**实验八**

**题目：**

|  |
| --- |
| 自由设计一款控制台小游戏，实现人机互动或用户之间互动玩耍。 |

**设计理念：**

|  |
| --- |
| 特点：   1. 使用了面向对象编程的核心概念 2. 包含随机事件系统 3. 加入了打字机效果增加趣味性 4. 使用emoji让显示更生动 5. 包含资源管理系统 6. 有随机生成的星球名称 7. 包含多种互动事件 8. 有简单的战斗系统 9. 使用了异常处理 10. 有状态管理系统 |

|  |
| --- |
| 游戏玩法：   * 探索随机生成的星球 * 收集资源和星币 * 与外星生物互动 * 对抗太空海盗 * 在星球间跃迁 * 管理生命值和资源 |

**完整代码：**

SpaceGame.java：

|  |
| --- |
| Java import java.util.Scanner;  public class SpaceGame {  public static void main(String[] args) {  GameManager gameManager = new GameManager();  gameManager.startGame();  } } |

GameManager.java：

|  |
| --- |
| Java import java.util.Scanner; import java.util.Random; import java.util.concurrent.TimeUnit;  public class GameManager {  private Player player;  private Scanner scanner;  private Random random;  private SpaceShip spaceShip;  private Planet currentPlanet;  private boolean isGameRunning;   public GameManager() {  scanner = new Scanner(System.*in*);  random = new Random();  isGameRunning = true;  }   public void startGame() {  showIntro();  initializeGame();  gameLoop();  }   private void showIntro() {  System.*out*.println("\n🌟 欢迎来到星际探险家 🌟");  System.*out*.println("在这个充满未知的宇宙中，你将成为一名勇敢的探险家...");  System.*out*.println("准备好开始你的星际冒险了吗？\n");  TypeWriter.*slowPrint*("加载宇宙数据中...", 50);  System.*out*.println();  }   private void initializeGame() {  System.*out*.print("请输入你的探险家名字: ");  String name = scanner.nextLine();  player = new Player(name);  spaceShip = new SpaceShip("宇宙漫游者号");  currentPlanet = PlanetGenerator.*generateRandomPlanet*();  }   private void gameLoop() {  while (isGameRunning && player.getHealth() > 0) {  showMainMenu();  int choice = getUserChoice();  processChoice(choice);  }  endGame();  }   private void showMainMenu() {  System.*out*.println("\n==========================");  System.*out*.println("当前位置: " + currentPlanet.getName());  System.*out*.println("==========================");  System.*out*.println("1. 探索当前星球 🔍");  System.*out*.println("2. 查看状态 📊");  System.*out*.println("3. 跃迁到新星球 🚀");  System.*out*.println("4. 使用物品 🎒");  System.*out*.println("5. 结束探索 🏁");  System.*out*.println("==========================");  }   private int getUserChoice() {  while (true) {  try {  System.*out*.print("请选择行动 (1-5): ");  return Integer.*parseInt*(scanner.nextLine());  } catch (NumberFormatException e) {  System.*out*.println("请输入有效的数字！");  }  }  }   private void processChoice(int choice) {  switch (choice) {  case 1:  exploreCurrentPlanet();  break;  case 2:  showStatus();  break;  case 3:  warpToNewPlanet();  break;  case 4:  useItem();  break;  case 5:  isGameRunning = false;  break;  default:  System.*out*.println("无效的选择！");  }  }   private void exploreCurrentPlanet() {  TypeWriter.*slowPrint*("正在探索" + currentPlanet.getName() + "...", 50);  System.*out*.println();   int eventType = random.nextInt(4);  switch (eventType) {  case 0:  findResources();  break;  case 1:  encounterAlien();  break;  case 2:  findTreasure();  break;  case 3:  spacePirates();  break;  }  }   private void findResources() {  int resources = random.nextInt(20) + 1;  System.*out*.println("💎 发现了" + resources + "单位的稀有矿物！");  player.addResources(resources);  }   private void encounterAlien() {  String[] aliens = {"👽 和善的格莱普人", "🤖 机械族商人", "👾 神秘的星云生物"};  String alien = aliens[random.nextInt(aliens.length)];  System.*out*.println("遭遇了" + alien + "!");   if (random.nextBoolean()) {  int reward = random.nextInt(15) + 5;  System.*out*.println("它送给你" + reward + "单位的能量晶体！");  player.addResources(reward);  } else {  System.*out*.println("它友好地向你挥手告别。");  }  }   private void findTreasure() {  String[] treasures = {"📦 神秘的空间宝箱", "🎁 远古文明遗物", "💫 星际宝藏"};  String treasure = treasures[random.nextInt(treasures.length)];  int value = random.nextInt(30) + 10;  System.*out*.println("发现了" + treasure + "! 价值" + value + "星币！");  player.addMoney(value);  }   private void spacePirates() {  System.*out*.println("⚠️ 遭遇太空海盗！");  int damage = random.nextInt(20) + 5;  player.takeDamage(damage);  System.*out*.println("在战斗中受到了" + damage + "点伤害！");   if (random.nextBoolean()) {  int loot = random.nextInt(25) + 5;  System.*out*.println("但是你成功击退了海盗，并缴获了" + loot + "星币！");  player.addMoney(loot);  }  }   private void showStatus() {  System.*out*.println("\n=== 探险家状态 ===");  System.*out*.println("👤 名字: " + player.getName());  System.*out*.println("❤️ 生命值: " + player.getHealth() + "/100");  System.*out*.println("💰 星币: " + player.getMoney());  System.*out*.println("💎 资源: " + player.getResources());  System.*out*.println("🚀 飞船: " + spaceShip.getName());  }   private void warpToNewPlanet() {  TypeWriter.*slowPrint*("启动跃迁引擎...", 50);  System.*out*.println();  currentPlanet = PlanetGenerator.*generateRandomPlanet*();  System.*out*.println("已到达新的星球: " + currentPlanet.getName());  }   private void useItem() {  if (player.getResources() >= 10) {  System.*out*.println("使用10单位资源制造医疗包...");  player.useResources(10);  player.heal(30);  System.*out*.println("恢复了30点生命值！");  } else {  System.*out*.println("资源不足！需要10单位资源。");  }  }   private void endGame() {  System.*out*.println("\n=== 探险结束 ===");  System.*out*.println("探险家: " + player.getName());  System.*out*.println("最终收集的星币: " + player.getMoney());  System.*out*.println("最终收集的资源: " + player.getResources());  System.*out*.println("感谢您的探索！");  } } |

Player.java：

|  |
| --- |
| Java public class Player {  private String name;  private int health;  private int money;  private int resources;   public Player(String name) {  this.name = name;  this.health = 100;  this.money = 0;  this.resources = 0;  }   public String getName() {  return name;  }   public int getHealth() {  return health;  }   public int getMoney() {  return money;  }   public int getResources() {  return resources;  }   public void takeDamage(int damage) {  health -= damage;  if (health < 0) health = 0;  }   public void heal(int amount) {  health += amount;  if (health > 100) health = 100;  }   public void addMoney(int amount) {  money += amount;  }   public void addResources(int amount) {  resources += amount;  }   public void useResources(int amount) {  resources -= amount;  } } |

SpaceShip.java:

|  |
| --- |
| Java public class SpaceShip {  private String name;  private int fuel;   public SpaceShip(String name) {  this.name = name;  this.fuel = 100;  }   public String getName() {  return name;  } } |

Planet.java:

|  |
| --- |
| Java public class Planet {  private String name;  private String type;   public Planet(String name, String type) {  this.name = name;  this.type = type;  }   public String getName() {  return name;  }   public String getType() {  return type;  } } |

PlanetGenerator.java：

|  |
| --- |
| Java import java.util.Random;  public class PlanetGenerator {  private static final String[] *PREFIXES* = {"阿尔法", "贝塔", "伽马", "德尔塔", "艾普斯", "奥米伽"};  private static final String[] *SUFFIXES* = {"星", "星球", "行星", "卫星"};  private static final String[] *TYPES* = {"岩石", "气态", "冰冻", "熔岩", "丛林"};  private static Random *random* = new Random();   public static Planet generateRandomPlanet() {  String name = *PREFIXES*[*random*.nextInt(*PREFIXES*.length)] + "-" +  (*random*.nextInt(999) + 1) + " " +  *SUFFIXES*[*random*.nextInt(*SUFFIXES*.length)];  String type = *TYPES*[*random*.nextInt(*TYPES*.length)];  return new Planet(name, type);  } } |

TypeWriter.java:

|  |
| --- |
| Java public class TypeWriter {  public static void slowPrint(String text, long millisPerChar) {  for (char c : text.toCharArray()) {  System.*out*.print(c);  try {  Thread.*sleep*(millisPerChar);  } catch (InterruptedException e) {  Thread.*currentThread*().interrupt();  }  }  System.*out*.println();  } } |

**代码运行截图：**













