Kathford international college of engineering and management

A project report on noughts and crosses

**Submitted to: Submitted by:**

**CSIT Department Bina Balami  
 Priyanka Gyawali**

**Reema Prajapati**

**Sandhya Gauro**

**Table of contents**

ACKNOWLEDGEMENT

ABSTRACT

1. INTRODUCTION 6
   1. Background
   2. Overview of Noughts and Crosses
      1. Rules
      2. Other variants
2. OBJECTIVES 7

2.1 Academic Objectives

2.2 Program Objectives

3. PROBLEM ANALYSIS 8

3.1 Noughts and Crosses board

3.2 User interface

4. METHODODLOGY 9

4.1 Project development

4.1.1 Selection of idea

4.1.2 Research

4.1.3 Design

4.1.4 Coding

4.1.5 Report

4.2 Programming tools used

4.2.1 C++ programming language

4.2.2 Code::Blocks IDE

5. ALGORITHMS AND FLOWCHART 11

5.1 Algorithm

5.2 Main flowchart of the program

6. SOURCE CODE 13

7. RESULT 19

8. EPILOGUE 20

8.1 Limitations and possible improvements

8.2 Conclusion

8.3 References

**ACKNOWLEDGEMENT**

This report was prepared as a partial fulfillment of the academic requirements of the Third Semester of B.Sc.CSIT. Effort has been made to ensure that the report is accurate and professional as far as possible.

We are very grateful to the Department of CSIT of IOE Kathford International College for providing us with this great opportunity to develop a project. It has greatly enhanced our knowledge of the course and has given us the valuable experience of research and teamwork, which will definitely be of great help for our career development.

We would like to express our deepest gratitude to our project coordinator, Mekh Raj Joshi, for his constant support and supervision on the project development and presentation.

Similarly, our special thanks goes to all our friends and well-wishers who provided us with their valuable comments, suggestions, corrections, critiques and their encouragement which was obviously a big boost of energy for the completion of the project.

Although there may be many who remain unacknowledged in this humble note of gratitude, there are none who remain unappreciated.

Bina Balami, 13

Priyanka Gyawali, 25

Reema Prajapati, 27

Sandhya Gauro, 37

**ABSTRACT**

This is the technological era—the age of the computers. Everyone wants their work to be done digitally from a child’s nursery rhymes to space exploration, everything is done on computers. Likewise, nowadays people are being fond of playing games in computer. This is because playing games in computer is a way more interesting.

“**Noughts and Crosses**” is a very popular two-player turn-wise board game that barely anyone in the world is unknown about. This game has different names in different places. In Nepal, it is popular as “**Aalu Cross” (आलु क्रस).**

In this project, we have created the board game “Noughts and Crosses” using the **object oriented programming language,C**++. It can be played between any two users.

We’ve developed the program for the Windows platform using C++ language. Although we’ve not used any graphics library in this project, we’ve been able to create a simple but appealing user interface.

We hope to provide its users with entertainment as well as challenge.

**1. INTRODUCTION**

**1.1 Background**

One of the most interesting and enthusiastic topic in Computer Programming is the “Game Programming”. No programmer can be called a good programmer if he/she hasn’t written a game. Games are not just for fun. It increases creativity, thinking capacity, body-mind synchronization, and decision and reasoning ability.

As teenagers, we are no doubt interested in making games than any other applications. Since we didn’t have much programming knowledge while proposing the project, it seemed like the popular board game “Noughts and Crosses” was the suitable game to build as a project because it is not a very complex game and it can be developed with not so lengthy code.

**1.2 Overview of Noughts and Crosses**

“Noughts and Crosses” is a simple paper-and-pencil board game. It is one of the most common and popular board games in the world. There are hardly any people in the world who hasn’t at least heard of this game. It is played all over the world, may be with a different name. It is also called “Tic-tac-toe”, whereas in Nepal, this game is known as “Aalu Cross”, and is mostly played by young children.

**1.2.1 Rules**

1. The game is played between two players on a board with 3x3 gird of squares.

2. The two players are represented by ‘O’ and ‘X’.

3. The players place (or draw) ‘X’ and ‘O’ alternatively on the board until the game ends (i.e. when one of the players win or the game is drawn)

4. If a player draws three-in-a-row (horizontally, vertically or diagonally) then that player wins.

5. If all the nine squares are filled without a winner, then the game is a tie.

**1.2.2 Other Variants**

Noughts and Crosses can also be played on a 4x4 or 5x5 grid where the players try to draw four-in- a-row and five-in-a-row respectively. It can even be played on a bigger grid such as 10x10, but for such large grids the objective usually remains to obtain five-in-a-row.

**2. OBJECTIVES**

We can broadly categorize our objectives into two main categories. They are as

follows:

**2.1 Academic Objectives**

• To fulfill the partial requirement of our course.

• To learn the object-oriented programming (OOP) paradigm using C++ programming

language.

• To learn to develop software in a managed way.

• To gain research and team work experiences.

* 1. **Program Objectives**

The main objectives of this project are:

* To develop a computer game of ‘Noughts and Crosses’.
* To use different techniques of C++ programming language to make the game interesting and interactive.
* To create a simple but appealing User Interface without using any graphics library.

**3. PROBLEM ANALYSIS**

Problem analysis is an important step is developing any project. It is generally the beginning of a project. After selecting the topic of the project, it is the step of researching about the problems that need to be solved in the project. It generally requires digging deep into the subject matter and knowing as much as you can about it so that your project covers all the necessary elements of the topic.

This step was, however, not too difficult for our ‘noughts an crosses’ project as the game is a simple one and we’re all very familiar with it. We know how it is played and knew exactly what was needed to be done. The main problems that we needed to solve to build a fully functional, intelligent and good-looking game are as follows:

**3.1 Noughts and crosses board**

The first and foremost problem for making the game of ‘Noughts and crosses’ is to represent the board in a data format. The board is actual is a 3x3 matrix, which could be represented by a two-dimensional array in programming. But for such a small matrix, using a simple one-dimensional array of 9 elements seemed to be more efficient and easy to program.

Again, there is the question of how to represent the players and blank spaces. In real, noughts and crosses, players are represented by crosses ‘X’ and circles ‘O’. We thought representing the players with the respective characters ‘X’ and ‘O’ for players and gave each cells a number for blank spaces. This way the players can simply mark the cells by entering the number of the cell.

**3.2 User Interface**

For any application of any type, user interface is one of the key components that determines how well the application will be perceived. For the same type of programs, users will opt to use the one that’s easy on the eyes. Although functionality is very important too, ease-of-use is not any less important.

Creating a good user-interface while being able to display all the necessary things is not that easy. Displaying the noughts and crosses board itself is a challenge. Since we didn’t have any knowledge of graphics programming at the time, and our project didn’t need special graphical elements expect some lines, we decided to create simple user interface using only ASCII characters. There are enough characters in the ASCII table for us to create the lines and characters that we need. So, we didn’t need to use any graphics library.

**4. METHODOLOGY**

**4.1 Project Development**

The project was developed in about a months’ time. Our methodology can be briefly summarized as follows:

**4.1.1 Selection of idea**

While researching about ideas to do a project on, we came across this concept of making a computer game of noughts and crosses. Then we researched about it if it was feasible and we decided it to be a very good and optimal idea to do a project on.

**4.1.2 Research**

After deciding the project idea, we all worked on collecting as much information about it as possible. We searched rigorously through the web for documents relating to the game although we knew the game very well. During this period, we also learned few other algorithms of this game using various concepts.

**4.1.3 Design**

Before actual coding, we all worked on having a proper design. We discussed about our interfaces and the classes required for object-oriented design. We preferred graphics library for the design but since we had no knowledge about graphics, we made a simple user friendly graphic.

**4.1.4 Coding**

After the design had been fixed, we all worked on learning the necessary algorithms for developing the game. We used the C++ programming language for the object oriented programming paradigm.

**4.1.5 Report**

After the project was developed enough to fulfill our primary objectives, we finalized it and prepared this project report.

**4.2 Programming tools used**

We have used many tools, algorithms, libraries, or concepts to build this project.

Some of the significant ones are as follows:

**4.2.1 C++ programming language**

C++ is a general-purpose multi-paradigm programming language created by Bjarne Stroustrup. It is an improvement of the C programming language. It has *imperative*, *Object-oriented* and *generic* programming features, while also providing the facilities for low-level memory manipulation. It can be used to build almost any type of software, whether it be system software, application software, or even embedded software.

We used C++ as our programming language because of course requirements and also because we learned it most recently and it’s our favorite programming language right now.

**4.1.2 Code::Blocks IDE**

Code::Blocks is a full-featured IDE (Integrated Development Environment) aiming to make the individual developer (and the development team) work in a nice programming environment offering everything he/they ever need from a program of that kind.

For this project, we used the Code::Blocks version 13.12. And really it provided almost everything thing we needed to code the program. It made the task of coding easier.

**5. ALGLORITHMS AND FLOWCHARTS**

**5.1 Algorithm**

1. Start

2. Initialize the variables

3. Draw the board

4. Enter the cell to mark

5. Mark the cell

6. Check the win case

If win then display the winner and go to step 9

Else go to step 8

7. Check the draw case

8. Check the draw

If draw then display draw and go to step 9

Else go to step 3

9. End

**5.2 Main Flowchart of the program**

Board.plot();

Initialize variables

Is it a draw?

YES

NO

NO

Display draw

Display winner

Is it a win?

Input for selecting cell

Whose.turn();

Drawboard();

Checkwin();

YES

**6. SOURCE CODE**

#include <iostream>

#include<windows.h>

class Board

{

char cell[9];

bool turn;

int moves;

public:

Board(bool t = true):

cell{'0','1','2','3','4','5','6','7','8'},

turn(t),

moves(0)

{}

int plot(unsigned int x)

{

if ( cell[x] != 'O' && cell[x] != 'X')

{

if(turn)

{

cell[x] = 'O';

}

else

{

cell[x] = 'X';

}

int ret = checkWin();

if(ret > 0)

{

return int(turn)+1;

}

else if(ret < 0)

{

return -2;

}

turn = !turn;

return 0;

}

else

{

return -1;

}

}

int checkWin()

{

moves++;

//win requires more than 4 moves

if(moves > 4)

{

//top row

if (cell[0] == cell[1] && cell[1] == cell[2])

return 1;

//middle row

else if (cell[3] == cell[4] && cell[4] == cell[5])

return 1;

// bottom row

else if (cell[6] == cell[7] && cell[7] == cell[8])

return 1;

//left column

else if (cell[0] == cell[3] && cell[3] == cell[6])

return 1;

//middle column

else if (cell[1] == cell[4] && cell[4] == cell[7])

return 1;

//right column

else if (cell[2] == cell[5] && cell[5] == cell[8])

return 1;

//backward diagonal

else if (cell[0] == cell[4] && cell[4] == cell[8])

return 1;

//forward diagonal

else if (cell[2] == cell[4] && cell[4] == cell[6])

return 1;

//all slots full but no win = draw

else if(moves == 9)

return -1;

//no win but moves left = playing

else

return 0;

}

return 0;

}

void draw()

{

using namespace std;

cout<<"\n\n";

cout << " | | " << endl;

cout << " " << cell[0] << " | " << cell[1] << " | " << cell[2] << endl;

cout << "\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_" << endl;

cout << " | | " << endl;

cout << " " << cell[3] << " | " << cell[4] << " | " << cell[5] << endl;

cout << "\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_" << endl;

cout << " | | " << endl;

cout << " " << cell[6] << " | " << cell[7] << " | " << cell[8] << endl;

cout << " | | " << endl << endl;

}

char whoseTurn()

{

if(turn)

return 'O';

else

return 'X';

}

};

int main()

{

SetConsoleTitle("NOUGHTS AND CROSSES");

std::cout<<"\n\nNOUGHTS AND CROSSES"<<std::endl;

std::cout<<"-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-="<<std::endl;

Board board;

while(1)

{

board.draw();

std::cout << "TURN:: " << board.whoseTurn() << std::endl;

std::cout << "\nENTER THE CELL YOU WANT TO MARK:: ";

int in;

std::cin >> in;

if(in>8 || in<0)

{

std::cout<<"\nInvalid Move!!! Enter from (0-8) only! "<<std::endl;

continue;

}

int win = board.plot(in);

if( win > 0)

{

std::cout<<"--------------------------------------";

std::cout<< "\n\n CONGRATS!!!!! WINNER IS = " << board.whoseTurn()<<" !!!!!!!!!!!!!"<<std::endl;

board.draw();

break;

}

else if(win == -1)

{

std::cout<< "\nInvalid move. Move again." << std::endl;

}

else if(win == -2)

{

std::cout<<"--------------------------------------";

std::cout<< std::endl << "\n DRAW!" << std::endl;

board.draw();

return 0;

}

}

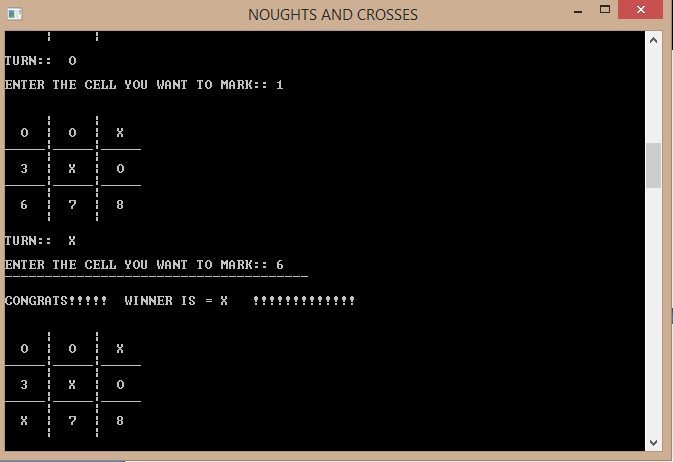
return 0;

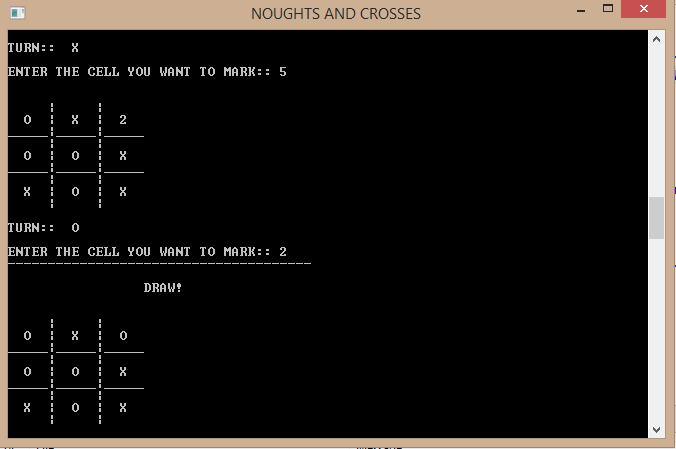
}

**7. RESULT**

Hence by using object oriented programming paradigm and C++ programming language we created a computer game of noughts and crosses. Any two user can play the game and entertain themselves. The output screenshots are as follows.

****

****

****

**8. Epilogue**

**8.1 Limitations and Possible Improvements**

As we all know, nothing in this world is 100% perfect. Everything has flaws and limitations. These are reasons behind all the advancements in the history of mankind. Similarly, developing a project like this for the first time was a really challenging as well as interesting task. The project may not be able to fulfill the entire requirement. Here is list of the limitations and possible ways to eliminate them.

* Our project implements the simplest form of Noughts and Crosses. There are different variants of noughts and crosses that are even more challenging and fun to play. Due to time and knowledge limitation, we couldn’t quite implement them in our project. But it can be done in near future.
* The project doesn’t implement audio and visual effects which would have made the project even more fascinating. We can use a graphics library such as <graphics.h>, SDL, etc. to greatly enhance the GUI and make the game even more fun to play.
* This game can only be played between any two users. Hence the human player cannot play with the computer which would be more interesting. Also, it doesn’t include any networking features. Hence, it can only be played in a single computer in a single window. We can add the network multiplayer game mode through which two users in different computers connected to a common network can play against each other.

**8.2 Conclusion**

This project entitled “Noughts and crosses” is thus a console game implementation of the actual Noughts and crosses board game. Hence, this project is successfully completed and the game was developed with a simple user-interface that can be played between any two users.

We learned a lot of new things and got a lot of valuable new experiences by doing this project. It has planted the all necessary ‘Programmer Mind’ in our brain that will help us succeed in the programming field later in our career.

**8.3 References**

The following websites, blogs and books helped us a lot in solving the problems and answering our queries during the development of this project.

1. http://www.flipkarma.com

2. http://en.wikipedia.org

3. http://boardgames.about.com