um Pokémon com HP alto que não esteja em desvantagem. Caso todos os pokemons e movimentos do Pokémon atual do agente estejam em desvantagem, ele irá utilizar um movimento aleatório.



*Figura 03 – Decisões finais do agente*

1. **Métodos especiais**

* **getEffectiveMove –** Recebe como parâmetro uma lista de movimentos, os tipos de um Pokémon, e uma bool “effective”. O método compara todos os movimentos com os tipos de um Pokémon, retornando o movimento que possua maior Base Power. O parâmetro “Effective” é utilizado para decidir se o método deve retornar um movimento super-efetivo ou um movimento normal. Caso todos os movimentos não sejam efetivos, retorna *Null*.
* **getEffectivePokemon –** Recebe como parâmetro dois arrays: um contendo os tipos de um pokemon, e outro contendo uma lista de pokemons. Compara os tipos do primeiro array com os tipos de todos os pokemons do segundo array, e retorna um Pokémon do segundo array que possua vantagem contra os tipos do primeiro array. Caso nenhum Pokémon seja efetivo, irá retornar um Pokémon que não esteja em desvantagem. Caso mais de um Pokémon esteja nesta condição, irá retornar o Pokemon com maior HP.
* **isPokemonEffective –** Recebe como parâmetros um array com os tipos do inimigo e um Pokémon (preferencialmente o Pokémon atual do agente). O método testa os tipos dos dois pokemons, retornando *True* caso os tipos do inimigo sejam super-efetivos contra o segundo Pokémon, e *false* caso o primeiro Pokémon esteja em desvantagem ou caso os tipos sejam neutros.