Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра автоматизованих систем управління

Зображення, що містить текст

Автоматично згенерований опис

**Звiт**

до лабораторної роботи № 3

з дисципліни

# “ Прикладне програмування”

Виконав: студент групи ОІ-11СП

**Богдан Заяць**

Прийняв:

**МельникР.В.**

Львів – 2024

**Завдання:**

Напишіть свою реалізацію консольної гри “Битва дроїдів”

**Мінімальний набір вимог до програми:**

1. Створіть базовий клас Droid, від якого будуть походити інші підкласи (види дроїдів), які будуть відрізнятися різними характеристиками. Мінімальний набір характеристик: name, health, damage.

2. Додайте можливість різних видів бою: 1 на 1, або команда на команду.

3. Класи потрібно грамотно розкласти по пакетах.

4. У програмі має бути консольне меню. Мінімальний набір команд:

− створити дроїда (обраного виду);

− показати список створених дроїдів;

− запустити бій 1 на 1 (вибрати дроїдів, які будуть змагатися);

− запустити бій команда на команду (сформувати команди суперників з дроїдів, яких ви створили у першому пункті);

− записати проведений бій у файл;

− відтворити проведений бій зі збереженого файлу;

− вийти з програми.

**Додаткове завдання**

**Використайте свою фантазію, щоб гра вийшла цікавою для Вас і для глядача**.  
Жодних обмежень немає.

**Програма:**

Main.java

package edu.bogdan.zaiats.lab3.main;  
  
import edu.bogdan.zaiats.lab3.game.Team;  
import edu.bogdan.zaiats.lab3.game.factory.DroidTypes;  
  
import java.util.List;  
import java.util.Random;  
  
import static edu.bogdan.zaiats.lab3.game.factory.DroidFactory.*makeDroid*;  
import static java.util.stream.IntStream.*range*;  
  
class Main {  
 private final static int *ROUNDS* = 69;  
  
 public static void main(String[] args) {  
 var rng = new Random();  
 Team winner = null;  
 var team1 = new Team("Big Diapers", List.*of*(  
 *makeDroid*(DroidTypes.*BATTLE*),  
 *makeDroid*(DroidTypes.*BATTLE*),  
 *makeDroid*(DroidTypes.*IMPOSTER*),  
 *makeDroid*(DroidTypes.*SHAMAN*),  
 *makeDroid*(DroidTypes.*SHAMAN*),  
 *makeDroid*(DroidTypes.*BATTLE*),  
 *makeDroid*(DroidTypes.*BATTLE*),  
 *makeDroid*(DroidTypes.*BATTLE*)  
 ));  
 var team2 = new Team("Suckers", List.*of*(  
 *makeDroid*(DroidTypes.*BATTLE*),  
 *makeDroid*(DroidTypes.*BATTLE*),  
 *makeDroid*(DroidTypes.*SHAMAN*),  
 *makeDroid*(DroidTypes.*SHAMAN*),  
 *makeDroid*(DroidTypes.*IMPOSTER*),  
 *makeDroid*(DroidTypes.*BATTLE*),  
 *makeDroid*(DroidTypes.*BATTLE*),  
 *makeDroid*(DroidTypes.*BATTLE*)  
 ));  
  
 var rounds = 0;  
 for (var round : *range*(1, *ROUNDS* + 1).toArray()) {  
 rounds = round;  
 if (team1.count() == 0 || team2.count() == 0) {  
 break;  
 }  
 var firstIdx = rng.nextInt(team1.count());  
 var secondIdx = rng.nextInt(team2.count());  
 var first = team1.getMember(firstIdx);  
 var second = team2.getMember(secondIdx);  
 team1.interact(firstIdx, second);  
 team2.interact(secondIdx, first);  
  
 team1.interact(firstIdx, team1.getMember(rng.nextInt(team1.count())));  
 team2.interact(secondIdx, team2.getMember(rng.nextInt(team2.count())));  
  
 team1.cleanCorpses();  
 team2.cleanCorpses();  
  
  
 if (team1.count() == 0 && team2.count() > 0) {  
 winner = team2;  
 break;  
 } else if (team2.count() == 0 && team1.count() > 0) {  
 winner = team1;  
 break;  
 }  
 }  
  
  
 System.*out*.println("Total rounds: " + rounds);  
 if (winner != null) {  
 System.*out*.print("Winner: ");  
 *printTeam*(winner);  
 } else {  
 System.*out*.println("Draw");  
 *printTeam*(team1);  
 *printTeam*(team2);  
 }  
 }  
  
 private static void printTeam(Team team) {  
 System.*out*.println(team);  
 }  
}

Team.java

package edu.bogdan.zaiats.lab3.game;  
  
import edu.bogdan.zaiats.lab3.game.base.CanAttack;  
import edu.bogdan.zaiats.lab3.game.base.CanHeal;  
import edu.bogdan.zaiats.lab3.game.base.Droid;  
import edu.bogdan.zaiats.lab3.game.base.Vulnerable;  
import edu.bogdan.zaiats.lab3.game.droids.Imposter;  
  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Random;  
  
import static java.util.stream.IntStream.*range*;  
  
public class Team {  
 private final String name;  
 private ArrayList<Droid> team;  
 private final Random random = new Random();  
  
 public Team(String name, List<Droid> team) {  
 this.team = new ArrayList<>(team);  
 this.name = name;  
 }  
  
 public int count() {  
 return team.size();  
 }  
  
 public Droid getMember(int index) {  
 return team.get(index);  
 }  
  
 public void interact(int idx, Droid second) {  
 var first = team.get(idx);  
 if (second == null) {  
 return;  
 }  
  
 boolean areTeammates = team.contains(second);  
  
 switch (first) {  
 case Imposter sus -> {  
 if (areTeammates) {  
 var chance25 = random.nextInt(4) == 0;  
 if (chance25) {  
 sus.attack(second);  
 } else {  
 sus.heal(second);  
 }  
 } else {  
 sus.attack(second);  
 }  
 }  
 case CanAttack attacker -> {  
 if (!areTeammates) {  
 attacker.attack(second);  
 }  
 }  
 case CanHeal shaman -> {  
 if (areTeammates) {  
 shaman.heal(second);  
 }  
 }  
 default -> {  
 }  
 }  
 }  
  
 public String getName() {  
 return this.name;  
 }  
  
 @Override  
 public String toString(){  
 return this.getName() + "\n" + this.team.toString()  
 .replace("},","},\n")  
 .replace('[',' ')  
 .replace(']',' ');  
 }  
  
 public void cleanCorpses() {  
 this.team.removeIf(droid -> droid.getHealth() <= 0 );  
 }  
}

imposter.kt

package edu.bogdan.zaiats.lab3.game.droids  
  
import edu.bogdan.zaiats.lab3.game.base.CanAttack  
import edu.bogdan.zaiats.lab3.game.base.CanHeal  
import edu.bogdan.zaiats.lab3.game.base.Droid  
import edu.bogdan.zaiats.lab3.game.base.Vulnerable  
  
class Imposter(private var health: Int,private val healthBoost: Int) : Droid(health), CanAttack, CanHeal {  
 override fun attack(target: Vulnerable?) {  
 if (this == target){  
 return;  
 }  
 val damage = (target?.*health* ?: 0) / (1..4).*random*()  
 target?.receiveDamage(damage)  
 }  
  
 override fun heal(target: Vulnerable?) {  
 if (this == target){  
 return;  
 }  
 var health = 45 / (1..45).*random*()  
 target?.receiveHealth(health)  
  
 }  
  
 override fun toString(): String {  
 return "Shaman{" +  
 "health=" + health +  
 '}'  
 }  
}

BattleDroid.java

package edu.bogdan.zaiats.lab3.game.droids;  
  
import edu.bogdan.zaiats.lab3.game.base.CanAttack;  
import edu.bogdan.zaiats.lab3.game.base.Droid;  
import edu.bogdan.zaiats.lab3.game.base.Vulnerable;  
  
public class BattleDroid extends Droid implements Vulnerable, CanAttack {  
 private final int damage;  
  
 public BattleDroid(int health, int damage) {  
 super(health);  
 this.damage = damage;  
 }  
  
 @Override  
 public void attack(Vulnerable target) {  
 target.receiveDamage(this.damage);  
 }  
  
 @Override  
 public String toString() {  
 return "BattleDroid{" +  
 "damage=" + damage +  
 ", health=" + health +  
 '}';  
 }  
}

Shanam.java

package edu.bogdan.zaiats.lab3.game.droids;  
  
import edu.bogdan.zaiats.lab3.game.base.CanHeal;  
import edu.bogdan.zaiats.lab3.game.base.Droid;  
import edu.bogdan.zaiats.lab3.game.base.Vulnerable;  
  
public class Shaman extends Droid implements CanHeal {  
 private final int healthBoost;  
  
 public Shaman(int health, int health\_boost) {  
 super(health);  
 this.healthBoost = health\_boost;  
 }  
  
 @Override  
 public void heal(Vulnerable target) {  
 if (this.equals(target)){  
 return;  
 }  
 target.receiveHealth(healthBoost);  
 }  
  
  
 @Override  
 public String toString() {  
 return "Shaman{" +  
 "health=" + health +  
 '}';  
 }  
}

interfaces

package edu.bogdan.zaiats.lab3.game.base;  
  
public interface CanAttack {  
 void attack(Vulnerable target);  
}

package edu.bogdan.zaiats.lab3.game.base;  
  
public interface CanHeal {  
 void heal(Vulnerable target);  
}

package edu.bogdan.zaiats.lab3.game.base;  
  
public interface Vulnerable {  
 void receiveDamage(int damage);  
  
 void receiveHealth(int health);  
  
 int getHealth();  
}

Droid.java

package edu.bogdan.zaiats.lab3.game.base;  
  
public class Droid implements Vulnerable{  
 protected final int maxHealth;  
 protected int health;  
  
  
 public Droid(int maxHealth) {  
 this.maxHealth = maxHealth;  
 this.health = maxHealth;  
 }  
  
 @Override  
 public void receiveDamage(int damage) {  
 this.health -= damage;  
 if (this.health < 0) {  
 this.health = 0;  
 }  
 }  
  
 @Override  
 public void receiveHealth(int health) {  
 this.health += health;  
  
 if ( this.health > this.maxHealth){  
 this.health = maxHealth;  
 }  
 }  
  
 @Override  
 public int getHealth() {  
 return this.health;  
 }  
}

DroidTypes.java

package edu.bogdan.zaiats.lab3.game.factory;  
  
public enum DroidTypes {  
 *BATTLE*,  
 *SHAMAN*,  
 *IMPOSTER*}

DroidFactory.java

package edu.bogdan.zaiats.lab3.game.factory;  
  
import edu.bogdan.zaiats.lab3.game.base.Droid;  
import edu.bogdan.zaiats.lab3.game.droids.BattleDroid;  
import edu.bogdan.zaiats.lab3.game.droids.Imposter;  
import edu.bogdan.zaiats.lab3.game.droids.Shaman;  
  
public class DroidFactory {  
 public static Droid makeDroid(DroidTypes type) {  
 return switch (type) {  
 case *BATTLE* -> new BattleDroid(45, 20);  
 case *SHAMAN* -> new Shaman(45, 15);  
 case *IMPOSTER* -> new Imposter(64,15);  
 };  
 }  
}

**Результати:**

**1)**

Total rounds: 34

Winner: Suckers

BattleDroid{damage=20, health=6},

Shaman{health=64}

**2)**

Total rounds: 30

Winner: Suckers

Shaman{health=19},

BattleDroid{damage=20, health=24},

BattleDroid{damage=20, health=12}

**3)**

Total rounds: 26

Winner: Suckers

Shaman{health=17},

Shaman{health=64}

**4)**

Total rounds: 30

Winner: Big Diapers

BattleDroid{damage=20, health=3},

Shaman{health=64},

Shaman{health=45}

**5)**

Total rounds: 39

Draw

Big Diapers

Suckers

**Висновок:**

В цій лабораторній роботі я вивчив основи Java.