

Documentação do trabalho de PDS1

Descrição Geral

Este algoritmo implementa uma simulação de batalha Pokémon entre dois jogadores. Cada jogador pode ter até 100 pokémons, que possuem as características de ataque, defesa, vida e tipo. As batalhas seguem um sistema de turnos, onde os pokémons se atacam alternadamente e o vencedor é o jogador que derrotar todos os pokémons do adversário.

Este programa foi desenvolvido utilizando a linguagem C e ele lê os dados de entrada de um arquivo, processa as informações, realiza a batalha, e exibe os resultados, incluindo os pokémons sobreviventes do vencedor e os derrotados de ambos os jogadores.

Estrutura do Algoritmo

1. Leitura dos Dados:

- Os dados são lidos de um arquivo chamado dadospok.
- A primeira linha contém dois números inteiros, representando o número de Pokémon do jogador 1 e do jogador 2.
- Cada linha subsequente descreve os atributos de um Pokémon: Nome, Ataque, Defesa, Vida e Tipo.

2. Simulação da Batalha:

- A batalha ocorre em turnos alternados:
 - O Pokémon do jogador 1 ataca o Pokémon do jogador 2.
 - Em seguida, o Pokémon do jogador 2 contra-ataca.
- O dano é calculado com base no ataque do atacante, na defesa do defensor e em modificadores de tipo:
 - Se o tipo do atacante for "forte" contra o tipo do defensor, o ataque é aumentado em 20%.
 - Se for "fraco", o ataque é reduzido em 20%.
 - Em casos neutros, o ataque não sofre alteração.

- Quando um Pokémon tem sua vida reduzida a 0 ou menos, ele é derrotado, e o próximo Pokémon da equipe entra na batalha.

3. Exibição dos Resultados:

- O programa identifica o jogador vencedor.
- Exibe os Pokémon sobreviventes do vencedor e os Pokémon derrotados de ambos os jogadores.

Decisões de Implementação

1. Estruturas de Dados:

- Uma estrutura pokemon foi utilizada para armazenar os atributos de cada Pokémon.
- Arrays são utilizados para representar os times dos dois jogadores.

2. Funções Principais:

- calcular_modificador: Determina o modificador de tipo entre atacante e defensor.
- simular_batalha: Gerencia o fluxo da batalha entre os dois jogadores.
- ler_dados: Lê os dados do arquivo de entrada e inicializa os times.

3. Validações:

- Caso o arquivo de entrada não seja encontrado, o programa exibe uma mensagem de erro e encerra.
- O cálculo de dano garante que pelo menos 1 ponto de dano seja infligido, mesmo que a defesa do Pokémon adversário seja alta.

Exemplo de Entrada e Saída

Entrada (Arquivo dadospok):

```
3 3
Squirtle 12 8 15 agua
Vulpix 14 5 15 fogo
Onix 10 7 20 pedra
Golem 16 9 10 pedra
Charmander 8 8 12 fogo
Jolteon 10 7 12 eletrico
```

Saída:

```
dados do jogador 1:  
Squirtle 12 8 15 agua  
Vulpix 14 5 15 fogo  
Onix 10 7 20 pedra  
  
dados do jogador 2:  
Golem 16 9 10 pedra  
Charmander 8 8 12 fogo  
Jolteon 10 7 12 eletrico
```

iniciando a batalha:

```
Squirtle atacou Golem causando 3 de dano.  
Golem atacou Squirtle causando 8 de dano.  
Squirtle atacou Golem causando 3 de dano.  
Golem atacou Squirtle causando 8 de dano.  
Golem venceu Squirtle  
Vulpix atacou Golem causando 5 de dano.  
Vulpix venceu Golem  
Charmander atacou Vulpix causando 3 de dano.  
Vulpix atacou Charmander causando 6 de dano.  
Charmander atacou Vulpix causando 3 de dano.  
Vulpix atacou Charmander causando 6 de dano.  
Vulpix venceu Charmander  
Jolteon atacou Vulpix causando 5 de dano.  
Vulpix atacou Jolteon causando 7 de dano.  
Jolteon atacou Vulpix causando 5 de dano.  
Jolteon venceu Vulpix  
Onix atacou Jolteon causando 5 de dano.  
Onix venceu Jolteon  
jogador 1 venceu!  
pokemon sobreviventes do jogador 1:  
Onix
```

```
pokemon derrotados do jogador 1:  
Squirtle  
Vulpix
```

```
pokemon derrotados do jogador 2:  
Golem  
Charmander  
Jolteon
```

