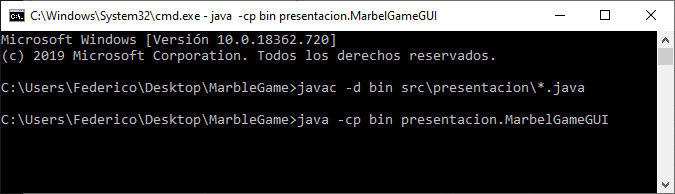
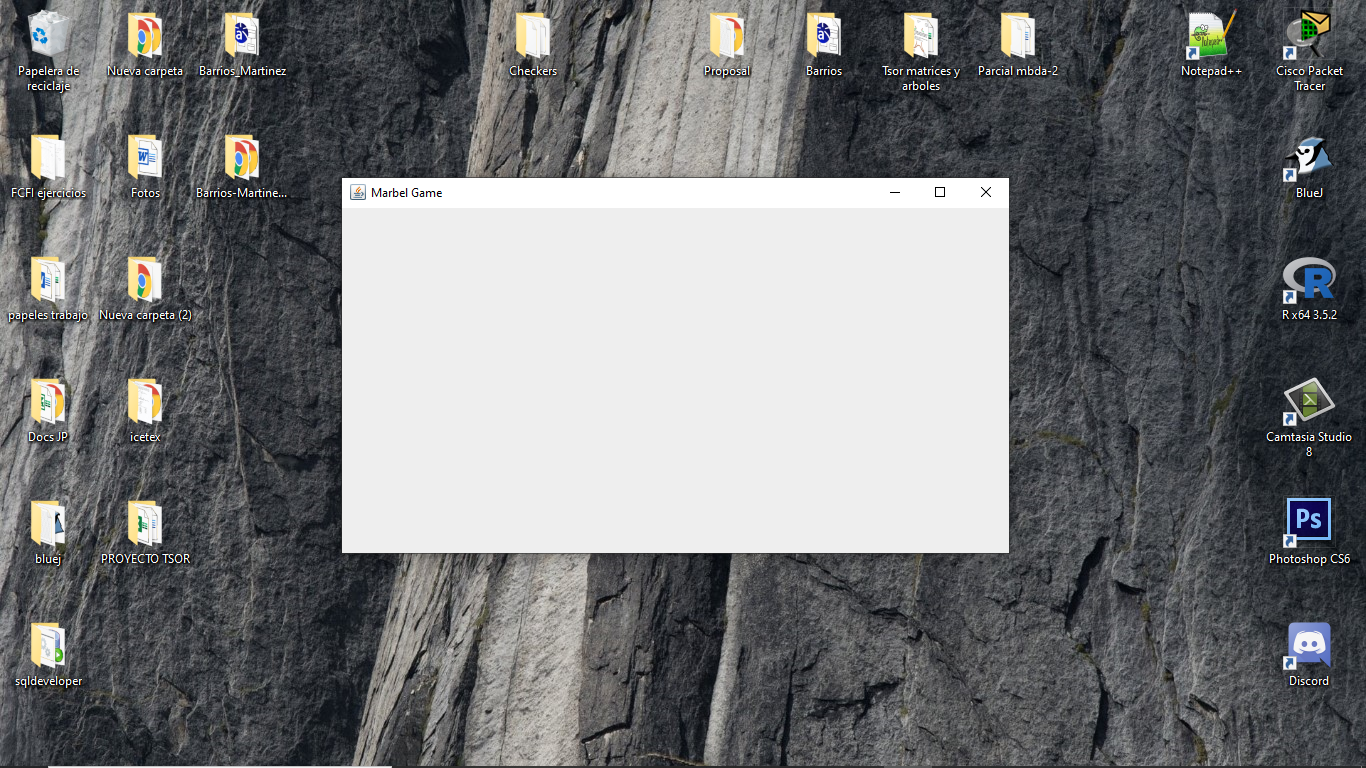
Ciclo 0

1.





2.

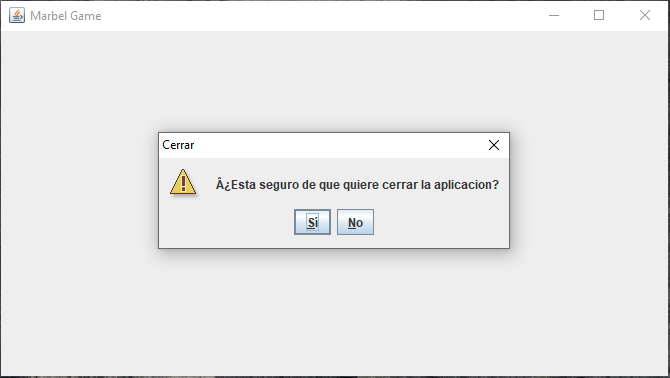


3.

* La ventana si se cierra, pero la ejecución no termina.
* Para terminar la ejecución se puede usar el método setDefaultCloseOperation que sirve para determinar que sucede cuando se oprime al hacer clic en el botón de cerrar, entonces ponemos el parámetro **EXIT\_ON\_CLOSE** para que se cierre y finalice la ejecución al cerrar la ventana.

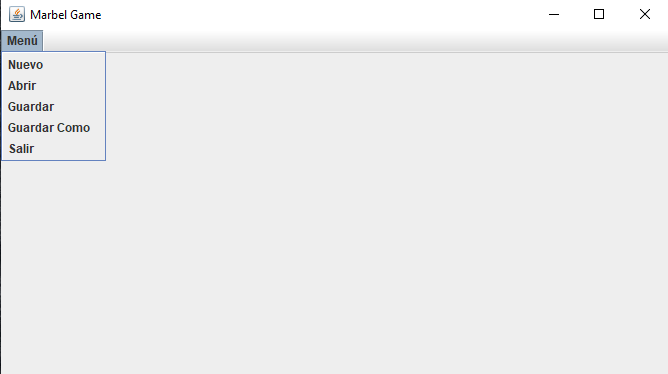
4. setDefaultCloseOperation sirve para determinar lo que debe suceder cuando se haga clic en el botón de cerrar la ventana.

5.

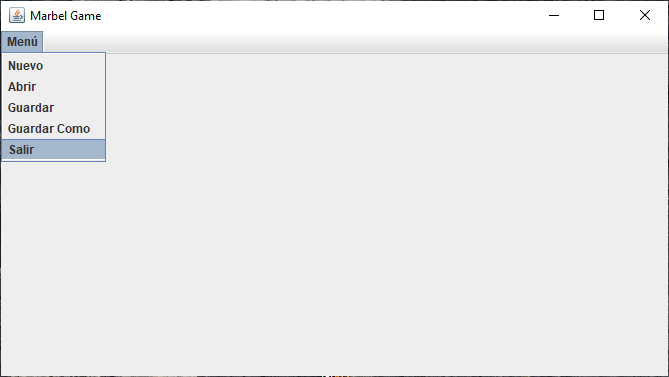


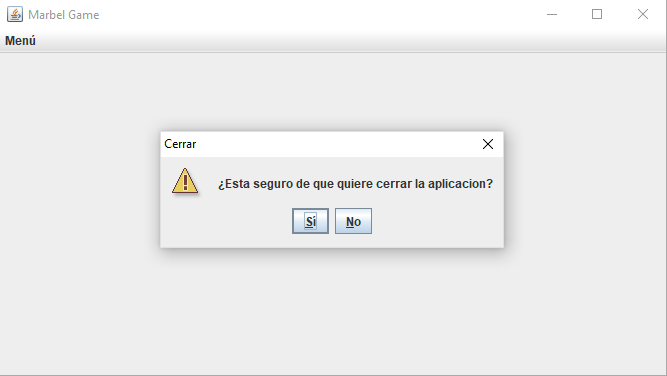
Ciclo 1

2.



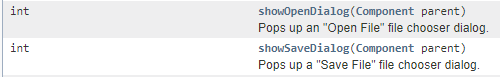
3.





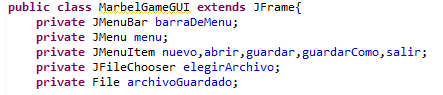
Ciclo 2

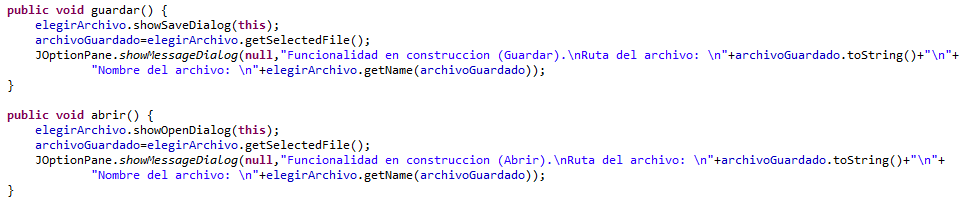
1.



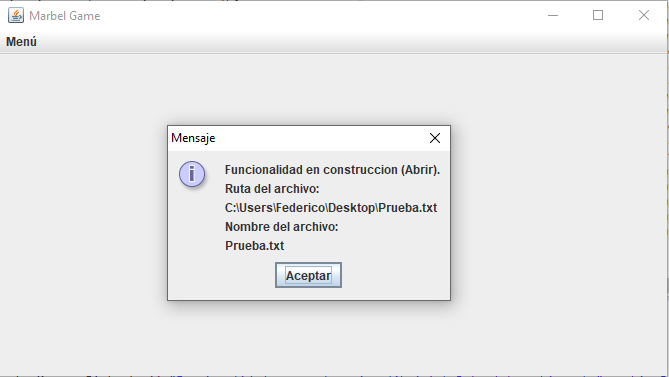


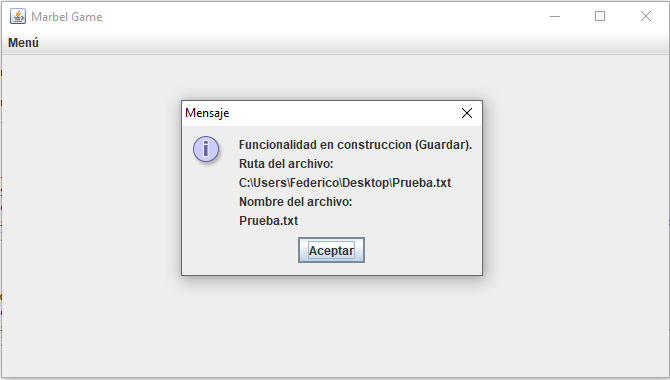
2.





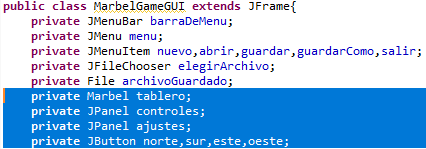
3.





Ciclo 3

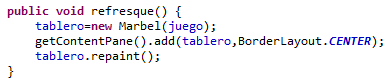
1.

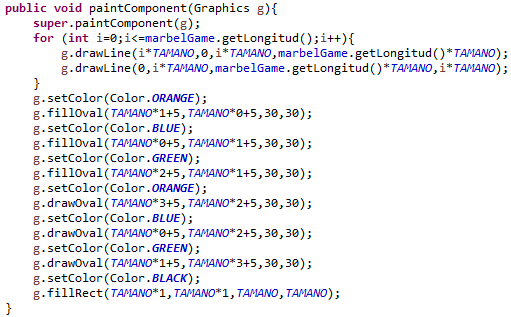


Marbel es una clase que extiende de JPanel.

2.

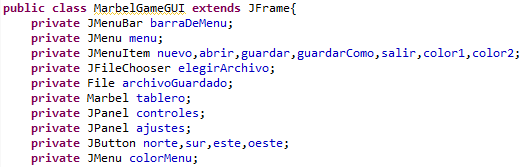






Ciclo 4

1.

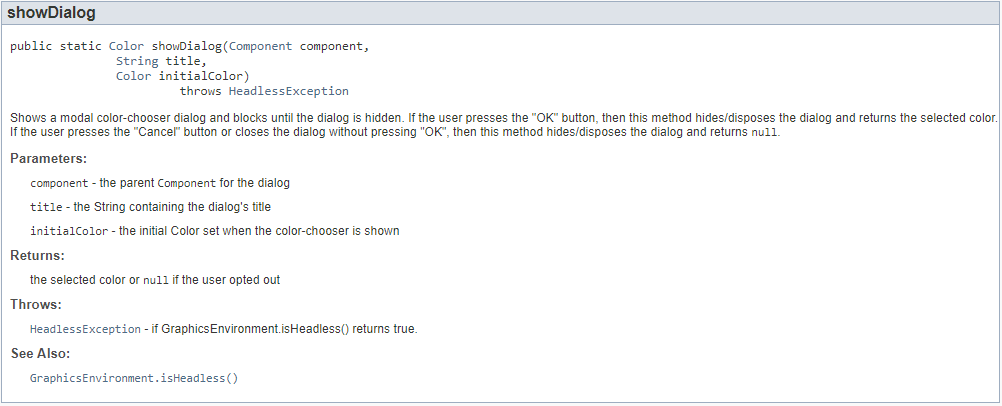


colorMenu: Para cambiar de color decidí agregar a la barra de menú una nueva opción para poder realizar esta acción.

color1: Este item del menú Color es para cambiar el color de fondo del tablero.

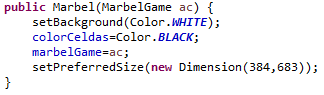
Color2: Este item del menú Color es para cambiar el color de las celdas del tablero.

2.



3.

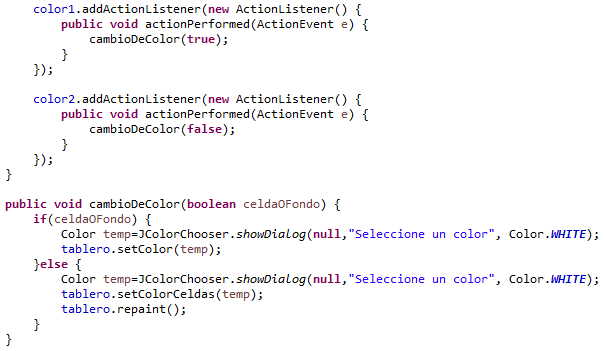
Al inicio los colores son blanco y negro.



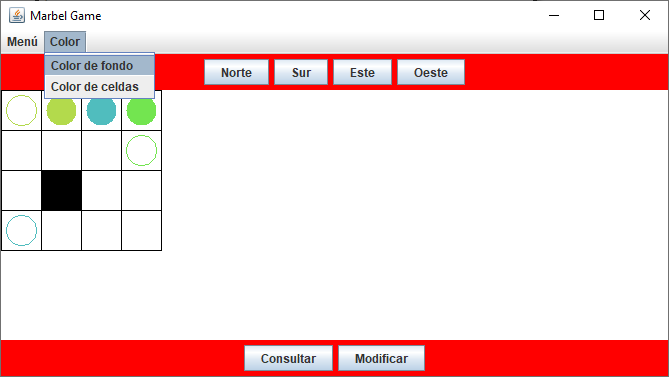
Se añaden los componentes al menú.

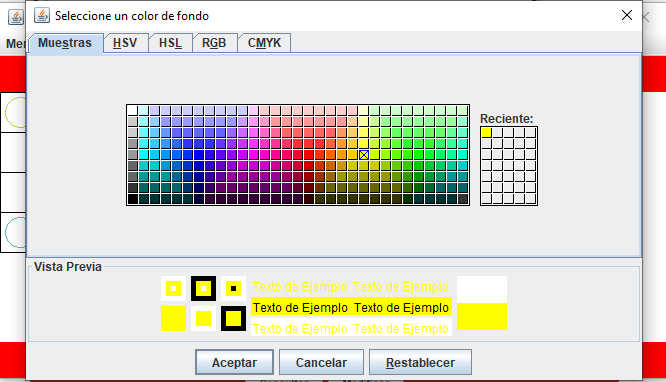


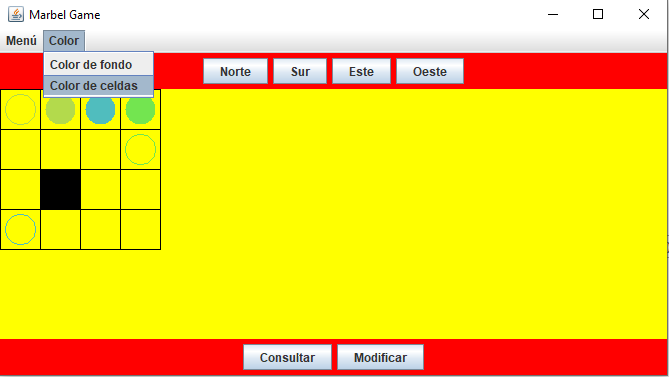
Se hacen los Listeners.

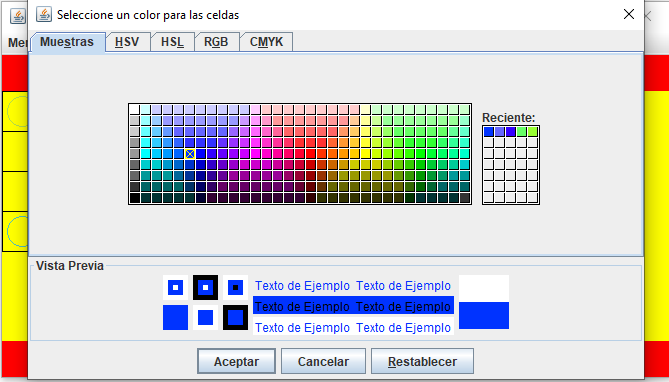


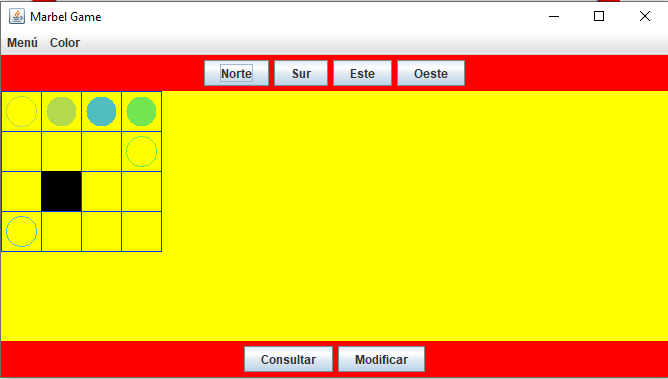
4.







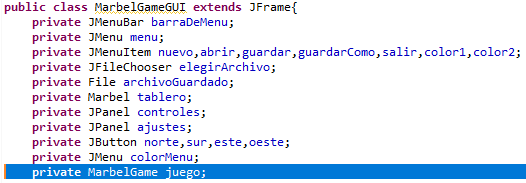




Ciclo 5

Ciclo 6

1.



2.

El método refresque delega para que la clase Marbel que es un JPanel sea la que pinte sobre el panel tablero. Para este punto se mejoró el código para que se pinte cualquier tablero realizado por la capa de aplicación.



3.

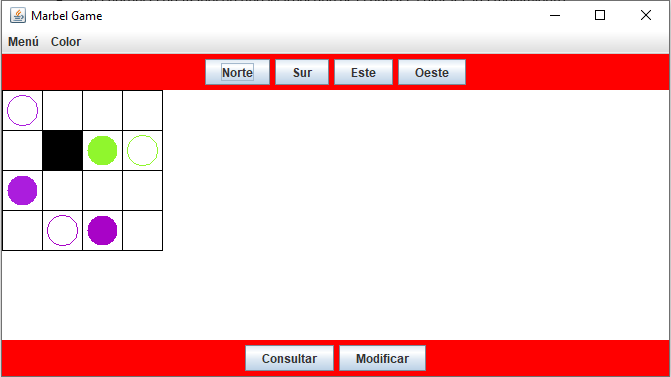
Para aplicar este caso de uso se necesitan los siguientes elementos:

* Un objeto de tipo MarbelGame (Aplicación) que es el que vamos a pintar y sobre el que se van a realizar los movimientos del juego.
* Los listeners para cada botón (Norte, sur, este, oeste) para que hagan el movimiento que debe hacer cada uno.
* Una herencia de JPanel de tipo Marbel que es la que va a pintar cada movimiento.

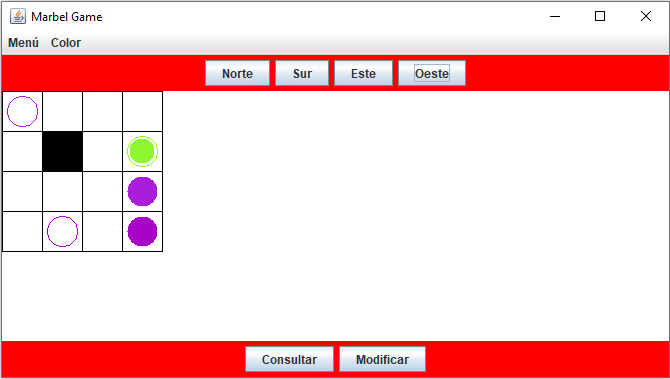
4. Ver código.

5.

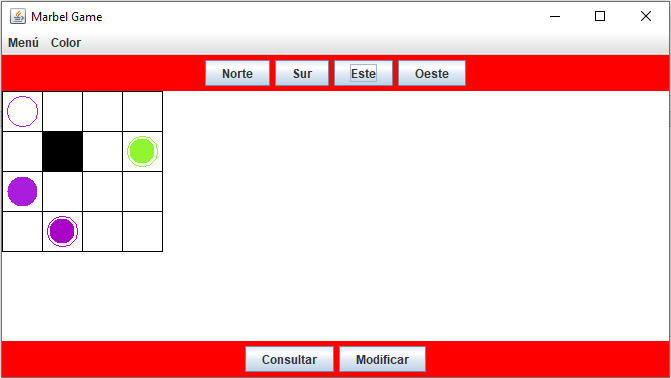
Tablero inicial:



Oprimimos el botón oeste:



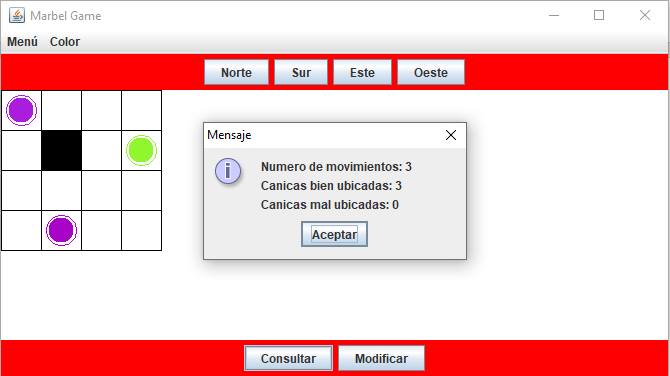
Oprimimos el botón este:



Oprimimos el botón sur:



Oprimimos el botón consultar:



Ciclo 8

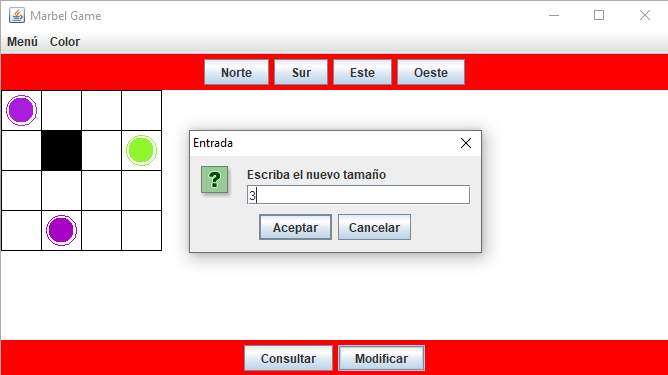
1.

* Un JButton que al oprimirlo se ejecute la acción modificar.
* Tres JOptionPane.showInputDialog, en cada uno se va a preguntar el nuevo tamaño, el numero de canicas y el número de barreras.
* La función modificar en la clase de la capa de aplicación.

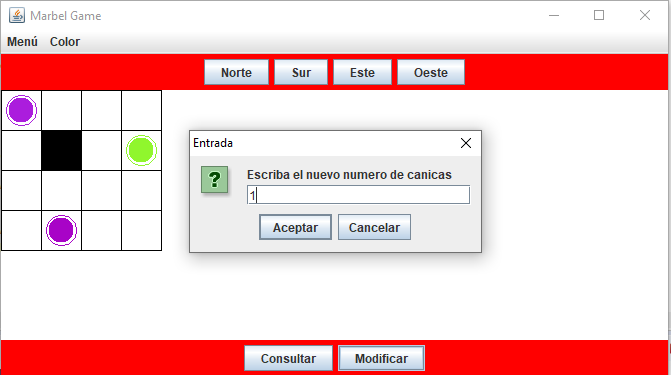
2. Ver código

3.

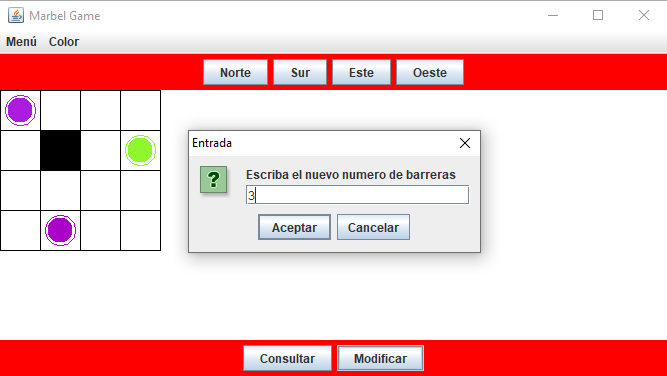
Al oprimir el botón modificar se abre el primer cuadro de dialogo para que se ingrese el tamaño nuevo:



Al dar aceptar se abrirá el siguiente dialogo para escribir el número de canicas:



Luego se abrirá el dialogo para escribir el número de barreras:



Después de esto tendremos un nuevo tablero con las características que especificamos:



Retrospectiva:

1. ¿Cuál fue el tiempo total invertido en el laboratorio por cada uno de ustedes? (Horas/Hombre)

20 horas

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

Casi completo

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

Porque al tener en cuenta las pruebas de aceptación, marcamos un lineamiento para el código, es decir, se sabe para donde ir.

1. ¿Cuál consideran fue su mayor logro? ¿Por qué? ¿Cuál consideran que fue su mayor problema? ¿Qué hicieron para resolverlo?

El mayor logro fue conseguir que el tablero sea representado gráficamente, ya que tuve muchos problemas para hacerlo.

1. ¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los resultados?

Me comprometo a seguir trabajando para mejorar la calidad de las actividades realizadas.