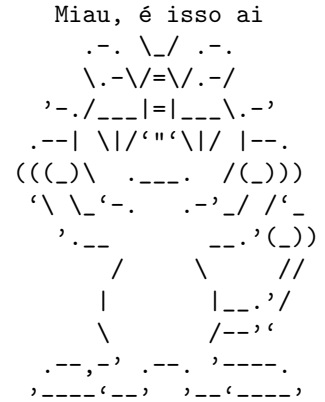
Relatório de Projeto - Psychomon

Guilherme Schützer - NUSP 8658544
Renato Geh - NUSP 8536030
Ricardo Lira - NUSP 8536131
Yan Couto - NUSP 8536023

1 Primeira Fase



Na primeira fase do projeto, primeiramente planejamos o que fariamos e começamos a primeira parte do diagrama de classes. Após debater e discutir como cada parte desa fase iria se interagir, e com o diagrama feito, ficou mais fácil a programação e implementação das ideias.

Inicialmente fizemos a base do programa, incluindo a definição de um pokemon, seus atributos, seus ataques, o que é um ataque, e como estes seriam usados. Ainda sem implementar as fórmulas de combate ou outras funções mais específicas, criamos duas classes que auxiliariam em organizar as batalhas e dados para o programa. A classe "Pokedex" serve para "coletar" e armazenar os pokemons e seus respectivos atributos/ataques, enquanto a classe "Pokestadium" tem como objetivo ser um criador de batalhas, que organiza quais pokemons lutam, e inicializa o combate.

Com essa base já preparada, começamos a planejar as lutas ("pokebattles"), que já teriam uma interface simples implementada para o usuário, de forma que seja visível o combate entre dois pokemons. A classe "Pokebattle" então definiria quem começaria (o pokemon mais rapido), e organizaria os turnos de cada pokemon, mostrando os movimentos possíveis e imprimindo alguns de seus atributos e do seu oponente.

Implementar o dano foi um pouco trabalhoso, a fórmula não é tão simples quanto pensávamos, mas tirando escrever toda a tabela de type effectiveness o resto foi bem rápido. Como já tinhamos deixado preparado o cálculo do

ataque antes, não foi necessário mudar nada fora da classe ataque, o que ajudou bastante.

Apesar de vários testes terem sido feitos durante todo esse processo, ao terminar o sistema de batalha mais simples, começamos a criar pokemons com a finalidade de testar o combate. Essa criação inicial foi feita "criativamente", inventando nomes e atributos aleatórios, sem se basear em pokemons existentes. Conforme fossemos arrumando alguns erros e bugs, começamos a pegar dados de sites como bulbapedia.com para nossos pokemons, adaptando as informações ao formato que nosso programa aceita como input.

2 Segunda Fase

A segunda fase do projeto foi bem diferente. Já tinhamos toda a base da primeira fase feita, o grande desafio foi aprender a usar novas bibliotecas para conseguir fazer tudo aquilo funcionar 'online'. Tinhamos pouca familiaridade com XML, então foi por ali que começamos, já que o servidor e o cliente precisavam daquilo para funcionar.

Depois de pesquisar algumas bibliotecas e tutoriais de como criar, ler e validar XML's em Python, decidimos usar a *lxml*, que também permite a validação dos XML's usando um xsd, que ao avançar no projeto, se mostrou muito útil

para encontrar erros, especialmente quando começamos a criar o servidor e o cliente.

Porém, tivemos uma grande dificuldade de usar essa biblioteca no Windows, tentando até mesmo compilar o seu código fonte, sem sucesso. Resolvemos portanto desenvolver o projeto no Linux, onde está biblioteca já vem instalada por padrão.

Tirando essa dificuldade inicial, o resto de XML's não foi muito trabalhoso, apenas esforço braçal.

Depois de pronta essa parte, já conseguimos um programa que cria XML's a partir de Pokemons e Pokemons a partir de XML's. Então começamos a criação da interface servidor/cliente. Para o servidor usamos a biblioteca *flask*, que tornou bem fácil a criação de um servidor funcional. O fluxo da batalha foi um pouco diferente do normal, por isso tivemos que adaptar parte do programa para isso. Para o cliente usamos *requests*, e o processo foi bem parecido com a batalha normal.

A interação entre os dois demorou um pouco para ficar boa, ocorreram algumas dificuldades para organizar o fluxo de dados, mas com um pouco de tempo conseguimos resolver. A primeira batalha usando dois terminais diferentes foi bem divertida.

Um 'problema' que encontramos é que fazendo as coisas exatamente como é dito no enunciado do EP, não é possível saber o HP inicial e PP's iniciais dos ataques da parte do servidor (a menos que *gambiarras* sejam utilizadas) e muito menos da parte do cliente. Além disso, o cliente fica sem nenhuma indicação de quais ataques foram usados pelo servidor e quanto dano os ataques causaram (além das informações adicionais).

Por último, também tivemos que arrumar o idioma do EP, já que em partes a comunicação com o usuário se dava em inglês e outras em português. Decidmos mudar, por fim, tudo para o inglês.