Design & Concept by:

Chinmay Phutela | 13103467

Shaurya Sanger | 13103471

Shubham Kumar Jain | 13103481

Batch: B2
Group Id: G3

Jaypee Institute of Information Technology, Sector-62 Noida

Contents

Introduction

2048 is a highly addictive arcade game that functions on the simple principle of addition. The objective is to attain the number 2048 using the combination of arrow keys thus sliding the tiles on the 4X4 grid. When two tiles with same number collide they merge to give a single tile of the number twice the one that collided.

After every iteration a new tile randomly appears in an empty spot with a base value of 2 or 4. The game is won when the combination of 2048 is achieved on the grid.

2048 is an extremely challenging puzzle game which is sure to rack the brain of every player. Spastically speaking only 10% of the people who have played the game have been able to attain 2048 without any unfair means !

How to Install

Hardware Requirements :

Minimum Requirements of 64MB RAM

System Requirements :

Windows XP/Vista/7
Turbo C or C++ for Windows XP/Vista
Turbo C++ 3.0.7.8beta for Windows 7

Step 1: Install the required softwares if not present.

Step 2: Then Install the2048 setup. After installation has been completed, copy the 2048.c file onto your TC and set the location for initiating the graphics.

Step 3: Hurray! You have successfully installed your game and can play it freely now.

Problem Statement

2048 is simple yet addictive puzzle game which keeps the user on its toes and in the last few weeks has attained a cult status. When we first set out to make this game we had no idea of the challenges we'd be facing.

One of the basic problems was implementing a multi link list which would cater as the grid for our game. The next hurdle was figuring out the logic for the up,down,left and right shift which would serve as the four basic moves for the game. This initial step turned out to be more challenging than we thought. The shift function took a lot of time to make and debug. It was the pivotal part of the game and had to be functioning before we could do anything else.

To lend the game a certain authenticity it was decided to colour the tiles based on the number they were holding. Tiles of a higher sum would get a relatively lighter colour as compared to tiles of lower value. In order to make the game more straight forward level were introduced.

Searching C?

The Program enrolls through a tremendous use of Data Structures like multi-link list and file handling and graphics to make our game most effecient.

The entire gameplot has been set up on a multi link list consisting all up, down, left & right directions which lead each tile to another. Thus whenever the user gives the input in the form of an arrow key, the link list is traversed through the above links and accordingly, the function is performed.

Graphics too have been used for the loading and the main screens. Mouse Programming further helps the user easily navigate through the game.

File Handling has been used to evaluate and store the user_score and also displaying the highest score.

Understanding the Program

The structure of each tile in the program consists of 6 members :

DaTa to determine the tile number;
Location to determine the tile location and color;
up, down, left & right directional arrows to other tiles;

The Program includes in total 19 user-defined functions including main function. Here are some enlisted as follows :

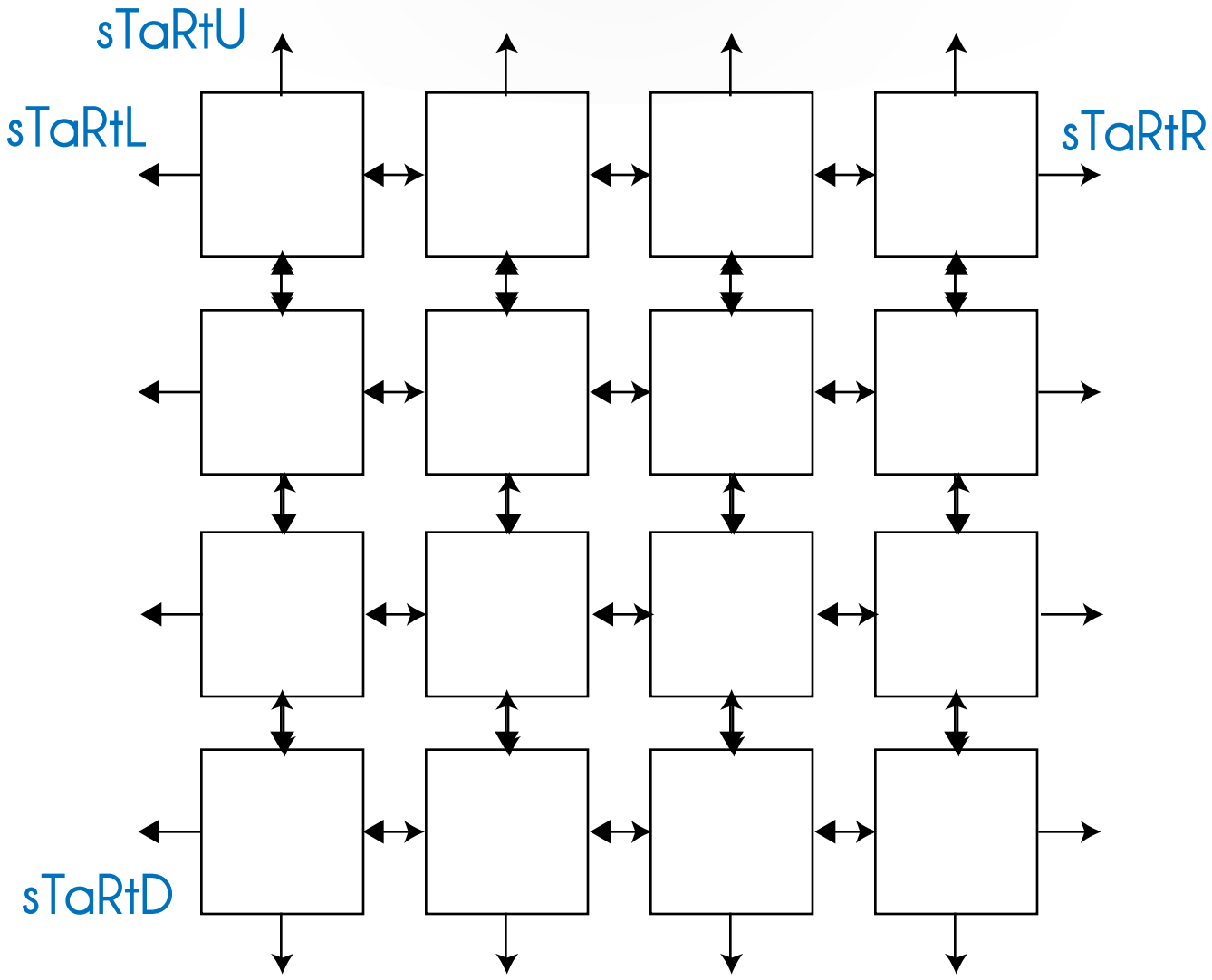
int USERSCREEN() ;
int Modes() ;
int Instructions() ;
int Statistics() ;
int Architects() ;
int Scoreboard(int score) ;
int Evaluate(int score) ;

} To handle the loading & main screen operations.

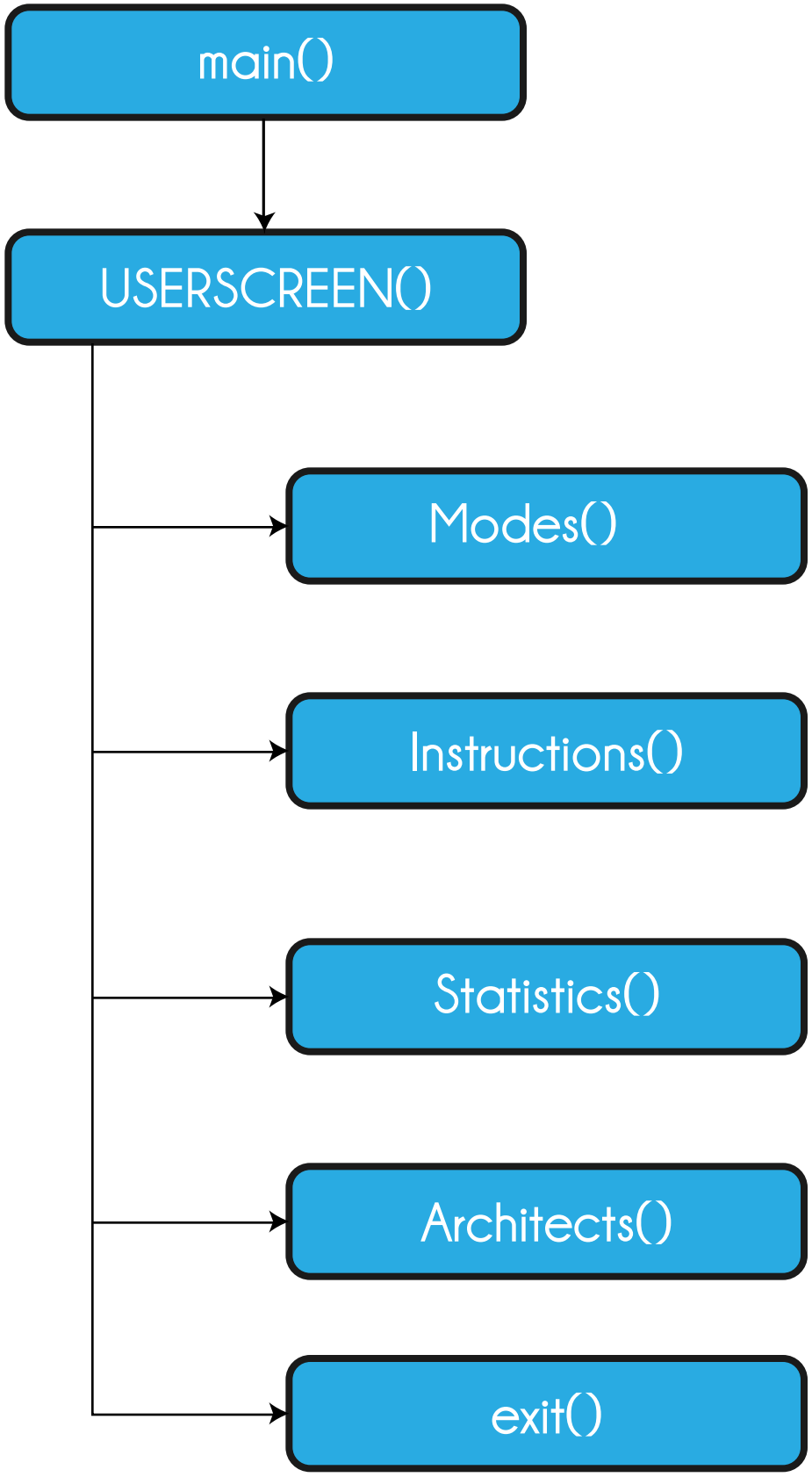
```
int main() ;

int CrEaTe_MaIn() ;                    //To create the multi link list
int SetLocation() ;                    //To give location to each tile
int DiSpLaY() ;                        //To display the link list
int CoLoR_sTyLe(int DaTa, int Location) ;

int fUnCtIoN(int mode) ;               //To play the game
GaMe_OvEr(int mode) ;                  //To check for game_status

int GeNeRaTe_RaNdOm() ;                //To generate random numbers

void sHiFt_uP(TiLe *P) ;               //Functions for shifting
void sHiFt_dOwN(TiLe *P) ;
void sHiFt_lEfT(TiLe *P) ;
void sHiFt_rIgHt(TiLe *P) ;
```

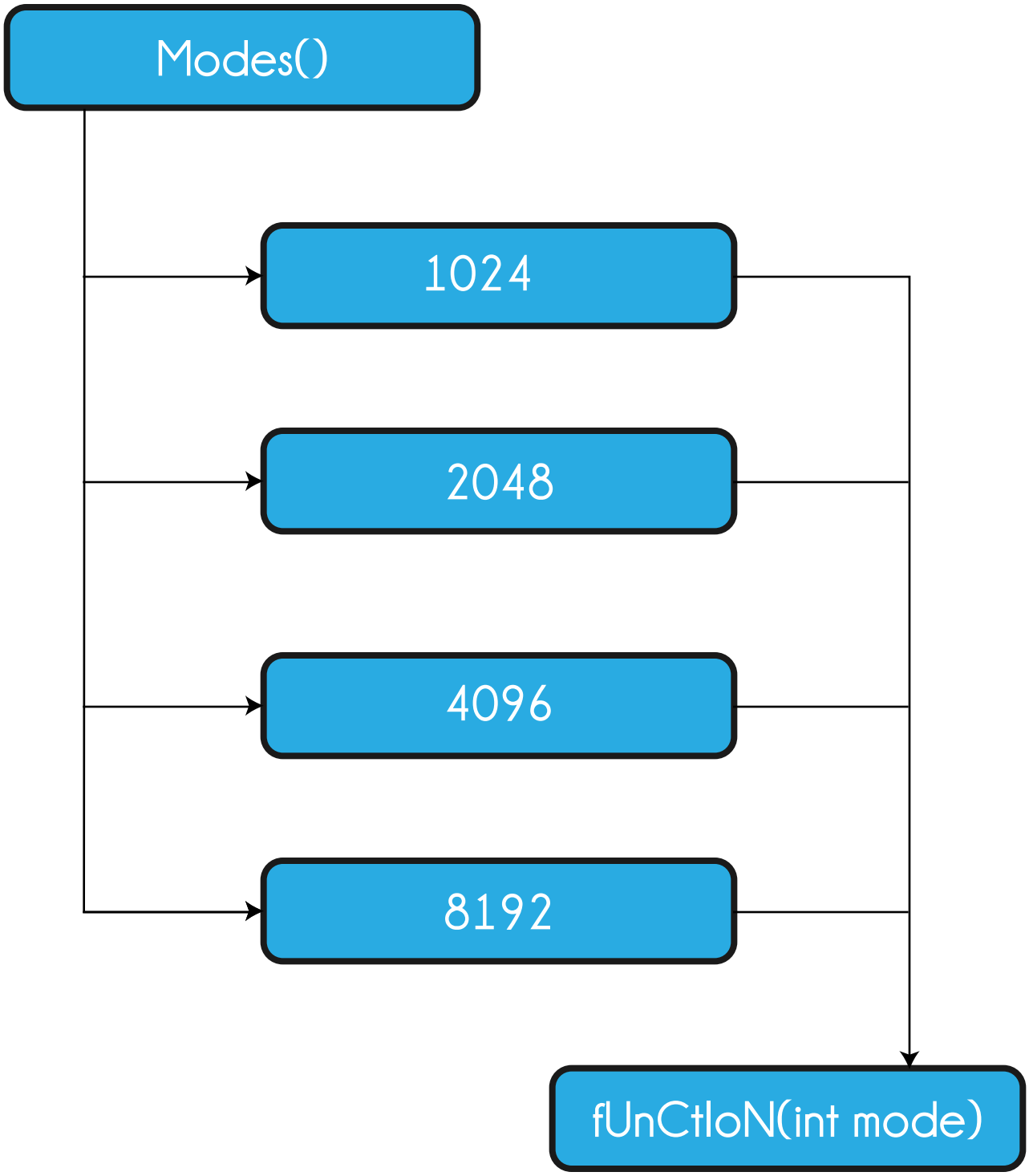
Various other functions have also been used for mouse programming which have not been enlisted here.

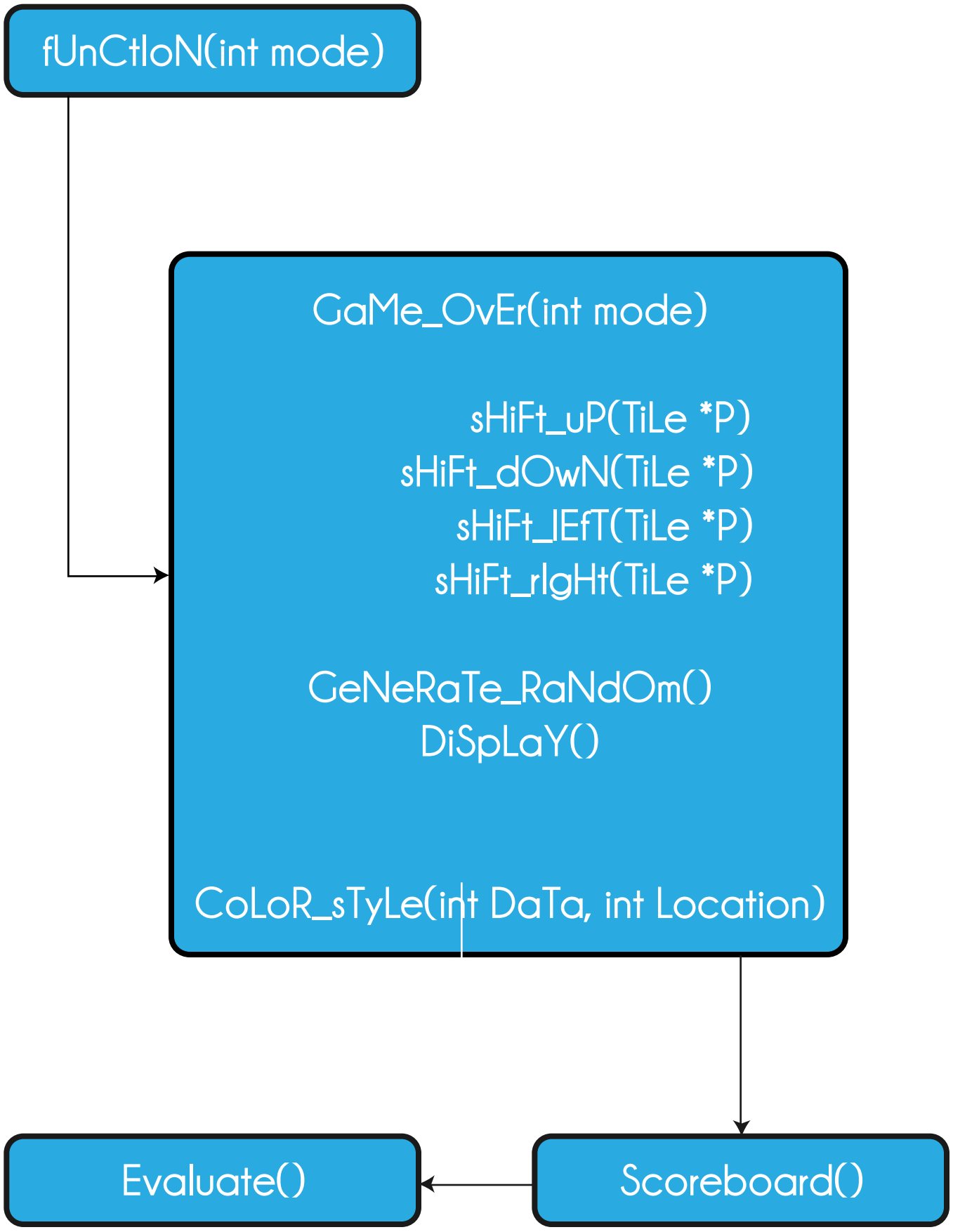
Flowcharts



(multi link list structure of the game)


main()
USERSCREEN()
Modes()
Instructions()
Statistics()
Architects()
exit()


Modes()
1024
2048
4096
8192
fUnCtIoN(int mode)


fUnCtIoN(int mode)
GaMe_OvEr(int mode)
sHiFt_uP(TiLe *P)
sHiFt_dOwN(TiLe *P)
sHiFt_lEfT(TiLe *P)
sHiFt_rIgHt(TiLe *P)
GeNeRaTe_RaNdOm()
DiSpLaY()
CoLoR_sTyLe(int DaTa, int Location)
Evaluate()
Scoreboard()

The End :)