Design Pattern Project

Topic: Sudoku Game Design

Project No. - 6

Manish Soni - 01FB16ECS193

Vaibhav Pundir - 01FB16ECS431

Objective:

The aim of this project is to design/develop a game using the design patterns. The game we propose to build is a Single Player Sudoku game. The intention of this project is not on which algorithm to be used to generate the sudoku board or to solve the sudoku games and neither the game will be an AI-driven where the computer solves the sudoku game rather it is a user-level game where a user can solve a sudoku puzzle and the intention is to use the best possible Design Patterns to design the game. In addition to this, the user will be able to store the game at any point in time and can choose to continue at a later point in time. Also, the option of getting the hint will be available to the user.

About the Game:

The Sudoku game is a single-player board game with 81 square cells distributed as 9 rows and 9 columns. These 81 cells are divided into 9 boxes with each box containing 9 cells. Random numbers ranging from 1 to 9 are placed on the board. This arrangement is what is known as the Sudoku puzzle. The aim of the game is to fill the board with numbers from 1 to 9, in such a way that, there exists only one instance of the numbers 1 to 9, in every column, row and box on the board. It is interesting to note that every Sudoku puzzle has one and only one unique solution [4]. The player is given an unsolved puzzle and is required to solve it by filling the empty cells with relevant numbers. There are various techniques to solve the

Sudoku puzzle but in this project deals with the design aspect of games rather than these techniques.

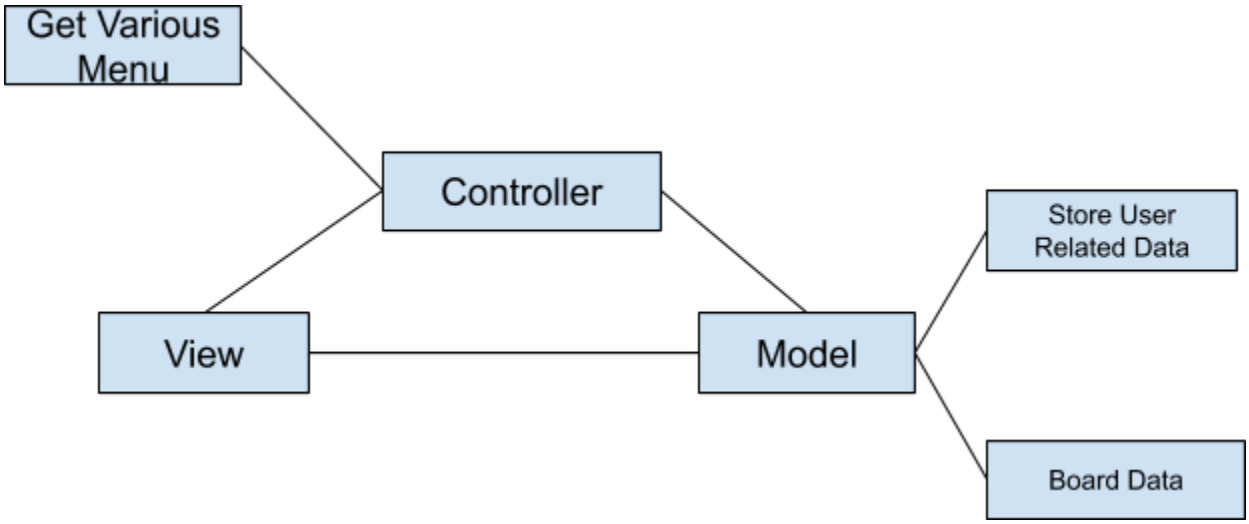
Game Design:

The main components of the Sudoku game are the model, view and controller components, which together make up the Model-View-Controller (MVC) pattern.
These are the main components of the games. The MVC architecture makes it possible to loosely couple the graphical representation from the core logic of the game. There could possibly be more than one view in an application.

Controller: Receives all the user events and delegates the same to the view and model depending on the requirements. Thus the controller layer seats in between the view and the model.

View: The view which the user is able to see(The UI part)

Model: The model stores the user-related data, game-related data etc.



Design Pattern Intended to Use:

1. **Singleton:** Allowing only one game to be active at a time, allowing only one player to play a game on one device.
2. **Decorator:** Adding of run time behaviour to the game, like if a user wants to change some settings for the game like colour etc.
3. **Factory Pattern:** For creating various displays for the user, Ex: Creating a new game, Get the list of all the games played by the user.
4. **Memento Pattern:** For restoring the current stage of the game to the previous (Undo action)
5. **State Pattern:** For changing the states of the individual cell.
6. **Iterator:** For Iterating over the board