

{introcomp}

Trabalho Computacional - Pokémon

## 1 Introdução

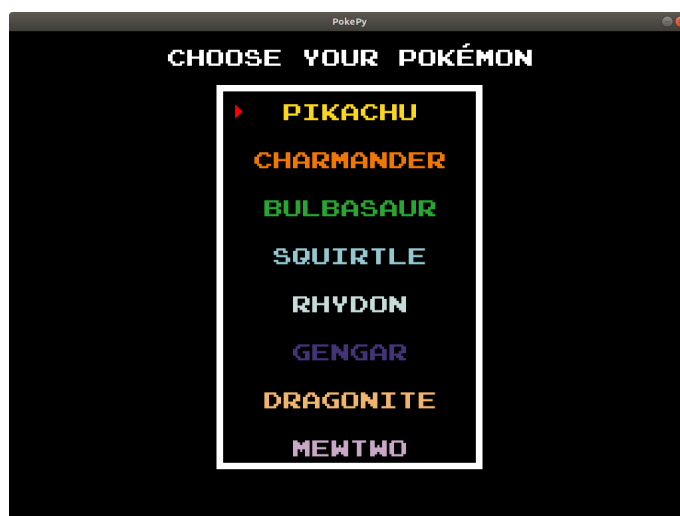
Ao longo do curso, diversos conceitos de programação foram lecionados. No módulo 2 demos e daremos ênfase às diversas aplicações que a linguagem Python pode oferecer. Um dos tópicos abordados foi a biblioteca Pygame, que permite uma representação gráfica com maior facilidade.

Utilizando esse recurso e todo o conteúdo das aulas, o trabalho consiste em um jogo baseado na série de jogos Pokémon, simulando uma batalha entre dois Pokémon. Abaixo forneceremos as especificações para a padronização dos trabalhos e alguns arquivos auxiliares que deverão ser utilizados no trabalho.

## 2 Especificações do Trabalho

O seu programa deve ser capaz de simular uma batalha entre dois Pokémon, escolhidos pelos usuários.

Primeiramente, o programa deve exibir uma tela inicial, em que cada jogador seleciona seu respectivo Pokémon. O menu deve funcionar com as setas convencionais do teclado em uma lista vertical e a confirmação deve ser feita pelo botão "Enter".



Após selecionados os dois Pokémon, a batalha deve ser iniciada. A estrutura visual deve simular a dos próprios jogos (de preferência, jogos mais antigos como o Fire Red, Leaf Green ou Emerald).



No canto esquerdo inferior tem-se um display onde são exibidas as mensagens produzidas por cada Pokémon, seja ataques, estado (vivo, morto) ou decisões dos jogadores.

No canto direito inferior, quatro opções devem estar disponíveis para o jogador:

- Lutar: exhibe os quatro golpes do Pokémon especificado no display;
- Bolsa: exhibe os itens na bolsa do treinador (**ponto extra**);
- Pokémon: exhibe a lista de Pokémon selecionáveis (**ponto extra**);
- Fugir: o treinador encerra a batalha e foge.

A opção atual deve ser destacada de alguma forma para facilitar a navegação do jogador.

Na parte central da tela temos a área de batalha, onde se situam os dois Pokémon.

O Pokémon selecionado deve ficar no canto esquerdo inferior, enquanto o adversário fica no oposto. Ao lado de cada um, deve ser exibido uma janela com seu respectivo nome, level e vida (em barra e número).

As imagens e sons que serão utilizados no trabalho estão disponíveis **aqui**, e ao longo do prazo podemos modificar ou incluir arquivos.

O trabalho poderá ser feito de maneira **INDIVIDUAL** ou **EM DUPLA**. O idioma do jogo e do código é opcional, preferencialmente em português ou inglês (lembrando que a legibilidade pode influenciar bastante na nota!). O prazo para entrega se encerra no dia **6 DE DEZEMBRO DE 2019**.

### 3 Especificações em relação ao Pygame

A biblioteca utilizada permitiria grandes variações entre trabalhos, portanto definimos o tamanho da janela como 800 pixels de largura por 600 pixels de altura, de maneira que a maior parte dos monitores conseguem suportar. Caso haja algum impedimento em relação a essa especificação, comunique a equipe o quanto antes!

As dimensões utilizadas durante a batalha para as janelas, tamanho dos pokémons, barras, fontes entre outras devem simular o máximo possível as reais do jogo original.

Quanto a tela inicial de escolha de Pokémon e a tela de encerramento do jogo, use sua criatividade! Easter Eggs e funcionalidades escondidas são bem vistas, desde que tenham bom senso. Qualquer recurso que for considerado ofensivo resulta em **nota 0!**

### 4 Boas Práticas

Como para boa parte dos alunos é o primeiro projeto de médio porte na programação, é imprescindível que se estabeleçam costumes para que a boa prática seja prolongada ao longo de novos códigos.

**Utilize funções!** É possível realizar o trabalho sem sequer fazer uma função, porém é algo extremamente desagradável e **muito mais difícil**. As funções são fundamentais para a descoberta de erros ao longo do programa e para maior apresentabilidade do código.

**Comente seu código!** "Código autoexplicativo" não existe. Coloque o máximo possível de comentários para que qualquer pessoa com o mínimo conhecimento de programação possa ser capaz de entender.

**Programação Orientada a Objetos!** Utilize dos conceitos lecionados ao longo do módulo 2 para desenvolver o trabalho, estruturando seu código

da melhor maneira possível, dividindo-o em classes e métodos.

## 5 Avaliação

A avaliação do trabalho divide-se em duas partes, sendo uma delas os pontos obrigatórios e a outra de pontos extras. A nota final será calculada considerando os dois valores, tendo os pontos obrigatórios um peso sobre os pontos extra:

$$NOTA = notaObrigatria + \frac{notaExtra * notaObrigatria}{10}$$

Os valores de cada parte do trabalho estão dispostos nas tabelas a seguir:

Requisitos Obrigatório	Pontuação
<b>Orientação a Objetos</b> - Seu trabalho deve ser modularizado e organizado de acordo com a programação orientada a objeto. Divida cada unidade adequadamente em classes e seus respectivos métodos.	1pt
<b>Comentários</b> - É indispensável que seu código esteja o mais explicado possível. Apesar da baixa pontuação, um código mal explicado pode acabar acarretando em penalização em outros tópicos.	0,5pts
<b>Funções</b> - Utilize funções de maneira adequada, visando um código bem objetivo, com funções de tamanho reduzido e nomes coerentes.	0,5pts
<b>Menu Inicial</b> - O menu inicial deve conter uma lista vertical de Pokémons, sendo navegável pelas setas e a selecionável por "Enter".	1pt
<b>Exibir Interface</b> - A interface é constituída pela área gráfica da batalha. O posicionamento dos quadros inferiores (preenchidos), Pokémons, barras com nome, status e level.	2pts
<b>Display do canto direito inferior</b> - O display deve exibir as quatro opções especificadas, obrigatoriamente funcionando a função de Fugir e tendo alguma reação na função Lutar.	1pt
<b>Sistema de Batalha</b> - O display inferior direito deve ter a função de luta implementada, exibindo os golpes disponíveis no display inferior esquerdo. O display inferior esquerdo deve exibir as mensagens de luta, como os golpes e estados de cada Pokémon. Os golpes devem causar dano no Pokémon adversário (alterando a barra de vida), as rodadas devem ser alternadas e o fim de jogo deve ser considerado.	3pts
<b>Animações</b> - O Pokémon golpeado deve exibir uma animação, como uma alteração rápida de cor ou um deslocamento. Quando um Pokémon perde toda sua vida, deve ser exibido também uma animação específica.	1pt
<b>TOTAL</b>	<b>10pts</b>

---

Requisitos Extras	Pontuação
<b>Animação da Barra de Vida</b> - Faça a vida decair vagarosamente quando o Pokémon receber um golpe.	1pt
<b>Música e Sons</b> - Toque uma música no menu inicial e também as músicas características durante a batalha (tanto a de início quanto a final).	1pts
<b>Easter Eggs</b> - Easter Eggs, desde que <b>NÃO SEJAM OFENSIVOS DE MANEIRA ALGUMA</b> são bem vistos, tendo uma avaliação quanto ao grau de complexidade.	1pts
<b>Pokémons Shiny</b> - Deve haver uma pequena probabilidade de que um Pokémon selecionado seja do tipo Shiny.	0,5pts
<b>Bolsa com Poções</b> - O botão "Mochila" do menu inferior direito deve funcionar, exibindo a mochila do jogador com alguma poção. Se selecionada, a vida do Pokémon deve ser aumentada.	1pts
<b>PVE</b> - O jogo deve ter suporte para apenas um jogador, tendo um adversário que escolha os golpes.	1pt
<b>Arquivos</b> - O código deve ser particionado em arquivos de maneira organizada e concisa.	1pt
<b>TOTAL</b>	<b>6,5pts</b>

As especificações do trabalho estão sujeitas a alterações, sendo estas comunicadas em aula e pelo grupo do WhatsApp.

Sugerimos que comece o trabalho quanto antes, não é uma tarefa trivial. Utilize das aulas como apoio principalmente nos conceitos de Orientação a Objetos e na biblioteca Pygame. Solucionaremos o máximo de dúvidas possíveis, mas pedimos paciência pela alta carga que o curso de Engenharia de Computação nos proporciona (principalmente no final do período) e pela maior complexidade das dúvidas, que por vezes são inviáveis pelas plataformas online convencionais.

**PIKA PIKA!** (mãos à obra!)

**OBS:** O trabalho deverá ser enviado por e-mail até o dia 06/12/2019, para o endereço eletrônico [introcomp@inf.ufes.br](mailto:introcomp@inf.ufes.br). O assunto do e-mail deverá ser Trabalho Introcomp - <NOME DO ALUNO>