

UNIVERSIDAD SAN CARLOS DE GUATEMALA
IPC1
LABORATORIO

MANUAL TECNICO

JOSE ROBERTO MEJICANO LARA
202112542

Primero empezamos declarando e inicializando nuestras constantes y variables en el juego

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4   */
5  package packman;
6
7  /**
8   *
9   * @author cvgjo
10  */
11 import java.util.Scanner;
12 import java.util.Random;
13
14 public class Packman {
15
16     // Constantes del juego
17     private static final char FANTASMA = '@';
18     private static final char PREMIO_SIMPLE = '0';
19     private static final char PREMIO_ESPECIAL = '$';
20     private static final char PARED = 'X';
21     private static final char PACMAN = '<';
22     private static final char VACIO = ' ';
23
24     // Variables de juego
25     private static String nombreUsuario;
26     private static int puntaje = 0;
27     private static int vidas = 3;
28     private static char[][] tablero;
29     private static int filas, columnas;
30     private static int pacmanFila, pacmanColumna;
31     private static int premiosRestantes = 0;
32     private static boolean juegoActivo = false;
33     private static boolean juegoPausado = false;
34
35     // Historial de partidas (máximo 100 partidas)
36     private static String[] historialNombres = new String[100];
37     private static int[] historialPuntos = new int[100];
```

Luego nuestro historial de partida con vectores y nuestro scanner para poder leer datos tambien empezamos a dibujar nuestro menu y sus cases.

comenzamos con el main y a pedir el nombre del usuario e inicializar las variables restantes

```
70         default:
71             System.out.println("Opcion no valida");
72         }
73     } while(opcion != 3);
74 }
75
76 private static void iniciarJuego() {
77     // Solicitar nombre de usuario
78     System.out.print("Ingrese su nombre de usuario: ");
79     nombreUsuario = scanner.nextLine();
80
81     // Configurar tablero
82     configurarTablero();
83
84     // Inicializar juego
85     punaje = 0;
86     vidas = 3;
87     juegoActivo = true;
88     juegoPausado = false;
89
90     // Posicionar Pac-Man
91     posicionarPacMan();
92
93     // Iniciar loop del juego
94     loopJuego();
95 }
96
97 private static void configurarTablero() {
98     char tipoTablero;
99     //opciones menu para iniciar
100    do {
101        System.out.print("Seleccione tipo de tablero (P) Pequeño 5x6, (G) Grande 10x10: ");
102        tipoTablero = scanner.nextLine().toUpperCase().charAt(0);
103    } while(tipoTablero != 'P' && tipoTablero != 'G');
104
105    if(tipoTablero == 'P') {
106        filas = 5;
```

dibujamos nuestro tablero

```
Source History ▾ 103     } while(tipoTablero != 'P' && tipoTablero != 'G'); 104     if(tipoTablero == 'P') { 105         filas = 5; 106         columnas = 6; 107     } else { 108         filas = 10; 109         columnas = 10; 110     } 111 112     tablero = new char[filas][columnas]; 113     inicializarTableroVacio(); 114 115     int totalEspacios = filas * columnas; 116     int maxPremios = (int)(totalEspacios * 0.4); 117     int maxParedes = (int)(totalEspacios * 0.2); 118     int maxTrampas = (int)(totalEspacios * 0.2); 119 120     // Solicitar cantidades 121     int cantidadPremios = solicitarCantidad("premios", 1, maxPremios); 122     int cantidadParedes = solicitarCantidad("paredes", 1, maxParedes); 123     int cantidadTrampas = solicitarCantidad("trampas", 1, maxTrampas); 124 125     // Colocar elementos aleatoriamente 126     colocarElementos(PREMIO_SIMPLE, cantidadPremios, true); 127     colocarElementos(PARED, cantidadParedes, false); 128     colocarElementos(FANTASMA, cantidadTrampas, false); 129 130     premiosRestantes = cantidadPremios; 131 } 132 133 private static void inicializarTableroVacio() { 134     for(int i = 0; i < filas; i++) { 135         for(int j = 0; j < columnas; j++) { 136             tablero[i][j] = VACIO; 137         } 138     } 139 }
```

con la información ingresada solicitamos las cantidades y que esos valores se vayan a la variable para poder dibujar el tablero

```
133 134     private static void inicializarTableroVacio() { 135         for(int i = 0; i < filas; i++) { 136             for(int j = 0; j < columnas; j++) { 137                 tablero[i][j] = VACIO; 138             } 139         } 140     } 141 142     private static int solicitarCantidad(String tipo, int min, int max) { 143         int cantidad; 144         do { 145             System.out.print("Ingrese cantidad de " + tipo + " (" + min + "-" + max + "): "); 146             cantidad = scanner.nextInt(); 147             scanner.nextLine(); 148         } while(cantidad < min || cantidad > max); 149         return cantidad; 150     } 151 152     private static void colocarElementos(char elemento, int cantidad, boolean esPremio) { 153         int colocados = 0; 154 155         while(colocados < cantidad) { 156             int fila = random.nextInt(filas); 157             int columna = random.nextInt(columnas); 158 159             if(tablero[fila][columna] == VACIO) { 160                 if(esPremio && random.nextBoolean()) { 161                     tablero[fila][columna] = PREMIO_ESPECIAL; 162                 } else { 163                     tablero[fila][columna] = elemento; 164                 } 165                 colocados++; 166             } 167         } 168     } 169 }
```

```
Source History ▾ 169
170  □     private static void posicionarPacMan() {
171      int fila, columna;
172
173  □     do {
174         mostrarTablero();
175         System.out.print("Ingrese fila para Pac-Man (0-" + (filas-1) + "): ");
176         fila = scanner.nextInt();
177         System.out.print("Ingrese columna para Pac-Man (0-" + (columnas-1) + "): ");
178         columna = scanner.nextInt();
179         scanner.nextLine();
180     } while(fila < 0 || fila >= filas || columna < 0 || columna >= columnas || 
181             tablero[fila][columna] == PARED);
182
183     pacmanFila = fila;
184     pacmanColumna = columna;
185     tablero[pacmanFila][pacmanColumna] = PACMAN;
186 }
187
188  □     private static void loopJuego() {
189      while(juegoActivo && vidas > 0 && premiosRestantes > 0) {
190          mostrarPanelControl();
191          mostrarTablero();
192
193          if(!juegoPausado) {
194              System.out.print("Movimiento (8:arriba, 5:abajo, 6:derecha, 4:izquierda, F:Pausa): ");
195              String input = scanner.nextLine();
196
197              if(input.equalsIgnoreCase("F")) {
198                  pausarJuego();
199              } else {
200                  procesarMovimiento(input);
201              }
202          }
203      }
204  }
205
206  □     // Terminar partida
207  □     if(vidas == 0) {
208      System.out.println("Game Over! Te quedaste sin vidas.");
209  } else if(premiosRestantes == 0) {
210      System.out.println("Felicitaciones!!! Has ganado la partida.");
211
212      registrarPartida();
213  }
214
215  □     private static void mostrarPanelControl() {
216      System.out.println("\n==== PANEL DE CONTROL ====");
217      System.out.println("Usuario: " + nombreUsuario);
218      System.out.println("Puntaje: " + puntaje);
219      System.out.println("Vidas: " + vidas);
220      System.out.println("Premios restantes: " + premiosRestantes);
221      System.out.println("=====");
222  }
223
224  □     private static void mostrarTablero() {
225      System.out.println("\n" + "=".repeat(columnas * 2 + 1));
226      for(int i = 0; i < filas; i++) {
227          System.out.print(" | ");
228          for(int j = 0; j < columnas; j++) {
229              System.out.print(tablero[i][j] + " ");
230          }
231          System.out.println(" | ");
232      }
233      System.out.println(" " + "=".repeat(columnas * 2 + 1));
234  }
235
236  □     private static void procesarMovimiento(String tecla) {
237      int nuevaFila = pacmanFila;
238      int nuevaColumna = pacmanColumna;
239
240      switch(tecla) {
241          case "8": nuevaFila = (pacmanFila - 1 + filas) % filas; break;
```

```
241         case "8": nuevaFila = (pacmanFila - 1 + filas) % filas; break;
242         case "5": nuevaFila = (pacmanFila + 1) % filas; break;
243         case "6": nuevaColumna = (pacmanColumna + 1) % columnas; break;
244         case "4": nuevaColumna = (pacmanColumna - 1 + columnas) % columnas; break;
245         default: System.out.println("Tecla no valida"); return;
246     }
247
248     // Verificar si hay pared
249     if(tablero[nuevaFila][nuevaColumna] == PARED) {
250         System.out.println("Hay una pared No puedes moverte alli.");
251         return;
252     }
253
254     // Mover Pac-Man
255     tablero[pacmanFila][pacmanColumna] = VACIO;
256
257     // Verificar elementos en la nueva posición
258     char elemento = tablero[nuevaFila][nuevaColumna];
259
260     if(elemento == FANTASMA) {
261         vidas--;
262         System.out.println(";Te encontraste con un fantasma! Vidas restantes: " + vidas);
263     } else if(elemento == PREMIO_SIMPLE) {
264         puntaje += 10;
265         premiosRestantes--;
266         System.out.println(";Premio simple! +10 puntos");
267     } else if(elemento == PREMIO_ESPECIAL) {
268         puntaje += 15;
269         premiosRestantes--;
270         System.out.println(";Premio especial! +15 puntos");
271     }
272
273     pacmanFila = nuevaFila;
274     pacmanColumna = nuevaColumna;
275     tablero[pacmanFila][pacmanColumna] = PACMAN;
276 }
277 }
```

```
277
278     private static void pausarJuego() {
279         juegoPausado = true;
280         int opcion;
281
282         do {
283             System.out.println("\n==== JUEGO PAUSADO ====");
284             System.out.println("3. Regresar al juego");
285             System.out.println("4. Terminar partida");
286             System.out.print("Seleccione opcion: ");
287
288             opcion = scanner.nextInt();
289             scanner.nextLine();
290
291             switch(opcion) {
292                 case 3:
293                     juegoPausado = false;
294                     break;
295                 case 4:
296                     juegoActivo = false;
297                     juegoPausado = false;
298                     System.out.println("Partida terminada.");
299                     break;
300                 default:
301                     System.out.println("Opcion no valida");
302             }
303         } while(opcion != 3 && opcion != 4 && juegoPausado);
304     }
305
306     private static void registrarPartida() {
307         if(historialCount < 100) {
308             // Desplazar registros para mantener orden inverso (más reciente primero)
309             for(int i = historialCount; i > 0; i--) {
310                 historialNombres[i] = historialNombres[i-1];
311                 historialPuntos[i] = historialPuntos[i-1];
312             }
313         }
314     }
315 }
```

```
310         historialNombres[i] = historialNombres[i-1];
311         historialPuntos[i] = historialPuntos[i-1];
312     }
313
314     historialNombres[0] = nombreUsuario;
315     historialPuntos[0] = puntaje;
316     historialCount++;
317 }
318 }
319
320 private static void mostrarHistorial() {
321     if(historialCount == 0) {
322         System.out.println("\nNo hay partidas registradas.");
323         return;
324     }
325
326     System.out.println("\n==== HISTORIAL DE PARTIDAS ===");
327     System.out.println("Usuario\t\tPuntos");
328     System.out.println("-----");
329
330     for(int i = 0; i < historialCount; i++) {
331         System.out.println(historialNombres[i] + "\t\t" + historialPuntos[i]);
332     }
333     System.out.println("=====");
334
335     System.out.print("Presione Enter para continuar...");
336     scanner.nextLine();
337 }
338 }
```