

ALDO SAÚL VÁSQUEZ MOREIRA

CARNET: 202109754

FACULTAD DE INGENIERÍA

ESCUELA DE CIENCIAS Y SISTEMAS

LABORATORIO DE INTRODUCCIÓN A LA PROGRAMACIÓN 1



MANUAL TÉCNICO

Manual Técnico PAC-MAN

Para el desarrollo del código de esta versión de consola del juego PAC-MAN se utilizó una estructura secuencial.

Métodos Utilizados

Únicamente se creó un método llamado `menuPrincipal()` debido a la gran cantidad de veces que es necesario redireccionar al usuario a la pantalla principal durante el uso del juego.

```
public static void main(String[] args) {  
    int contador = -1;  
    menuPrincipal(contador);  
}
```

Dicho método tiene un parámetro llamado “contador” el cual cumple la función de almacenar la cantidad de veces que se ejecuta el juego para así poder almacenar los datos del jugador en un vector en la posición que el contador indique. Asimismo, el contador se inicializa con el valor -1 ya que al ejecutarlo por primera vez este adquirirá el valor 0, ya que como bien se sabe, todo arreglo (ya sea unidimensional o bidimensional) inicia en la posición 0.

A continuación, se muestra la estructura del método mencionado:

```
//Creando el método del menú principal
public static void menuPrincipal(int contador) {
    Scanner lector = new Scanner(System.in);
    String[] nombresJugador = new String[10];
    int[] edadJugador = new int[10];
    int[] punteoJugador = new int[10];
    int[] movimientosJugador = new int[10];
    int opcion;

    do {
        System.out.println("*****");
        System.out.println("* -----Bienvenido a Pacman-----*");
        System.out.println("*                                     *");
        System.out.println("* Seleccione una opción:           *");
        System.out.println("*                                     *");
        System.out.println("* [1] Jugar                         *");
        System.out.println("* [2] Historial                     *");
        System.out.println("* [3] Salir                         *");
        System.out.println("*****");
        opcion = lector.nextInt();

        switch (opcion) {
```

Se puede ver que en este da inicio el juego, permitiéndole al usuario escoger entre tres opciones. Dicha opción es leída e interpretada por medio de una estructura selectiva (switch)

En caso de que la respuesta sea 1 se ejecuta la parte principal del código. Donde el usuario determina las dimensiones del arreglo bidimensional, así como el desarrollo de la partida.

Puntos importantes:

- Creación de tablero: Para la creación del tablero se hizo uso de un arreglo bidimensional, asignando como dimensiones los datos ingresados por el

usuario almacenados previamente en dos variables. Además de diversas condiciones dentro de la estructura iterativa para cumplir con los requerimientos del juego.

```
//Creando la matriz para mostrar el tablero
int salida1 = 0, salida2 = 0, salida3 = -1;
if (filas % 2 == 0) {
    salida1 = (filas / 2) - 1;
    salida2 = salida1 + 1;
} else {
    salida1 = (((filas - 1) / 2)) - 1;
    salida2 = salida1 + 1;
    salida3 = salida2 + 1;
}
String[][] tablero = new String[filas][columnas];
for (int i = 0; i < filas; i++) {
    for (int j = 0; j < columnas; j++) {
        if (i == 0 || i == (filas - 1) || j == 0 || j == (columnas - 1)) {
            if (i == salida1 || i == salida2 || i == salida3) {
                tablero[i][j] = " ";
            } else {
                tablero[i][j] = "*";
            }
        } else {
            tablero[i][j] = " ";
        }
    }
}
```

- Generación de ítems aleatorios: Para la generación de ítems aleatorios nuevamente se hizo uso de la estructura selectiva if para cumplir con los requerimientos del juego, tales como: evitar que se sobrescriban dichos elementos, así como evitar que estos se coloquen en las entradas del tablero, etc. Esto por medio uso de la librería Random.

```

int porcentajeColumnas = (int) (cantidadColumnas * 0.05);
for (int i = 0; i < porcentajeColumnas; i++) {
    do {
        pfAsteriscoI = numeroRandom.nextInt( bound: filas - 2) + 1;
        pcAsteriscoI = numeroRandom.nextInt( bound: columnas - 2) + 1;
        tablero[pfAsteriscoI][pcAsteriscoI] = "+";
    }while (tablero[pfAsteriscoI][pcAsteriscoI].equals(" ") && pcAsteriscoI != 0 && pcAsteriscoI != (columnas-1));
}
do {
    pfAsterisco = numeroRandom.nextInt( bound: filas - 2) + 1;
    pcAsterisco = numeroRandom.nextInt( bound: columnas - 2) + 1;
    tablero[pfAsterisco][pcAsterisco] = "u";
}while (tablero[pfAsterisco][pcAsterisco].equals(" "));
do {
    pfDolar = numeroRandom.nextInt( bound: filas - 2) + 1;
    pcDolar = numeroRandom.nextInt( bound: columnas - 2) + 1;
    tablero[pfDolar][pcDolar] = "$";
}while ( tablero[pfDolar][pcDolar].equals(" "));
do {
    pfArroba = numeroRandom.nextInt( bound: filas - 2) + 1;
    pcArroba = numeroRandom.nextInt( bound: columnas - 2) + 1;
    tablero[pfArroba][pcArroba] = "@";
}while (tablero[pfArroba][pcArroba].equals(" "));
do {
    pfPacman = numeroRandom.nextInt( bound: filas - 2) + 1;
    pcPacman = numeroRandom.nextInt( bound: columnas - 2) + 1;
    tablero[pfPacman][pcPacman] = "V";
}while (tablero[pfPacman][pcPacman].equals(" "));

```

- **Movimiento del PAC-MAN:** Se hizo uso de estructuras selectivas, así como diversas condicionantes. Sin embargo, el punto importante es que dicha estructura permitía modificar la posición de la fila o columna en la que el PAC-MAN se posiciona.

```

switch (movimiento) {
    case "W":
        int pfPacman2 = pfPacman - 1;
        memoriaDato = tablero[pfPacman2][pcPacman];
        if (memoriaDato.equals("*")) {
            tablero[pfPacman][pcPacman] = "V";
            tablero[pfPacman2][pcPacman] = "*";
            pfPacman2 = pfPacman2 + 1;
        } else {
            tablero[pfPacman][pcPacman] = " ";
            tablero[pfPacman2][pcPacman] = "V";
            for (int i = 0; i < filas; i++) {
                for (int j = 0; j < columnas; j++) {
                    if (i == pfPacman2 & j == pcPacman) {
                        tablero[i][j] = tablero[pfPacman2][pcPacman];
                        if (memoriaDato.equals("@")) {
                            punteo = punteo + 10;
                            tablero[pfPacman2][pcPacman] = "V";
                            do {
                                pfArroba = numeroRandom.nextInt( bound: filas - 2) + 1;
                                pcArroba = numeroRandom.nextInt( bound: columnas - 2) + 1;
                                tablero[pfArroba][pcArroba] = "@";
                            } while (tablero[pfArroba][pcArroba].equals(" "));
                            tablero[i][j] = tablero[pfPacman2][pcPacman];
                        }
                    }
                }
            }
            if (memoriaDato.equals("$")) {
                punteo = punteo + 15;
                tablero[pfPacman2][pcPacman] = "V";
            }
        }
    }
}

```

En el ejemplo adjunto se puede observar que al presionar la tecla “W” la posición fila del PAC-MAN se reduce en 1, es decir, va hacia arriba. Con esta idea general se procedió con el resto de botones.

- Generación de historial: Únicamente se hizo uso de vectores unidimensional, estructuras iterativas y selectivas para cumplir con los requerimientos del juego. Asimismo, aquí tiene participación importante la variable “contador” del método principal, puesto que cada uno de los datos ingresados se almacenan en la posición que esta variable indica. Luego, únicamente se “imprimen”.

Además, también se validó de ingresar datos aleatorios que en los espacios en el que el vector estuviera vacío, ya que de no ser así se obtenía un error dado que no se puede “imprimir” en consola posiciones vacías.

```

case 2:
    System.out.println("-----HISTORIAL-----");

    //Para llenar espacios vacios
    for (int i = 0; i < 10; i++) {
        if (edadJugador[i] == 0) {
            nombresJugador[i] = "----";
            edadJugador[i] = 0;
            punteoJugador[i] = 0;
            movimientosJugador[i] = 0;
        }
    }

    //Imprimir historial
    for (int i = 0; i < 10; i++) {
        if (edadJugador[i] == 0) {
            System.out.print("");
        } else {
            System.out.println("Nombre: " + nombresJugador[i]);
            System.out.println("Edad: " + edadJugador[i]);
            System.out.println("Punteo: " + punteoJugador[i]);
            System.out.println("Movimientos: " + movimientosJugador[i]);
            System.out.println("*****");
        }
    }
    System.out.println(" ");
    break;

```

Librerías utilizadas:

- **Librería Scanner:** Proveniente del paquete java.util. Se utilizó para obtener las entradas de los tipos de datos primitivos.
- **Librería Random:** Proveniente del paquete java.util. Se utilizó para generar números pseudo-aleatorios.