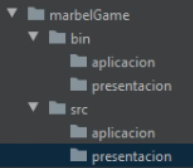
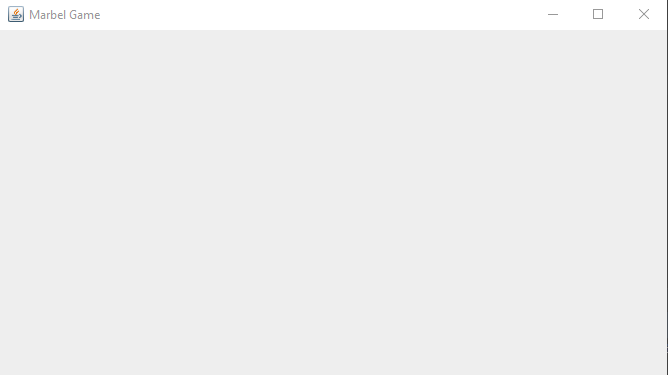
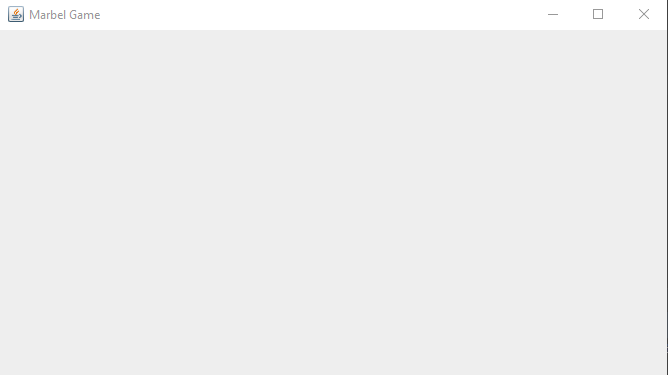
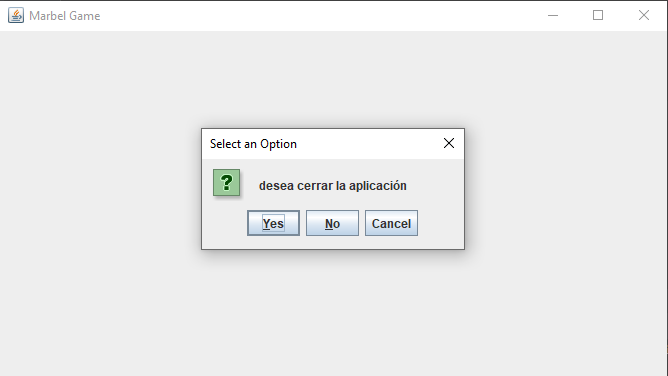
# **Desarrollo**

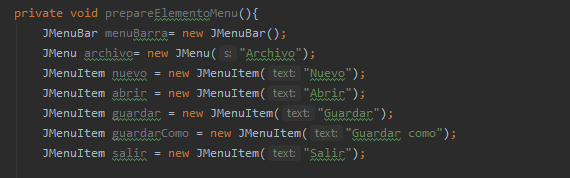
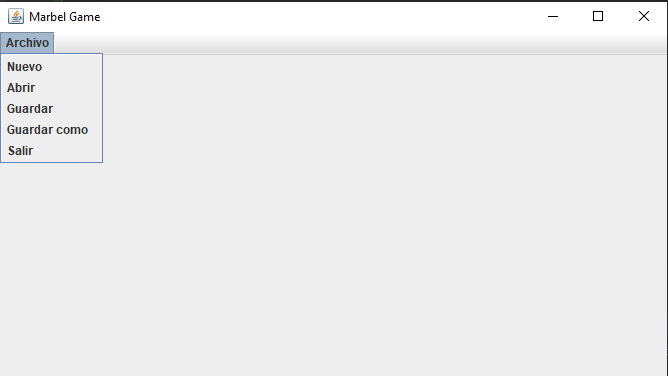
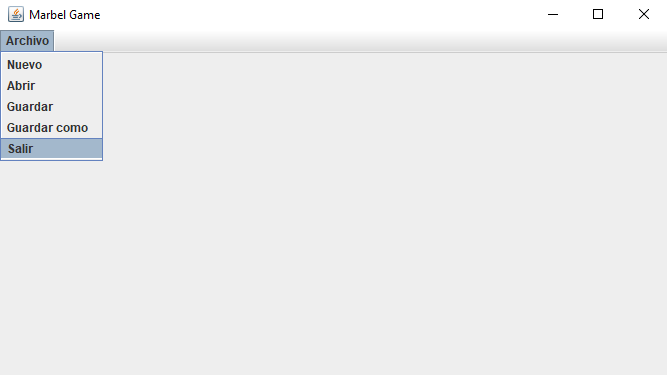
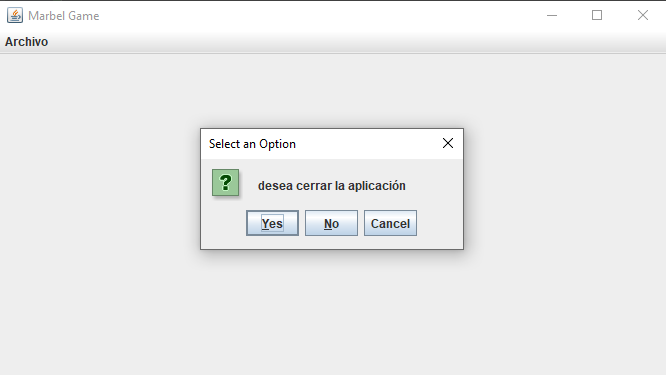
## **Directorios**

1. Se creó la estructura de directorios:

## **Ciclo 0: Ventana vacía – Salir**

1. ventana inicial de MarbelGame
2. Modificación del tamaño de la ventana ocupando un cuarto de pantalla y ubicada en el centro.
3. Inicialmente, cuando cerramos la ventana, la ejecución del programa no termina, esto se debe a que el comportamiento del JFrame por defecto al cerrar la ventana es HIDE\_ON\_CLOSE, sin embargo, necesitamos que su comportamiento sea EXIT\_ON\_CLOSE.
4. Podemos usar la función *setDefaultCloseOperation* de la clase JFrame para cambiar su comportamiento al cerrar, en este caso se cambiaría de HIDE\_ON\_CLOSE a EXIT\_ON\_CLOSE.
5. Usamos el método *ShowConfirmDialog* de la clase *JOptionPane*, allí podemos poner el cuadro de confirmación para que se confirme la salida del programa.

## **Ciclo 1: Ventana con menú – Salir**

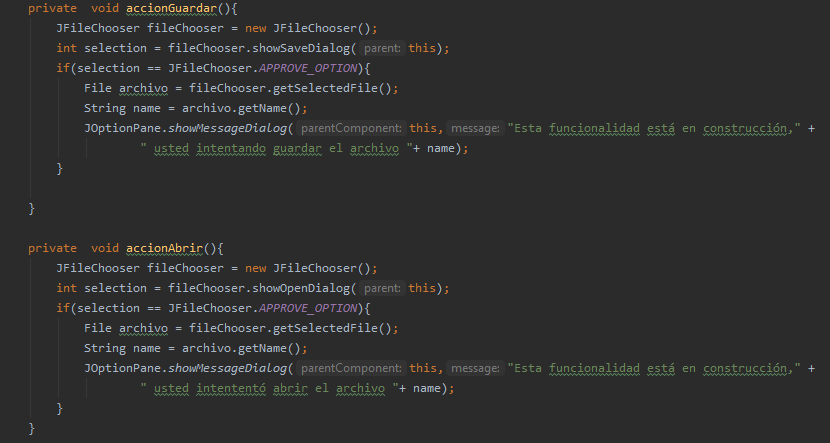
1. Se definieron los elementos necesarios para el menú, sin embargo, aún no se han asignado y no es funcional.
2.  Se asignaron los elementos del menú y este fue el resultado.
3. Reutilizamos la funcionalidad de cerrar la ventana y se la aplicamos al botón “Salir”

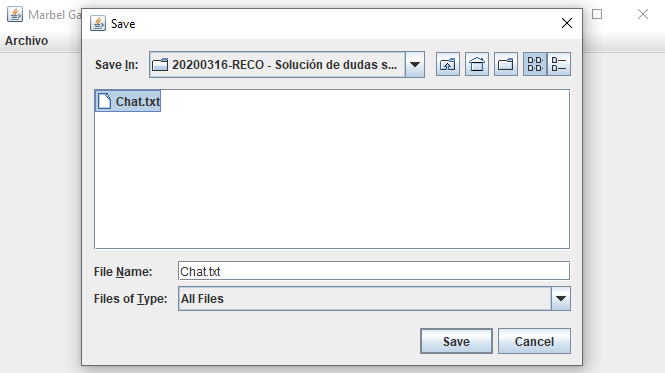
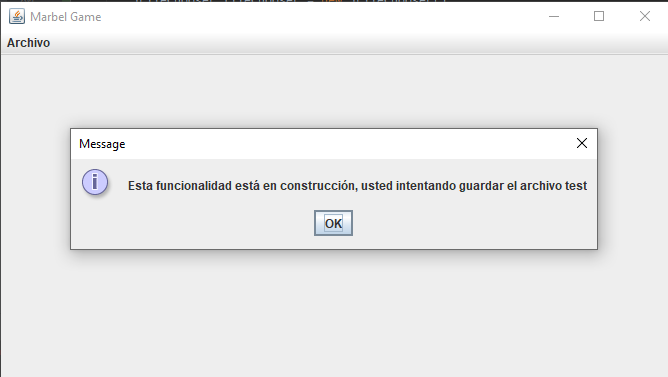
## **Ciclo 2: Salvar y abrir**

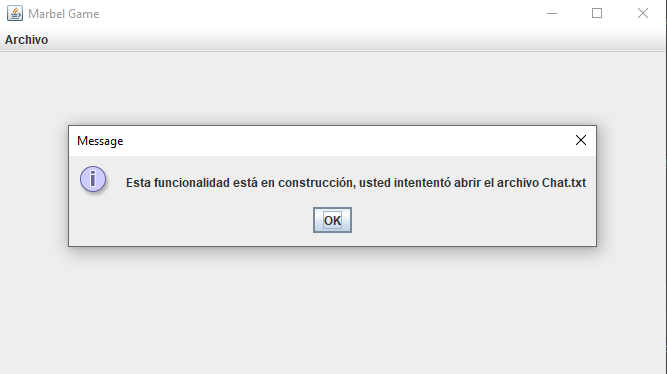
1. *showOpenDialog*: Crea una ventana para abrir un archivo.

*showSaveDialog*: Crea una ventana para guardar un archivo.

*getSelectedFile:* Retorna el archivo seleccionado.

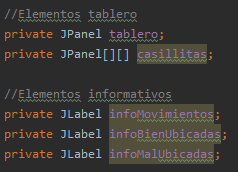
1. Se implementó de la siguiente forma, mostrando los respectivos mensajes para cada acción.
2. Abrir y aguardar archivo.

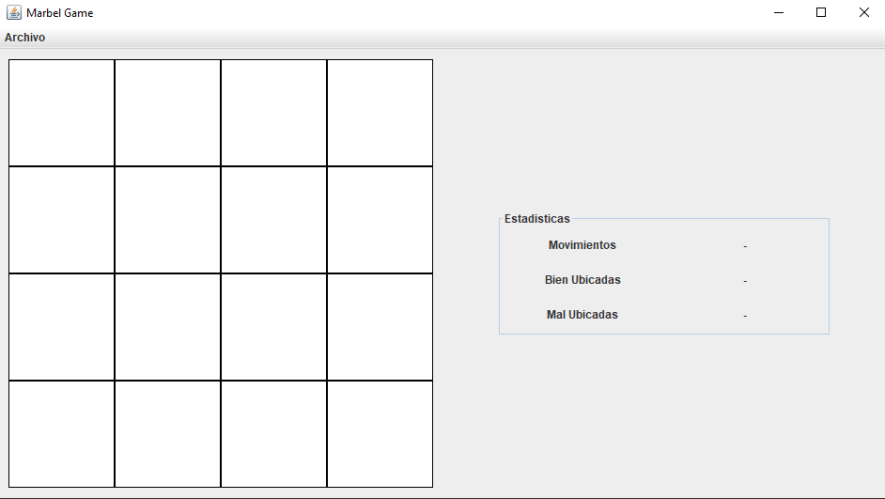
Guardar:

Abrir:

## **Ciclo 3: Forma de la ventana principal**

1. atributos privados necesarios

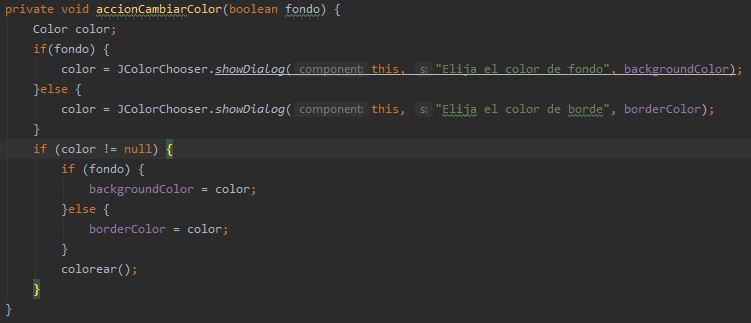


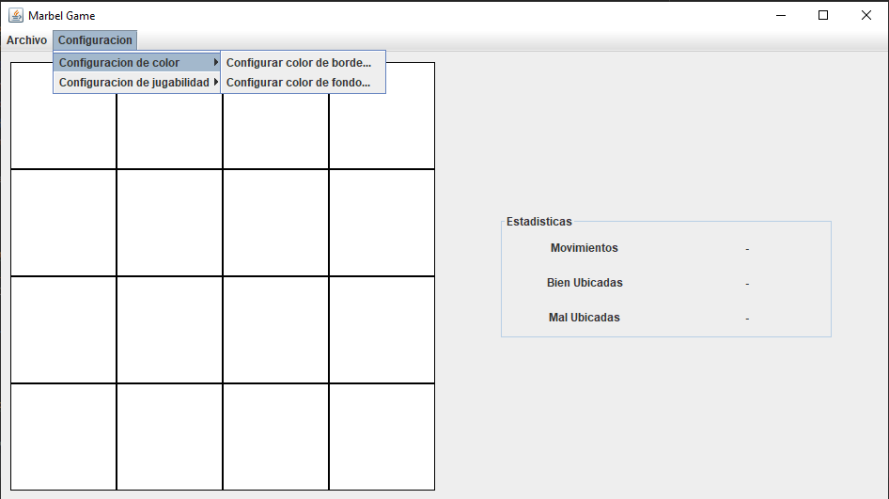
2. tablero inicial.

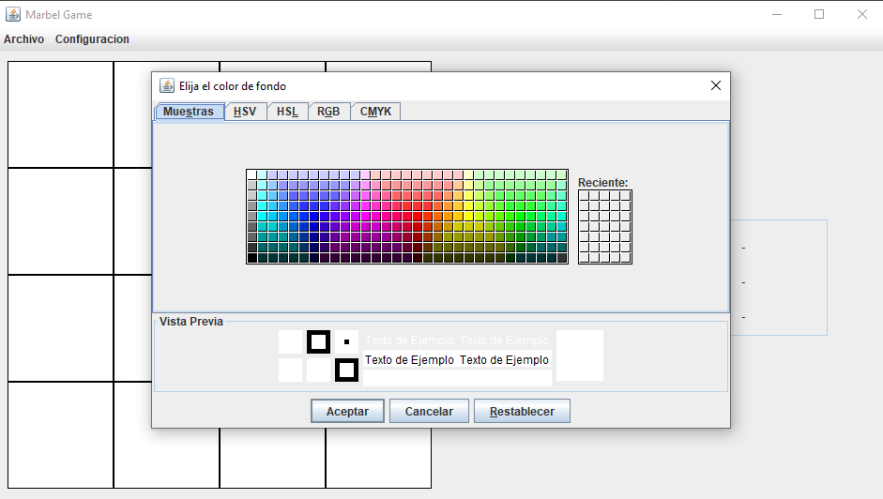
## **Ciclo 4: Cambiar color**

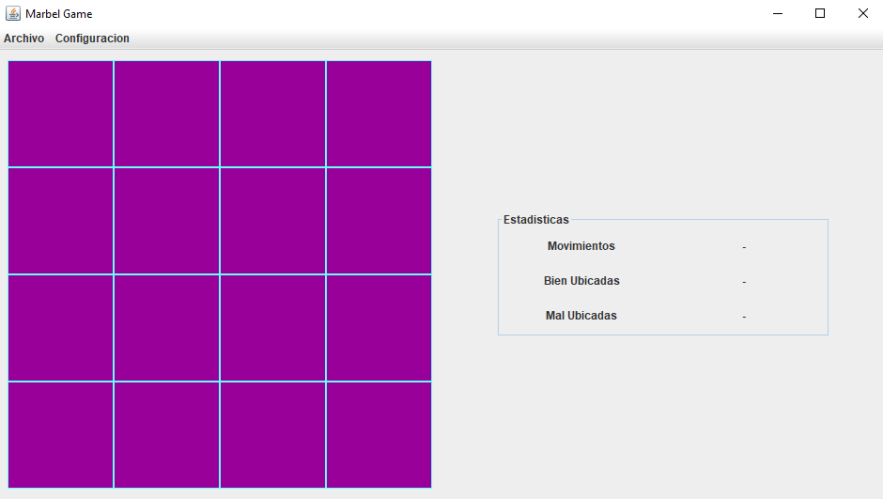
1. las acciones fueron implementadas en el menú de juego, se utilizaron dos menús uno para configurar el color de borde y el color de fondo, cada uno de ellos con un oyente, donde se implemente la clase JColorChooser para seleccionar el color.
2. comportamiento de JColorChooser y showDialog.

* JColorChooser : proporciona un panel de controles diseñados para permitir a un usuario manipular y seleccionar un color.
* showDialog : Es un método estático el cual muestra un cuadro de diálogo de selección de color modal y se bloquea hasta que se oculte el cuadro de diálogo. Si el usuario presiona el botón "Aceptar", este método oculta / elimina el cuadro de diálogo y devuelve el color seleccionado. Si el usuario presiona el botón "Cancelar" o cierra el cuadro de diálogo sin presionar "Aceptar", este método oculta / elimina el cuadro de diálogo y devuelve el valor nulo.

1. se implementó de la siguiente forma los elementos necesarios ora cambiar el color del tablero.

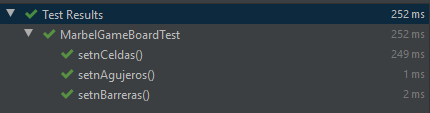
1. cambiar color del tablero



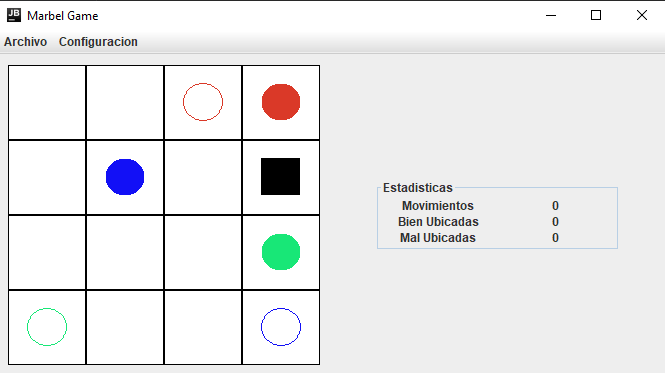


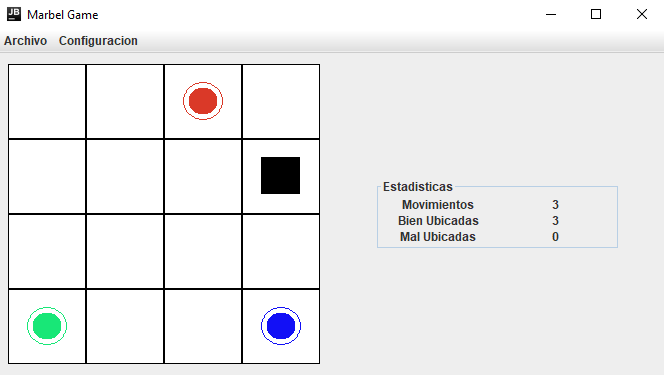
## **Ciclo 5: Modelo MarbelGame**

1. – *código*--
2. Pruebas

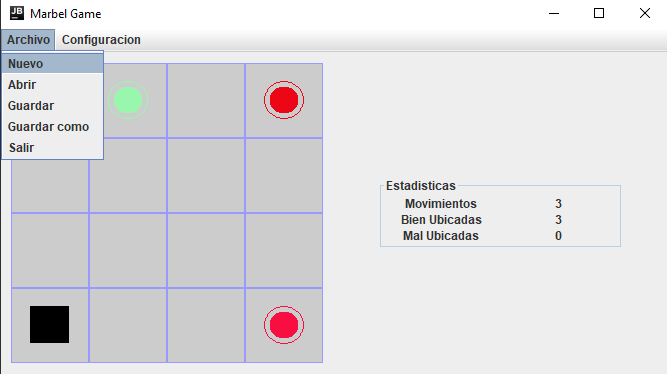
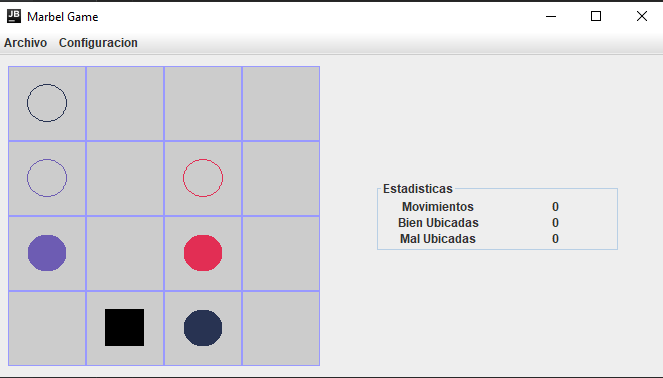


## **Ciclo 6: Jugar**

1. – *código*--
2. – *código*--
3. Se requiere colocar un MarbelGameGUI en la capa de presentación e implementar unos oyentes para las flechas del teclado, donde se realizan los diferentes movimientos de las canicas.
4. – *código*--
5. Pantalla de implantación de juego

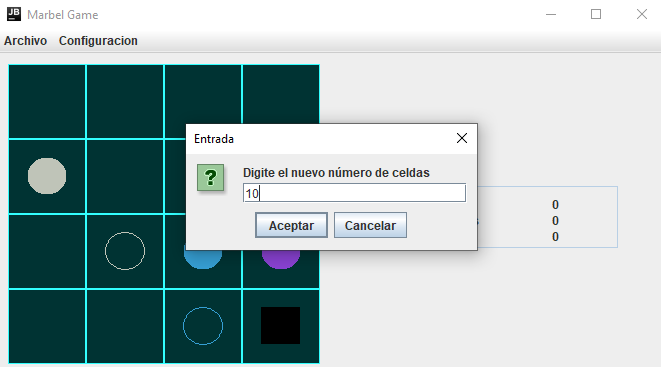
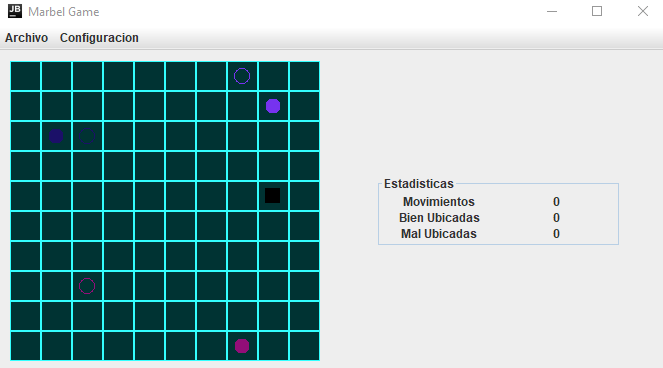


## **Ciclo 7: Reiniciar**

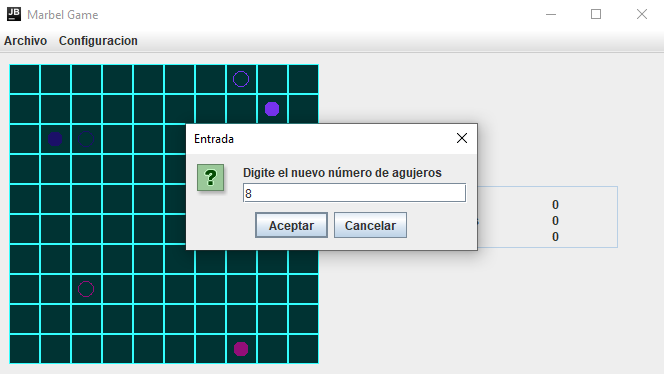
1. – *código*--
2. – *código*—
3. Reiniciar el juego

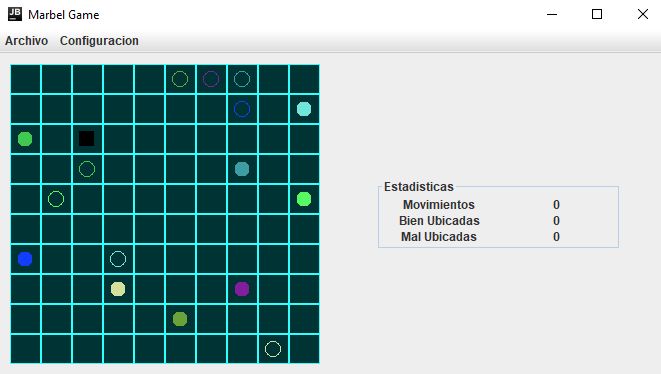
## **Ciclo 8: Cambiar la configuración del juego**

1. se utilizaron tres atributos cada uno para el numero de Caldas, agujeros y canicas en la clase MabelGameBoad, adicional se implementaron tres métodos en la clase MarbelGameGUI, para modificar el valor de cada uno de los tres elementos dependiendo del valor digitado por el usuario.
2. – *código*--
3. configuración del tablero

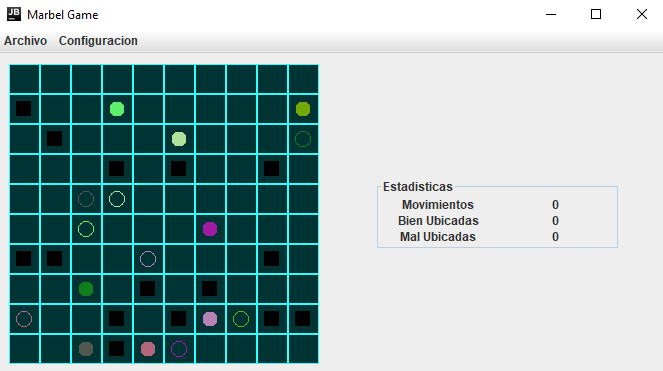
*número celdas:*

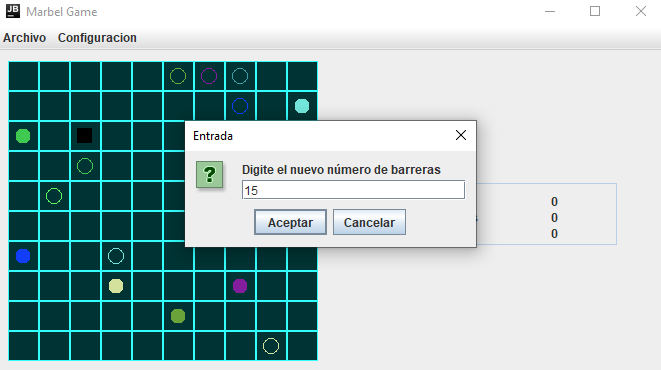
*Numero de agujeros:*

**

**

*Numero de barreras:*

**

**

## **RETROSPECTIVA**

1. *¿Cuál fue el tiempo total invertido en el laboratorio por cada uno de ustedes?*

(10/10)

2. *¿Cuál es el estado actual del laboratorio? ¿Por qué?*

Completo

3. *Considerando la práctica XP del laboratorio ¿por qué consideran que es importante?*

Las pruebas unitarias y de aceptación ya que validamos el correcto funcionamiento del laboratorio, donde verificamos si hay algún fallo para corregirlo y garantizar que los requerimientos solicitados hayan sido cumplidos satisfactoriamente.

4. *¿Cuál consideran fue su mayor logro? ¿Por qué? ¿Cuál consideran que fue su mayor*

*problema? ¿Qué hicieron para resolverlo?*

Implementar la parte de la interfaz del juego, debido a que había muchas conficiones en la implantación, luego se realizó una investigación acerca de el funcionamiento de cada componente a utilizar y su correcta implementación.

5. ¿*Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los*

*resultados?*

Plantear diferentes soluciones he implementaciones para el correcto funcionamiento.