TABLE OF CONTENTS 1

I.Acknowledgement………..………………………………………………………………2

II.Abstract………………………………….……………………………………………….3

III.Chapters

1.Introduction………………….………………………………………………………………4

2.Literature Survey…………………………………………………………………………….5

3.Correlating theoretical concept to practical Implementation……………………………….6

4.Flow diagram/design of the system and Implementation…………………………………...

IV. Results/Snapshots

V. conclusions

VI. Bibliography

ACKNOWLEDGEMENTS

We four students of Computer science Engineering of 3rd semester are presenting our “CONNECT 4” project report as part of our curriculum.

I would like to express my sincere gratitude to our object oriented programming

(oop) mentor Dr. Vijaya Shetty for giving us the project that helped us learn many things.

KUMUDA K

RITIKA SUBUDHI

SHEETAL S HARSHINI

SHREYA A HEGDE

ABSTRACT

The mini project is a simulation of Funskool's connect 4 game using C++ Object oriented programming.

Connect Four is a two-player strategy game played on a 7-column by 6-row board. Each player has a colour and drops successively a disc of his colour in one column, the disc falls down to the lowest empty cell of the column. The first player to make an alignment of four discs of his colour wins.

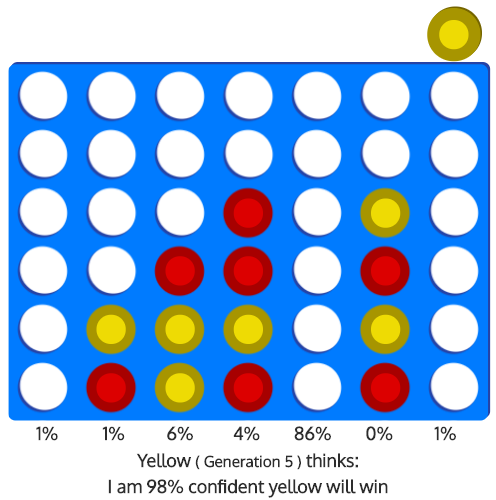
****

Fig a: connect 4 game

INTRODUCTION

Connect 4 is a fun vertical strategy two player connection game in which the players first choose a colour (or a digit) and then takes turns dropping one coloured disc (or a digit) from the top into a seven-column, six-row vertically suspended grid. The pieces fall straight down, occupying the lowest available space within the column. The player who gets four discs (or digit assigned to him/her) either in a horizontally, vertically or diagonally is declared as the winner of the game. Though the game looks simple, it is a complex game because in a classical “Connect Four” game the number of possible game board positions when all 0 to 42 pieces/discs have occupied their position is 4,531,985,219,092 combinations.

LITERATURE SURVEY

CO-RELATING THEORETICAL CONCEPT TO PRACTICAL IMPLEMENTATION

We are simulating connect 4 game in this Object Oriented Programming-C++ project.

The summary of OOP’s concepts that have been implemented in this project are as follows:

I.CLASSES :

The building block of C++ that leads to Object Oriented programming is a **Class**. It is a user defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance/object of that class.

SYNTAX:

class classname{

//data members

//data functions

};

Syntax for creating object:

classname object\_name;

Access specifiers:

 Access specifiers define how the data members and functions of a class can be accessed.

In C++, there are three access specifiers:

* public - members are accessible from outside the class.
* private - members cannot be accessed from outside the class, however the member functions can access them.
* protected - members cannot be accessed from outside the class, however the member functions can access them, however, they can be accessed in inherited classes.

In our program

II. FRIEND FUNCTION:

A normal function can made to access private members of a class by declaring it as friend.

A friend function of a class is defined outside that class scope but it has the right to access all private and protected members of the class.

A friend function is definition and call is same as normal functions/subroutines

SYNTAX:

class classname

{

public: friend return\_type function\_name(parameter list);

};

return\_type function\_name(parameter list)

{

//function body

}

III.FILE HANDLING

Files are used to store data in a storage device permanently. File handling provides a mechanism to store the output of a program in a file and to perform various operations on it.

A stream is an abstraction that represents a device on which operations of input and output are performed. A stream can be represented as a source or destination of characters of indefinite length depending on its usage.

In C++ we have a set of file handling methods. These include ifstream, ofstream, and fstream. These classes are derived from fstream base and from the corresponding iostream class. These classes, designed to manage the disk files, are declared in fstream and therefore we must include fstream and therefore we must include this file in any program that uses files.

In C++, files are mainly dealt by using three classes fstream, ifstream, ofstream.

* ofstream: This Stream class signifies the output file stream and is applied to create files for writing information to files
* ifstream: This Stream class signifies the input file stream and is applied for reading information from files.
* fstream: This Stream class can be used for both read and write from/to files.