```
    from ensurepip import version
import string
import time
from tkinter import N
from turtle import delay
from unicodedata import name
import numpy as np
import sys
import random
import os

#Delay print intg

def delay_print(s):
    #print one character at a time

    for c in s:
        sys.stdout.write(c)
        sys.stdout.flush()
        time.sleep(0.05)

def clearConsole():
    command = 'clear'
    if os.name in ('nt', 'dos'):    # If Machine is running on Windows,
use cls
        command = 'cls'
    os.system(command)

#creating a class
class MonkeMon:
    def __init__(self, name, types, moves, EVs,
health='===================='):
#save variables as attributes for each MonkeMon object
        self.name = name
        self.types = types
        self.moves = moves
        self.attack = EVs['ATTACK']
        self.defense = EVs['DEFENSE']
        self.health = health
        self.bars = 20 #amount of health bars
```

```
    def fight(self,Monkemon2):
        #Allow two Monkemon to fight eachother.

        print("\n-----MONKEMON BATTLE-----")
        print(f'\n{self.name}')
        print("TYPE/", self.types)
        print("ATTACK/", self.attack)
        print("DEFENSE/", self.defense)
        print("LVL/", 3*(1+np.mean([self.attack,self.defense])))
        print("\nVS")
        print(f'\n{Monkemon2.name}')
        print("TYPE/", Monkemon2.types)
        print("ATTACK/", Monkemon2.attack)
        print("DEFENSE/", Monkemon2.defense)
        print("LVL/", 3*(1+np.mean([Monkemon2.attack,Monkemon2.defense])))

        time.sleep(4.5)



        version =
['Fire','Water','Grass','Fighter','Psychic','Steel','Dark','Normal','Guild
ed',]
        for i,k in enumerate(version):
            if self.types == k:
                if Monkemon2.types == k:
                    string_1_attack = '\nIt wasnt very effective...'
                    string_2_attack = '\nIt wasnt very effective...'


                #MonkeMon 2 is STRONG
                if Monkemon2.types == version[(i+1)%3]:
                    Monkemon2.attack *= 2
                    Monkemon2.defense *= 2
                    self.attack /= 2
                    self.defense /=2
                    string_1_attack = '\nIts not very effective...'
                    string_2_attack = '\nIts super effective!'
```

```
                if Monkemon2.types == version[(i+2)%3]:
                    self.attack *= 2
                    self.defense *= 2
                    Monkemon2.attack /= 2
                    Monkemon2.defense /= 2
                    string_1_attack = '\nIts super effective!'
                    string_2_attack = '\nIts not very effective...'

                else:
                    self.attack *= 1
                    self.defense *= 1
                    Monkemon2.attack /= 1
                    Monkemon2.defense /= 1
                    string_1_attack = f'\n {self.name} managed to land its
attack!'
                    string_2_attack = string_1_attack


        while(self.bars > 0) and (Monkemon2.bars > 0):
            print(f"\n{self.name}\tHLTH\t{self.health}")
            print(f"{Monkemon2.name}\tHLTH\t{Monkemon2.health}\n")

            print(f"Go {self.name}!")
            for i, x in enumerate(self.moves):
                print(f"{i+1}.", x)
            index = int(input('Pick a move: '))

            delay_print(f"\n{self.name} used {self.moves[index-1]}!")

            time.sleep(1)
            delay_print(string_1_attack)
            Monkemon2.bars -= self.attack
            Monkemon2.health = ""
            for j in range(int(Monkemon2.bars+.1*Monkemon2.defense)):
                Monkemon2.health += "="
```

```
            time.sleep(1)
            print(f"\n{self.name}\tHLTH\{self.health}")
            print(f"{Monkemon2.name}\tHLTH\{Monkemon2.health}\n")
            time.sleep(3)


            #check to see if monkemon fainted
            if Monkemon2.bars <= 0:
                delay_print(f"\n...{Monkemon2.name} fainted!")
                break


               #Monkemon2's turn
            print(f"Go {Monkemon2.name}!")
            for i, x in enumerate(Monkemon2.moves):
                print(f"{i+1}.", x)
            index = int(input('Pick a move: '))
            delay_print(f"\n{Monkemon2.name} used
{Monkemon2.moves[index-1]}!")
            time.sleep(1)
            delay_print(string_2_attack)

            #determine damage
            self.bars -= Monkemon2.attack
            self.health = ""
            for j in range(int(self.bars+.1*self.defense)):
                self.health += "="

            #check if fainted
            time.sleep(1)
            print(f"{Monkemon2.name}t\tHLTH\t{Monkemon2.health}")
            print(f"{Monkemon2.name}t\tHLTH\t{Monkemon2.health}\n")
            time.sleep(.5)
            if self.bars <= 0:
                delay_print("\n..." + self.name + ' fainted')
                break
```

```
if __name__ == '__main__':
    #Creating MonkeMon
    Golden_Monke = 'Golden Monke', 'Guilded', ['Daddys Credit card',
'Flex', 'Cash Money', 'Cash Rain'], {'ATTACK':50, 'DEFENSE':20}
    Lava_Monke = 'Lava Monke', 'Fire',['Flaming Burst', 'Tag', 'Scorching
Palm', 'Falling Fire'], {'ATTACK':20, 'DEFENSE':15}
    Daisy09 = 'Daisy09', 'Water', ['Song of Demons', 'Triangle Symbol',
'Haunting Tune', 'Straight-bone walk'], {'ATTACK':40, 'DEFENSE':25}
    The_Noob = 'The Noob', 'Figher', ['The Basics', 'Slow Walk',
'Standstill', 'Bounce'], {'ATTACK':25, 'DEFENSE':10}
    The_Tryhard = 'The Tryhard', 'Fighter', ['Wall Running', 'Wall
Climbing', 'Pure Sweat', 'Hospital Flip'], {'ATTACK':35, 'DEFENSE':20}
    The_Modder = 'The Modder', 'Psychic', ['Time Warp Clap', 'Flight',
'Long Arms Slap', 'Banning Bash'], {'ATTACK':45, 'DEFENSE':30}
    Jmancurly = 'Jmancurly','Steel', ['Monster Master', 'Little Kid
Legion', 'Lava form preperation', 'MONKE MONKE MONKE!'], {'ATTACK':35,
'DEFENSE':40}
    Toxic_Kid = 'Toxic Kid', 'Fire', ['Toxicity', 'Youre Bad!', 'Loser!',
'Racial Slurs'], {'ATTACK':45, 'DEFENSE':25}
    Discord_Moderator = 'Discord Moderator','Dark', ['Food', 'Diabetic
Destroyer', 'Get banned noob!', 'Obese Obliteration'], {'ATTACK':30,
'DEFENSE':30}
    Giga_Monke = 'Giga Monke', 'Steel', ['Lift up', 'Chad Skills',
'Respectfulness', 'Muscle Flex'], {'ATTACK':50, 'DEFENSE':50}
    The_Ghost_of_PBBV = 'PBBV', 'Dark', ['Mods', 'Siren Blare',
'Straight-bone walk', 'Red Eyed Monke'], {'ATTACK':60, 'DEFENSE':40}
    The_Brancher = 'The Brancher', 'Grass', ['Trees of speed', 'Branching
Bash', 'Leaping Leafswarm', 'Soaring Salad Strike'], {'ATTACK':45,
'DEFENSE':20}
    Tiktoker_Monke = 'Tiktoker Monke', 'Psychic', ['Fake Juke', 'Special
Effects', 'Shitty Song', 'Beg for Follows'], {'ATTACK':35, 'DEFENSE':40}
    The_Jefflings = 'The Jefflings', 'Dark', ['Jeffling Strike',
'Connected Crash', 'Familial Finale', 'Familial Feast'], {'ATTACK':40,
'DEFENSE':40}
```

```
    Background_Noise_Monke = 'Background Noise Monke', 'Fighting',
['Parents Arguing','Dog Barking', 'Toddler Scream', 'Controller Smack'],
{'ATTACK':35, 'DEFENSE':20}
    Broken_Controller_Monke = 'Broken Controller Monke', 'Normal', ['Wall
Punch','Pet Smack', 'Controller Catastrophe', 'Table Chipper'],
{'ATTACK':30, 'DEFENSE':30}
    The_Music_Monke = 'Music Monke', 'Gilded', ['Vibe', 'Earrape', 'Doom
Music', 'Sonic Blare'], {'ATTACK':30, 'DEFENSE':40}
    Loop_Monke = 'Monke Gif', 'Psychic', ['RGB Flash', 'Colorful Crassh',
'Looping Dance', 'Boogey of Reddit'], {'ATTACK':45, 'DEFENSE':55}
    Pimp_Chimped = 'Pimp Chimpen', 'Psychic', ['Swag', 'Rich Ravage',
'Cash Rain', 'Cash Money'], {'ATTACK': 30, 'DEFENSE': 20}
    Sad_Jeff = 'Sad Jeff', 'Water', ['Sadness', 'Depression','Tears of
Woe', 'Depressive Offense'], {'ATTACK':30, 'DEFENSE':60}
    Swamp_Monke = 'Swamp Monke', 'Dark', ['Mutated Mash', 'Bite', 'Feast
of Flesh', 'Seaweed Slash'], {'ATTACK':30, 'DEFENSE':45}
    Gen_Z = 'Gen Z Monke', 'Normal', ['Tiktok', 'FNAF Reference', 'Poppy
Playtime Reference', 'Floss'], {'ATTACK':38, 'DEFENSE':35}
    Millenial_Monke = 'Millenial Monke', 'Steel', ['Memes', '90s Kid',
'Pandemic Experience', 'Bad Choices'], {'ATTACK':25, 'DEFENSE':30}
    Boomer_Monke = 'Boomer Monke', 'Grass', ['Back in my day...', 'Hatred
of Spouce', 'Pollution', 'Even More Bad Choices'], {'ATTACK':50,
'DEFENSE':25}
    Big_Boomer_Monke = 'Boomer Monke EX', 'Grass', ['Generational God',
'Pollution Inventor', 'Recycling Inventor', 'Gen Z Shame'], {'ATTACK':100,
'DEFENSE':50}









    Monkemon_list = [Golden_Monke, Lava_Monke, Daisy09, The_Noob,
The_Noob, The_Tryhard, The_Modder, Jmancurly, Toxic_Kid,
Discord_Moderator, Giga_Monke, The_Brancher,
```

```
    Tiktoker_Monke, The_Jefflings, Background_Noise_Monke,
Broken_Controller_Monke, The_Music_Monke, Pimp_Chimped, Sad_Jeff,
Swamp_Monke, Gen_Z, Millenial_Monke, Boomer_Monke,
    Big_Boomer_Monke, The_Ghost_of_PBBV, Loop_Monke]
    print('Look at this example of Golden Monke at the top of the screen,
you can see its name, moves, and you can see its ATTACK and its DEFENSE.
Each MonkeMon has a different ATTACK, DEFENSE, and a different set of
moves. Pick whichever one you want by typing a number from 1-26.')
    print("Which Monkemon would you like to start with?\n")
    #print(list(Monkemon_list))
    count = 0

    print(Monkemon_list[0])
    # Iterates through the Monkemon list, and gives a UI of available
Monkemon
    for i in Monkemon_list:
        count += 1
        print(f'{count}. {i[0]}')
        print('\n')
    chosen = input('Type Monkemon Number \n\n---> ')
    # Passing required info into the Monkemon class
    starting_Monkemon =
MonkeMon(Monkemon_list[int(chosen)-1][0],Monkemon_list[int(chosen)-1][1],M
onkemon_list[int(chosen)-1][2],Monkemon_list[int(chosen)-1][3])
    #starting_Monkemon.fight(starting_Monkemon)
    # Generates a random Monkemon to fight

    def random_generator(Mon_list=Monkemon_list):
        random_chosen = random.choice(Mon_list)
        fighting_Monkemon =
MonkeMon(random_chosen[0],random_chosen[1],random_chosen[2],random_chosen[
3])
        return fighting_Monkemon
    while True:
        battle = starting_Monkemon.fight(random_generator())
        # Detects if user has lost, and end the game
        if battle == False:
            delay_print("\n\nYou have lost the game. Better luck next
time!")
            break
```

```
        delay_print("\n\nCongratulations, moving on to the next battle")
```

This is the code for MonkeMon Python, and is outdated. I will put the source code for MonkeMon 2D here.