

PREGUNTA 1

The screenshot shows the Eclipse IDE with the `Game.java` file open. The code is as follows:

```

99     Cell nextCell = board.getCells()[row][col];
100
101     return nextCell;
102 }
103
104 public static void main(String[] args)
105 {
106
107     System.out.println("Going to start game");
108
109     Cell initPos = new Cell(0, 0);
110     Snake initSnake = new Snake(initPos);
111     Board board = new Board(10, 10);
112     Game newGame = new Game(initSnake, board);
113     newGame.gameOver = false;
114     newGame.direction = DIRECTION_RIGHT;
115
116     for (int i = 0; i < 5; i++) {
117         if (i == 2)
118             newGame.board.generateFood();
119         newGame.update();
120         if (i == 3)
121             newGame.direction = DIRECTION_RIGHT;
122         if (newGame.gameOver == true)
123             break;
124     }
125 }
126 }
127

```

The Variables window on the right shows the following variables and their values:

Name	Value
args	String[0] (id=19)
initPos	Cell (id=28)
initSnake	Snake (id=30)
board	Board (id=32)
newGame	Game (id=33)
i	0

The Console window at the bottom shows the following output:

```

Game (1) [Java Application] /home/estudiante/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_16.0.2.v20210721-1149/jre/bin/
Going to start game
Going to update the game
Going to find next cell
Going to check for Crash
Snake is moving to 0 1

```

The screenshot shows the Eclipse IDE with the `Game.java` file open. The code is the same as in the previous screenshot.

The Variables window on the right shows the following variables and their values:

Name	Value
args	String[0] (id=19)
initPos	Cell (id=28)
initSnake	Snake (id=30)
board	Board (id=32)
newGame	Game (id=33)
i	1

The Console window at the bottom shows the following output:

```

Game (1) [Java Application] /home/estudiante/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_16.0.2.v20210721-1149/jre/bin/
Going to start game
Going to update the game
Going to find next cell
Going to check for Crash
Snake is moving to 0 1
Going to update the game
Going to find next cell
Going to check for Crash
Snake is moving to 0 2

```

Si i vale 2 se genera la comida

```
99     Cell nextCell = board.getCells()[row][col];
100
101     return nextCell;
102 }
103
104 public static void main(String[] args)
105 {
106
107     System.out.println("Going to start game");
108
109     Cell initPos = new Cell(0, 0);
110     Snake initSnake = new Snake(initPos);
111     Board board = new Board(10, 10);
112     Game newGame = new Game(initSnake, board);
113     newGame.gameOver = false;
114     newGame.direction = DIRECTION_RIGHT;
115
116     for (int i = 0; i < 5; i++) {
117         if (i == 2)
118             newGame.board.generateFood();
119         newGame.update();
120         if (i == 3)
121             newGame.direction = DIRECTION_RIGHT;
122         if (newGame.gameOver == true)
123             break;
124     }
125 }
126 }
127 }
```

Name	Value
update() returned	(No explicit return value)
args	String[0] (id=19)
initPos	Cell (id=28)
initSnake	Snake (id=30)
board	Board (id=32)
newGame	Game (id=33)
i	2

Game (1) [Java Application] /home/estudiante/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_16.0.2.v20210721-1149/jre/bin/jav
Going to update the game
Going to find next cell
Going to check for Crash
Snake is moving to 0 1
Going to update the game
Going to find next cell
Going to check for Crash
Snake is moving to 0 2
Going to generate food
Food is generated at: 8 3
Going to update the game
Going to find next cell
Going to check for Crash
Snake is moving to 0 3

Si i vale 3 la dirección es a la derecha

The screenshot shows an IDE with a Java file named `Game`. The code implements a simple Snake game logic. The `main` method initializes the game state and enters a loop that updates the game state and finds the next cell. The console output shows the game's progress, including the snake's movement and food generation.

```
100
101     return nextCell;
102 }
103
104 public static void main(String[] args)
105 {
106
107     System.out.println("Going to start game");
108
109     Cell initPos = new Cell(0, 0);
110     Snake initSnake = new Snake(initPos);
111     Board board = new Board(10, 10);
112     Game newGame = new Game(initSnake, board);
113     newGame.gameOver = false;
114     newGame.direction = DIRECTION_RIGHT;
115
116     for (int i = 0; i < 5; i++) {
117         if (i == 2)
118             newGame.board.generateFood();
119         newGame.update();
120         if (i == 3)
121             newGame.direction = DIRECTION_RIGHT;
122         if (newGame.gameOver == true)
123             break;
124     }
125 }
126 }
127
```

The console output shows the game's progress:

```
Game (1) [Java Application] /home/estudiante/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_16.0.2.v20210721-1149/jre/bin
Going to update the game
Going to find next cell
Going to check for Crash
Snake is moving to 0 2
Going to generate food
Food is generated at: 8 3
Going to update the game
Going to find next cell
Going to check for Crash
Snake is moving to 0 3
Going to update the game
Going to find next cell
Going to check for Crash
Snake is moving to 0 4
```

Si i vale 4 se sale del bucle

The screenshot shows the Eclipse IDE with the following components:

- Editor:** Snake.java is open, showing the main method. The code is as follows:

```
99     Cell nextCell = board.getCells()[row][col];
100
101     return nextCell;
102 }
103
104 public static void main(String[] args)
105 {
106     System.out.println("Going to start game");
107
108     Cell initPos = new Cell(0, 0);
109     Snake initSnake = new Snake(initPos);
110     Board board = new Board(10, 10);
111     Game newGame = new Game(initSnake, board);
112     newGame.gameOver = false;
113     newGame.direction = DIRECTION_RIGHT;
114
115     for (int i = 0; i < 5; i++) {
116         if (i == 2)
117             newGame.board.generateFood();
118         newGame.update();
119         if (i == 3)
120             newGame.direction = DIRECTION_RIGHT;
121         if (newGame.gameOver == true)
122             break;
123     }
124 }
125 }
126 }
127 }
```
- Variables View:** Shows the current state of variables:

Name	Value
args	String[] (id=19)
initPos	Cell (id=28)
initSnake	Snake (id=30)
board	Board (id=32)
newGame	Game (id=33)
i	4
- Console View:** Shows the output of the program:

```
Game (1) [Java Application] /home/estudiante/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_16.0.2.v20210721-1149/jre/
Going to generate food
Food is generated at: 8 3
Going to update the game
Going to find next cell
Going to check for Crash
Snake is moving to 0 3
Going to update the game
Going to find next cell
Going to check for Crash
Snake is moving to 0 4
Going to update the game
Going to find next cell
Going to check for Crash
Snake is moving to 0 5
```

En el menú de variables cambio el valor de i para que sea 4 cuando depuro en la línea 120. Termina y se sale del bucle.

The screenshot shows the Eclipse IDE with the following components:

- Editor:** Displays the `Snake.java` file. The code includes a `main` method that initializes a game and enters a `for` loop. Line 120, `if (i == 3)`, is highlighted in green.
- Variables Window:** Located on the right, it shows the current state of variables. The variable `i` is highlighted with a value of 4. Other variables include `update() returned` (No explicit return value), `args` (String[0] (id=19)), `initPos` (Cell (id=20)), `initSnake` (Snake (id=23)), `board` (Board (id=26)), and `newGame` (Game (id=28)).
- Console:** Located at the bottom, it shows the output of the program: `Game (1) [Java Application] /home/estudiante/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_16.0.2.v20210721-1149/jre/bin/java (20 J`, `Going to start game`, `Going to update the game`, `Going to find next cell`, `Going to check for Crash`, and `Snake is moving to 0 1`.

PREGUNTA 3.

ID Caso Prueba	Nombre	Descripción	Precondiciones	Resultado esperado	Resultado obtenido	Comentarios
ID1	Comprobar que se actualiza get.Head	Si es distinto de gameOver y la direccion es distinta de Direction_none	gameOver false direccion distinto a direction_none	snake.getHead	snake.getHead	
ID2	Comprobar que se actualiza gameOver = true	Si es distinto de gameOver y la direccion es distinta de Direction_none , direccion sera direction_none, gameOver sera true	gameOver false direccion distinto a direction_none			