



Project from Portugal

Erasmus/IP on Learning
Computer Programming in
Virtual Environment 2012-
2013

Project Description

The aim of this project is the development of the 4InALine game. The concept of the game is to arrange four discs in a row (in our case since we use a text-based interface it depends on the player symbol). The player discs need to be lined up. Horizontal, vertical, and diagonal lines are valid. There are two players and only one player plays at a time.

The board game is represented as:

GAME BOARD						
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	x	o	-	-	-
-	o	x	o	x	-	-
1	2	3	4	5	6	7

Development

The project is developed using C language and as IDE we recommend [Codeblocks](#).

The project implementation will be focused on the following main goals:

- define the data structures and other configurations that support the game
- UI (text-based)
- game loop with persistence (text files to store the game state)
- develop the behaviour of the player. The game must support different playing modes (e.g. Person vs Person and Person vs Computer)
- highscores for the five best players, possibly considering the number of movements
- statistics will be stored in files

First, the main goal will be the implementation of the game logic. Teams have to implement the basic playing mode, which is Player vs. Player. The game will also support other types of playing modes, Player vs. Computer, etc.

Project Source code:

The 4InALine Tournament project is available at the svn repository:

<https://code.google.com/p/erasmus-portugal-project/>

Main Data Structures:

This variables are declared in the Global.h file

```
#define HEIGHT 6    //number of lines
#define WIDTH 7     //number of columns
#define EMPTY '-'   //empty position in the board
#define PLAYERNAME 20
#define J1 'X'      //represents the players' 1 symbol to be used in the board
#define J2 'O'

typedef struct Gamer
{
    int num;
    int moves;    // We need to keep track of the number of moves made
    char name[PLAYERNAME];
} Gamer;

typedef struct
{
    int numGames; //total played games
    int numWins;  // number of games won
    int totalNumMoves; // the total number of moves made by the player
    int totalScore; // the total score achieved by this player through all
his games
    char name[PLAYERNAME] ; //the name of the player
} GamerStat;

typedef char board[WIDTH][HEIGHT];
```

Testing Phase

Project specification - small approach (only the most important methods)

Aim of the Function:	This function should validate the move of the player. Checks if it is in the limits of the board. source code in file Play.c
Function Name:	int validate(board p,int col);
Input parameters:	board and column
Output parameters:	1 if valid, 0 if not

Aim of the Function:	Tries to see if the last player move makes him win. source code in file Play.c
Function Name:	int checkWin(board p, int j, int k,int paint);
Input parameters:	j = column, k = row, p = board, paint = like a flag to see if the player's movement is painted on the board
Output parameters:	1 if player wins, 0 if not

Aim of the Function:	This function gives a score to the player based on their total moves. The higher the number of moves, the lower of score. source code in file Statistics.c
Function Name:	int calculateScore(int moves);
Input parameters:	moves = number of moves the player made in the game
Output parameters:	score based on the number of moves

Aim of the Function:	Saves the game state into a file. source code in file Play.c
Function Name:	void savegame (Gamer player1, Gamer player2,board p,int numTurns,int freeSpaces,Gamer currentgamer)

Input parameters:	player 1 and player 2 are the players; p is the board, numTurns is the number of turns; freeSpaces is the amount of free spaces left in the board currentgamer is the player who is actually playing
Output parameters:	void function, but it saves to a file the 2 players' name, the board, the turn number, the number of free spaces in the board, the number of moves of player 1 and player 2 and whose turn was it.

Aim of the Function:	This function finds the first free space (line number) in the board. source code in file Play.c
Function Name:	int findLine(board p,int col)
Input parameters:	col is the column that the player selected to put the token p is the board
Output parameters:	number of the line

How is the code organizer?

File Name	Description
Board.h and Board.c	Initialize the board and show it
Global.h	defines the global variables to be used in the project
IA.h and IA.c	Different levels of difficulty of the IA
Play.h and Play.c	Contains the gaming validations
Statistics.h and Statistics.c	Stores all the data about the game statistics
TeamPT.h and TeamPT.c	Welcome screen
main.c	main function of the game
utilities.h and utilities.c	Utility functions to improve the interface of the game
Credits.h and Credits.c	Display the name of the developer and the main goal of the game

Developer Team:

Jorge Gonçalves (PT)
Rafael Caetano (PT)
Alexandre Daussy (FR)
Giacomo Corpino (IT)
Claudio Baffi (IT)
Don Gourdet (NL)
Otteran Ludachris (NL)

Teacher:

Marisa Maximiano
Vitor Távora