

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Ingeniería en Ciencias y Sistemas

Laboratorio de Introducción a la Programación

Sección B



Práctica 1

Juego Pacman en Java

Manual Técnico

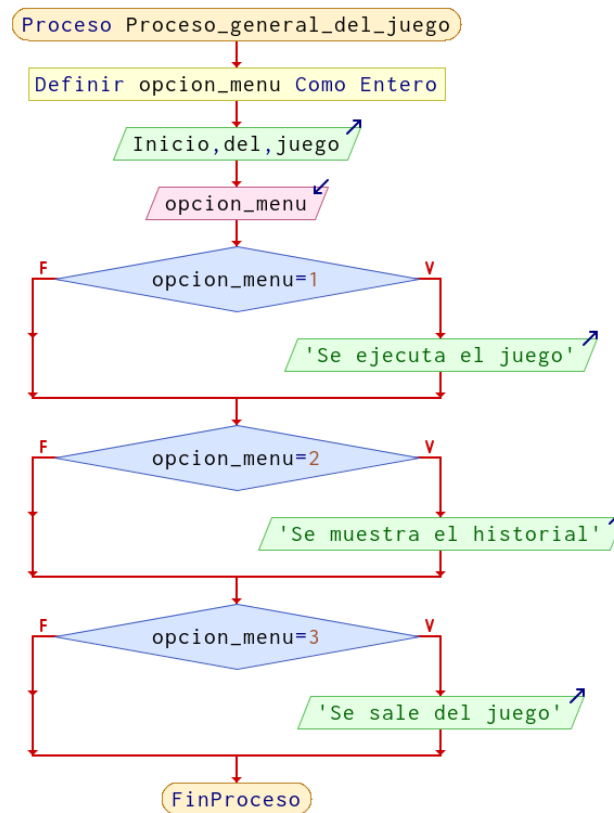
Anyelo Gustavo Hernández Ayala

201807398

Guatemala, 13 febrero de 2022

Estructura base del juego

Dividiremos la explicación del programa en 3 partes: la ejecución del juego, mostrar el historial y salir del programa.



Declaración de Variables

Se declaran las variables filas, columnas y un contador para realizar la generación de la tabla. También se hace el objeto rand para generar los valores aleatorios.

```
// Estructura base para el movimiento de Pacman.
int filas, columnas, contador = 0;
String movimiento;
boolean exit = false;

Random rand = new Random();
```

Se declararán las variables PosX y PosY para poder mover el Pacman dentro del tablero. Y unos contadores para utilizar While para generar aleatoriamente las frutas, paredes y el Pacman dentro del tablero.

```
Posicion inicial de pacman
int PosX, PosY;
Generador de paredes
int columnaRandom, filaRandom;
int contador_pacman = 0;
Para guardar el historial
int contador_datos = 0;
Fruta de 10 puntos
int contador_fruta_1 = 0; // @
Fruta de 15 puntos
int contador_fruta_2 = 0; // $
Fruta de -10 puntos
int contador_fruta_3 = 0; // #
```

```

Arreglos para el historial.
String[] nombre = new String[50];
int[] edad = new int[50];
int[] puntaje = new int[50];
int[] movimientos = new int[50];

```

Se crearon los arreglos correspondientes a la edad, nombre, puntaje y movimientos, con un valor máximo de 50.

Ejecución del juego

Luego de elegir la opción 1, el programa pedirá que ingrese sus datos: nombre, edad y la dimensión de la tabla, esta como mínimo será de 8x8.

Creación de las tablas

Para la creación de las tablas se usaron las funciones «for» y de arreglos para generar una matriz. Lo separe en partes, de modo que su asignación sea secuencial.

```

// Parte de arriba
for(int fila = 0; fila < filas; fila++)
{
    tablero[fila][0] = "*";
}

```

Este bloque asigna el valor de “*” en las primeras celdas de la matriz, de 0x0 hasta 0xN.

Luego para hacer la parte de en medio, realice un for dentro de otro for para que asignara valor de “*” o de espacios en blanco. Entre estos for se realizo una condición, si el ciclo for recorre la mitad de la matriz y si esta matriz es de columnas par o impar, dejara unos espacios entre las columnas de modo que quede simétrica y sirvan luego para realizar los portales en el cuál el Pacman se moverá.

```

// Espacio para el portal en las columnas (columna izquierda)
if(columna != (columnas / 2))
{
// Hace simetricas las columnas
if(columnas % 2 == 0)
{
    tablero[0][columnas/2 - 1] = " ";
}
else{
    tablero[0][columnas/2 - 1] = " ";
    tablero[0][columnas/2 + 1] = " ";
}
    tablero[0][columna] = "*";
}
else{
    tablero[0][columna] = " ";
}

```

Luego de asignar los valores de en medio, realizamos un ultimo for para asignar los valores del final.

```
// Parte de abajo
for(int fila = 0; fila < filas; fila++)
{
    tablero[fila][tablero[0].length - 1] = "*";
}
```

Posicion aleatoria del pacman, muros y las frutas

Para generar el pacman en cualquier posicion valida se tuvo que escribir dentro de una función while, y asignar un contador previamente. Dentro del While vemos que tenemos un If, el cual revisa que no este sobrescribiendo valores existentes, y si se cumple todo, asignara el valor del Pacman en el tablero.

```
// Decide la posicion inicial de pacman
do{
    PosX = rand.nextInt(maximoFilas + minimoFilas) + minimoFilas; //filas = 10 | 0, 1, 2,...,9
    PosY = rand.nextInt(maximoColumnas + minimoColumnas) + minimoColumnas;
    if(!"#".equals(tablero[PosX][PosY]) &&
        !".".equals(tablero[PosX][PosY]) &&
        !"$".equals(tablero[PosX][PosY]) &&
        !"@".equals(tablero[PosX][PosY]) &&
        !"V".equals(tablero[PosX][PosY]))
    {
        tablero[PosX][PosY] = "V";
        prueba[PosX][PosY] = "V";
        contador_pacman++;
    }
}while(contador_pacman <= 0);
```

Bien, ahora para generar los demas objetos aleatoriamente se uso este mismo bloque de código varias veces con la excepcion de que se cambio la variable contador_pacman por la correspondiente, y la asignación del valor en el tablero.

```
// Añade la fruta de 10 puntos
do{
    columnaRandom = rand.nextInt(maximoColumnas + minimoColumnas) + minimoColumnas;
    filaRandom = rand.nextInt(maximoFilas + minimoFilas) + minimoFilas;
    if(!"#".equals(tablero[filaRandom][columnaRandom]) &&
        !"."equals(tablero[filaRandom][columnaRandom]) &&
        !"$".equals(tablero[filaRandom][columnaRandom]) &&
        !"@".equals(tablero[filaRandom][columnaRandom]) &&
        !"V".equals(tablero[filaRandom][columnaRandom]))
    {
        tablero[filaRandom][columnaRandom] = "@";
        prueba[filaRandom][columnaRandom] = "@";
        contador_fruta_1++;
    }
}while(contador_fruta_1 <= 0);
```

Considero que se pudo haber hecho uso de funciones, pero ese no es el propósito de esta práctica, así que tuvo que ser de esta manera.

Impresión del tablero y el menu de puntaje

Esta parte imprime un menu arriba del tablero para que se pueda ver a tiempo real el puntaje del juego y la cantidad de movimientos realizados.

```
System.out.println("-----");
System.out.println(CYAN + "Nombre: " + VERDE + nombre[contador_datos] + ANSI_RESET);
System.out.println(CYAN + "Puntaje: " + MORADO + puntaje[contador_datos] + ANSI_RESET);
System.out.println(CYAN + "Movimientos: " + ROJO + movimientos[contador_datos] + ANSI_RESET);
System.out.println("-----");
```

Ahora la siguiente parte imprime la tabla, utilizando dos For para que imprima los valores 1 a 1 desde el inicio. También se le agrego colores para que

```
// Impresión de la tabla
for(int columna = 0; columna < columnas; columna++){
    for(int fila = 0; fila < filas; fila++){
        if(".".equals(tablero[fila][columna])){
            System.out.print(AZUL + tablero[fila][columna] + " " + ANSI_RESET);
        }else if("#".equals(tablero[fila][columna])){
            System.out.print(ROJO + tablero[fila][columna] + " " + ANSI_RESET);
        }else if("$".equals(tablero[fila][columna])){
            System.out.print(VERDE + tablero[fila][columna] + " " + ANSI_RESET);
        }else if("@".equals(tablero[fila][columna])){
            System.out.print(CYAN + tablero[fila][columna] + " " + ANSI_RESET);
        }else if("V".equals(tablero[fila][columna])){
            System.out.print(AMARILLO + tablero[fila][columna] + " " + ANSI_RESET);
        }else{
            System.out.print(tablero[fila][columna] + " ");
        }
    }
    System.out.println("");
}
```

Movimiento del Pacman

Para el movimiento del Pacman se escribieron condiciones para que revise si, en este caso, la posición de arriba sea "*", si es "*" la siguiente posición, el Pacman no se moverá y añadirá +1 al valor del movimiento. Luego si no es igual a "*" revisa que sea igual a cualquier fruta, hará el movimiento y después añadirá o quitará puntos dependiendo de cual sea la fruta.

Si ninguno de estos casos se cumple, solo se moverá el Pacman en la dirección seleccionada, en este caso, moverse hacia arriba. Para el caso de que se use el portal, se añadió una condición extra, sólo en las teclas seleccionadas "a" o "d".

```

Mueve hacia arriba
if("w".equals(movimiento) || "W".equals(movimiento))
{
    if(tablero[PosX][PosY - 1] == "X")
    {
        PosY = PosY;
        movimientos[contador_datos] += 1;
    }else if(tablero[PosX][PosY - 1] == "#")
    {
        tablero[PosX][PosY] = " ";
        PosY -= 1;
        tablero[PosX][PosY] = "V";
        movimientos[contador_datos] += 1;
        puntaje[contador_datos] -= 10;
        contador_fruta_3 = 0;
    }else if(tablero[PosX][PosY - 1] == "$")
    {
        tablero[PosX][PosY] = " ";
        PosY -= 1;
        tablero[PosX][PosY] = "V";
        movimientos[contador_datos] += 1;
        puntaje[contador_datos] += 15;
        contador_fruta_2 = 0;
    }else if(tablero[PosX][PosY - 1] == "@")
    {
        tablero[PosX][PosY] = " ";
        PosY -= 1;
        tablero[PosX][PosY] = "V";
        movimientos[contador_datos] += 1;
        puntaje[contador_datos] += 10;
        contador_fruta_1 = 0;
    }else{
        tablero[PosX][PosY] = " ";
        PosY -= 1;
        tablero[PosX][PosY] = "V";
        movimientos[contador_datos] += 1;
    }
}
}

```

```

Teletransportación del lado izquierdo
if(PosX == 0)
{
    if(tablero[filas - 1][PosY] == "X")
    {
        PosX = PosX;
        movimientos[contador_datos] += 1;
    }else if(tablero[filas - 1][PosY] == "#")
    {
        tablero[PosX][PosY] = " ";
        PosX = filas - 1;
        tablero[PosX][PosY] = "V";
        puntaje[contador_datos] -= 10;
        contador_fruta_3 = 0;
    }else if(tablero[filas - 1][PosY] == "$")
    {
        tablero[PosX][PosY] = " ";
        PosX = filas - 1;
        tablero[PosX][PosY] = "V";
        puntaje[contador_datos] += 15;
        contador_fruta_2 = 0;
    }else if(tablero[filas - 1][PosY] == "@")
    {
        tablero[PosX][PosY] = " ";
        PosX = filas - 1;
        tablero[PosX][PosY] = "V";
        puntaje[contador_datos] += 10;
        contador_fruta_1 = 0;
    }else{
        tablero[PosX][PosY] = " ";
        PosX = filas - 1;
        tablero[PosX][PosY] = "V";
    }
}
}

```

La lógica es la misma que la anterior, lo unico que cambia es el chequeo que se realiza en el arreglo.

De último tenemos que si la tecla seleccionada es “m”, esta cerrara el juego, dándole valor a la variable booleana como True y añadirá +1 al contador_datos para seguir ingresando valores nuevos en el historial.

```

Cierra la partida actual
else if("m".equals(movimiento) || "M".equals(movimiento))
{
    exit = true;
    contador_datos++;
}
else{
    System.out.println("Ingrese una tecla valida!");
}
}

```

Por último tenemos la validación si el puntaje del jugador llega a 100, gana, si llega a 0, pierde.

```
if(puntaje[contador_datos] >= 100){
    exit = true;
    contador_datos++;
    System.out.println(VERDE + ";HAS GANDO!" + ANSI_RESET);
}else if(puntaje[contador_datos] <= 0){
    exit = true;
    contador_datos++;
    System.out.println(ROJO + "Has perdido." + ANSI_RESET);
}
```

Y con esto termina el código para ejecutar el juego, ahora veremos cómo se realizó el historial.

Historial

Si realizamos una partida y luego seleccionamos la opción 2, de ver el historial, podremos ver los datos de las partidas anteriores.

```
else if(menu_opcion == 2)
{
    Impresion del historial
    for(int i = 0; i <= nombre.length -1; i++)
    {
        if(nombre[i] != null) //Solo muestra los valores que no sean cero.
        {
            System.out.println("-----");
            System.out.println(CYAN + "Nombre del jugador: " + VERDE + nombre[i] + ANSI_RESET);
            System.out.println(CYAN + "Edad: " + VERDE + edad[i] + ANSI_RESET);
            System.out.println(CYAN + "Puntaje: " + MORADO + puntaje[i] + ANSI_RESET);
            System.out.println(CYAN + "Movimientos: " + ROJO + movimientos[i] + ANSI_RESET);
        }
    }
    System.out.println("-----");
}
```

Se utilizó un for para que recorra todos los datos en los arreglos. Luego se escribió una condición, si el nombre en el arreglo i, no es igual a null, este imprimirá los valores que no sean null en el arreglo, ya que es de tipo String, y si no tiene valores, este retornara como Null.

Salir del juego

De último si la opción seleccionada fue 3(salir), saldra del juego y notificará al usuario que se salió.

```
else if(menu_opcion == 3)
{
    System.out.println("Se salió del programa.");
    System.exit(0);
}else{
    System.out.println("Ingrese una opción válida.");
}
```