

O jogo é uma adaptação do famoso Flappy Bird, no qual o jogador controla um pássaro que deve passar por canos que se movem na tela, evitando colidir com eles. O jogador deve pressionar a barra de espaço para fazer o pássaro voar e desviar dos obstáculos. O jogo termina quando o pássaro colide com um cano ou com o chão, porém nesse jogo ao invés do pássaro, é um morcego em um cemitério desviando de tumulos.

O jogo é escrito em Python e utiliza a biblioteca Pygame para lidar com gráficos e interação com o usuário. O código tem as seguintes dependências/bibliotecas:

pygame

os

random

O código começa definindo as dimensões da tela em que o jogo será executado, largura e altura, que são armazenadas nas constantes `TELA_LARGURA` e `TELA_ALTURA`, respectivamente. Em seguida, são carregadas as imagens que serão usadas no jogo, que estão localizadas na pasta `imgs`. As imagens são carregadas e redimensionadas usando o método `pygame.transform.scale2x`, que aumenta o tamanho da imagem em uma proporção de 2x.

As imagens carregadas são:

`IMAGEM_CAIXAO`: a imagem do cano que o morcego deve desviar;

`IMAGEM_CHAO`: a imagem do chão que aparece na parte inferior da tela;

`IMAGEM_BACKGROUND`: a imagem de fundo que aparece atrás dos canos e do morcego;

`IMAGENS_BAT`: uma lista com as três imagens do morcego, que são usadas para criar a animação de batimento de asas.

Depois, é criada uma instância da classe `pygame.font.Font`, que é usada para exibir os pontos do jogador na tela. A fonte usada é a Arial, com tamanho 45.

A seguir, é definida a classe `bat`, que representa o morcego. A classe possui os seguintes atributos:

`x`: a posição horizontal do morcego na tela;

`y`: a posição vertical do morcego na tela;

`angulo`: o ângulo de rotação do morcego;

`velocidade`: a velocidade vertical do morcego;

`altura`: a altura em que o morcego estava antes de começar a cair;

`tempo`: o tempo desde o último pulo do morcego;

`contagem_imagem`: o tempo desde a última troca de imagem do morcego na animação de batimento de asas;

`imagem`: a imagem atual do morcego.

A classe `bat` possui os seguintes métodos:

pular: faz o morcego pular, alterando sua velocidade vertical e tempo;

mover: move o morcego verticalmente, atualizando sua posição e ângulo de acordo com sua velocidade;

desenhar: desenha o morcego na tela, aplicando a animação de batimento de asas e a rotação;

get_mask: retorna uma máscara de colisão para o morcego, que é usada para detectar colisões com os canos.

A seguir, é definida a classe `Cano`, que representa um cano. A classe possui os seguintes atributos:

x: a posição horizontal do cano na tela;

altura: a altura do cano;

pos_topo: a posição vertical do topo do cano;

pos_base: a posição do chão ou base;

Além disso, a classe `Cano` possui um método `definir_altura` que é chamado no construtor para definir a altura do cano superior e inferior, a distância entre eles, e suas respectivas posições na tela. O método `mover` atualiza a posição do cano na tela de acordo com a velocidade estabelecida. O método `desenhar` desenha o cano superior e inferior na tela.

Por fim, a função `jogo` cria as instâncias da classe `bat` e `Cano`, e os desenha na tela continuamente enquanto o jogo estiver em execução. A função também verifica se houve colisão entre o morcego e os canos, atualiza a pontuação do jogador, e exibe a pontuação na tela. A função utiliza o módulo `time` para controlar a velocidade de atualização do jogo, e a função `pygame.quit()` é chamada ao final do jogo para fechar a janela do Pygame.