# Display the world’s borders

Modify the **SnakeWorld** constructor**:**

public SnakeWorld()

{

super(25, 20, 32);

addObject(new Border(), 0, 0);

}

Second version of the **SnakeWorld** constructor**:**

public SnakeWorld()

{

super(25, 20, 32);

for (int x = 0; x < getWidth(); x ++ ) {

addObject(new Border(), x, 0);

addObject(new Border(), x, getHeight()-1);

}

for (int y = 1; y < getHeight()-1; y ++) {

addObject(new Border(), 0, y);

addObject(new Border(), getWidth()-1, y);

}

}

# Display Bobby’s head

Add the following **property** in the class **SnakeWorld**:

private LinkedList<SnakeBody> snake = new LinkedList<SnakeBody>();

Add the following at the beginning of the class **SnakeWorld**:

import java.util.\*;

Add the following lines in the **SnakeWorld constructor**:

SnakeBody body = new SnakeBody();

snake.add(body);

addObject(body, 2, 2);

# Bobby’s Movement

Add the following lines in the **SnakeWorld**class **properties**:

private int dx = 1;

private int dy = 0;

Add this method in the **SnakeWorld**class:

public void act()

{

//on remplace l'image de la tête

SnakeBody head = snake.getLast();

head.setImage("tail.png");

//crée une nouvelle tête

SnakeBody newHead = new SnakeBody();

int newHeadX = head.getX()+dx;

int newHeadY = head.getY()+dy;

//ajoute la nouvelle tête à la liste et au world

addObject(newHead, newHeadX, newHeadY);

snake.add(newHead);

}

# Limit the snake tail size

Add the following property in the **SnakeWorld**class:

private int tailCounter = 5;

Add the following lines in the **act()** method of the **SnakeWorld**class:

if (tailCounter == 0) {

SnakeBody tail = snake.removeFirst();

removeObject(tail);

} else {

tailCounter--;

}

# Change of direction

Add this method to the **SnakeWorld**class:

private void changeDirection() {

if (Greenfoot.isKeyDown("left") && dx == 0 ) {

dx = -1;

dy = 0;

} else if (Greenfoot.isKeyDown("right") && dx == 0 ) {

dx = 1;

dy = 0;

} else if (Greenfoot.isKeyDown("down") && dy == 0 ) {

dx = 0;

dy = 1;

} else if (Greenfoot.isKeyDown("up") && dy == 0 ) {

dx = 0;

dy = -1;

}

}

Add the following line in the **act()** method of the **SnakeWorld** class (at the beginning of the method):

changeDirection();

# Manage collisions

Add this property to the **SnakeWorld**class:

private boolean dead = false;

Add the following lines in the **act()** method of the **SnakeWorld**class:

if (dead) {

return;

}

Add this method to the **SnakeWorld**class:

public void dead() {

dead = true;

}

Add the following lines to the **act()** method of the **SnakeWorld**class(make sure to copy this code at the right place) :

List<Block> blocks = getObjectsAt(newHeadX, newHeadY, Block.class);

for(Block block : blocks) {

block.collision(this);

}

Add this method to the **Block** class:

public void collision(SnakeWorld world) {

world.dead();

}

# Add an apple

Add the following lines in the **SnakeWorld**constructor:

Apple apple = new Apple();

addObject(apple,

Greenfoot.getRandomNumber(getWidth()-2)+1,

Greenfoot.getRandomNumber(getHeight()-2)+1);

# Collision with an apple

Add this method to the **Apple** class:

public void collision(SnakeWorld world) {

world.grow(2);

setLocation(

Greenfoot.getRandomNumber(getWorld().getWidth()-2)+1,

Greenfoot.getRandomNumber(getWorld().getHeight()-2)+1);

}

Add this method to the **SnakeWorld**class:

public void grow(int i) {

tailCounter = tailCounter + i;

}

# Add sounds

Add the following line in the **collision** method of the **Apple** class:

Greenfoot.playSound("slurp.mp3");

Add the following line in the **collision** method of the **Block** class:

Greenfoot.playSound("dead.mp3");