

Taller 5 Patrones
Julian Mondragon
202221122
Universidad de los Andes

Game of Life

Este repositorio se puede encontrar en el siguiente link:

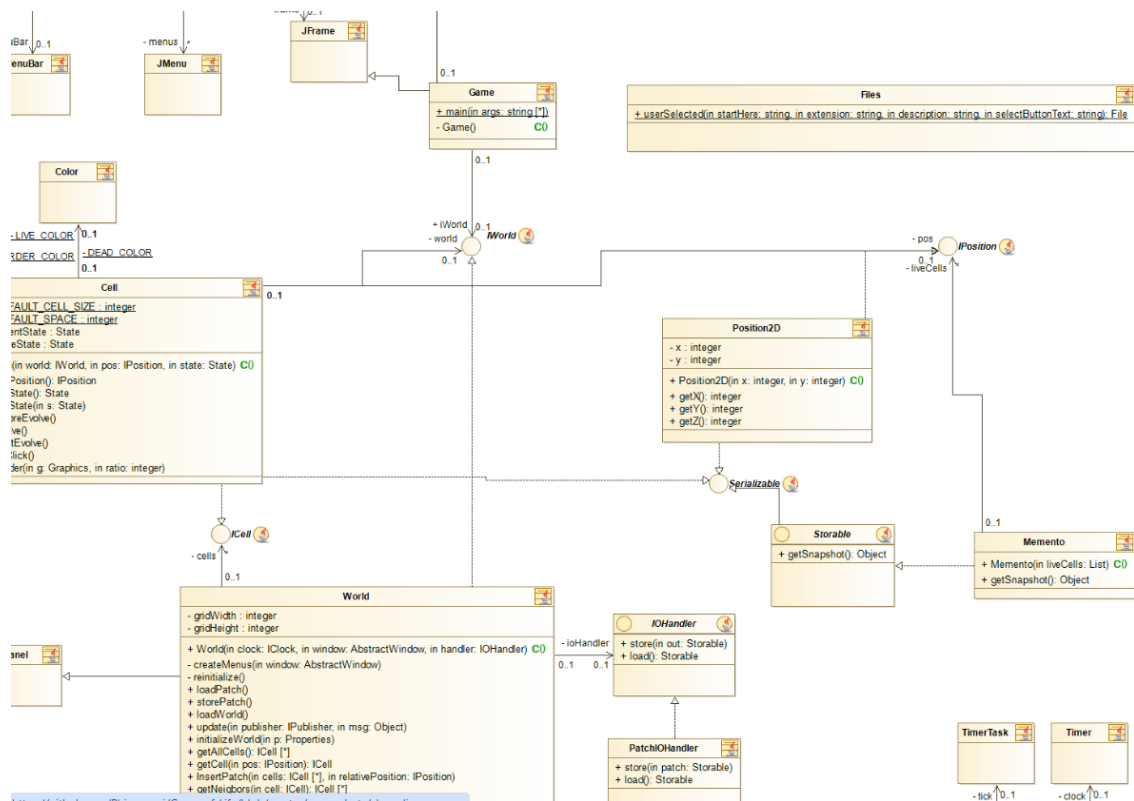
<https://github.com/Phiponacci/Game-of-Life>

Este repositorio esta 100% hecho en Java y trata sobre el Juego de la vida. Este es un juego creado por el matematico Conway para simular como unas pocas interacciones ciertas celulas pueden crear grandes cambios en el entorno. El juego consiste en celdas enendidas o apagadas que son capaces de cambiar de estado dependiendo que las celdas que tengan a su al rededor. Las reglas son sencillas, todo se centra en que si una celda viva tiene 2 o mas celdas vivas a su al rededor, esta se mantiene viva porque de lo contrario muere. Mientras que una celda muerta, necesita exactamente 3 celdas vivas a su al rededor para que pueda revivir.

Lo dificil de implementar este juego es el momento de guardar las configuraciones en las que se encuentren cada celula ya que estas configuraciones no solo se tienen que guardar, sino que todo el resto de celulas deben tener acceso a estas para poder verificar si cambiar de estado o no.

El patron que se va a analizar en este caso, es el patron de Memento, el cual es un patron de diseño cuyo objetivo principal es proporcionar la capacidad de capturar el estado interno de un objeto de manera que el objeto pueda ser restaurado a este estado más tarde, sin revelar la estructura interna del objeto. Este patron suele ser utilizado comúnmente cuando se quiere implementar la funcionalidad de deshacer (undo) en una aplicación. Permite que un objeto capture su estado interno en un objeto llamado "Memento", y luego este Memento se puede almacenar externamente. Posteriormente, el objeto original puede ser restaurado a ese estado anterior utilizando el Memento almacenado.

Además de la funcionalidad del undo, el patrón Memento también se utiliza en situaciones donde se necesita tomar instantáneas (snapshots) de un objeto en diferentes momentos para implementar la recuperación de estado o la persistencia.



Como se puede ver en el UML hecho por los autores del código, existe una clase llamada Memento que hereda los métodos de la interfaz Storable que se encarga de guardar snapshots o en otras palabras estados de cada objeto célula. El patrón memento se ve utilizado cuando estas snapshots son usadas para definir que células siguen con vida. Esto con el objetivo de que se pueda acceder a dicha información para que el resto de las células puedan cambiar de estado dependiendo de estas células vivas que se almacenan en memento.

```
1    package com.leo.dev.gol.tools.io;
2
3    import edu.emp.gl.gol.model.IPosition;
4
5    import java.util.List;
6
7    /**
8     * @author Leo
9     */
10   ✓ public class Memento implements Storable {
11       private final List<IPosition> liveCells;
12
13       public Memento(List<IPosition> liveCells) {
14           this.liveCells = liveCells;
15       }
16
17       @Override
18       public Object getSnapshot() {
19           return liveCells;
20       }
21   }
```

Una gran ventaja de implementar este patron es que al guardar estos estados de las celulas vivas, se esta haciendo mucho mas facil la impliementacion del codigo para hacer que celulas muertas revivan. Esto porque no se tiene que revisar todo el tablero casilla por casilla y que asi las comparaciones o restricciones para que alguna celula arbitraria reviva, termine siendo mas ineficiente.

Sin embargo, no siempre un snapshot es una buena opcion ya que memento podria tener como desventaja que se aumenta la complejidad de implementacion del codigo a la hora de guardar estos estados. Esto significa que el hecho de agregar la nueva clase memento, puede hacer mas dificil alguna modificacion futura que se pueda hacer al codigo en un futuro.

Con esto en mente, una posible solucion al problema de almacenar los estados de la celula, puede ser guardar alguna clase de booleano por cada casilla que representa si esta viva o muerta y que esta lista se almacene la clase game. Esto evitara la implementacion del

patron de memento ya que con solo una estructura de datos sencilla se esta manejando el guardado de la informacion.