

IMPLEMENTACIÓN DEL CODIGO:

Declaración de variables en públicas para poder darle uso a las misma para que pueda ser globales y que se puedan usar en diferentes bloques de códigos, así mismo también hay en algunas variables en donde se les esta asignando un valor como tal según el tipo que sea.

```
10 //Variables que serviran para almacenar la informacion:
11 //MEDIDAS DEL TABLERO PEQUEÑO
12 private static final int FILAS_PEQUEÑO = 5;
13 private static final int COLUMNAS_PEQUEÑO =6;
14 //MEDIDAS DEL TABLERO GRANDE
15 private static final int FILAS_GRANDE =10;
16 private static final int COLUMNAS_GRANDE=10;
17 //TIPO DE PREMIOS QUE TENRA EL TABLERO
18 private static final String fantasma = "@";
19 private static final String premio = "0";
20 private static final String premio_especial = "$";
21 private static final String pared = "X";
22 private static final String pacman = "<";
23 //OBJETO DE LAS LIBRERIAS DE JAVA
24 private static Scanner sc = new Scanner(System.in);
25 private static Random rand = new Random();
26 //VARIABLES DEL TOTAL DE VIDAS: para poder visualizar el menu de punteo
27 public static int vidas = 3;
28 public static int premiosRestante;
29 public static int punteo;
30 //Historial de Juego
31 public static String[] historialNombre = new String[3];
32 public static int[] historialPuntos = new int[3];
33 public static int totalPartidas;
```

Método Main: Donde se ejecutará toda la parte del código que está escrito adentro

Ciclo que funcionara para que el usuario pueda escoger cualquiera de las opciones que están en el menú principal que se mostrara, y según lo que el usuario ingreso eso mismo hara.

```
35 public static void main(String[] args) {
36     //Variables para almacenar info. del usuario
37     String nombreUsuario = "";
38     int opcionInicio = 0;
39     String tipoTablero = "";
40
41     //CICLO DEL MENU PRINCIPAL
42     while(opcionInicio!=3){
43         System.out.println("Bienvenido a pacman.");
44         System.out.println("Elija las opciones");
45         System.out.println("Elija 1 para crear el tablero");
46         System.out.println("Elija 2 para ver los puntos");
47         System.out.println("Elija 3 para salir");
48         opcionInicio = sc.nextInt();
49         //METODO DE LA OPCION QUE ELIJA EL USUARIO:
50         opcionElegida(opcionInicio, nombreUsuario, tipoTablero);
51     }
52 }
```

Método que tomara la decisión según el usuario haya ingresado con diferentes casos escritos.

CASO 1:

Crear el tablero: En caso que el usuario quiera realizar un nuevo tablero le pedirá el nombre, y tiene que indicar que tipo desea implementar del mismo.

Dependiendo del cual desea el usuario, tiene una condición para los dos tipos de tablero que hay ya que no tiene las mismas longitudes en donde se le están asignando las longitudes.

```
switch (opcionInicio) {  
    //LAS OPCIONES CON SUS DIFERENTES METODOS DEL MENU  
    case 1:  
        //CASO 1: CREAR TABLERO: Ingresando su nombre, tipo de Tablero, interaccion del usuario.  
        System.out.println("Ingrese su nombre: ");  
        nombreUsuario = sc.next();  
        //TIPO DE TABLERO  
        System.out.println("B. Tablero de 5x6");  
        System.out.println("G. Tablero de 10x10");  
        System.out.println("Tipo de Tablero: ");  
        tipoTablero = sc.next();  
        int filas = 0;  
        int columnas = 0;  
        if(tipoTablero.equals("P")){  
            filas = FILAS_PEQUEÑO;  
            columnas = COLUMNAS_PEQUEÑO;  
        } else if(tipoTablero.equals("G")){  
            filas = FILAS_GRANDE;  
            columnas = COLUMNAS_GRANDE;  
        } else {  
            System.out.println("Tipo inválido, se usará pequeño por defecto");  
        }  
        //CANTIDAD DE PREMIOS, PAREDES, FANTASMAS QUE TENDRA EL TABLERO.  
        System.out.println("Por favor, ingresa los siguientes valores:");  
    }
```

Asignación de los diferentes premios, trampas para que se dispersen en el tablero que se va a crear y en base a eso también no tiene que exceder de un limite de la cantidad que ingrese.

```
numeroPremios = asignarCantidades("premios", 0.4, filas, columnas);  
numeroPremiosEspecial = asignarCantidades("premios Especiales", 0.4, filas, columnas);  
  
//Asigna paredes  
numeroParedes = asignarCantidades("paredes", 0.2, filas, columnas);  
  
//Asigna trampas  
numeroTrampas = asignarCantidades("fantasmas", 0.4, filas, columnas);  
  
System.out.println("Creando tablero...");
```

Ciclo para que vaya llenando la matriz, en base a las filas, columnas que se vayan a ingresar

```
//MATRIZ DIMENSIONAL DONDE SE ALMACENARAN LAS FILAS Y COLUMNAS  
String[][] tablero = new String[filas][columnas];  
//CICLO DONDE SE VA IR RELLENANDO LA MATRIZ  
for(int i = 0; i < tablero.length; i++){  
    for(int j = 0; j < tablero[0].length; j++){  
        tablero[i][j]= " ";  
    }  
}
```

Llamada al método que va ir colocando los diferentes premios, paredes, fantasmas en la matriz

```

108 //ASIGNACION DE LOS PREMIOS, PAREDES, FANTASMAS.
109 //PREMIOS
110 colocarElemento(tablero, premio, numeroPremios);
111 colocarElemento(tablero, premio_especial, numeroPremiosEspecial);
112 //PAREDES
113 colocarElemento(tablero, pared, numeroParedes);
114 //FANTASMAS
115 colocarElemento(tablero, fantasma, numeroTrampas);
116 //MOSTRANDO EL TABLERO
117 imprimirTablero(tablero);
118 System.out.println();

```

Bucle para el posicionamiento del personaje donde el usuario deberá asignar en que parte del tablero desea posicionarse.

```

120 //ALMACENAMIENTO DE LOS DIFERENTES MOVIMIENTOS DEL PERSONAJE.
121 int personajeFila;
122 int personajeColumna;
123 //BUCLE PARA QUE MUESTRE EN DONDE VA IR COLOCANDO AL PACMAN
124 do{
125     System.out.println("Elegir donde desea colocar al personaje:");
126     System.out.print("Filas: ");
127     personajeFila = sc.nextInt();
128     System.out.print("Columnas: ");
129     personajeColumna = sc.nextInt();
130     //EN CASO QUE NO SEA VALIDA LA FILA Y LA COLUMNA
131     if(verificacionPosicionPersonaje(personajeFila, personajeColumna, tablero) == false){
132         System.out.println("Posición no valida, escoge otro lugar");
133     }
134 }while(verificacionPosicionPersonaje(personajeFila, personajeColumna, tablero) == false);
135
136 System.out.println("Personaje colocado exitosamente!");
137 tablero[personajeFila-1][personajeColumna-1] = "P";

```

Ciclo que funcione para que el usuario vaya interactuando y moviendo el personaje (Arriba, Abajo, Derecha, Izquierda)

```

142 String movimientoPersonaje;
143 boolean jugando = true;
144 do{
145     //Menú de nombre del usuario, vidas, puntos
146     EstadoJuego(nombreUsuario);
147     imprimirTablero(tablero);
148     //MOVIMIENTOS QUE SE PUEDEN IR REALIZANDO
149     System.out.println("Mueve tu personaje");
150     System.out.println("8. Arriba");
151     System.out.println("5. Abajo");
152     System.out.println("6. Derecha");
153     System.out.println("4. Izquierda");
154     System.out.print(">");
155     movimientoPersonaje = sc.next();

```

Posible pausa que pueda realizar el usuario cuando esté realizando los movimientos; si en dado caso quiere finalizar la partida o desea rede anudar la partida

```

157 //METODO SI EN UN DADO CASO DESEA PAUSAR EL JUEGO
158 if(movimientoPersonaje.equals("F")){
159     mostrarMenuPausa();
160     int opcionPausa;
161     //POSIBLES OPCIONES QUE EL USUARIO PUEDE HACER DURANTE LA PAUSA DEL JUEGO
162     System.out.println("Ingrese una opción Correcta: ");
163     opcionPausa = sc.nextInt();
164     System.out.println("");
165     //POSIBLES OPCIONES EN EL MENU DE PAUSA
166     switch (opcionPausa) {
167         case 3:
168             System.out.println("Regresando...");
169             break;
170         case 4:
171             System.out.println("Juego Terminado...");
172             //PARA QUE VAYA ALMACENANDO EL NOMBRE, EL TOTAL DE PUNTOS
173             historialNombre[totalPartidas] = nombreUsuario;
174             historialPuntos[totalPartidas] = punteo;
175             totalPartidas = totalPartidas +1;
176             jugando = false;
177             break;
178         default:
179             System.out.println("Opción Invalida");
180     }
181     continue;
182 }

```

Posibles casos en donde el juego termine, en caso que se llegue a quedar sin vidas, o logre ganar comiendo todos los premios seleccionados.

```

183 //LLAMADA DE METODO MOVER PERSONAJE CON SUS PARAMETROS
184 moverPersonaje(tablero,posicion,movimientoPersonaje);
185 //CONDICIÓN PARA FINALIZAR LA PARTIDA SI EN DADO CASO SE QUEDA SIN VIDA
186 //O SE COME TODOS LOS PREMIOS
187 if(vidas<=0){
188     System.out.println("Te quedastes sin vidas!");
189     jugando = false;
190     historialNombre[totalPartidas] = nombreUsuario;
191     historialPuntos[totalPartidas] = punteo;
192     totalPartidas = totalPartidas +1;
193 }
194 else if(premiosRestante == 0){
195     System.out.println("Ganaste!");
196     historialNombre[totalPartidas] = nombreUsuario;
197     historialPuntos[totalPartidas] = punteo;
198     totalPartidas = totalPartidas +1;
199     jugando = false;
200 }
201 }while(jugando);
202 break;
203

```

CASO 2: Donde el usuario podrá ver la historia de las partidas que ha tenido o simplemente si desea dar por terminado el juego.

```

203         break;
204     case 2:
205         //MOSTRAR EL HISTORIAL DE LOS JUEGOS DEL USUARIO
206         mostrarHistorial();
207         break;
208     case 3:
209         //TERMINARA EL PROGRAMA COMO TAL
210         opcionInicio = 0;
211         System.out.println("El Programa a Finalizado");
212         break;
213     default:
214         System.out.println("Opción inválida. Intenta de nuevo");
215     }
216 }
217

```

Método implementado para el movimiento del personaje que se encargara de la sentencia lógica para que se pueda mover por la derecha, izquierda, arriba o abajo.

```

219 public static boolean moverPersonaje(String[][] tablero, int[] posicion, String direccion){
220     int filaActual = posicion[0];
221     int columnaActual = posicion[1];
222
223     int nuevaFila = filaActual;
224     int nuevaColumna = columnaActual;
225     //DEPENDIENDO DEL MOVIMIENTO DEL USUARIO
226     switch(direccion){
227     case "8": //Arriba
228         nuevaFila = nuevaFila -1;
229         break;
230     case "6": //Derecha
231         nuevaColumna = nuevaColumna +1;
232         break;
233     case "5": //Abajo
234         nuevaFila = nuevaFila +1;
235         break;
236     case "4": //Izquierda
237         nuevaColumna = nuevaColumna -1;
238         break;
239     default:
240         System.out.println("Movimiento no válido");
241         return false;
242     }
243 }

```

Verificación de los bordes, para que pueda pasar de un borde inferior al superior y viceversa

```

243 //Verificar Límites; DE LAS COLUMNAS
244 //Cantidad de filas, columnas del tablero
245 int filas = tablero.length;
246 int columnas = tablero[0].length;
247 //VERIFICACION DE LIMITE DE LA FILAS
248 if(nuevaFila < 0){
249     nuevaFila = filas - 1; //Sale arriba -> aparece abajo
250 }else if(nuevaFila >= filas){
251     nuevaFila = 0; //Sale abajo -> aparece arriba
252 }
253 //VERIFICACION DE LIMITES DE LA COLUMNAS
254 if (nuevaColumna < 0) {
255     nuevaColumna = columnas - 1; // Sale izquierda -> aparece derecha
256 }
257 else if (nuevaColumna >= columnas) {
258     nuevaColumna = 0; // Sale derecha -> aparece izquierda
259 }

```

Implementación de verificar si en caso llega a chocar con algo en el movimiento nuevo que está realizando el usuario, en caso que choque con un fantasma llegue restarle puntos y si en dado caso come premios que le vaya sumando la cantidad de premios.

```

261 //EN CASO QUE A TOPA CON ALGO EN EL MOVIMIENTO NUEVO QUE SE LE ESTA DANDO
262 if(tablero[nuevaFila][nuevaColumna].equals(fantasma)){
263     System.out.println("=====");
264     System.out.println("Te atrapó un fantasma!");
265     //SE VA RESTANDO LA VIDA
266     vidas = vidas - 1;
267     tablero[nuevaFila][nuevaColumna] = " ";
268 }
269 } else if(tablero[nuevaFila][nuevaColumna].equals(premio)){
270     System.out.println("=====");
271     System.out.println("Atrapaste un Premio!");
272     //SUMA DE PUNTEO
273     punteo = punteo + 10;
274     premiosRestante = premiosRestante - 1;
275 }
276 }else if(tablero[nuevaFila][nuevaColumna].equals(pared)){
277     System.out.println("=====");
278     System.out.println("Te Atrapó una Pared");
279     return false;
280 }else if(tablero[nuevaFila][nuevaColumna].equals(premio_especial)){
281     System.out.println("=====");
282     System.out.println("Atrapaste un Premio Especial");
283     //SUMA DE PREMIOS ESPECIALES
284     punteo = punteo + 15;
285     premiosRestante = premiosRestante - 1;
286 }
287 //IMPRIMIENDO LA NUEVA POSICIÓN Y ELIMINANDO LOS PREMIOS O CON LO QUE TOPO.
288 tablero[filaActual][columnaActual] = " ";
289 tablero[nuevaFila][nuevaColumna] = pacman;
290
291 posicion[0] = nuevaFila;
292 posicion[1] = nuevaColumna;
293
294 return true;
295 }

```

Implementación del método pausa, en donde lo único que hará es mostrar las posibles opciones cuando el programa este pausado.

```
296 //METODO EN CASO QUE EL USUARIO PONGA EN PAUSA EL JUEGO
297 public static void mostrarMenuPausa() {
298     System.out.println("===== JUEGO EN PAUSA =====");
299     System.out.println("3. Regresar");
300     System.out.println("4. Terminar partida");
301
302 }
```

Implementación del método para que muestre el tablero y depende del usuario, porque existen dos tipos de tableros en los cuales se puede jugar.

```
304 //METODO PARA IMPRIMIR EL TABLERO SEGUN EL TIPO DE FILA Y COLUMNA
305 public static void imprimirTablero(String[][] tablero) {
306     int filas = tablero.length;
307     int columnas = tablero[0].length;
308     // Borde superior
309     for (int j = 0; j < columnas + 2; j++) {
310         System.out.print("--");
311     }
312     System.out.println();
313     // Filas del tablero
314     for (int i = 0; i < filas; i++) {
315         System.out.print("|");
316
317         for (int j = 0; j < columnas; j++) {
318             System.out.print(tablero[i][j] + " ");
319         }
320         System.out.println("|");
321     }
322     // Borde inferior
323     for (int j = 0; j < columnas + 2; j++) {
324         System.out.print("--");
325     }
326     System.out.println();
327 }
```

Método que el usuario podrá ver las vidas que tiene disponibles y la cantidad de premios que ha ido ganando.

```
328 //Impresion del tabla de nombre del usuario, vidas, punteo
329 public static void EstadodeJuego(String nombreUsuario) {
330     System.out.println("=====");
331     System.out.println("Usuario: " + nombreUsuario );
332     System.out.println("Punteo: " + punteo);
333     System.out.println("Vidas: " + vidas );
334     System.out.println("=====");
335 }
336
```

Implementación del método que servirá para verificar la posición del usuario.


```

336
337 //Verificar la posicion del personaje en donde se va ir posicionando
338 public static boolean verificacionPosicionPersonaje(int x, int y, String[][] tablero){
339     if(tablero[x-1][y-1].equals(" ")){
340         return true;
341     }
342     return false;
343
344 }

```

Método que servirá para implementar las cantidades de los premios según el tipo de tablero que se vaya a jugar.

```

345 //Asignación de los premios del tablero pequeño
346 public static int asignarCantidades(String objeto, double porcentaje, int filas, int columnas){
347     boolean bandera = false;
348     int cantidad = 1;
349     while(bandera == false){
350         System.out.print("Elige la cantidad de: "+objeto+": ");
351         cantidad = sc.nextInt();
352         bandera = true;
353         int maxPremios = (int)(filas * columnas * porcentaje);
354         if (cantidad < 1 || cantidad > maxPremios) {
355             System.out.println("Por favor ingresa una cantidad correcta.");
356             bandera = false;
357         }
358     }
359     return cantidad;
360
361 }

```

Método que servirá para que se vayan almacenando los diferentes objetos con su respectiva cantidad que se le asigne.

```

363 public static void colocarElemento(String[][] tablero, String simbolo, int cantidad) {
364     int colocados = 0;
365     int filas = tablero.length;
366     int columnas = tablero[1].length;
367
368     while (colocados < cantidad) {
369         int fila = rand.nextInt(filas);
370         int columna = rand.nextInt(columnas);
371
372         if (tablero[fila][columna].equals(" ")) {
373             tablero[fila][columna] = simbolo;
374             colocados++;
375         }
376     }
377 }

```

Implementación del método que servirá para ir almacenando y guarde el punto del usuario y nombre del mismo

```

378 //HISTORIAL DE PARTIDAS
379 public static void mostrarHistorial(){
380     System.out.println("===== HISTORIAL DE PARTIDAS =====");
381     //EN CASO QUE ESTE EN 0 LAS PARTIDAS
382     if(totalPartidas == 0){
383         System.out.println("No hay movimientos registrados.");
384         return;
385     }
386     //EN CASO QUE NO ESTE EN 0 EL HISTORIAL DE PARTIDAS; imprimira los datos
387     for(int i = totalPartidas - 1; i >= 0; i--){
388         System.out.println(historialNombre[i] + " - Puntos: " + historialPuntos[i]);
389     }
390     System.out.println("=====");
391 }
392 }

```


