

Министерство образования Республики Беларусь
Учреждение образования
Белорусский государственный университет информатики и радиоэлектроники

Отчёт по лабораторной работе
игра «Змейка»
по дисциплине
“Кроссплатформенное программирование”

Выполнил студент
учебной группы 450503
Козлов А.А.

Проверил преподаватель
Кухарчук И.В.

Минск 2016

СОДЕРЖАНИЕ

1 ОПИСАНИЕ ИГРЫ.....	3
1.1 Игровой процесс.....	3
1.2 Скриншоты игры.....	3
2 ДИАГРАММА КЛАССОВ.....	5
2.1 Обоснование компоновки классов.....	6
3 БЛОК-СХЕМА.....	7
4 НОТАЦИЯ.....	8
5 МНОГОПОТОЧНОСТЬ.....	9
6 СОРТИРОВКА.....	10
7 СТАТИСТИКА.....	11
8 ПЕРЕВОД НОТАЦИИ К ПОЛЬЗОВАТЕЛЬСКОМУ ВИДУ.....	12
9 ПРИЛОЖЕНИЕ А Сгенерированная документация.....	13

1 ОПИСАНИЕ ИГРЫ

1.1 Игровой процесс

Компьютерная игра, возникшая в середине 1970-х.

Простая игра, где змейка собирает яблоки . Правила компьютерной игры очень просты:

Игрок управляет длинным, тонким существом, напоминающим змею, которое ползает по плоскости (как правило, ограниченной стенками), собирая еду (или другие предметы), избегая столкновения с собственным хвостом и краями игрового поля. В некоторых вариантах на поле присутствуют дополнительные препятствия. Каждый раз, когда змея съедает кусок пищи, она становится длиннее, что постепенно усложняет игру. Игрок управляет направлением движения головы змеи (обычно 4 направления: вверх, вниз, влево, вправо), а хвост змеи движется следом. Игрок не может остановить движение змеи.

1.2 Скриншоты игры



Рисунок 1.1 - Главное меню игры

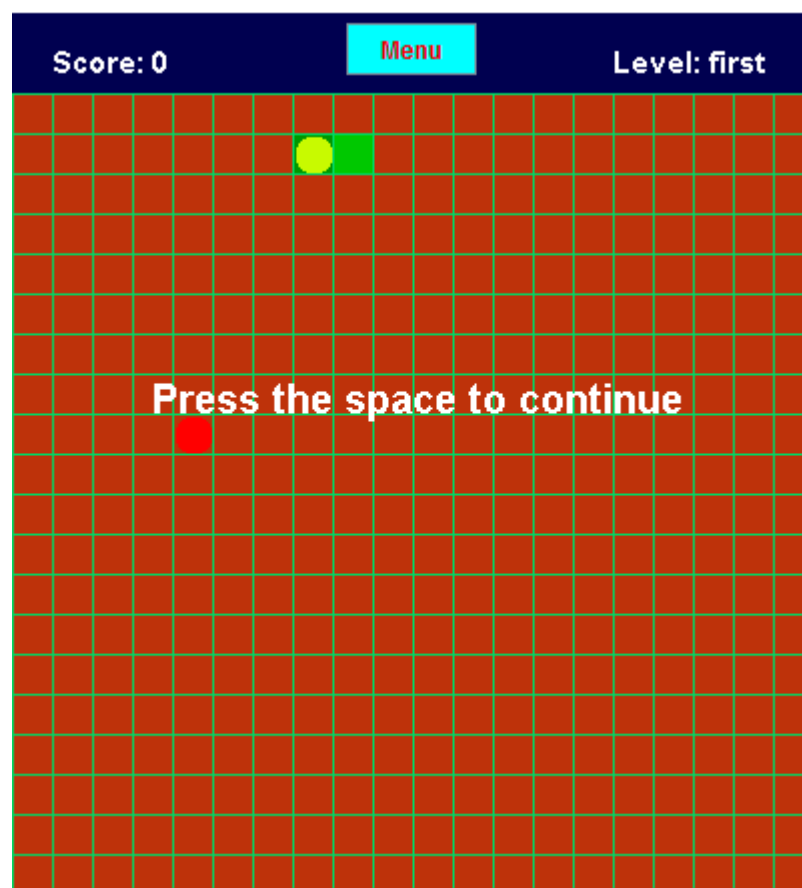


Рисунок 1.2 – Игровое поле

2 ДИАГРАММА КЛАССОВ

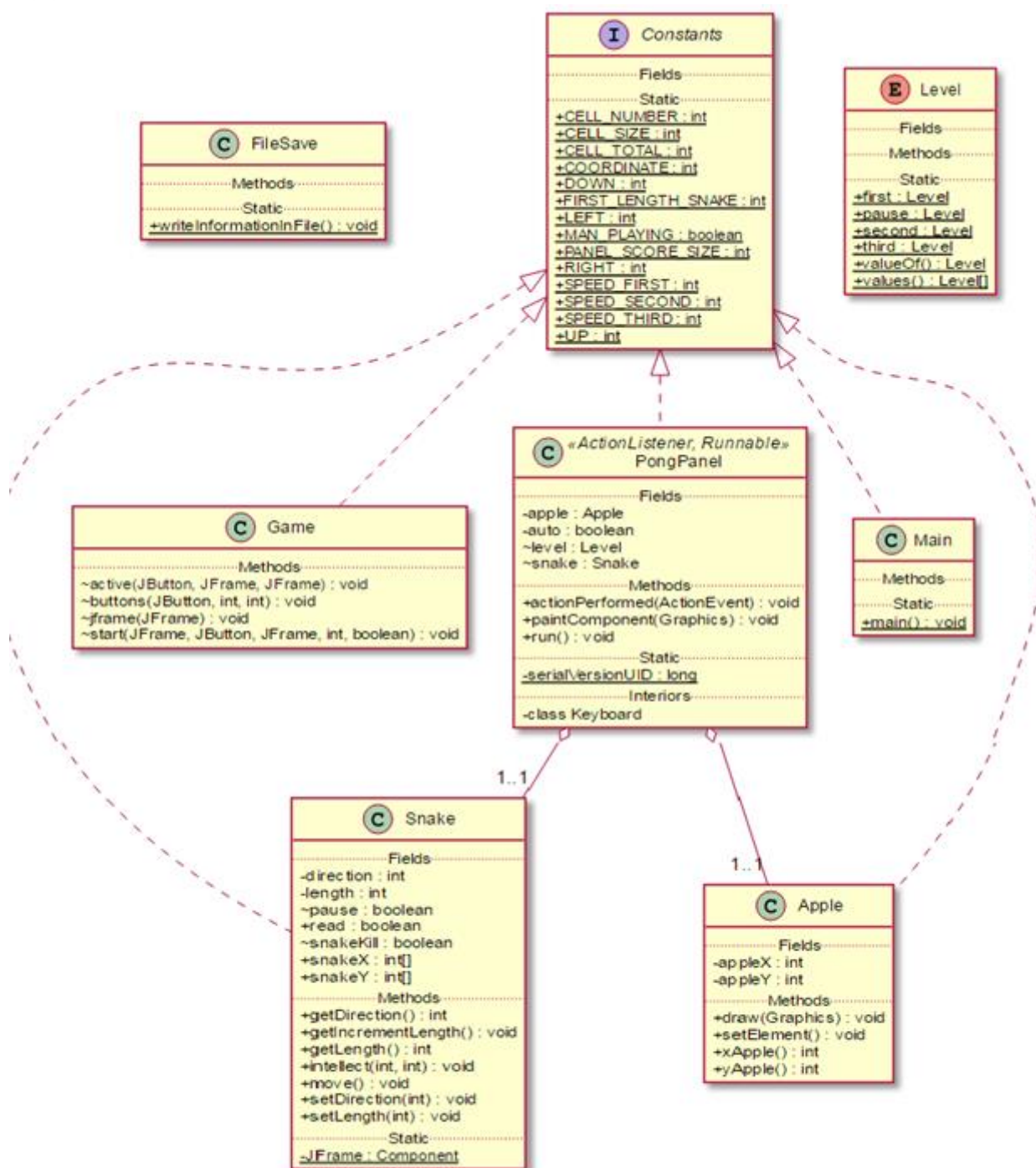


Рисунок 2.1 – Диаграмма классов

2.1 Обоснование компоновки классов

Программа содержит 8 основных классов, один класс для запуска программы и три класса для сортировки игр.

В классе Apple происходит создание и рисование яблока для игры. Яблоко формируется рандомно.

Интерфейс Constants содержит константы, которые в дальнейшем используются в других классах.

Класс Level содержит уровни игры, которые пользователь выбирает в главном меню. Этот класс ничего не наследует.

Класс Main является основным. Здесь происходит создание объектов других классов. Этот класс использует константы класса Constants.

Класс Menu делает переход между игровыми сценами.

Класс Snake является основным для игры. Здесь происходит передвижение и рисование змеи.

В классе SnakePanel происходит взаимодействие между остальными классами. Содержит необходимые слушатели панели, такие как ActionListener.

Класс OneStep содержит состояние каждого хода змейки. При перемещении змеи происходит передача новых данных в конструктор класса OneStep.

В классе ReadAndWrite происходит сохранение/чтение информации в файл. С помощью этого класса воспроизводится игра.

3 БЛОК-СХЕМА

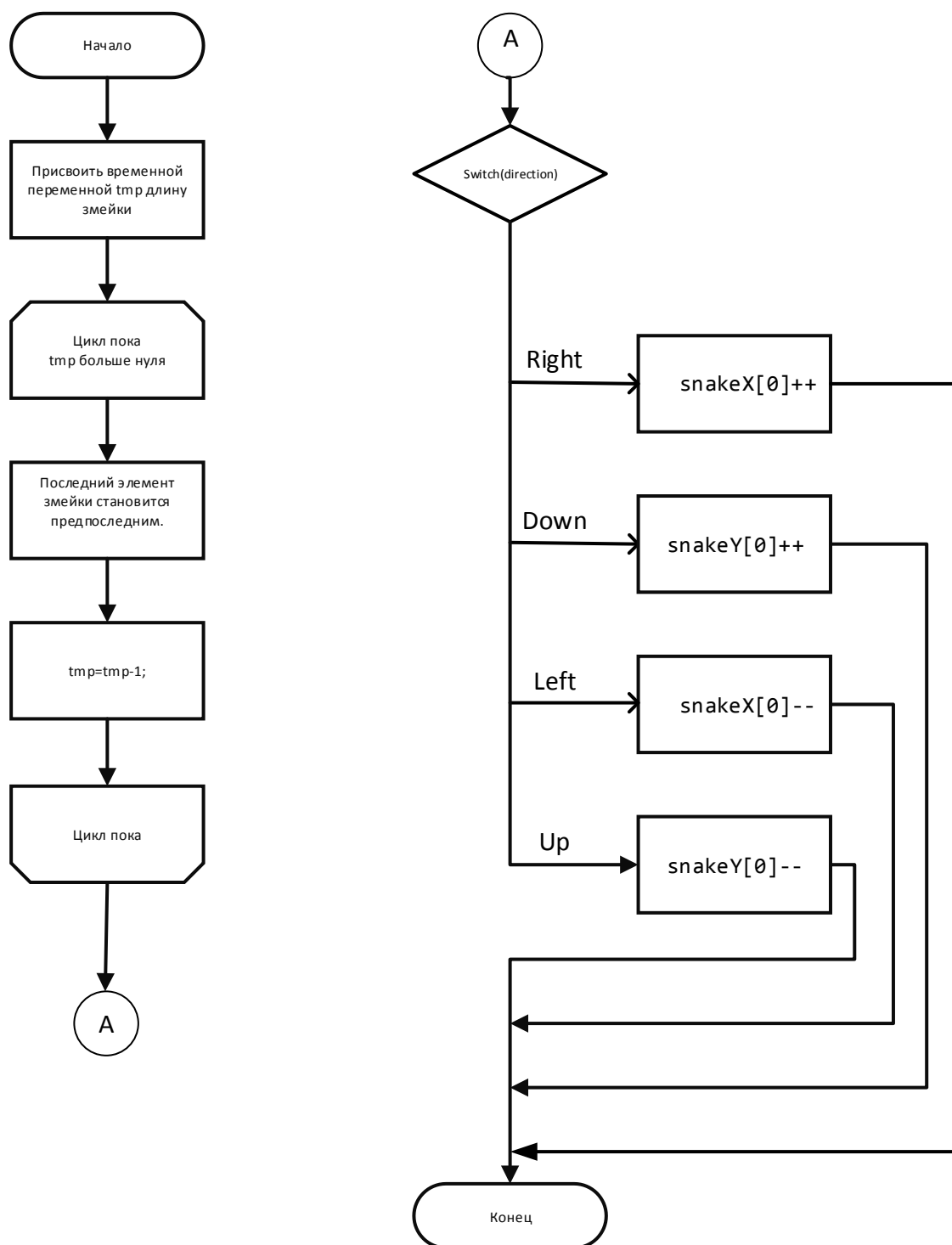


Рисунок 3.1 – Структурная схема механизма движения змейки

4 НОТАЦИЯ

Нотация должна обладать следующими возможностями:

1. воспроизведение сохранённой игры;
2. отображаться в поле игры;
3. сохраняться в файл

Для воспроизведения игры необходимо знать координаты объектов в каждом кадре. В связи с этим, координаты элементов удобно добавлять в список, из которого и в который будет записываться или считываться состояние игры.

В список покадрово добавляются координаты выбранных элементов. После окончания игры эти координаты записываются в файл. Этих данных достаточно для воспроизведения игры, поэтому именно они и были выбраны. При воспроизведении игры нужные данные считываются из файла в список и покадрово восстанавливают положение объектов в игре

1	length
2	snakeY
3	snakeX
4	appleX
5	appleY

Таблица 4.1 – Данные, записываемые в файл

5 МНОГОПОТОЧНОСТЬ

Snake является классом «графики игры» и отвечает за отрисовку, в то время как все вычисления производятся в классе SnakePanel, выполняются в отдельном потоке.

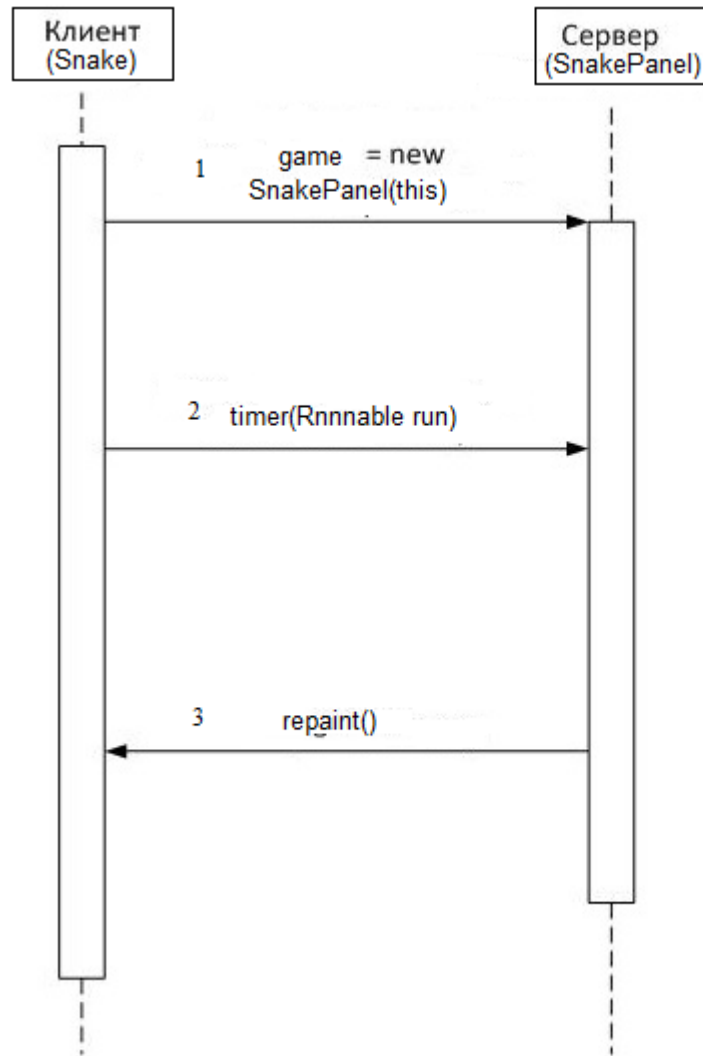


Рисунок 5.1 - Диаграмма последовательности

При помощи таймера вызывается функция `run()` с определенной периодичностью, в данной функции выполняются необходимые операции логики игры, далее в функции `run()` подается сигнал для перерисовки окна.

6 СОРТИРОВКА

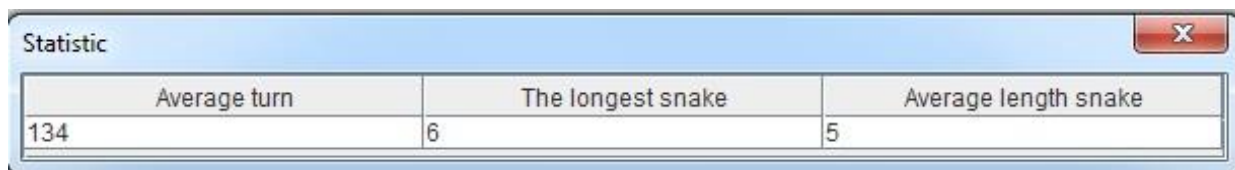
Сортировки игр были произведены по двум критериям: количеству сделанных ходов и длине змейки. Эти критерии характеризуют сложность прохождения игры и помогают лучше определиться с успехами в её прохождении. Java уступает Scala по производительности при выполнении больших объёмов работы, т.к. Scala имеет преимущество при выполнении алгоритмов с хвостовой рекурсией.

Таблица 6.1 – Время сортировки нотаций

	100 нотаций	1000 нотаций	10000 нотаций
Scala, мс	285-311	2345-2375	40895-40930
Java, мс	290-315	2386-2434	40914-40935

7 СТАТИСТИКА

Статистика была составлена по трём критериям: средняя длина, лучшая игра и среднее количество ходов за все игры.

A screenshot of a software window titled 'Statistic'. The window has a blue header bar with a red 'X' button in the top right corner. Below the header is a table with three columns: 'Average turn', 'The longest snake', and 'Average length snake'. The table contains one row of data with the values 134, 6, and 5 respectively.

Average turn	The longest snake	Average length snake
134	6	5

Рисунок 7.1 – Статистика игр

8 ПЕРЕВОД НОТАЦИИ К ПОЛЬЗОВАТЕЛЬСКОМУ ВИДУ

Для перевода нотации к пользовательскому виду был реализован следующий алгоритм: при сохранении игры, с помощью инструмента match case записывает в файл определённое сообщение, соответствующее каждому ходу игрока.

```
Kozlov_Game_Snake  
coordinates snakeX9.0  
coordinates snakeX 3.0  
length snake1  
coordinates snakeX10.0  
coordinates snakeX 3.0  
length snake1  
coordinates snakeX11.0  
coordinates snakeX 3.0  
length snake1  
coordinates snakeX12.0  
coordinates snakeX 3.0  
length snake1  
coordinates snakeX13.0  
coordinates snakeX 3.0  
length snake1  
coordinates snakeX14.0  
coordinates snakeX 3.0  
length snake1  
coordinates snakeX14.0  
coordinates snakeX 4.0  
length snake1  
coordinates snakeX14.0  
coordinates snakeX 5.0  
length snake1  
coordinates snakeX14.0  
coordinates snakeX 6.0  
length snake1  
coordinates snakeX14.0  
coordinates snakeX 7.0  
...
```

Рисунок 8.1 – Пример приведения нотации к пользовательскому виду

ПРИЛОЖЕНИЕ А

Сгенерированная документация

На следующей картинке приведён пример сгенерированной документации. Для её создания код комментировался определённым образом, а затем применялся инструмент Javadoc.

Class Snake

java.lang.Object
com.alexey.game.Snake

All implemented interfaces:
Constants

```
public class Snake
extends java.lang.Object
implements Constants

This class draws a snake and describes her movement
```

Field Summary

Fields

Modifier and Type	Field and Description
int	length
int []	snakeX
int []	snakeY

Fields inherited from interface com.alexey.game.Constants

CELL_NUMBER, CELL_SIZE, CELL_TOTAL, COORDINATE, DOWN, FIRST_LENGTH_SNAKE, LEFT, MAX_PLAYING, PANEL_SCORE_SIZE, RIGHT, SPEED_FIRST, SPEED_SECOND, SPEED_THIRD, UP, X_ONE, Y_ONE

Constructor Summary

Constructors

Constructor and Description
Snake (int x0, int y0) Constructor

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
void	draw (java.awt.Graphics g)
int	getDirection ()
void	getIncrementLength () This method increment length
int	getLength ()

int	getLength()
int	getSnakeXBody()
int	getSnakeYBody()
void	intellect(int posX, int posY)
void	move() This method allows the snake to move
void	setDirection(int direction)
void	setLength(int length)
void	setSnakeXBody(int snakeXBody)
void	setSnakeYBody(int snakeYBody)
Methods Inherited from class java.lang.Object	
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait	

Field Detail

length
public int length
snakeX
public int[] snakeX
snakeY
public int[] snakeY

Constructor Detail

Snake
public Snake(int x0, int y0)
Constructor

Method Detail

getLength
public int getLength()
Returns: length
getIncrementLength
public void getIncrementLength()
This method increment length
getDirection
public int getDirection()
Returns: direction
setDirection
public void setDirection(int direction)
Parameters: direction -
setLength
public void setLength(int length)
Parameters: length -
draw
public void draw(java.awt.Graphics g)
Parameters: g - This method draws a snake
intellect
public void intellect(int posX, int posY)

