МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ КОМП’ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра економічної кібернетики

та інформатики

Комплексне практичне індивідуальне завдання

з дисципліни

«Конструювання програмного забезпечення»

на тему: «Реалізація гри “Змійка” на Java»

Виконав:

студент групи ПЗС-32

Кибук Д.О.

Перевірив:

Сирник О. Й.

Тернопіль 2015

ЗМІСТ

Введення

1.Історія створення гри «змійка»;

2.Цілі та завдання;

3.Теоритичні відомості;

4.Лістинг програми;

Висновок

Список використаної літератури

**Введення**

Snake (Пітон, Удав, Змійка та ін.) - комп'ютерна гра, що виникла в середині або в кінці 1970-х.

Гравець управляє довгою, тонкою істотою, що нагадує змію, яка повзає по площині (як правило, обмеженою стінками), збираючи їжу (або інші предмети), уникаючи зіткнення з власного хвоста і країв ігрового поля. У деяких варіантах на полі присутні додаткові перешкоди. Кожен раз, коли змія з'їдає шматок їжі, вона стає довшою, що поступово ускладнює гру. Гравець управляє напрямком руху голови змії (зазвичай 4 напрямки: вгору, вниз, вліво, вправо), а хвіст змії рухається слідом. Гравець не може зупинити рух змії.

1. Історія створення гри «змійка»

Першою грою цього плану був ігровий автомат Hustle, випущений фірмою Gremlin в 1977 році. Одна з перших реалізацій (можливо, перша) на домашньому комп'ютері була зроблена в 1979 році на комп'ютері TRS-80, автор F. Seger (Німеччина). Незабаром після неї вийшла версія для Commodore VIC-20, під назвою Worms. Мікрокомп'ютерна версія гри Hustle була випущена Milton Bradley для комп'ютера TI-99 / 4A в 1980 році.

У число відомих варіантів Пітона входить гра Nibbles, яка деякий час входила в комплект MS-DOS.

Тривимірна версія Пітона існує для телефону Nokia N70 і наступних телефонів Nokia.

2 Цілі і завдання

Мета: навчитися створювати аплет

Завдання: Створити аплет для реалізації нижченаведених завдань. При написанні програми обов'язково використовувати методи багатопотокового програмування. Гра «Змійка». Управління клавішами. Спочатку «змійка» складається з двох ланок, на екрані з'являються предмети їжі. При «поїданні» з «змійка» у довжину збільшується на одну ланку. Закінчення гри - торкання межі екрану або «укус» себе самої. Врахувати проходження всього шляху всіма ланками «змійки».

3.Теоритичні відомості

Игра. На початку гри виводиться вікно, в якому я помістив посилання для початку гри. Пропонується на вибір 4 різних типи гри. Відмінність полягає у швидкості. Після кліка по ній починається процес гри. На монітор виводиться прямокутник, в якому і відбувається весь основний процес гри. Внизу даного прямокутника ми можемо побачити Кількість набраних очок. Вони відображають успішність проходження гри. Чим більше швидкість гри, тим швидше набираються очка. Об'єктами в моїй розробці служать ланки змійки. Для активної адаптації і сприйняття, і реалізації програми я використовував графічні об'єкти. Вони з'являються випадковим чином на екрані. Якщо Ви врізаєтеся змійкою в стінку, або в саму себе, то гра автоматично закінчується з виведенням статистичних даних.

Клас (class) Будівельний блок Java-додатків, клас є групою елементів даних (членів), з якими пов'язані програми (методи), що виконують операції над цими даними. Пакет PL / SQL в чомусь схожий на клас, за винятком того, що ви не можете створювати окремий об'єкт з пакету, тому він більше схожий на статичний Java-клас.

Типи даних (datatypes) Все в Java визначається через клас, крім елементарних типів даних. Так, String є класом, тому коли ви оголошуєте змінну типу String, ви створюєте об'єкт, заснований на цьому класі. Елементарні типи даних Java включають чотири типи цілих, а також boolean, char, double, і float.

Реалізація (instantiation) Для створення приватного примірника з більш загальної структури, потрібно оголосити об'єкт як екземпляр або реалізацію класу. У мові PL / SQL ви можете оголосити тип запис (TYPE) і потім оголосити екземпляр запису, заснований на цьому типі.

Член (member) Клас може містити будь-яку кількість змінних, методів та інших класів (класів таких як в Java 1.1), які всі разом називаються членами класу. Клас може бути визначений усередині своїх членів, які містять інформацію про клас в цілому (статичні члени) або про реалізаціях об'єктів цього класу. Члени класу схожі на змінні рівня пакета в PL / SQL. Проте, кожного разу коли ви створюєте об'єкт з класу, ви отримуєте новий набір членів. Пакет має тільки одну реалізацію своїх даних під час сесії бази даних Oracle.

Метод (method) Метод - це іменована група операторів мови програмування Java, які пов'язані з певним класом і які можуть бути викликані для об'єктів, що є реалізаціями класу. У Java ви можете визначити статичні методи (також звані методами класу), які можуть бути доступні безпосередньо з класу без створення об'єкта. Метод main також завжди оголошується як статичний.

Об'єкт (object) Відомий також як екземпляр, об'єкт - це елемент виконуваного коду, який особливим чином реалізує основну структуру класу. Клас є загальним шаблоном типів (sorts), що надає структуру і правила для цієї структури. У загальному значенні об'єкт - це елемент даних, яким ви маніпулюєте у вашому додатку.

Пакет (package) Пакет - це елемент мови Java, використовуваний для групування споріднених класів під загальним ім'ям. Всі пакети в Java API згруповані в один пакет, званий java. Всередині пакету java знаходяться інші пакети, згруповані за функціональним можливостям, наприклад, пакет java.util містить основні класи утиліт.

4.Лістинг програми

package snake;

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

public class snake extends Applet implements Runnable

{

Image dot[]=new Image[400];

Image back;

Image     offI;

Graphics  offG;

int x[]= new int[400];

int y[]= new int[400];

int rtemp=1;

int game=1;

int level;

int z;

int n;

int count=0;

int score=0;

int add=1;

Button b= new Button("Beginner");

Button i= new Button("Intermediate");

Button p= new Button("Professional");

Button X= new Button("Xtreamest");

String stemp;

String s;

String t;

boolean go[]=new boolean[400];

boolean left=false;

boolean right=false;

boolean up=false;

boolean down=false;

boolean started=false ;

boolean me=false;

Thread setTime;

public void init()

{

add(b);

add(i);

add(p);

add(X);

setBackground(Color.black);

back = getImage(getCodeBase(), "screan.gif");

for (z=0 ; z < 400 ; z++){dot[z] = getImage(getCodeBase(), "dot.gif");  }

}

public void update(Graphics g)

{

Dimension d = this.size();

if(offI == null)

{

offI = createImage(d.width, d.height);

offG = offI.getGraphics();

}

offG.clearRect(0, 0, d.width, d.height);

paint(offG);

g.drawImage(offI, 0, 0, null);

}

public void paint(Graphics g)

{

g.drawImage(back, 0, 0, this);

g.setColor(Color.white);

if(started)

{

g.setFont(new Font("Verdana", 1, 12));

t = "Score "+score+"";

g.drawString(t, 75, 220);

}

if(game==1)

{

g.setFont(new Font("Verdana", 1, 13));

s = "Select Mode";

g.drawString(s, 65, 30);

b.move(75, 50);

i.move(68, 90);

p.move(68, 130);

X.move(73, 170);

}

if((game==2)||(game==3))

{

if(!started)

{

g.setFont(new Font("Verdana", 1, 11));

t = "Use the key board arrows to move!";

g.drawString(t, 5, 215);

}

for (z=0 ; z <= n ; z++){  g.drawImage(dot[z],x[z],y[z],this);  }

me=true;

}

if(!me)

{

g.setFont(new Font("Verdana", 1, 11));

t = "by Cumka, especially for Tywkanchik";

g.drawString(t, 5, 215);

}

if(game==3)

{

g.setFont(new Font("Verdana", 1, 13));

s="Game Over";

g.drawString(s, 65, 60);

}

}

public void run()

{

for(z=4 ;z <400 ; z++) { go[z]=false;}

for(z=0 ; z<4 ; z++) { go[z]=true;x[z]=91;y[z]=91;}

n=3;

game=2;

score=0;

b.move(70, -100);

i.move(70, -100);

p.move(70, -100);

X.move(70, -100);

left=false;

right=false;

up=false;

down=false;

locateRandom(4);

while(true)

{

if (game==2)

{

if ((x[0]==x[n])&&(y[0]==y[n])){go[n]=true;locateRandom((n+1));score+=add; }

for(z = 399 ; z > 0 ; z--)

{

if (go[z])

{

x[z] = x[(z-1)]; y[z] = y[(z-1)];

if ((z>4)&&(x[0]==x[z])&&(y[0]==y[z])){ game=3; }

}

}

if(left){ x[0]-=10; }

if(right){ x[0]+=10; }

if(up){ y[0]-=10; }

if(down){ y[0]+=10; }

}

if(y[0]>191){y[0]=191;game=3;}

if(y[0]<1){y[0]=1;game=3;}

if(x[0]>191){x[0]=191;game=3;}

if(x[0]<1){x[0]=1;game=3;}

if (game==3)

{

if (count <(1500/level)) { count++; } else { count=0;game=1;repaint();setTime.stop(); }

}

repaint();

try{setTime.sleep(level);}

catch(InterruptedException e){}

}

}

public void locateRandom(int turn)

{

rtemp=(int)(Math.random()\*20);

x[turn]=((rtemp\*10)+1) ;

rtemp=(int)(Math.random()\*20);

y[turn]=((rtemp\*10)+1);

n++;

}

public boolean keyDown(Event e, int key)

{

if ((key == Event.LEFT) &&(!right)){left = true; up = false; down = false;if(!started)started=true;}

if ((key == Event.RIGHT) && (!left)){right = true; up = false; down = false;if(!started)started=true;}

if ((key == Event.UP) && (!down)){ up = true; right = false; left = false;if(!started)started=true;}

if ((key == Event.DOWN) && (!up)){down = true; right = false; left = false;if(!started)started=true;}

return true;

}

public boolean action(Event event, Object obj)

{

stemp = (String) obj;

if(stemp.equals("Beginner"))

{

add=2;

level=100;

setTime = new Thread(this);

setTime.start();

return true;

}

if(stemp.equals("Intermediate"))

{

add=5;

level=70;

setTime = new Thread(this);

setTime.start();

return true;

}

if(stemp.equals("Professional"))

{

add=10;

level=40;

setTime = new Thread(this);

setTime.start();

return true;

}

if(stemp.equals("Xtreamest"))

{

add=20;

level=20;

setTime = new Thread(this);

setTime.start();

return true;

}

return false;

}

}

Висновок

У результаті вивчення мови Java була розроблена і створена гра "Змійка". Вона може бути яскравим прикладом для відображення головних, принципових можливостей цієї мови. У моїй розробці використовуються динамічно-активні принципи мови Java. У програму інтегровані графічні об'єкти. Програма є логічною і максимально інформативною.

**Список використаної літератури**

1. - <http://ru.sun.com/java/books/online/>
2. - <http://ru.wikipedia.org/wiki/>
3. Книга Х.М. Дейтел, П.Дж. Дейтел, С.И. Сантри: Технологии программирования на Java. Том 1. Графика, JavaBeans.