МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ КОМП’ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра комп’ютерних наук

**КОМПЛЕКСНО-ПРАКТИЧНЕ**

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

**з навчальної дисципліни “Конструювання програмного забезпечення”**

**На тему: «** **Калькулятор боїв до онлайн гри Consulwar»**

**Виконав:** ст.гр. ПЗС-31

Ліхота О.І.

**Перевірив:** Сирник О.Й.

ТНЕУ– 2015

**План**:

Вступ

1. Суть поставленої задачі
2. Гра та особливості, програма
3. Висновок

Список використаних джерел

**Вступ**

Всі ми хоч раз в житті, грали ігри, ми прагнемо бути найкращими, приділяємо час і частині з нас це виходить, іншим же, ні. Часто в думках кожного з нас появляється ідея «Класно би було, якщо б програма могла грати замість мене, я б міг стати найкращим без лишніх зусиль». Завдання для мене було написати калькулятор, завдяки якому можна би було обраховувати результати бою, яка з команд виграє, а яка програє, які втрати будуть. Пізніше на зібраних знаннях було написано скріпт, який грає сам.

**Суть поставленої задачі**

Програмовані калькулятори мають можливість вводити і виконувати програми користувача. Мають велику кількість регістрів пам'яті (10 і більше). По функціональності наближаються до найпростіших комп'ютерів. В залежності від моделі програмовані калькулятори мають кілька типів вбудованих мов програмування: найпростіша (запам'ятовує тільки натискання клавіш без виведення на екран); така, що відображає коди команд (зазвичай використовуючи зворотний польський запис), Бейсік або власна мова програмування, наближена до Бейсік.

Завданням було написання калькулятора для онлайн гри Consulwar, щоб легко можна було прораховувати дані бою та не витрачати лишніх ресурсів, як у грі, так і свого особистого часу.

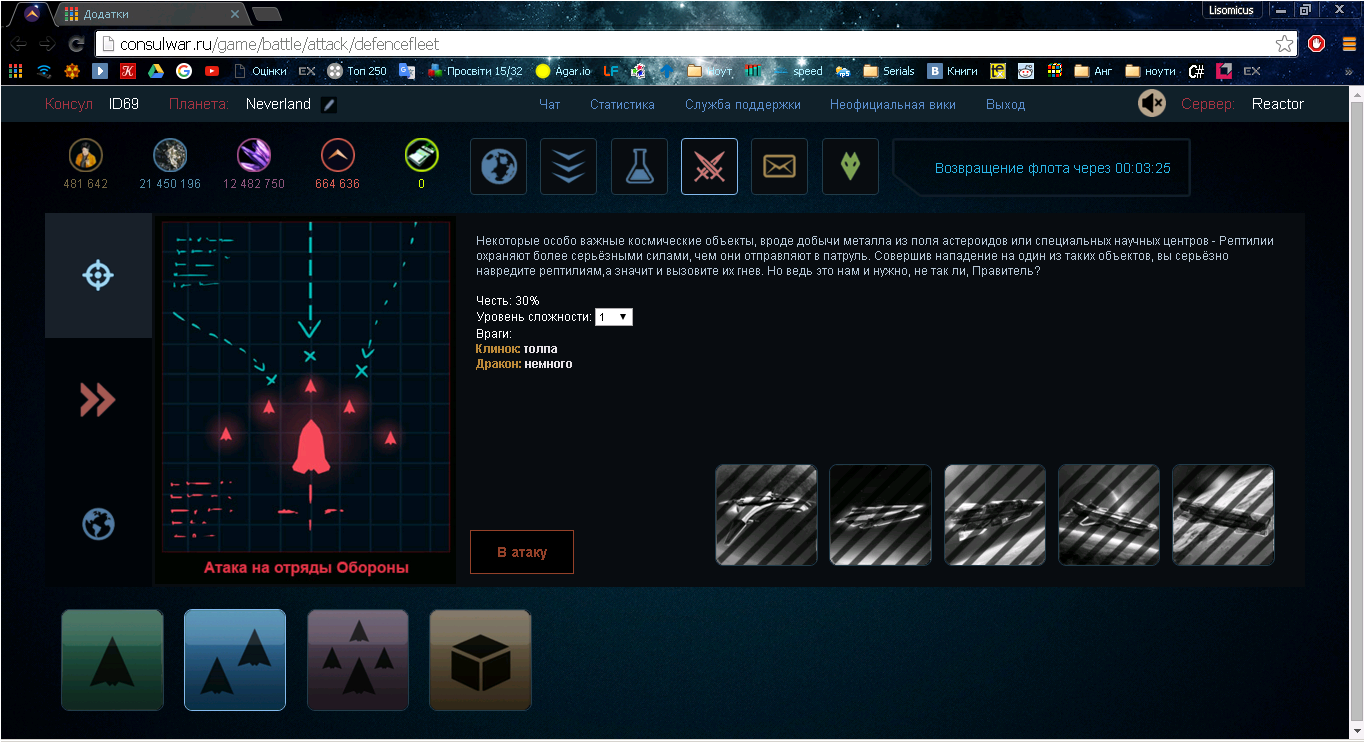
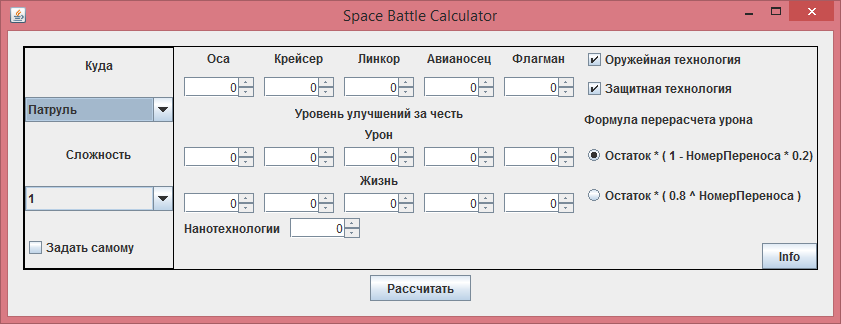


Рисунок 1.-Головна сторінка гри

  
 Рисунок 2.-Головне вікно калькулятора

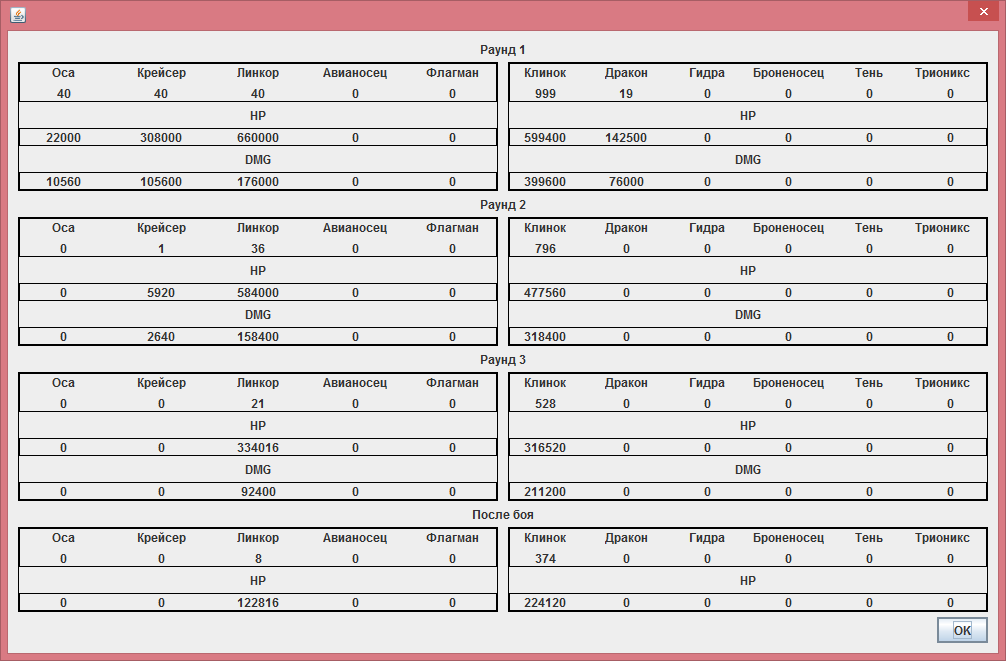


Рисунок 3.-Обрахунок втрат сторін

**Гра та особливості, програма**

Суть гри полягає у тому, що вам потрібно свати Землю, від загарбників, а саме від Рептилій, посилаючи війська на Землю. Ми граємо за людей, бої проводяться автоматично.

В коді програми потрібно було описати кожного юніта, його запас життя, його силу нападу. Обрахувавши остаточні дані, вивести результат бою, враховуючи різноманітні бонуси та технології.

Код є громіздким, тому розглянемо лише кілька класів:

Клас ShipGroup:

public abstract class ShipGroup {  
  
 private Integer attack;  
 private Integer health;  
 private Integer count;  
 private List<Class<? extends ShipGroup>> priorities;  
  
 private Integer currentAttack = 0;  
 private Integer currentHealth = 0;  
 private Boolean isDead = false;  
  
 protected ShipGroup(Integer attack, Integer health, List<Class<? extends ShipGroup>> priorities, Integer count) {  
 this.attack = attack;  
 this.health = health;  
 this.count = count;  
 this.priorities = priorities;  
 currentAttack = count \* getAttack();  
 currentHealth = count \* getHealth();  
 }  
  
 public Integer getAttack() {  
 return attack;  
 }  
  
 public Integer getHealth() {  
 return health;  
 }  
  
 public Integer getCount() {  
 return count;  
 }  
  
 public List<Class<? extends ShipGroup>> getPriorities() {  
 return priorities;  
 }  
  
 public Integer getCurrentAttack() {  
 return currentAttack;  
 }  
  
 public void setCurrentAttack(Integer currentAttack) {  
 this.currentAttack = currentAttack;  
 }  
  
 public void recalculateCurrentAttack() {  
 currentAttack = count \* getAttack();  
 }  
  
 public Integer getCurrentHealth() {  
 return currentHealth;  
 }  
  
 public void setCurrentHealth(Integer currentHealth) {  
 this.currentHealth = currentHealth;  
  
 count = currentHealth / getHealth();  
  
 if (currentHealth % getHealth() > 0) {  
 count++;  
 }  
 }  
  
 public void kill() {  
 currentHealth = 0;  
 isDead = true;  
 count = 0;  
 }  
  
 public boolean isDead() {  
 return isDead;  
 }  
}

КласBattleship:

public class Battleship extends HumanShipGroup {  
 private static final int *ATTACK* = 4000;  
 private static final int *HEALTH* = 15000;  
 private static final int *HONOR\_ATTACK\_BONUS\_PER\_LVL* = 20;  
 private static final int *HONOR\_HEALTH\_BONUS\_PER\_LVL* = 40;  
  
 public static Integer *honorAttackLvl* = 0;  
 public static Integer *honorHealthLvl* = 0;  
  
 private static final List<Class<? extends ShipGroup>> *priorities* = new ArrayList<>(6);  
 static {  
 *priorities*.add(Hydra.class);  
 *priorities*.add(Armadillo.class);  
 *priorities*.add(Dragon.class);  
  
 *priorities*.add(Blade.class);  
 *priorities*.add(Shadow.class);  
 *priorities*.add(Trioniks.class);  
 }  
 public Battleship(int count) {  
 super(*ATTACK*, *HEALTH*, *priorities*, count);  
 }  
  
  
 @Override  
 public Integer getHonorAttack() {  
 return *HONOR\_ATTACK\_BONUS\_PER\_LVL* \* *honorAttackLvl*;  
 }  
  
 @Override  
 public Integer getHonorHealth() {  
 return *HONOR\_HEALTH\_BONUS\_PER\_LVL* \* *honorHealthLvl*;  
 }  
}

Клас Armadillo:

public class Armadillo extends ShipGroup {  
 private static final int *ATTACK* = 1000;  
 private static final int *HEALTH* = 40000;  
  
 private static final List<Class<? extends ShipGroup>> *priorities* = new ArrayList<>(5);  
 static {  
 *priorities*.add(Wasp.class);  
 *priorities*.add(Cruiser.class);  
 *priorities*.add(Battleship.class);  
  
 *priorities*.add(Carrier.class);  
 *priorities*.add(Flagship.class);  
 }  
  
 public Armadillo(int count) {  
 super(*ATTACK*, *HEALTH*, *priorities*, count);  
 }  
}

Клас Оса він же Wasp:

public class Wasp extends HumanShipGroup {  
 private static final int *ATTACK* = 240;  
 private static final int *HEALTH* = 500;  
 private static final int *HONOR\_ATTACK\_BONUS\_PER\_LVL* = 5;  
 private static final int *HONOR\_HEALTH\_BONUS\_PER\_LVL* = 5;  
  
 public static Integer *honorAttackLvl* = 0;  
 public static Integer *honorHealthLvl* = 0;  
  
­­­ private static final List<Class<? extends ShipGroup>> *priorities* = new ArrayList<>(6);  
 static {  
 *priorities*.add(Blade.class);  
 *priorities*.add(Dragon.class);  
 *priorities*.add(Armadillo.class);  
   
 *priorities*.add(Hydra.class);  
 *priorities*.add(Shadow.class);  
 *priorities*.add(Trioniks.class);  
 }  
  
 public Wasp(int count) {  
 super(*ATTACK*, *HEALTH*, *priorities*, count);  
 }  
  
 @Override  
 public Integer getHonorAttack() {  
 return *HONOR\_ATTACK\_BONUS\_PER\_LVL* \* *honorAttackLvl*;  
 }  
  
 @Override  
 public Integer getHonorHealth() {  
 return *HONOR\_HEALTH\_BONUS\_PER\_LVL* \* *honorHealthLvl*;  
 }  
}

Весь проект я добавлю додатком, там можна буде подивитись інші класи.

**Висновок**

Використовуючи стандартні бібліотеки що надаються такими IDE як NetBeans чи IntelliJIdea ми можемо з легкістю відправляти різного роду запити (POST, GET та інші). Це дозволяє нам навіть імітувати роботу браузера або навіть написати свій інтернет-браузер.

**Список використаних джерел**:

1. [Електронний ресурс]. – Режим доступу: http://www.msdn.com.
2. [Електронний ресурс]. – Режим доступу: http://www.codenet.ru
3. [Електронний ресурс]. – Режим достпу: http://professorweb.ru/
4. [Електронний ресурс]. – Режим достпу: http://uk.wikipedia.org/
5. С#: Пер. С англ./Дейтел Х.,Дейтел П., Листфилд Дж., Нието Т., Йегер Ш., Златкина М. – СПб.: БХВ-Петербург, 2006. – 1056 с.:ил.
6. Джеффри Рихтер - CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке С#. Мастер-класс./Пер. с англ. — М.: Издательство «Русская Редакция»; СПб.: Питер, 2007. — 656 стр. : ил.
7. Кристиан Нейгел, Билл Ивьен, Джей Глинн, Карли Уотсон, Морган Скиннер - С# 2008 и платформа .Net 3.5 для профессионалов.
8. [Електронний ресурус]. – Режим доступу: http://www.codeabbey.com/