10th October

Time spent: 2 hours in the lab discussing the topic for our project and finalized the topic of Sudoku game. I mentioned the method of generating a Sudoku by using the brute force method and backtracking.

Time spent: 1 hour thinking and designing of basic functions that I would require irrespective of the logic to generate a Sudoku. I found I required the function for checking if a number had any conflict with the rules of Sudoku at any particular row and column.

11th October

Received the function equal\_array from akanksha to be used later in the program.

13TH October

Time spent: 3 hours

I finalized the logic for generating the Sudoku and began designing functions I decided it would be done with three functions SolveSudoku, NoConflict and FindUnassignedPos. The function NoConflict was the easiest so I began from there I wrote a simple code to achieve this. Aditya Sing and Akeek sent me their versions of the NoConflict function and the one written by akeek was the best so I decided to use it.

Then I started designing the FindUnassignedPos. This was tricky and finally the function was returning the first position without a number in a 9x9 array.

Finally I decided to put all the functions the 9x9 array into a class called Sudoku.

16th October

Time spent: 3 hours

I Designed and wrote the code for the SolveSudoku function.

Time spent: 2 hour

I tested and debugged the program.

17th October

Time spent: 2 hours in the Cs lab. Showed the team the generation and discussed the further functionalities. We wanted to put a stopwatch but the idea was dropped. Also we decided to add a difficulty level.

23rd October

Time spent: 3 hours

Now I wrote the function that would be called by the user to generate the Sudoku and the functions for removing numbers from the generated Sudoku. I wrote the function to check uniqueness and the function that removed numbers from random positions.

26th October

Time spent: 1 hour

I now retyped the entire program with all unnecessary lines removed and made it into a header file.

27th October

Time spent: 2 hours

I tested and debugged the entire code. Removed a few errors and tried to reduce generation time.

31st October

Time spent: 2 hour

In Cs lab. Aditya Singh showed us the program for the graphics part he had written. We discussed the graphics and decided to go with what Aditya Singh had written. His program contained bitmaps instead of multiple windows.

1st November

Time spent: 30min

After removing printing the generation time was considerably reduced also made it so that you only had to call the generate function and it would set the parameters to the required Sudoku.

5th November

Time spent: 1hour 30 min

Met with Aditya Singh at 10pm and changed the input from cin as it does not work in ezwindows. We made buttons with numbers on them and you click on the number and wherever you click on the Sudoku grid the cell will be set to that number.

6th November

Time spent: 2hours met with Akanksha, Afsar and Aditya.

Aditya Afsar and I typed the entire program for the graphics while akanksha created the Bitmaps.

Time spent: 4hours

We then went to the Cs lab where we had booked an extra lab slot. We finalized the program and integrated the parts of the graphics written by us. Then we ran into a problem that the bitmaps were resolution dependant and the centimeters changed for different resolutions on the bitmaps. I thought we should rewrite the entire graphics with render lines and rectangles but the team thought it would take long. Then we decided on a fixed resolution and started changing and scaling the range for the click of mouse and the window size.

7th November

Time spent: 2hours met in Cs lab

Completed the scaling. Instead of converting the integer to string during rendertext of the inputs we converted it to a character array and used another version of the renderText function. This solved the problem that we could not properly input.

11th November

Time Spent: 30 min

Met with team-ajay kumar absent.

We did the peer review and discussed the project report.