Documentação Jogo da Cobrinha

Na primeira etapa foi criada uma estrutura básica procedural e em seguida foram definidas variáveis globais de largura e altura da tela, cores, velocidade e posição inicial da cobra. Esta foi representada por um vetor, ou seja, uma lista que armazenava as coordenadas de cada segmento. Essa estrutura possibilitou adicionar ou remover elementos dinamicamente, simulando o crescimento da cobra quando ela come a comida. A movimentação foi feita calculando a nova posição da cabeça e inserindo-a no início do vetor. Deste modo, caso a cobra não comesse, o último elemento era removido, mantendo o seu tamanho constante. O jogo é controlado pelo usuário por eventos do Pygame, verificando teclas pressionadas para alterar a direção. Ele termina quando a cobra colide com as bordas ou com ela mesma. Essa versão não tinha placar nem tela de Game Over, mas já demonstrava o uso de vetores para manipulação dinâmica.

Já na segunda etapa a estrutura foi organizada com funções para separar responsabilidades. Foram criadas funções como loop_jogo e tela_game_over. Adicionamos o placar, exibindo a pontuação na tela e a pontuação era incrementada sempre que a cobra comia a comida, e exibida no canto superior da tela. Também foi implkementada uma tela de Game Over, permitindo reiniciar o jogo com a tecla Enter ou sair com a tecla P. Nessa fase houve a integração com o arquivo de configuração *conf.ini*, o que permitiu que o usuário definisse controles sem alterar o código.

A terceira etapa foi a migração para orientação a objetos, criando a classe JogoCobrinha. Essa mudança trouxe encapsulamento, reutilização e facilidade de manutenção. A classe concentra toda a lógica do jogo, e seus atributos representam o estado, como largura e altura da tela, lista da cobra, direção, pontuação e posição da comida. Deste modo, o método carregar_configuracoes lê o arquivo conf.ini e carrega as configurações globais, como resolução e teclas. A lógica do jogo foi distribuída em métodos: reiniciar_jogo pra resetar o estado do jogo, tratar_eventos pra processar as entradas do usuário por meio das teclas, atualizar pra movimentar a cobra e verificar colisões, desenhar pra renderizar o jogo na tela, tela_game_over pra exibir a tela de fim de jogo e executar que controla o loop principal do jogo. Assim sendo, cada função acaba interagindo com os atributos da classe e o encapsulamento é mantido.

Por fim, a evolução do código teve linha lógica: primeiro, foi a versão mínima com tudo em um bloco, apenas movimento e colisão; depois, uma versão intermediária

com funções, placar e tela de Game Over; por fim, uma versão orientada a objetos, com encapsulamento, integração completa com conf.ini e maior organização. Cada etapa trouxe modularidade e clareza. Comecei definindo variáveis globais e depois evoluí pra outras atributos encapsulados em uma classe. O vetor ficou como estrutura central, porém o modo como ele era manipulado ficou mais organizado.