Bomberman Document

The game has 10 .h files, bomb.h , player.h, lifeplus.h, key.h, gift.h, game.h, bombplus.h, board.h, backend.h and GraphicBomber.h that we are going to explain briefly about each one.

Bomb.h

 private attributes : x , y , leftRange, rightRange, upRange, downRange, exTime, time (all integers) the ...Range attributes specify the span of bomb expliosion, exTime determines how many seconds the program should wait for the bomb and explodes it, time is a temporary variable. X and Y are the ones to locate the bomb whenever related key is pressed.

Public :

1. bool bombLocated (whether the bomb is located or not (it's an attribute !))

2. { void setX(int x) , int getX(), void setY(int y), int getY() : Determination of the bomb's location }

3. { void setLeftRange(int leftRange) , int getLeftRange(), void setRightRange(int rightRange), int getRightRange(), void setUpRange(int upRange) , int getUpRange(), void setDownRange(int downRange) , int getDownRange() : Determination of the bomb's span of explosion(in four directions) }

4. { void setExTime(int exTime) , int getExTime() : Determines the explosion time }

5. void bombIsLocated (Key &key ,Player &player , bool bo) : key specifies a specific key to locate the bomb , when the bomb is located, it can kill both players but this is for a single game that's why we have &player here, bool bo : naturaly when we press a key and we want it to work, we need to call the graphics, but here we wanted to prevent that and we pass a bool variable of graphics to this function.

6. void bombIsExploded () : Sees whether bomb is exploded or not.

7. void boardUpdate(Board &board,Player &p1, Player &p2) : Based on the explosion of bombs , kills the players if they were in the limited area, and destroyes the walls (the ones that can be destroyed).

Player.h

(key.h and board.h are included)
private: int x , int y , int life

public:

1. void moveControl(Key &key , Board &board) : gets the keys and determines that which arrow key is pressed and changes the player location on the board

2. { void setLife (int life) , int getLife() : Do changes to player's life attribute .}

3. { void setX(int x) , int getX(), void setY(int y), int getY() : the x and y of the player's location }

Key.h

key is a struct with four attributes (bool up, bool down, bool left, bool right), in its constructor at first we give make all of them FALSE. The whole point of key.h is to check whether arrow keys are pressed or not.

Gift.h

protected : int x , int y (this class has 2 children : LifePlus , BombPlus)

public:

1. { void setX(int x) , int getX (), void setY(int y), int getY() : Determines the location of the gift on the board }

2. virtual void rangAdder(Bomb &bomb) : This is related to BombPlus child, this function will increase the span of bomb's explosion in four direction.

3. virtual void addLife(Player &player) : This one is related to LifePlus child, this function will increase the life attribute of the Player class.

LifePlus.h

As mentioned earlier, this class is inherited from class Gift.
Public :
An empty constructor and void addLife(Player &player)

BombPlus.h

As mentioned above, this class is inherited from class Gift.
Public:
An empty constructor and void rangAdder(Bomb &bomb)

Board.h

Note : the game has a map (can have multiple ones), the map is in a file that the program reads the map from file and make it. (the exable walls(walls that can be destroyed) , the unexable walls(walls that can't be destroyed) and grass)

public :

1. char ** game_board (a multi-dimensional array that takes whatever is in the map file)

2. void setWallType(std::string filename) : The function that handles reading from file.

GraphicBomber.h

Everything related to graphics are in this class.

Private :

1. RenderWindow* window
2.RectangleShape** rectangle (each tile is a rectangle)
3.Texture brick , Texture exAbleWall , Texture noWall1, Texture noWall2

Public:

1. void LoadTexture (Texture &texture) : We got 4 same functions to load the textures.

2. void load () : Loading

3. bool rightKeyPressed () , bool upKeyPressed () , bool downKeyPressed() , bool leftKeyPressed() : All of these are for checking whether arrow keys are pressed or not

4. void closeWindow : closes the window

5. void setTextureC.... (RectangleShape** rectangle,int i , int j) : Four same functions, and they check the array and set the right texture to that part of array.

6. void setBoard(Board &board) : setting board.

7. void drawRect (RectangleShape **r,int i , int j) : draws rectangles as tiles.

8. void setPositionR(RectangleShape **r,int x,int y,int i ,int j) : setting the position of rectangles.

9. void rectSetsize(int i ,int j,int x,int y) : the size of the rectangles.

10. void show(Board &board ,Player &player ,Bomb &bomb) : Showing the player, the bomb and the whole board. This is called in an infinite while in the Backend class.
This class shows the game in every moment.

Game.h

This is the main class of game (rules and moves) , where game runs, and player.h , GraphicBomber.h , bomb.h, gift.h, bombplus.h and lifeplus.h are all included.

Public:

1. void Board_cout (Board &board) : prints the map to the terminal.

2. Key* setKeys(GraphicBomber &graphic ,Key &key ,Player &player) : setting specific keys for moves. (up , down , left and right)

3. Gift ** locateGift(Board &board) : locates the gifts places.

4. void run (GraphicBomber &graphic ,Key &key , Player &player , Board &board ,Bomb &bomb ,Gift**gift) : This function is called in an infinite while in the run function of Backend class. It's the whole game without graphic.

Backend.h

This is the class where Graphic class and Game class are connected. As the Private attributes we make objects from all of parent classes (Game game , Player player, Board board , Key key , Bomb bomb , GraphicBomber graphic , Gift** gift [there are multiple gifts]).

Public:

1. Backend () : newing Gift** .

2. void run () : There is an infinite while that graphic.show(...) and game.run(...) are called within it. And out of the while , we load the map from file , calling game.Board_Cout(board) to print the map to terminal, also locating the gift on board → gift = game.locateGift(board)

Group :
Mehraveh Ahmadi
Mahdieh Asna-ashari
Mahsa Razavi