

MAC0242 - Laboratório de Programação II
Prof. Dr. Alfredo Goldman

Projeto da disciplina - Etapa 3

Visão geral

Nosso projeto da disciplina será a criação de um jogo de batalha, estilo Pokémon, utilizando **Python 3** com orientação a objetos para isso. Ele será composto de três etapas:

1. Modelagem da batalha em modo texto
2. Comunicação ponto a ponto via rede para batalhas
3. Inteligência artificial para batalhas

O objetivo é ao fim do semestre organizar um campeonato entre os grupos. Então desde o começo pensem na melhor modelagem possível para maximizar o aproveitamento de código entre as fases.

Em todas as etapas será esperada a manutenção de **diagramas de classe** descrevendo a modelagem do seu programa, bem como uma **cobertura de testes** completa.

Tecnologias

Obrigatoriamente, os programas devem ser desenvolvidos com Python 3. O uso de bibliotecas já existentes do Python é livre, mas devem ser claramente indicadas em um arquivo *requirements.txt*¹.

Opcionalmente, mas altamente recomendado, é o uso do pacote virtualenv² para isolar essas dependências e garantir que seu ambiente de desenvolvimento seja facilmente replicável em outras máquinas (lembrando que precisamos executar seu programa nas nossas máquinas para poder corrigir!).

Para o servidor da web da segunda etapa uma boa opção é o Flask³. Esta é apenas uma sugestão e não uma determinação. Façam como julgarem mais adequado.

¹ https://pip.pypa.io/en/latest/user_guide.html#requirements-files

² <http://virtualenv.readthedocs.org/en/latest/virtualenv.html#usage>

³ <http://flask.pocoo.org/>

Esclarecimentos e ajustes sobre a Etapa 1

- Lembrando que, exceto onde especificado, estamos usando a geração 1 (*Red/Blue/Green/Yellow*) como base.
- Níveis de prioridade dos ataques: o ataque acontece no turno do pokémon, ou seja, ele irá atacar e dar dano independente do ataque escolhido pelo oponente.
 - Ataques que consumam mais de um turno, da mesma forma, estão fora do escopo.
- Modificadores de tipo: usaremos a tabela de modificadores de tipo da geração 1⁴, mesmo que ela contenha erros que depois foram arrumados.
- Itens e ataques que mudem algum atributo do pokémon continuam fora do escopo do projeto.
- Sigam as sugestões de implementação e definições dos enunciados.

Esclarecimentos e ajustes sobre a Etapa 2

- *Struggle* é representado pelo id zero nas requisições, desde que o servidor só permita utilizá-lo se o pokémon estiver sem PP
- Esquema XML ajustado disponível junto com o enunciado
 - ID dos movimentos é limitado de 1 a 4
 - Um ataque tem apenas um tipo
- Apesar de altamente recomendável, a validação do XML recebido é livre para os que quiserem fazê-la

Etapa 3

Com os resultados das etapas um e dois, agora temos uma modelagem rudimentar de uma batalha pokémon e dois programas diferentes podem se comunicar por rede para batalhar.

Agora a evolução desta etapa é bastante simples: seu programa deve tomar as decisões sobre os ataques que usar sozinho. Tanto cliente quanto servidor devem fazer isso.

A forma como isso será feito e os algoritmos envolvidos são livres para cada grupo usar sua imaginação. Mas claro que ocorrerá uma competição (com direito a chocolates \o/) entre os diferentes programas entregues. Os vencedores ganharão

⁴ http://bulbapedia.bulbagarden.net/wiki/Type/Type_chart#Generation_I

melhores notas (não se assustem que a diferença de nota do primeiro para o último não será extrema!).

Em seu relatório devem constar pelo menos:

- os comandos para iniciar seu programa em modo cliente e servidor
- os comandos para executar os testes automatizados
- como fazer seu programa jogar contra ele mesmo

Conselho

Procure seus colegas para testar seus clientes e servidores uns contra os outros. Além disso, será que seu programa consegue jogar contra ele mesmo?

O que deve ser entregue

- Todo o código fonte compactado em um arquivo **zip** ou **tar.gz**, inclusive testes de unidade que devem cobrir o código implementado
- Diagrama de classes descrevendo a implementação nos formatos **pdf** ou **png**
- Relatório
 - Nome e número USP de todos os integrantes do grupo
 - Descrição das dificuldades e desafios
 - Demais itens pedidos no enunciado e que julgarem necessários

Importante

- O trabalho é estritamente em grupos de no máximo 4 pessoas. Discussões entre os grupos serão mais do que bem vindas, mas plágio ou cola não serão tolerados. Veja a [política do Departamento de Ciência da Computação para casos de plágio ou cola](#).
- Escreva de forma clara e estruturada todos os seus códigos e relatórios. Organize e nomeie todos os arquivos entregues de forma que eles possam ser facilmente identificados. A avaliação levará em conta todas essas questões! Uma apresentação ruim, ou a falta de clareza, poderá prejudicar sua nota.
- O programa deve ser entregue por meio do sistema [Paca](#) em um arquivo zip ou tar.gz
- Apenas **um** integrante do grupo deve fazer a entrega

- Enquanto o prazo de entrega não expirar, você poderá entregar várias versões do mesmo exercício-programa. Apenas a última versão entregue será guardada pelo sistema. Encerrado o prazo, você perderá 1 ponto para cada hora de atraso. Não deixe para entregar seu exercício na última hora!
- Guarde uma cópia do seu exercício-programa pelo menos até o final do semestre.
- Caso existam problemas de relacionamento entre os grupos, o professor e os monitores devem ser comunicados prontamente, para que medidas possam ser tomadas a tempo.