**QUESTION-**

Complete the program to design a game engine that implements two different games based on users choice:-

1) Game of Life

2) Power Game

Following are the rules of the games-

1. Game of Life

The universe of the Game of Life is an infinite two-dimensional grid of square *cells*, each of which is in one of two possible states, *alive* or *dead*, or "populated" or "unpopulated". Every cell interacts with its eight neighbors, which are the cells that are horizontally, vertically, or diagonally adjacent. At each step in time, the following transitions occur:

1. Any live cell with fewer than two live neighbours dies, as if caused by underpopulation.
2. Any live cell with two or three live neighbours lives on to the next generation.
3. Any live cell with more than three live neighbours dies, as if by overpopulation.
4. Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction.

The rules continue to be applied repeatedly to create further generations.

2)Power Game

Consider the above problem but different rules:

1. The score of an individual is its own score plus half of the the sum of the score of neighbours.
2. If the score is greater than 1 and less than 9 then return the score else return 0.

Design a code to solve this problem efficiently.

**QUESTION-**

Refactor the code to minimize the repetition of the code. Modified code is given extract the common code and recode the project.

**Solution**

This design pattern is called **strategy design pattern**

The strategy pattern:

* Defines a family of algorithm
* Encapsulates each algorithm
* Makes the algorithm interchangeable within the family
* It unables us to select the algorithm at runtime

1. The strategy design pattern uses composition instead of inheritance.
2. In the strategy pattern, behaviors are defined as separate interfaces and specific classes that implement these interfaces.
3. The behavior can be changed without breaking the classes that use it, and the classes can switch between behaviors by changing the specific implementation used without requiring any significant code changes.

<<interface>>

Strategy

Algorithm()

Strategy 1

Game of Life

Algorithm1()

Strategy 2

Power Game

Algorithm2()

Context

Operation()

Strategy

For more info visit:

<https://en.wikipedia.org/wiki/Strategy_pattern>

<https://www.youtube.com/watch?v=nOX27nHmFKQ>