# Final Project Report

**Team Members:**

Karthik Narayan - 413031679

Ketki Mohit Naik - 953706796

Varshitha Naladala - 704002032

**Short Description**:

We are implementing the sudoku game. Sudoku is basically a number placement puzzle consisting of a grid of nine 3-by-3 squares, in which the numbers 1 to 9 must be placed so that each row, column and square only contains one instance of each number. The user is asked to fill the numbers in each of the boxes provided which are not prefilled from the input selector. The application has the ability to resume the game where the user previously left or start a new game.

**Project Layout:**

Sudoku puzzle

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | 5 |  | 1 |
|  |  |  | 1 | 9 |  |  |  | 2 |
| 5 | 1 | 7 |  | 2 |  | 8 |  |  |
| 6 |  | 1 |  |  |  | 2 | 4 |  |
|  |  | 5 |  |  |  | 7 |  |  |
|  | 4 | 2 |  |  |  | 9 |  | 5 |
|  |  | 9 |  | 5 |  | 4 | 8 | 3 |
| 3 |  |  |  | 4 | 2 |  |  |  |
| 4 |  | 8 |  |  |  |  |  |  |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | |

**Difference between project proposal and final project**

1. Selector not implemented in the final project – Input accepting mechanism was changed from mouse based to keyboard input. User is now required to enter the input in the format Row Number <Space> Col Number <space> Number which he/she wants to enter. Due to this there was no need for selector.
2. Not implemented ChangeBoxColor() – User inputs that are not valid will never be accepted and inserted in the sudoku grid. Therefore there was no need for this method whose objective was to change the fill color of the rectangle shape.
3. DecideUserOptions() not implemented – Game will be automatically resumed if it was previously saved by the user therefore there was no need to implement this method.
4. List of Filled Shapes – Based on the recommendation, we have removed list of filled shapes and maintained a single filled shapes object that is used to draw the UI.

**Description of classes:**

1. Main function:

The game is launched in this method. This class contains the reference to the BaseGame class.

It contains the following methods:

->determineGameType() method-

This method examines the file where game data is stored and decides whether to launch a new game or resume a previously saved game.

1. BaseGame class:

The BaseGame class is used to draw the sudoku. It continuously monitors the user inputs and fills the numbers in the box selected by the user. It validates the numbers entered by the user and checks for the repetition of the number in the same row or column or the block. It also checks whether the user has completed the puzzle or not.

It consists of fillshapes object to draw the boxes in the puzzle and inserts the values in each of the boxes.

The following are the methods used by the BaseGame class:

->A default constructor used to initialize

->displayUI() method:

It is used to draw the sudoku using fill shapes.

->insertNumberIntoRectShapes() method:

It is used to fill the number given by the user in a particular box.

->ValidateUserInput() method:

It is used to check if the number entered by the user is already present within the same row or column or block. It throws an exception if a junk data is entered user.

->IsGameComplete() method:

It is used to check whether the user has filled all the boxes or not.

->getUserInput() method:

Gets the input entered by the user in the console

-> errorAction() method:

Contains the code that will handle invalid data entered by the user

-> getCellNumber() method:

Returns appropriate cell number based on the row and col values

->WriteGameDataIntoFile() method:

It is used to write the numbers given by the user into a file inorder to resume the game.

->Overloading += operator to add the numbers selected by the user from the selector to the puzzle.

3)NewGame class:

This class is the child class of BaseGame class which launch a new puzzle.

This class has launchGame() method which calls the launchGame method in the BaseGame class.

->generateNewGame():

Displays a random game from one of the four games stored by the private member of this class called randomGame.

->parameterized constructor:

Accepts a file location and opens a file pointer to write data into that file.

5)ResumeGame class:

This class is the child class of BaseGame class which continues the game from where the user previously left.

It stores the part of the puzzle solved by the user before closing the game in a file.

This class has the following methods:

->Parameterized constructor:

It is used to accept the file location and open a file pointer to the game data file.

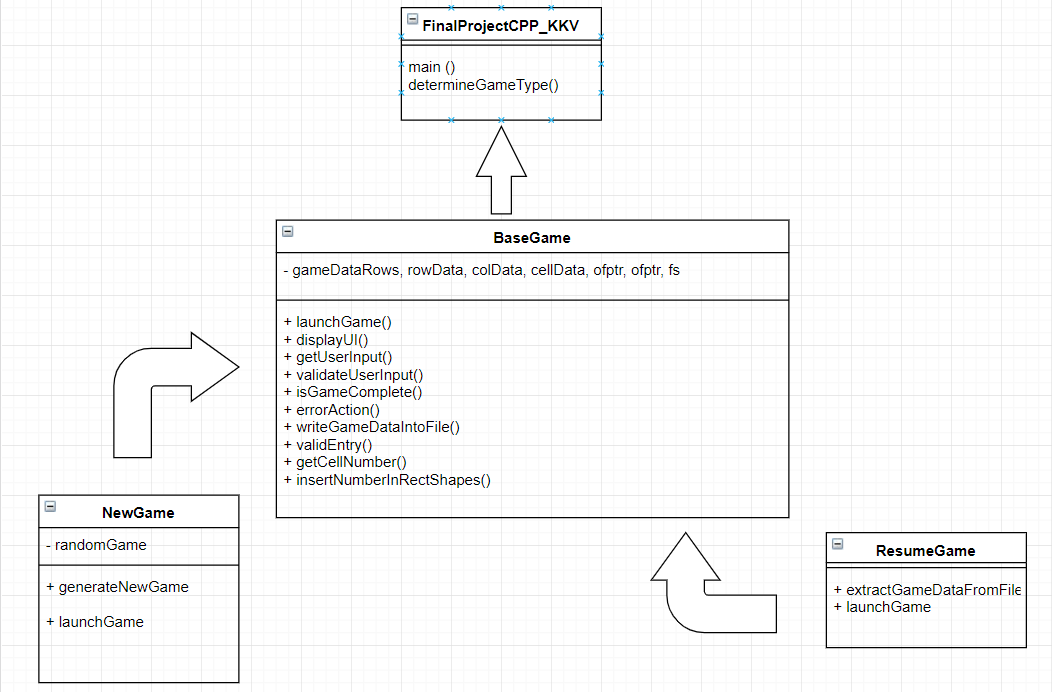
->LaunchGame method():

This method launches the launchGame method in the BaseGame class

-> extractGameDataFromFile():

Reads the contents from the game file and then fills the variable for storing sudoku grid.

**UML Diagram:**



**Description of Requirements fulfillment-**

1)Multiple classes -

Usage of 5 different classes for different implementations. They are as follows:

* BaseGame
* NewGame
* ResumeGame

2)Inheritance:

Inheritance is being implemented between BaseGame class and NewGame and ResumeGame classes. The Parent class is the BaseGame and the NewGame and ResumeGame are the children classes.

3)Composition:

Base Game class is composed of filled shapes

4)Polymorphism:

determineGameType function determines the type of game needs to be played and generates either a new game object or resume game object. This is then returned as base game pointer and then launchgame method on the returned pointer variable is invoked in the main function by downcasting the returned pointer variable.

5)Operator Overloading:

Operator Overloading is used in BaseGame class to add the inputs of the user to the puzzle.

6)Exception Handling:

Exceptions thrown when opening a file or writing or reading contents to a file is handled. Additionally any bad user data in the file or entered by the user in the console is handled in the main function.