import tkinter as tk

from tkinter import messagebox

import random

# Kart sınıfı

class Card:

def \_\_init\_\_(self, isim, hasar):

self.isim = isim

self.hasar = hasar

def \_\_str\_\_(self):

return self.isim

# Oyuncu sınıfı

class Player:

def \_\_init\_\_(self, isim):

self.isim = isim

self.el = []

self.puan = 0

def kart\_cek(self, kart):

self.el.append(kart)

def kart\_oyna(self, index):

return self.el.pop(index)

def puan\_ekle(self, puan):

self.puan += puan

# Oyun sınıfı

class Game:

def \_\_init\_\_(self, mode):

self.mode = mode

self.deste = [

Card("Pikachu", 50), Card("Bulbasaur", 45), Card("Charmander", 55),

Card("Squirtle", 48), Card("Jigglypuff", 42), Card("Meowth", 44),

Card("Psyduck", 46), Card("Snorlax", 49), Card("Eevee", 43), Card("Mew", 60)

]

self.ortadaki\_kartlar = random.sample(self.deste, 10)

self.oyuncu = Player("Oyuncu" if mode == "player\_vs\_computer" else "Bilgisayar1")

self.bilgisayar = Player("Bilgisayar" if mode == "player\_vs\_computer" else "Bilgisayar2")

self.tur\_sayisi = 0

def kart\_secim(self, oyuncu\_secim\_index, bilgisayar\_secim\_index):

oyuncu\_secim = self.ortadaki\_kartlar.pop(oyuncu\_secim\_index)

self.oyuncu.kart\_cek(oyuncu\_secim)

bilgisayar\_secim = self.ortadaki\_kartlar.pop(bilgisayar\_secim\_index - 1)

self.bilgisayar.kart\_cek(bilgisayar\_secim)

def kart\_sec\_ve\_devam\_et(self, oyuncu\_secim=None):

if self.mode == "computer\_vs\_computer":

oyuncu\_secim = random.randint(0, len(self.oyuncu.el) - 1)

oyuncu\_karti = self.oyuncu.kart\_oyna(oyuncu\_secim)

bilgisayar\_secim = random.randint(0, len(self.bilgisayar.el) - 1)

bilgisayar\_karti = self.bilgisayar.kart\_oyna(bilgisayar\_secim)

if oyuncu\_karti.hasar > bilgisayar\_karti.hasar:

self.oyuncu.puan\_ekle(5)

elif oyuncu\_karti.hasar < bilgisayar\_karti.hasar:

self.bilgisayar.puan\_ekle(5)

return oyuncu\_karti, bilgisayar\_karti

def oyun\_devam\_ediyor(self):

return bool(self.oyuncu.el or self.bilgisayar.el)

def kazanan(self):

if self.oyuncu.puan > self.bilgisayar.puan:

return f"{self.oyuncu.isim} oyunu kazandı! Toplam puan: {self.oyuncu.puan}, {self.bilgisayar.isim} puanı: {self.bilgisayar.puan}"

elif self.oyuncu.puan < self.bilgisayar.puan:

return f"{self.bilgisayar.isim} oyunu kazandı! Toplam puan: {self.bilgisayar.puan}, {self.oyuncu.isim} puanı: {self.oyuncu.puan}"

else:

return f"Oyun berabere! {self.oyuncu.isim} puanı: {self.oyuncu.puan}, {self.bilgisayar.isim} puanı: {self.bilgisayar.puan}"

# GUI oluşturma

class PokemonGameGUI:

def \_\_init\_\_(self, root):

self.root = root

self.root.title("Pokemon Kart Oyunu")

self.root.geometry("800x600")

self.mode\_label = tk.Label(self.root, text="Oyun modunu seçin:")

self.mode\_label.pack()

self.mode\_var = tk.StringVar()

self.mode\_var.set("player\_vs\_computer")

self.mode\_radio\_player = tk.Radiobutton(self.root, text="Oyuncu vs Bilgisayar", variable=self.mode\_var,

value="player\_vs\_computer")

self.mode\_radio\_player.pack()

self.mode\_radio\_computer = tk.Radiobutton(self.root, text="Bilgisayar vs Bilgisayar", variable=self.mode\_var,

value="computer\_vs\_computer")

self.mode\_radio\_computer.pack()

self.start\_button = tk.Button(self.root, text="Başlat", command=self.start\_game)

self.start\_button.pack()

def start\_game(self):

self.game = Game(self.mode\_var.get())

self.root.destroy()

self.show\_game\_ui()

def show\_game\_ui(self):

self.game\_window = tk.Tk()

self.game\_window.title("Pokemon Kart Oyunu")

self.game\_window.geometry("800x600")

self.player\_label = tk.Label(self.game\_window, text=self.game.oyuncu.isim)

self.player\_label.grid(row=0, column=0, padx=10, pady=10)

self.computer\_label = tk.Label(self.game\_window, text=self.game.bilgisayar.isim)

self.computer\_label.grid(row=0, column=2, padx=10, pady=10)

self.player\_listbox = tk.Listbox(self.game\_window, width=30, height=10)

self.player\_listbox.grid(row=1, column=0, padx=10, pady=10)

self.computer\_listbox = tk.Listbox(self.game\_window, width=30, height=10)

self.computer\_listbox.grid(row=1, column=2, padx=10, pady=10)

self.middle\_listbox = tk.Listbox(self.game\_window, width=30, height=10)

self.middle\_listbox.grid(row=1, column=1, padx=10, pady=10)

self.play\_button = tk.Button(self.game\_window, text="Kart Seç", command=self.choose\_cards)

self.play\_button.grid(row=2, column=1, padx=10, pady=10)

self.update\_ui()

def update\_ui(self, show\_damage=False):

self.player\_listbox.delete(0, tk.END)

self.computer\_listbox.delete(0, tk.END)

self.middle\_listbox.delete(0, tk.END)

for card in self.game.oyuncu.el:

display\_text = f"{card.isim} - Hasar: {card.hasar}"

self.player\_listbox.insert(tk.END, display\_text)

for card in self.game.bilgisayar.el:

display\_text = f"{card.isim} - Hasar: {card.hasar}"

self.computer\_listbox.insert(tk.END, display\_text)

for card in self.game.ortadaki\_kartlar:

display\_text = card.isim if not show\_damage else f"{card.isim} - Hasar: {card.hasar}"

self.middle\_listbox.insert(tk.END, display\_text)

def choose\_cards(self):

if self.game.tur\_sayisi < 3:

if self.game.mode == "computer\_vs\_computer":

while len(self.game.oyuncu.el) < 3:

player\_index = random.randint(0, len(self.game.ortadaki\_kartlar) - 1)

computer\_index = random.randint(0, len(self.game.ortadaki\_kartlar) - 1)

self.game.kart\_secim(player\_index, computer\_index)

self.update\_ui()

self.middle\_listbox.grid\_forget()

self.play\_button.config(text="Kart Oyna", command=self.play\_round)

else:

selected\_index = self.middle\_listbox.curselection()

if selected\_index:

player\_index = selected\_index[0]

computer\_index = random.randint(0, len(self.game.ortadaki\_kartlar) - 1)

self.game.kart\_secim(player\_index, computer\_index)

self.update\_ui()

if len(self.game.oyuncu.el) == 3 and self.middle\_listbox.winfo\_ismapped():

self.middle\_listbox.grid\_forget()

self.play\_button.config(text="Kart Oyna", command=self.play\_round)

elif self.game.tur\_sayisi == 3:

self.middle\_listbox.grid(row=1, column=1, padx=10, pady=10)

self.play\_button.config(text="Kart Seç", command=self.choose\_cards2)

self.update\_ui()

else:

self.show\_winner()

def choose\_cards2(self):

if self.game.mode == "computer\_vs\_computer":

while len(self.game.oyuncu.el) < 2:

player\_index = random.randint(0, len(self.game.ortadaki\_kartlar) - 1)

computer\_index = random.randint(0, len(self.game.ortadaki\_kartlar) - 1)

self.game.kart\_secim(player\_index, computer\_index)

self.update\_ui()

self.middle\_listbox.grid\_forget()

self.play\_button.config(text="Kart Oyna", command=self.play\_round2)

else:

selected\_index = self.middle\_listbox.curselection()

if selected\_index:

player\_index = selected\_index[0]

computer\_index = random.randint(0, len(self.game.ortadaki\_kartlar) - 1)

self.game.kart\_secim(player\_index, computer\_index)

self.update\_ui()

if len(self.game.oyuncu.el) == 2:

self.middle\_listbox.grid\_forget()

self.play\_button.config(text="Kart Oyna", command=self.play\_round2)

def play\_round(self):

if self.game.mode == "player\_vs\_computer":

selected\_index = self.player\_listbox.curselection()

if selected\_index:

player\_card, computer\_card = self.game.kart\_sec\_ve\_devam\_et(selected\_index[0])

messagebox.showinfo("Sonuç", f"Oyuncu Kartı: {player\_card} - Hasar: {player\_card.hasar}\nBilgisayar Kartı: {computer\_card} - Hasar: {computer\_card.hasar}")

self.update\_ui(show\_damage=True)

self.game.tur\_sayisi += 1

if not self.game.oyun\_devam\_ediyor():

self.choose\_cards()

else:

for \_ in range(len(self.game.oyuncu.el)):

player\_card, computer\_card = self.game.kart\_sec\_ve\_devam\_et()

messagebox.showinfo("Sonuç", f"Bilgisayar1 Kartı: {player\_card} - Hasar: {player\_card.hasar}\nBilgisayar2 Kartı: {computer\_card} - Hasar: {computer\_card.hasar}")

self.update\_ui(show\_damage=True)

self.game.tur\_sayisi += 1

if not self.game.oyun\_devam\_ediyor():

self.choose\_cards()

def play\_round2(self):

if self.game.mode == "player\_vs\_computer":

selected\_index = self.player\_listbox.curselection()

if selected\_index:

player\_card, computer\_card = self.game.kart\_sec\_ve\_devam\_et(selected\_index[0])

messagebox.showinfo("Sonuç", f"Oyuncu Kartı: {player\_card} - Hasar: {player\_card.hasar}\nBilgisayar Kartı: {computer\_card} - Hasar: {computer\_card.hasar}")

self.update\_ui(show\_damage=True)

self.game.tur\_sayisi += 1

if not self.game.oyun\_devam\_ediyor():

self.show\_winner()

else:

for \_ in range(len(self.game.oyuncu.el)):

player\_card, computer\_card = self.game.kart\_sec\_ve\_devam\_et()

messagebox.showinfo("Sonuç", f"Bilgisayar1 Kartı: {player\_card} - Hasar: {player\_card.hasar}\nBilgisayar2 Kartı: {computer\_card} - Hasar: {computer\_card.hasar}")

self.update\_ui(show\_damage=True)

self.game.tur\_sayisi += 1

if not self.game.oyun\_devam\_ediyor():

self.show\_winner()

def show\_winner(self):

winner\_message = self.game.kazanan()

messagebox.showinfo("Oyun Bitti", winner\_message)

self.game\_window.destroy()

if \_\_name\_\_ == "\_\_main\_\_":

root = tk.Tk()

app = PokemonGameGUI(root)

root.mainloop()