VIRTUAL PET SIMULATOR

NAME: G. SAI RAHUL BHAT

Table of Contents

1. Introduction

This project is a command-line based Virtual Pet Simulator written in Python. It allows the user to feed, play, and care for a pet by interacting with menus. The project simulates a pet's needs by adjusting hunger and happiness levels and includes a game-over condition if the pet is neglected.

2. Objective

The objective of this project is to practice using Python basics like functions, loops, conditionals, and dictionaries by simulating a pet's behavior.

3. Features Implemented

- User can feed pet from available food items (inventory).
- User can play with pet using available toys.
- Pet has attributes: hunger and happiness (0 to 100).
- Time passes and affects pet's mood and hunger.
- Game ends if pet is too hungry or sad.
- Menu-driven interface with input handling.

4. Code Explanation

Below is the complete Python code for the virtual pet simulator:

```
def feed(pet, food_items):
    print("\nAvailable food:")
    for idx, (item, value) in enumerate(food_items.items(),
start=1):
        print(f"{idx}. {item} (-{value} hunger)")
    choice = input("Choose food to give: ")
    try:
        choice = int(choice)
        item = list(food_items.keys())[choice - 1]
        pet["hunger"] -= food_items[item]
        pet["happiness"] -= 2
        print(f"You fed your pet some {item}.")
    except:
        print("Invalid choice.")
    if pet["hunger"] < 0:
        pet["hunger"] = 0
    if pet["happiness"] < 0:
        pet["happiness"] = 0

def play(pet, toy_items):
    print("\nAvailable toys:")
    for idx, (toy, fun) in enumerate(toy_items.items(), start=1):
        print(f"{idx}. {toy} (+{fun} happiness)")
    choice = input("Choose a toy to play with: ")
    try:
        choice = int(choice)
        toy = list(toy_items.keys())[choice - 1]
        pet["happiness"] += toy_items[toy]
        pet["hunger"] += 5
        print(f"You played with your pet using the {toy}.")
    except:
        print("Invalid choice.")
    if pet["happiness"] > 100:
        pet["happiness"] = 100
    if pet["hunger"] > 100:
```

```
        pet["hunger"] = 100

def show_status(pet):
    print("\n--- Pet Status ---")
    print("Hunger:", pet["hunger"])
    print("Happiness:", pet["happiness"])
    if pet["hunger"] > 80:
        print("Your pet is really hungry!")
    if pet["happiness"] < 20:
        print("Your pet looks very sad...")

def time_goes_by(pet):
    pet["hunger"] += 5
    pet["happiness"] -= 5
    if pet["hunger"] > 100:
        pet["hunger"] = 100
    if pet["happiness"] < 0:
        pet["happiness"] = 0

def game_over(pet):
    if pet["hunger"] >= 100:
        print("\nYour pet is starving! GAME OVER.")
        return True
    elif pet["happiness"] <= 0:
        print("\nYour pet is too sad... GAME OVER.")
        return True
    return False

def main():
    name = input("Name your pet: ")
    pet = {
        "name": name,
        "hunger": 50,
        "happiness": 50
    }

    food_items = {
        "Biscuits": 10,
        "Meat": 20,
        "Carrot": 5
```

```
    }

    toy_items = {
        "Ball": 10,
        "Frisbee": 15,
        "Puzzle": 5
    }

    steps = 0
    while True:
        print(f"\nWhat do you want to do with {pet['name']}?")
        print("1. Feed")
        print("2. Play")
        print("3. Check Status")
        print("4. Quit")

        option = input("Enter choice: ")

        if option == "1":
            feed(pet, food_items)
        elif option == "2":
            play(pet, toy_items)
        elif option == "3":
            show_status(pet)
        elif option == "4":
            print("Goodbye!")
            break
        else:
            print("Invalid option.")

        steps += 1
        if steps % 3 == 0:
            print("\nTime is passing...")
            time_goes_by(pet)

        if game_over(pet):
            break

main()
```

5. Output Screenshots

```
Name your pet: Shero

What do you want to do with Shero?
1. Feed
2. Play
3. Check Status
4. Quit
Enter choice: 1

Available food:
1. Biscuits (-10 hunger)
2. Meat (-20 hunger)
3. Carrot (-5 hunger)
Choose food to give: 2
You fed your pet some Meat.

What do you want to do with Shero?
1. Feed
2. Play
3. Check Status
4. Quit
Enter choice: 2

Available toys:
1. Ball (+10 happiness)
2. Frisbee (+15 happiness)
3. Puzzle (+5 happiness)
Choose a toy to play with: 2
You played with your pet using the Frisbee.

What do you want to do with Shero?
1. Feed
2. Play
3. Check Status
4. Quit
Enter choice: 3

--- Pet Status ---
Hunger: 35
Happiness: 63

Time is passing...

What do you want to do with Shero?
1. Feed
2. Play
3. Check Status
4. Quit
Enter choice: 4
Goodbye!
```

Figure 1: Sample Output after feeding and playing with the pet (*"Tested on Google Colab / Python 3.10"*)