VERSION 1.6

START OF game.PY SCRIPT

from player import Player

player = Player('John')

player.location = player.map.get\_location("burrows\_home")

player.play()

END OF game.PY SCRIPT

START OF player.PY SCRIPT

from map import Map

class Player:

    def \_\_init\_\_(self, name):

        self.name = name

        self.health = 100

        self.map = Map()

        self.location = self.map.get\_location("burrows\_home")

        print(f"Player's location: {self.location}")

    def is\_alive(self):

        return self.health > 0

    def travel\_to(self, location\_name):

        new\_location = self.map.get\_location(location\_name)

        if new\_location:

            self.location = new\_location

            self.location.visit\_location(self)

        else:

            print(f"The location {location\_name} doesn't exist!")

    def defend(self, enemy):

        self.health += 10

        enemy.attack(self)

        self.health -= max(0, enemy.attack\_value - 10)

    def attack(self, enemy):

        enemy.health -= 10

    def play(self):

        action = ''

        while action != 'exit' and self.is\_alive():

            print(f"You are currently at {self.location.name}.")

            print("You can travel to the following locations:")

            for i, location in enumerate(self.location.visitable\_locations, 1):

                print(f"{i}. {location}")

            action = input("Enter the number of the location you want to travel to, or 'exit' to quit: ")

            if action.isdigit():

                action = int(action)

                if 1 <= action <= len(self.location.visitable\_locations):

                    self.travel\_to(self.location.visitable\_locations[action-1])

                else:

                    print("Invalid input, please enter a number.")

            elif action != 'exit':

                print("Invalid input. Please enter a number or 'exit'.")

        if not self.is\_alive():

            print(f"{self.name} has fallen. Game Over.")

# END OF player.PY SCRIPT

END OF player.PY SCRIPT

START OF location.PY SCRIPT

class Location:

    def \_\_init\_\_(self, name, visitable\_locations):

        self.name = name

        self.visitable\_locations = visitable\_locations

    def visit\_location(self, player=None):

        print(f"You arrived at {self.name}.")

def visit\_location(self, player=None):

    print(f"Travelling to {self.name}...")

    def view\_options(self):

        print(f"You are currently at {self.name}.")

        print("You can travel to the following locations:")

        for i, location in enumerate(self.visitable\_locations, start=1):

            print(f"{i}. {location}")

END OF location.PY SCRIPT

START OF angry\_cat.PY SCRIPT

# angry\_cat.py

from animal\_enemy import Animal

import random

class AngryCat(Animal):

    def \_\_init\_\_(self):

        super().\_\_init\_\_('Angry Cat')

        self.confidence = random.randint(1, 20)

        self.anger = random.randint(1, 20)

        self.fear = random.randint(1, 20)

        self.luck = random.randint(1, 20)

        self.health = self.level \* 5  # specific health calculation for Angry Cat

        self.attack\_value = self.level \* 2.5  # specific attack value calculation for Angry Cat

    def attack(self, player):

        super().attack(player)

        if self.health < 75:

            self.confidence -= 5

        if self.health < 50:

            self.confidence -= 10

END OF angry\_cat.PY SCRIPT

START OF dragon.PY SCRIPT

# dragon.py

from animal\_enemy import Animal

import random

class Dragon(Animal):

    def \_\_init\_\_(self):

        super().\_\_init\_\_('Dragon')

        self.confidence = random.randint(1, 20)

        self.anger = random.randint(1, 20)

        self.fear = random.randint(1, 20)

        self.luck = random.randint(1, 20)

        self.health = self.level \* 10  # specific health calculation for Dragon

        self.attack\_value = self.level \* 5  # specific attack value calculation for Dragon

    def attack(self, player):

        super().attack(player)

        if self.health < 75:

            self.confidence -= 5

        if self.health < 50:

            self.confidence -= 10

END OF dragon.PY SCRIPT

START OF tree.PY SCRIPT

class Tree:

    def encounter(self, player):

        print("You walk past a tree.")

        action = input("Do you want to 1. Continue exploring or 2. Return Home? ")

        if action == '1':

            return "TheBush"

        elif action == '2':

            return "PlayerHome"

END OF tree.PY SCRIPT

START OF map.PY SCRIPT

class Map:

    def \_\_init\_\_(self):

        self.locations = ["burrows\_home", "burrows\_neighbourhood", "the\_bush", "burrows\_park"]

    def get\_location(self, location\_name):

        if location\_name in self.locations:

            module = \_\_import\_\_(location\_name)

            location = getattr(module, location\_name)()

            print(f"Retrieved location: {location.name}")

            return location

        else:

            print(f"Location '{location\_name}' not found.")

            return None

END OF map.PY SCRIPT

START OF fight.PY SCRIPT

# fight.py

import random

# fight.py

import random

def fight(player, enemy):

    while player.is\_alive() and enemy.is\_alive():

        print(f"\nPlayer's Health: {player.health}")

        print(f"{enemy.name}'s Health: {enemy.health}\n")

        action = input("Do you want to attack, defend, ignore, flee, or view stats? ")

        if action.lower() == "attack":

            player.attack(enemy)

            print(f"You attacked the {enemy.name}!")

        elif action.lower() == "defend":

            player.defend(enemy)

            print("You defended yourself!")

        elif action.lower() == "ignore":

            enemy.attack(player)

            print(f"The {enemy.name} attacked you!")

        elif action.lower() == "flee":

            print("You managed to flee!")

            return

        elif action.lower() == "view stats":

            enemy.display\_stats()

        else:

            print("Invalid action, please try again.")

            continue

        # Enemy action decision based on emotions and luck

        luck\_draw = random.randint(1, 20)

        if luck\_draw > enemy.luck and enemy.confidence > 10:

            # High confidence leads to attack

            print(f"The {enemy.\_\_class\_\_.\_\_name\_\_} is feeling confident!")

            enemy.attack(player)

        elif enemy.fear > 10:

            # High fear leads to inaction

            print(f"The {enemy.\_\_class\_\_.\_\_name\_\_} is too scared to do anything!")

        elif enemy.anger > 10:

            # High anger leads to reckless attack

            print(f"The {enemy.\_\_class\_\_.\_\_name\_\_} is furious!")

            enemy.attack(player)

        else:

            # Default action is to attack

            enemy.attack(player)

        print(f"The {enemy.\_\_class\_\_.\_\_name\_\_} attacked you!")

    # Print final health status after battle

    print(f"\nFinal Health Status:")

    print(f"Player's Health: {player.health}")

    print(f"{enemy.\_\_class\_\_.\_\_name\_\_}'s Health: {enemy.health}\n")

    if player.is\_alive():

        print(f"You defeated the {enemy.\_\_class\_\_.\_\_name\_\_}!")

    else:

        print(f"You were defeated by the {enemy.\_\_class\_\_.\_\_name\_\_}!")

END OF fight.PY SCRIPT

def fight(player, enemy):

    while player.is\_alive() and enemy.is\_alive():

        action = input("Do you want to attack, defend, or flee? ")

        if action.lower() == "attack":

            player.attack(enemy)

            print(f"You attacked the {enemy.\_\_class\_\_.\_\_name\_\_}!")

        elif action.lower() == "defend":

            player.defend(enemy)

            print("You defended yourself!")

        elif action.lower() == "flee":

            print("You managed to flee!")

            return

        else:

            print("Invalid action, please try again.")

            continue

        if enemy.is\_alive():

            enemy.attack(player)

            print(f"The {enemy.\_\_class\_\_.\_\_name\_\_} attacked you!")

    if player.is\_alive():

        print(f"You defeated the {enemy.\_\_class\_\_.\_\_name\_\_}!")

    else:

        print(f"You were defeated by the {enemy.\_\_class\_\_.\_\_name\_\_}!")

START OF the\_bush.PY SCRIPT

# the\_bush.py

from location import Location

import random

from fight import fight

from dragon import Dragon

from angry\_cat import AngryCat

from tree import Tree

from animal\_enemy import Animal  # import the new Animal class

class the\_bush(Location):

    def \_\_init\_\_(self):

        super().\_\_init\_\_('the\_bush', ["burrows\_home"])

    def explore(self, player):

        encounter\_chance = random.randint(1, 3)

        if encounter\_chance == 1:

            print("You encountered a Dragon!")

            fight(player, Dragon())

        elif encounter\_chance == 2:

            print("You encountered an Angry Cat!")

            fight(player, AngryCat())

        else:

            print("You encountered a tree.")

            next\_location = Tree().encounter(player)

            self.visitable\_locations = [next\_location]

        action = input("Do you want to 1. Continue exploring or 2. Return Home? ")

        if action == '2':

            self.visitable\_locations = ["burrows\_home"]

        else:

            self.visitable\_locations = ["the\_bush"]

    def visit\_location(self, player=None):

        if player is not None:

            self.explore(player)

        else:

            super().visit\_location()

END OF the\_bush.PY SCRIPT

START OF animal\_enemy.PY SCRIPT

import random

class Animal:

    def \_\_init\_\_(self, name):

        self.name = name

        self.level = random.randint(1,10)

        self.skill\_points = self.level \* 3

        self.attack\_skill = random.randint(1, self.skill\_points)

        self.agility\_skill = self.skill\_points - self.attack\_skill

    def is\_alive(self):

        return self.health > 0

    def attack(self, player):

        player.health -= self.attack\_value

    def display\_stats(self):

        print(f"{self.name}'s stats:")

        print(f"Level: {self.level}")

        print(f"Health: {self.health}")

        print(f"Attack: {self.attack\_skill}")

        print(f"Agility: {self.agility\_skill}")

END OF animal\_enemy.PY SCRIPT

START OF burrows\_neighbourhood.PY SCRIPT

# burrows\_neighbourhood.py

from location import Location

from angry\_cat import AngryCat

class burrows\_neighbourhood(Location):

    def \_\_init\_\_(self):

        super().\_\_init\_\_('burrows\_neighbourhood', ["burrows\_home", "burrows\_park"])

    def visit\_location(self, player=None):

        if player is not None:

            print("You are wandering around the Burrows neighbourhood.")

        else:

            super().visit\_location()

END OF Burrows\_neighbourhood..PY SCRIPT

START OF burrows\_home.PY SCRIPT

# burrows\_home.py

from location import Location

class burrows\_home(Location):

    def \_\_init\_\_(self):

        super().\_\_init\_\_('burrows\_home', ["burrows\_neighbourhood", "the\_bush"])

    def visit\_location(self, player=None):

        if player is not None:

            print("Welcome back home!")

        else:

            super().visit\_location()

END OF burrows\_home.PY SCRIPT

START OF burrows\_park.PY SCRIPT

# burrows\_park.py

from location import Location

class burrows\_park(Location):

    def \_\_init\_\_(self):

        super().\_\_init\_\_('burrows\_park', ["burrows\_neighbourhood", "burrows\_home"])

    def visit\_location(self, player=None):

        if player is not None:

            print("You walk around the park!")

        else:

            super().visit\_location()

END OF burrows\_park.PY SCRIPT

START OF XXX.PY SCRIPT

END OF XXX.PY SCRIPT