



Trabalho Computacional - IntroBattle

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 criar o sistema de batalha de um jogo de RPG. RPG é um dos gêneros mais populares de jogos eletrônicos, com os primeiros títulos datando de antes dos anos 80. Neles, o jogador controla as ações de um ou mais personagens e acompanha a história que o jogo conta. Alguns elementos que jogos deste gênero possuem são:

- Diferentes personagens com características, habilidades, história, itens e estatísticas únicas.
- A exploração do mundo onde se passa a história, com elementos fantásticos inspirados em fantasias da Idade Média ou elementos de ficção científica.
- Desenvolvimento da história, tanto do mundo quanto dos personagens principais e secundários.
- Desafios, principalmente através de combates e quebra-cabeças, que o jogador deve enfrentar para dar continuidade ao jogo.

Como seria demais abordar tudo isso em apenas um trabalho, vamos nos dedicar a implementar o sistema de batalha de um jogo de RPG. Para isso, na tela inicial, o jogador poderá escolher 3 heróis para seu grupo, que logo em seguida, enfrentará um necromante e sua última criação.

Abaixo forneceremos as especificações para a padronização dos trabalhos e alguns arquivos auxiliares que deverão ser utilizados no trabalho.

2 O Funcionamento da Batalha

Como dito anteriormente, o jogo consiste em uma batalha RPG na qual o jogador controla 3 personagens e deve derrotar o grupo de personagens inimigo para vencer.

Para explicarmos como o jogo deve funcionar, primeiro precisamos definir algumas regras. Todos os personagens, tanto os do jogador quanto os do inimigo, possuem algumas estatísticas em comum. Estas são:

- Pontos de Vida
- Ataque
- Defesa
- Velocidade

Os pontos de vida de um personagem representam a saúde e disponibilidade dele no combate. Costumam ser um valor positivo, e alcançar 0 pontos de vida significa que o personagem foi derrotado e não tem mais condições de continuar lutando.

O ataque de um personagem representa seu potencial ofensivo. Seja com lâminas afiadas, bolas de fogo ou flechas, este valor representa o quão bom este personagem é em atacar seus inimigos.

A defesa de um personagem representa seu potencial defensivo. Armadura pesada, bons reflexos ou uma magia de proteção são algumas das coisas que fazem com que um personagem tenha uma boa defesa, e é o que impede seus pontos de vida de irem a 0!

Por fim, velocidade. Um atributo mais simples, que rege a ordem em que o combate será realizado. Personagens com maior velocidade poderão agir primeiro, o que pode ser crucial para a estratégia de um determinado grupo.

Começando pelo personagem com a velocidade mais alta, e seguindo por ordem de velocidade, o jogo será realizado em turnos. Cada turno, um personagem poderá realizar uma determinada ação. As ações básicas exigidas para seu programa serão atacar e defender.

Escolher a ação **atacar** deve permitir ao personagem causar dano no inimigo baseado no seu ataque, enquanto escolher a ação **defender** deve permitir ao

2 O FUNCIONAMENTO DA BATALHA

personagem diminuir a quantidade de dano que ele sofrerá no começo do seu próximo turno.

Como é normal existir mais de um inimigo ao mesmo tempo durante um combate, seu programa também deverá ser capaz de permitir ao jogador escolher a qual inimigo atacar (mais sobre isso abaixo).

Ao realizar um ataque, o personagem atacante deverá subtrair um certo valor (dano) dos pontos de vida do alvo com base na seguinte fórmula:

$$Dano = Ataque \times \frac{50}{50 + Defesa}$$

(mas lembre-se de usar o ataque do personagem atacante e a defesa do personagem que está sendo atacado!).

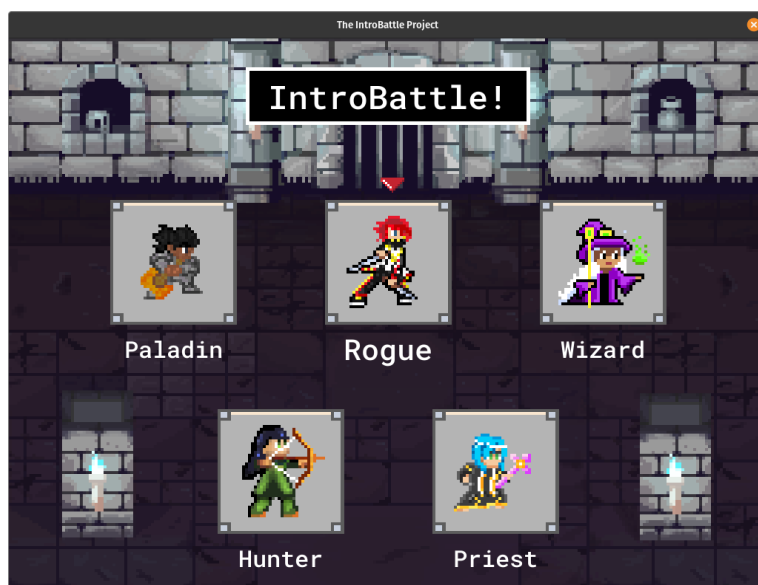
Ao realizar a ação de defesa, o personagem deverá ficar com o dobro do seu valor de defesa, fazendo com que ataques causem menos dano a ele. Este efeito se encerra no começo do seu próximo turno.

Realizar um ataque que faça os pontos de vida de um personagem chegarem a 0, derrota o personagem, fazendo com que ele se retire da batalha. O jogo se encerra quando um dos grupos não possuir mais personagens para lutar.

3 Especificações do Trabalho

Seu programa deve ser capaz de simular uma batalha entre dois grupos diferentes, um controlado pelo jogador e um controlado pelo próprio programa.

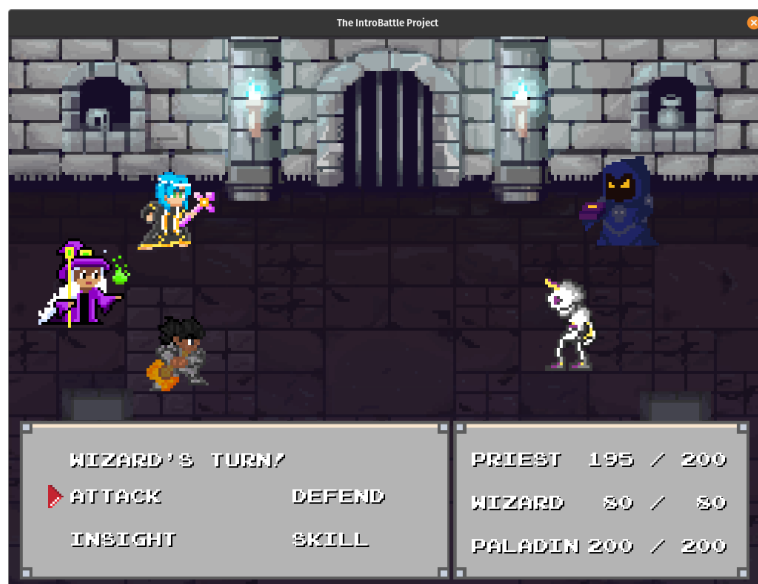
Primeiramente, o programa deve exibir uma tela inicial, em que o jogador escolhe os personagens que quer levar para batalha. O menu deve funcionar com as setas convencionais do teclado e possuir os personagens disponibilizados para escolha conforme a imagem abaixo.



Para selecionar os personagens, utilizaremos a tecla "Z". Faça a seta ficar em cima do personagem que quer selecionar, e lembre-se de não deixar o jogador selecionar o mesmo personagem mais de uma vez!

Após selecionar os 3 personagens do seu grupo, o jogador deve ser redirecionado para uma batalha entre o grupo escolhido e o grupo de inimigos.

A estrutura visual deve se parecer com a imagem abaixo.



No canto inferior esquerdo deve haver uma mensagem informando o personagem que deve jogar no turno atual. Além disso, caso seja um personagem do jogador, aparecerão as ações que o personagem poderá executar, seja atacar, defender ou utilizar ações especiais. Isso consiste no menu de escolhas que o jogador pode fazer!

No canto inferior direito, devem ser mostrados os nomes de cada um dos personagens do jogador, juntamente com seus pontos de vida. É necessário destacar a opção atual sempre que o jogador precisar realizar uma escolha/decisão, facilitando a jogabilidade. No nosso exemplo, há uma seta que aponta para a opção selecionada.

Na parte central da tela temos a área de batalha, onde se situam os dois grupos (tanto os personagens do jogador quanto os inimigos).

O grupo do jogador deve ficar na parte esquerda do campo de batalha, enquanto o necromante e seu esqueleto ficam no lado oposto. Como cada personagem atacará individualmente, é necessário explicitar qual personagem está atacando no momento, bem como o alvo de seu ataque. Novamente, utilizamos uma seta que aponta para qual inimigo o ataque será realizado.

Caso o jogador queira voltar em algum momento (por exemplo, caso ele tenha selecionado a opção de atacar, e enquanto está escolhendo o alvo, decide voltar e usar outra ação), permita que ele volte para o menu de seleção de ações com a tecla "X".

3 *ESPECIFICAÇÕES DO TRABALHO*

As imagens que serão utilizados no trabalho estão disponíveis [aqui](#). Ao longo do prazo do trabalho podemos modificar ou incluir arquivos nessa pasta.

O trabalho poderá ser feito de maneira **INDIVIDUAL**, em **DUPLA** ou **TRIO**. O idioma do jogo e do código deverá ser em português ou inglês (lembrando que a legibilidade pode influenciar bastante na nota!). O prazo para entrega se encerra no dia **11 DE DEZEMBRO DE 2021** (sendo passível de alterações então fique atento aos avisos no servidor do Discord).

4 Especificações em relação ao Pygame

A biblioteca utilizada permitiria grandes variações entre trabalhos, portanto definimos o tamanho da janela como 1024 pixels de largura por 768 pixels de altura, de maneira que a maior parte dos monitores conseguem suportar. Caso haja algum empecilho em relação a essa especificação, comunique a equipe o quanto antes!

As dimensões utilizadas durante a batalha para as janelas, tamanho dos personagens, fontes entre outras devem simular o máximo possível às imagens deste documento (não é necessário ficar exatamente igual, mas tente não deixar algo pequeno ou grande demais).

Quanto a tela inicial de escolha de personagens 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!**

5 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 que façam sentido 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, e dividindo-o em classes e métodos.

6 Envio do Trabalho

O trabalho deverá ser enviado por e-mail por um integrante do grupo até o dia 11/12/2021, para o endereço eletrônico introcomp@inf.ufes.br. O assunto do e-mail deverá ser Trabalho Introcomp - <NOME COMPLETO DOS ALUNOS>

Deve ser enviado um arquivo compactado, no formato .zip, contendo os arquivos .py, as imagens, fontes, sons e etc. usados na implementação do trabalho. Lembre-se de separar os arquivos em pastas para facilitar a correção do trabalho.

7 Avaliação

O trabalho vale 10 pontos. Além dos 10 pontos disponíveis através de requisitos obrigatórios, também disponibilizaremos requisitos extras! Esforços como adicionar efeitos sonoros e música, *easter eggs*, ações únicas, etc. proporcionarão ao(s) aluno(s) a chance de compensar possíveis problemas entre os requisitos obrigatórios.

Porém, para que essas atividades extras não desviem o foco da parte obrigatória do trabalho, adicionaremos um limitador à nota total. Este modificador será um valor entre 0 e 1 e é calculado da seguinte forma:

$$\text{modificador} = \frac{\text{nota obrigatória}}{10}$$

fazendo com que trabalhos que tenham desempenhado mal na parte obrigatória não consigam "passar por cima" dos requisitos obrigatórios através dos pontos extras. Por fim, a nota total será:

$$\text{Nota total} = \text{nota obrigatória} + (\text{modificador} \times \text{nota extra})$$

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

Requisitos Obrigatório	Pontuação
Orientação a Objetos - Seu trabalho deve ser modularizado e organizado de acordo com a programação orientada a objetos. Divida cada unidade adequadamente em classes e seus respectivos métodos.	1 ponto
Comentários - 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.	0,5 ponto
Funções - Utilize funções de maneira adequada, visando um código bem objetivo, com funções de tamanho reduzido e nomes coerentes.	0,5 ponto
Menu Inicial - O menu inicial deve conter uma interface com os personagens disponíveis no jogo, sendo navegável pelas setas do teclado e selecionável por "Z". Obs: caso seu jogo use alguma outra tecla, é necessário que coloque essa informação escrita ou em um arquivo texto ou em uma tela de instruções.	2 pontos
Exibir Interface - A interface é constituída pela área gráfica da batalha e a parte dos menus. Seu jogo deve mostrar os personagens, os inimigos e o menu. Como dito anteriormente, o menu precisa indicar o nome da classe de todos os personagens do jogador, além da vida atual e máxima deles. Além disso, também deve informar de quem é o turno atual, e caso seja do jogador, também deve disponibilizar as opções de ação (como atacar e defender).	2 pontos
Sistema de Batalha - As ações de atacar/defender devem funcionar. Os personagens devem agir conforme a ordem de agilidade deles. Um personagem que não tiver mais vida (morreu durante a batalha) não deve poder atacar. Quando um dos grupos (aliados ou inimigos) não tiver mais personagens, o jogo deve se encerrar e informar se o jogador ganhou ou perdeu.	4 pontos
TOTAL	10 pontos

Requisitos Extras	Pontuação
Música e Sons - Faça o jogo ter efeitos sonoros!	1 ponto
Easter Eggs - Easter Eggs, desde que NÃO SEJAM OFENSIVOS DE MANEIRA ALGUMA são bem vistos, tendo uma avaliação quanto ao grau de complexidade e quanto à criatividade . (O trabalho com Easter Egg mais legal normalmente é mostrado nas próximas edições do Introcomp!)	2 pontos
Ações Únicas - Faça com que cada classe tenha uma habilidade única além de atacar/defender. Elas ficam a critério dos membros do grupo, mas daremos algumas sugestões abaixo.	3 pontos
Tipos de ataque e defesa diferentes - Faça com que o ataque de cada personagem tenha um tipo de dano (físico ou mágico) dependendo de sua classe. Além disso, faça com que cada personagem tenha dois tipos de defesa: defesa física e defesa mágica. Dessa forma, quando um ataque acontecer, a defesa relacionada ao tipo de dano deve ser utilizada. Faça com que a ação de defesa dobre ambos os tipos de defesa do personagem.	2 pontos
Libra/análise - Faça com que um personagem seja capaz de analisar informações sobre um alvo. Ao analisar o alvo, exiba uma mensagem com o ataque (e tipo de dano caso tenha implementado), a defesa (ou defesas, caso tenha implementado), além da vida atual do alvo.	1 ponto
TOTAL	9 pontos

Para as ações únicas, algumas sugestões:

- Faça o mago usar uma magia que causa dano a todos os inimigos;
- Faça o guerreiro proteger seus aliados, aumentando a defesa deles por alguns turnos;
- Faça o clérigo curar um aliado;
- Faça o ladino causar mais dano com base na vida atual do alvo (fazendo ele causar mais dano com base na vida perdida do alvo).

8 Considerações Finais

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

Sugerimos que comece o trabalho o 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 os cursos de Engenharia e Ciência da Computação nos proporciona (principalmente no final do período) e pela maior complexidade das dúvidas, que por vezes são inviáveis pelas plataformas online convencionais.