

Relatório Programação

Aluno: Pedro Vinicius Mazo

Turma: 1G

Comecei fazendo o jogo, pela boa prática que é usado para criação de jogos com python: um arquivo principal main.py para executar o jogo, um arquivo game.py que será meio que a “Engine”, e o resto dos arquivos dentro suas respectivas pastas organizadas, essa boa prática foi feita com base em vídeos do Youtube, em que eles estavam organizando de uma forma parecida com essa.

- Importei todas as configurações do jogo antigo para o jogo atual, e deixei também em um módulo separado, chamado “configs.py”.
- O game.py segue uma forma parecida como experientes usam com o pygame, deixam um loop principal, onde: primeiro se atualiza, e depois é desenhado o jogo na tela.
- Criei a função RenderMap() para que não fique repetitivo escrever código, já que ela é usada em outros contextos também.

A primeira parte foi a renderização da matriz do mapa, usei uma forma parecida do código anterior para mostrar o mapa para o usuário, depois, foi a mostragem do Player, junto com a criação dele, criei a função CreatePlayer() para poder criar o jogador de uma forma externa, para não ficar com muito código em um arquivo só.

- Criei a classe Entidade (poderia ser opcionalmente abstrata), para ser uma superclasse de Player e Inimigo, onde cada um desses dois têm características diferentes, usando do polimorfismo para substituir as funções de Entidade, e fazer com que eles executem de formas diferentes, por serem objetos com propósitos diferentes.

O código printer.py foi criado para ter funções “extras” para deixar a experiencia melhor para o usuário, é um pacote com funções para colorir e limpar a cor terminal e também poder mostrar mensagens de forma animada.

- Dentro do arquivo default.py, temos a classe CenarioDefault, ela se refere ao cenário padrão do jogo, esse cenário pode ser trocado em qualquer momento, dois exemplos são no: inventario e batalhas, onde quando o usuário escolhe entrar em um desses modos, o cenário muda para um desses, em que dentro da classe Game no método run(), ele executa o update do cenário respectivo, é meio que um polimorfismo por duckTyping
- Como dito anteriormente, o código map.py, possui a função RenderMap(), que foi justamente executava dentro do cenário do inventário e da batalha para que seja mostrado ainda o mapa para o jogador.

Uma certa parte do código é focada para fazer os inimigos se moverem, nessa parte, escolhe um número aleatório, e escolhe esse mesmo número em forma de índice dentro do entities, se esse entities[índice] tiver o atributo “player” ele se move,

pois o inimigo é o único Objeto entre entities que recebe o atributo de "player", oque é uma forma de reconhecer isso.

O chatGPT foi usado apenas para gerar a matriz do mapa, para poupar tempo, nenhum código escrito foi feito por ele, além da matriz do mapa.

Foi criado o arquivo photos.py para guardar uma imagem visual de cada um dos inimigos, para deixar o jogo mais bonito, todas as imagens foram pegadas do mesmo site (<https://ascii.co.uk/art>), todas as vezes que o usuário atacar, ele verá a imagem do inimigo.

Quando o inimigo morrer, ele será reconstruído e trocado de posição apenas, para não ter como "zerar" o jogo, sempre terão mais inimigos.

Foi adicionado um contador de inimigos mortos pelo jogador, ele será exibido ao lado direito junto com outras informações.