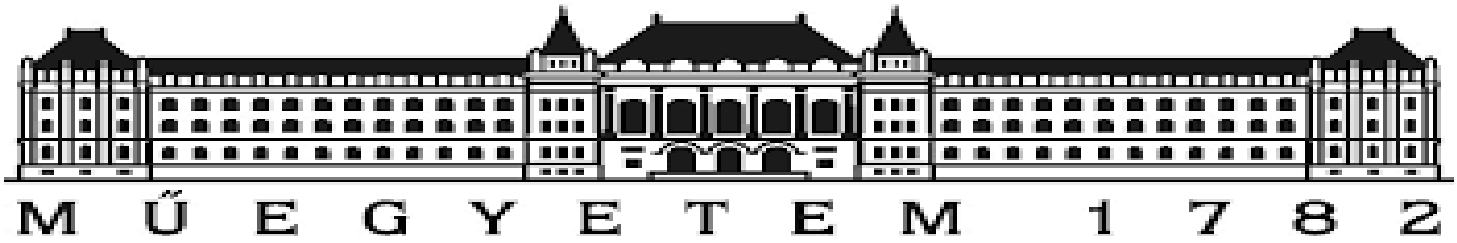


❖ MINESWEEPER DEVELOPER DOCUMENTATION

I. Overview of desired program

The player must clear a board containing hidden mines without detonating any of them, with help from clues about the number of neighbouring mines in each field.

II. Program explanation

The programme consist of one main class one subclasses and some supplementary function

class player{

diff: represent the difficulty

player():constructor of class player

int playagain():playagain offer the possibility to play another round after loss or win of previous one

void writestate(int x): depending on x writestate increment your winning or losing score

int difficulty():difficulty allow you to change difficulty setting

void states():state shows the number of wins or losses according to each difficulty

};

class map represent the game map and inherit some of the necessary function for the game from playe classr:

class Map:player {

d :represent the size of map

numberofbomb: represent the number of bomb in map

Map1:two dimension array that contain the location of bombs

Map2:two dimension array with which the player interact an try to clear

Map():constructor

~Map():destructor

Void create():initialize the maps

int introduction():open main screen allow the possibility to change configuration exit or play void draw():draw the map

int check(int x ,int y):check coordinate of case if it contains bombs if not it clears it and give number of adjacent bombs

int endgame():check if the map had been cleared

void play():utilize previous method to simulate game process

}

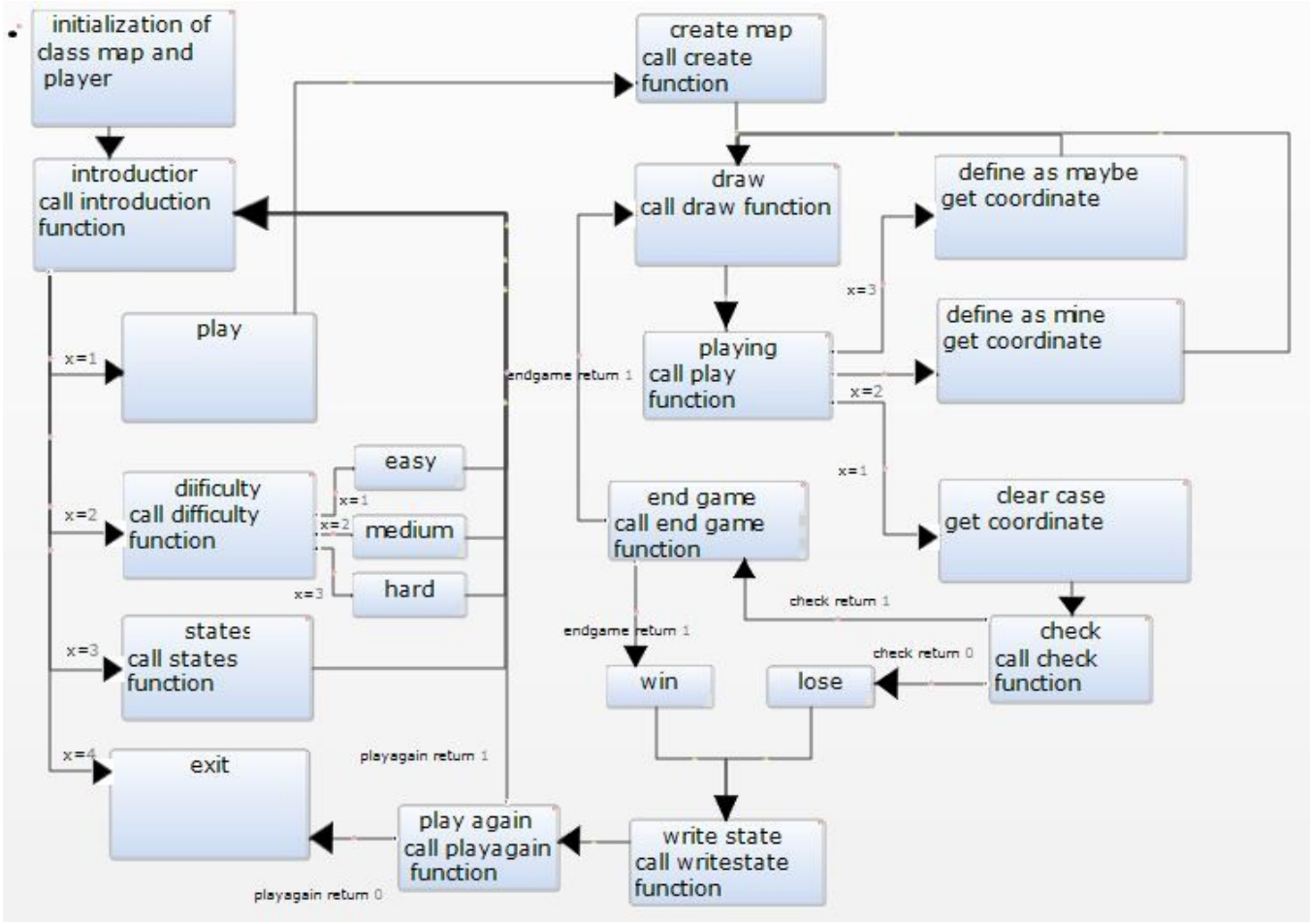
void newpage():this is an inline function that clear the screen

III. Implemented object-oriented programming technique

Inheritance (base and derived classes).

File management: saving data to file, loading data from file.

Exception management (try/catch).



IV. Testing

Testing was done manually by replaying the game and seeing if all cases are working