

Neste trabalho, desenvolvemos a adaptação de um jogo simples feito em python para o modelo de programação orientada a objetos. O jogo original funcionava inteiramente em um único arquivo, com todas as partes misturadas: criação do personagem, inventário, batalhas e menu. Apesar de funcionar, essa estrutura deixava o projeto desorganizado e pouco flexível para mudanças ou expansões.

No código inicial, as informações do personagem ficavam espalhadas, como vida, força, itens e a classe escolhida. As decisões eram feitas de forma direta e com pouca organização. O inventário era tratado de maneira muito simples e o combate possuía poucas variações.

Para melhorar isso, reorganizamos tudo usando o conceito de objetos. A primeira etapa foi transformar o personagem em uma estrutura própria, que passou a concentrar seus atributos e características. Cada classe do jogo foi separada em sua própria categoria, com valores e habilidades específicas. Isso deixou o sistema mais organizado e fiel ao estilo de um rpg.

O inventário, que antes era apenas uma lista comum, passou a fazer parte da estrutura do personagem. Assim, ficou claro que cada jogador possui e controla seu próprio conjunto de itens, tornando o jogo mais coerente.

Também criamos uma estrutura separada para representar os inimigos. Dessa forma, eles passaram a ter características variadas, tornando as batalhas menos repetitivas e mais interessantes.

Outra mudança importante foi a reorganização do funcionamento geral do jogo. O fluxo principal foi isolado em uma parte específica, responsável por chamar os sistemas de batalha, exploração e menu. Isso separou melhor as responsabilidades e deixou o código mais limpo e fácil de entender.

Além da organização estruturada, ampliamos a história do jogo. A introdução foi estendida, trazendo mais detalhes sobre a vila de Venarium, a névoa misteriosa e as ruínas antigas. Isso gerou uma ambientação mais envolvente e combinou melhor com o novo formato do jogo.

A inteligência artificial foi usada em alguns momentos, como para testar o código e apontar possíveis erros ou para rodar o jogo e usar todas as funções possíveis, assim, otimizando nosso tempo

Em conclusão, a adaptação para Programação Orientada a Objetos trouxe diversas melhorias. O jogo se tornou mais organizado, mais simples de ampliar e com uma aparência mais profissional. A divisão em partes menores e mais claras ajudou bastante na compreensão geral do projeto. O trabalho permitiu entender melhor como a POO funciona e por que ela é uma ferramenta tão útil para projetos maiores e mais complexos.