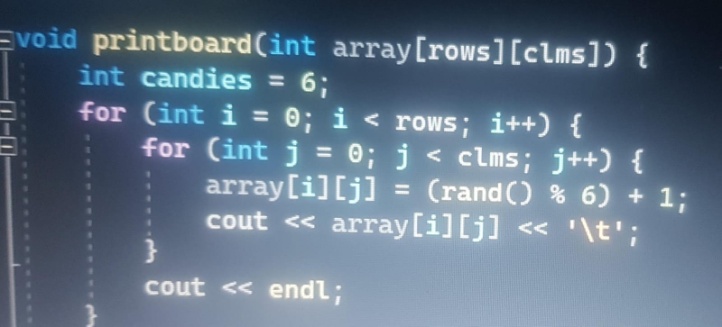
CANDY CRUSH

23L0901 - 23L0689

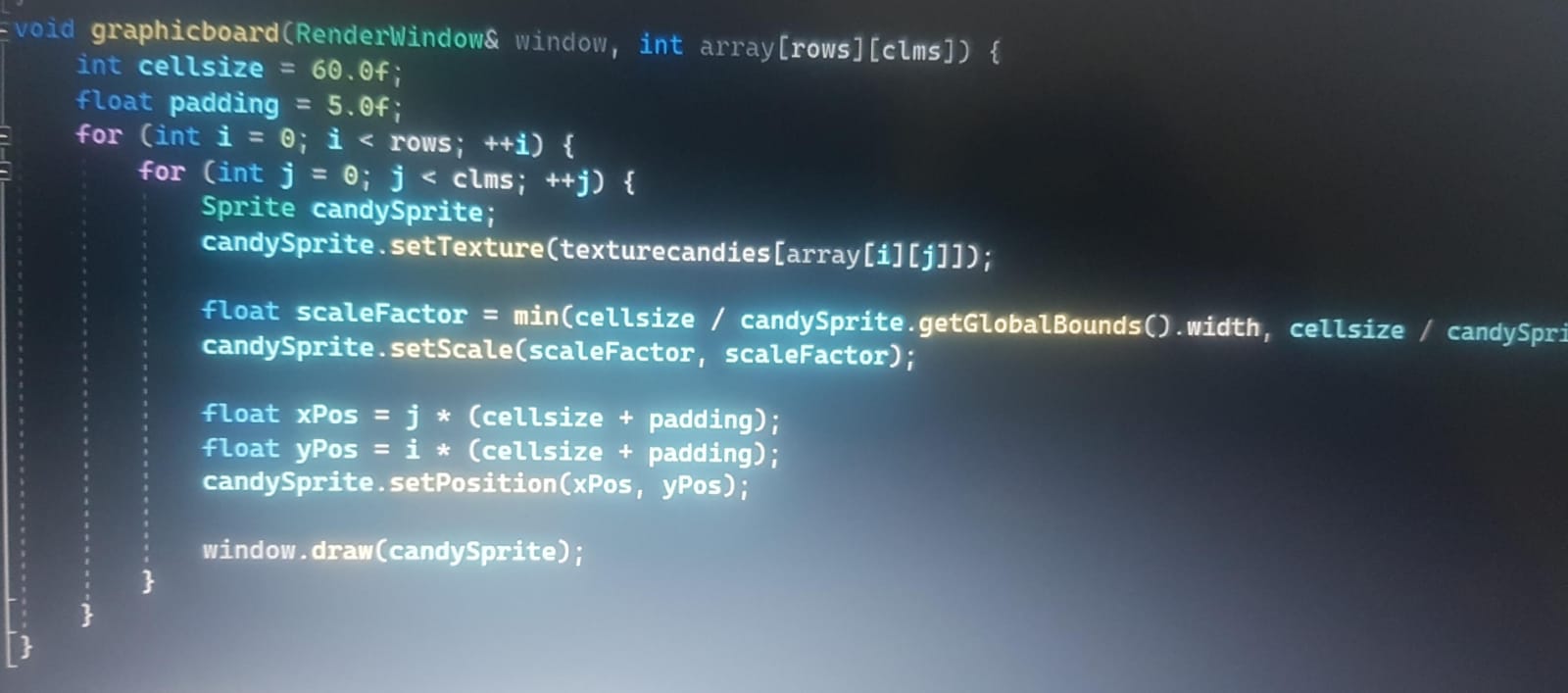
FUNCTIONS

We have declared two constant Integers which we can change according to the users requirements. (Render window )window represents the sfml screen.We have also adjusted the size of the window for the user.

**Printboard** is the first function we have used.This function runs 2 loops to make a 2d array and at every index a random number is generated and to generate a random number we have used rand() builtin function as in orignal game there are six candies so in formula of (rand%6+1) we have used six number which helps in generating number from 1 till 6 and each number represents different candy.

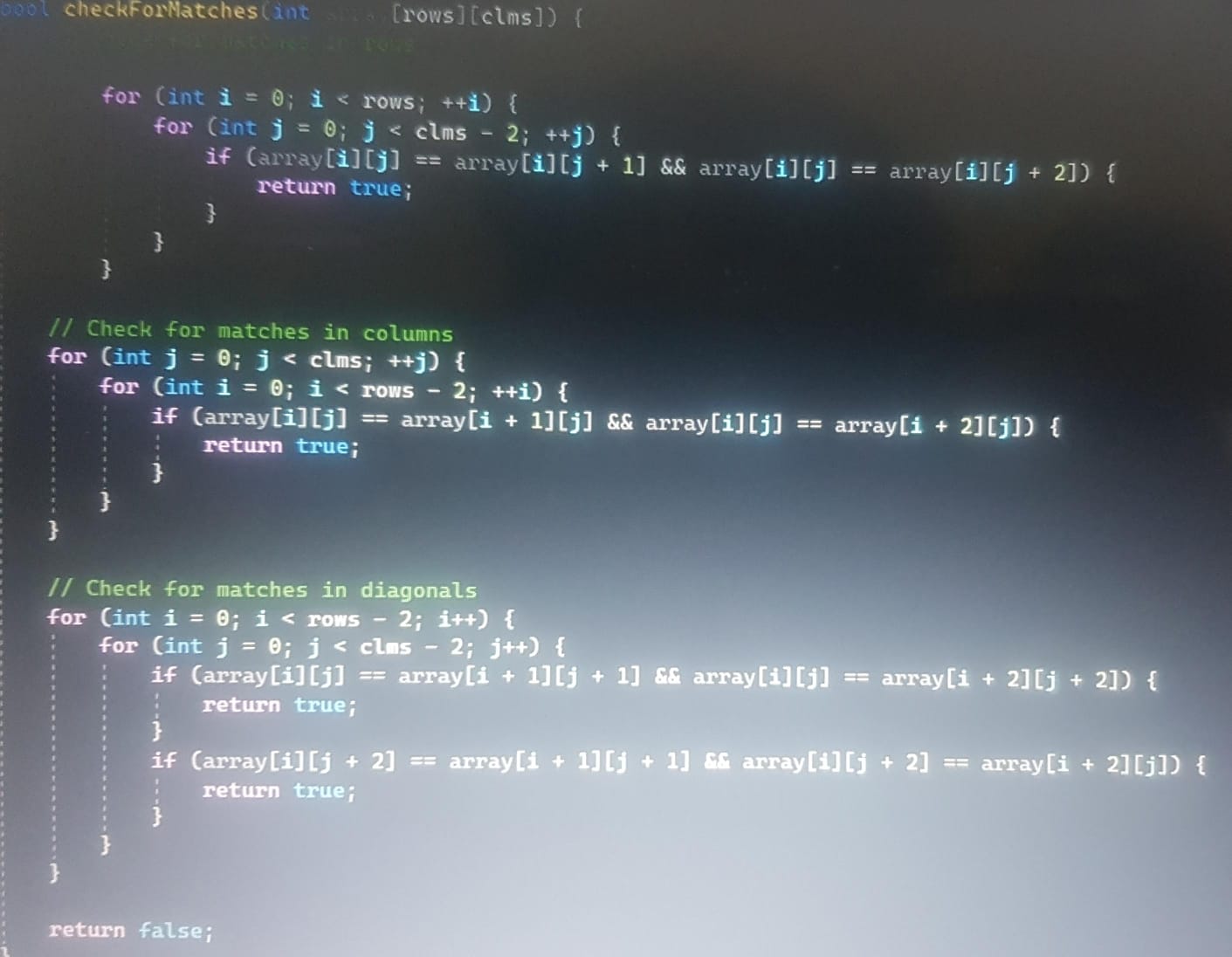


**Graphicboard** function’s primary purpose is to visually represent a candy board in SFML window. Each element denotes a specific type of candy. For each candy type, an SFML Sprite named candySprite is created and assigned the texture. It calculates a scaling factor to ensure that the candy sprite fit within a defined cell size on the board the sprite is scaled accordingly, and its position is determined based on the array indices, considering both the cell size and a specified padding.



**checkformatches** is the function which we have used in which a function has used loops to visit every index of the board the function checks column wise, row wise and diagonally as well.If any three consecutive index value are same

values ar same in a row or in a row it returns true as return type of my function is bool.the function also checks for diagonals and anti diagonals and if diagonals and if values at diagonals in any position gets match it returns true.



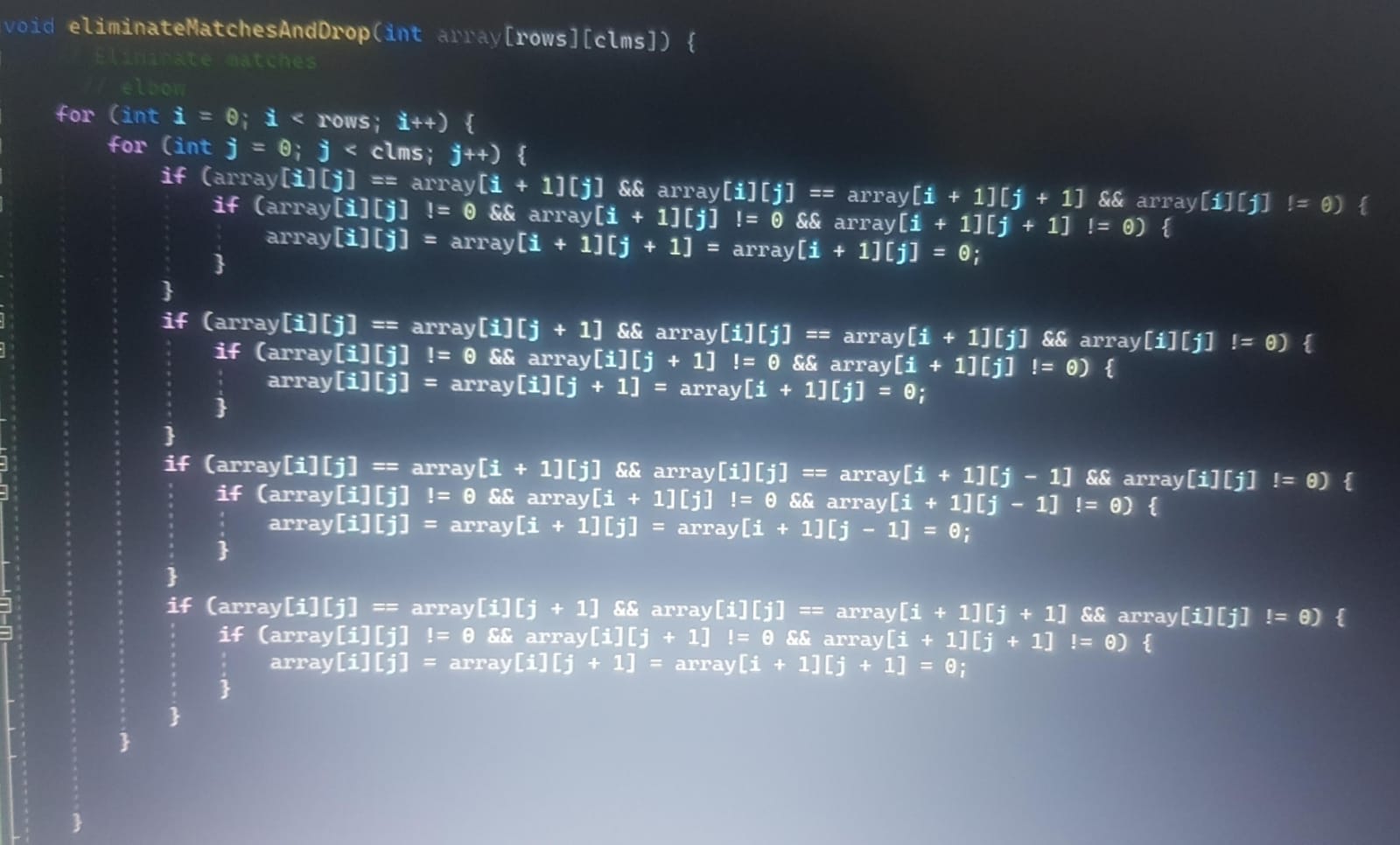
(eliminatecandies and drop) is the main core of my game.This function contains many functionailities like it helps in checking in checking matches and helping eliminate candies by assigning number 0 to that specific index where my match condition is satisfied.

The function checks for diagonals ,elbows ,it checks for three consecutive candies in row and coloumn. The function checks if any four consecutive index value contains same value it helps in generating a special candy of value 7 as well in a row or in column.

The function checks also checks if there is index which contains the value which is same for next next four consecutive indexes in a row or in a column which results in a special candy bomb getting form at that index which is represented by value of 8.

The function also checks if 4 index values making a diagonal or anti diagonal exist or not it it exist it assigns them with value of 0.The function also checks if any 5 index values of the board making a diagonal or antidiagonal is same or not if it is same it assigns value of 0 at that five indexes.

The function **eliminatematchesanddrop** is the main function of my game.The function checks for matches at indexs and eliminate them.



First it checks for elbow .2 loops are running to visit every index of board and the logic used is for four types of elbows and if a elbow pattern is made by three index point in such a way that the values at that index points are same it changes the value and give that indexes value of 0.In our game we have made a elbow of three candies .

Secondly The functions checks for 5 match .2 for loops are running and a check is made in every row that if a index[ i][j]exist and next four other index point contains same value as index [i][j]so value at index[ i][j]becomes 8 a value for my special candy and next four index value in a row gets value of 0.columns are checked the same way as rows and same criteria is used to check for 5 match in the column if it exists the the next four index in column is assigned with value of 0 and index [i][j]gets value of 8.

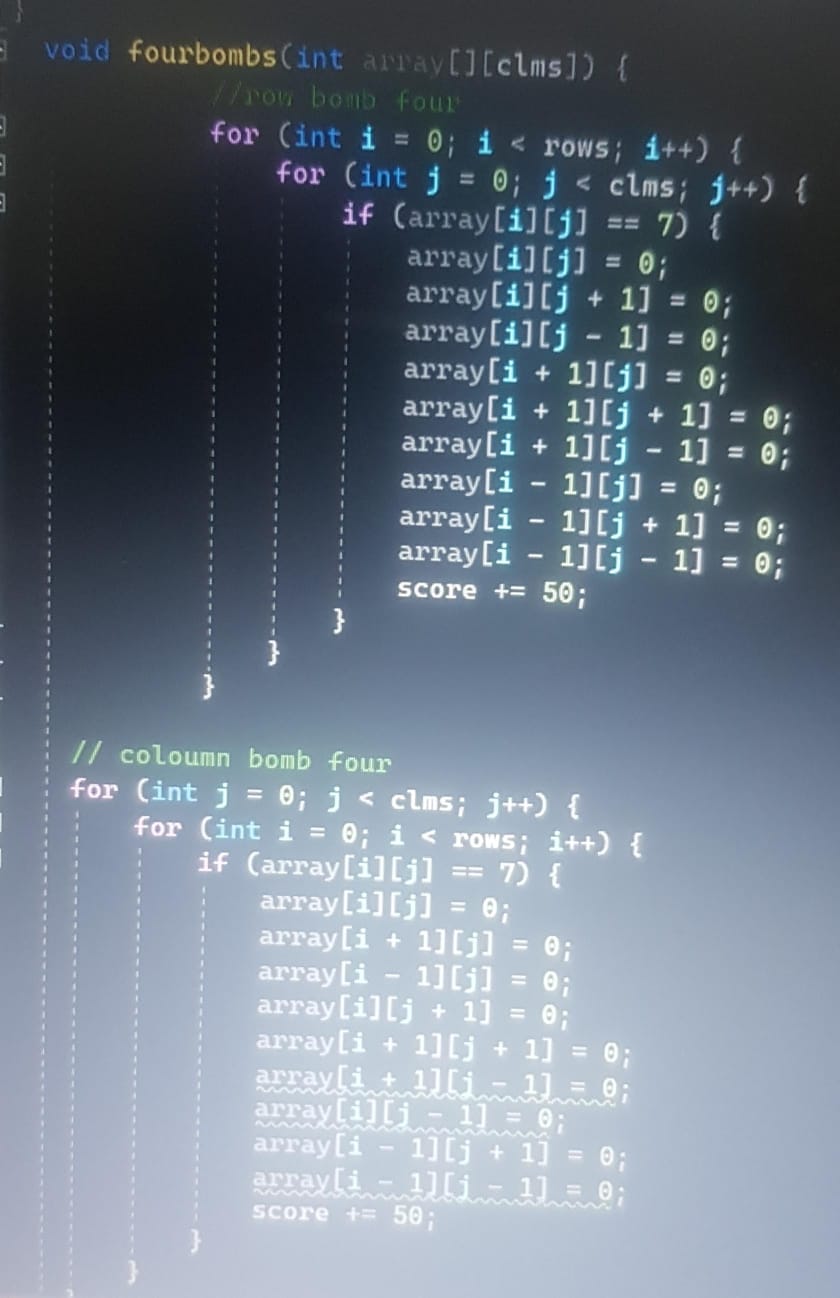
The functions also check for four match in a row and in column as well .Again 2  for loops are used which is used to visit every index point of the board and if a index[i][j] point exist in a row whose value is same as to the next three index values on the same row if it is True it assigns the index [i][j]with  value of 7 and other next consecutive index value with value 0.Same criteria is used for the column and if a index value [i][j] is same to next three consecutive index values in a column it assigns index value [i][j] with value 7 and next three comsecutive index in that particular column with value of 0.

The function checks for three matches in a row Or in a column. For check in a row the function runs 2 for loops to.vist every index in each row and it checks that if a index [i][j] exist whose value matches with next 2 consecutives index point in the same row if it is True it assigns all index [i][j] and next two consecutive index with value of 0.

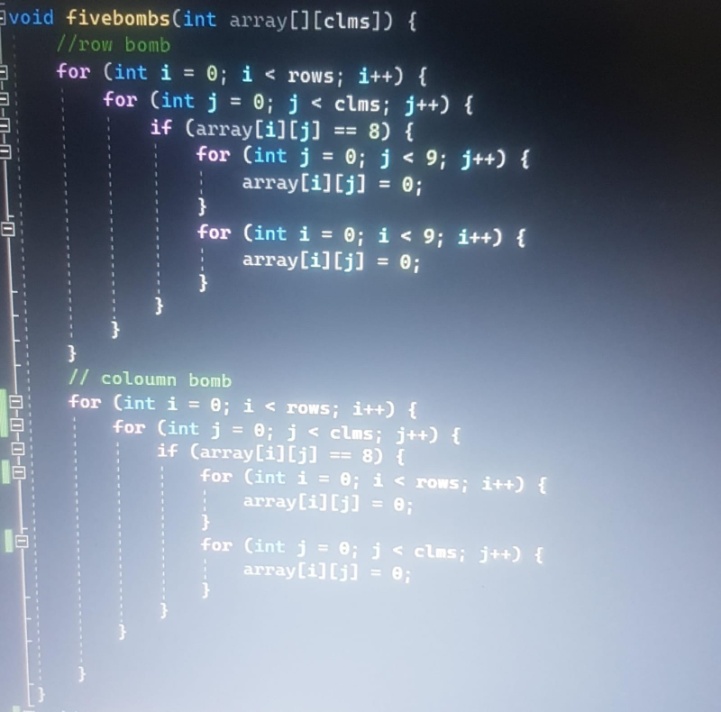
Same criteria is used to check match in a column and if a  value index point matches to next 2 index value in a particular column it assignment value 0 to three of that indexes.

The function also checks that 5 index making a diagonal and anti diagonal container the same value if it's True it assignment all that index making diagonal or anti diagonal with value 0.same it checks if any four indexes making a diagonal Or anti diagonal contains same value it assigns all four index with value 0.function seen if any three indexes makes a diagonal and antidiagonal and it contains the same value it assigns all three indexes with value 0.

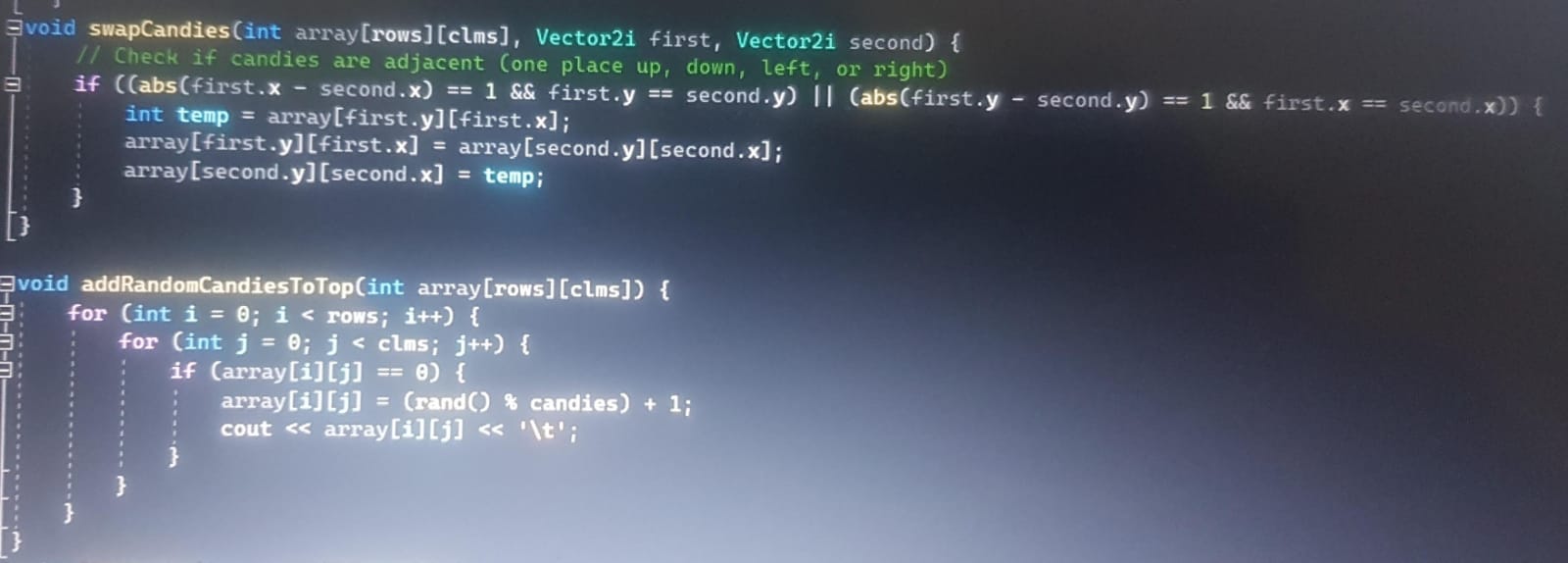
The next function is **fourbomb .**It is an void type function.The function checks every row index points by using 2 for loops and it checks a index point whose value is 7 .If it is True it pops it three by three region like a index [i][j] contains value of 7 then it assigns all index around this index making a 3×3 region including index [i][j] with value of 0.Same criteria is used to check in a column that if a index exist in a column with value 7 then the function assigns all the index making a 3 ×3 region including the index [i][j] with value 0.



The function **fivebomb** is a void type function. The function checks first row wise with help 2 for loops tocheck a index point with value of 8 if it exists it has a loop which runs in that particular row and assigns 0 value to all index in that row .another for loops is used which run in that particular column which contains index [i][j] and assigns all index at that column with value 0.The function also checks a index value with value of 8 column wise with The same criteria.

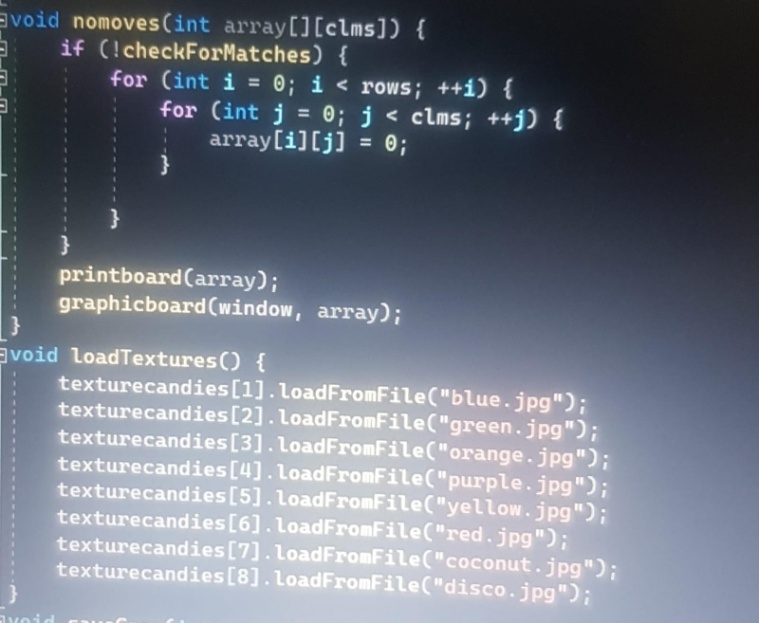


The **swapCandies** function plays a vital role in enabling the interaction between candies in a board. the function first checks if the pair of candies is adjacent, verifying that they are positioned either one up, down, left, or right from each other, then function starts the swapping process. By temporary variable, the values of the two candies are exchanged .On board the corresponding candies are showed according to numbers.



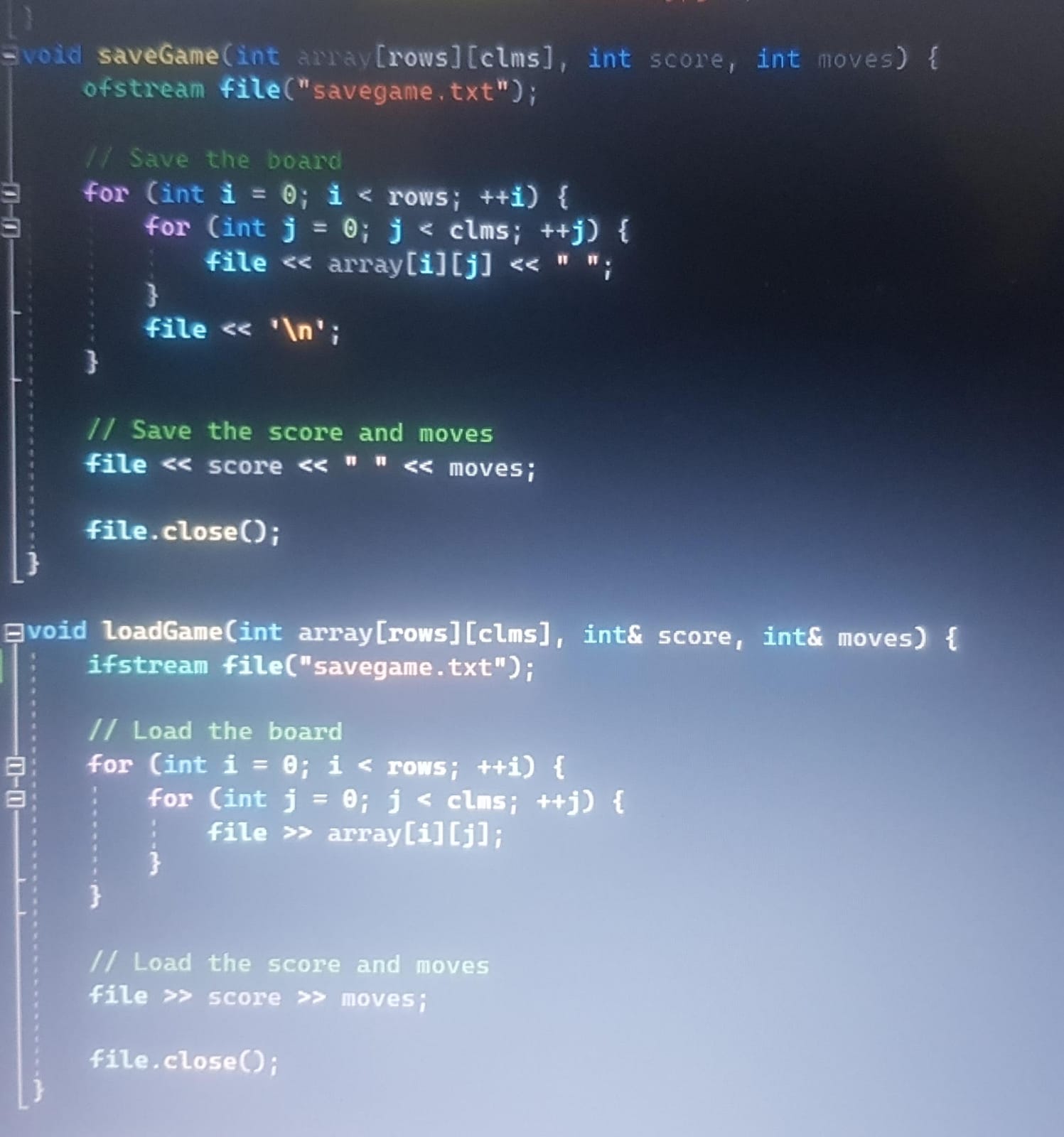
**Addrandomcandeistotop** is the function used to add generated new candies at the first row at every index .It used two.for loops and checks if row is 0 means first row it assigns all indexes in that row with a random value using rand function which is rand%6+1.

**Loadtextures** is a void function which is helped to load pictures to.the program and to the window in which different candies represent different textutes and all have a specific value and all of them are loaded from files as there path are decalred where they exist .



**Savegame** is a void function to save the current content of the game. The paremeter passed to is a the 2d array or my board game,score and moves is passed by reference .It uses file handling. The file savegame is opened by ofstream mode so.data can be written on it so all the 2d board is stores in it .Two for loops are used to store every index value in the file .The current score and moves are also stored in the file .

**Loadgame** is another function used to load the save game .parameters passed to.it is a 2d array or my game board,score and moves is passed by reference. It uses file.handling file is opened in reading mode 2 loops are used and to.move in a 2d board and every board index recipes value from.the file( savegame) .score and moves values are also given by this file .



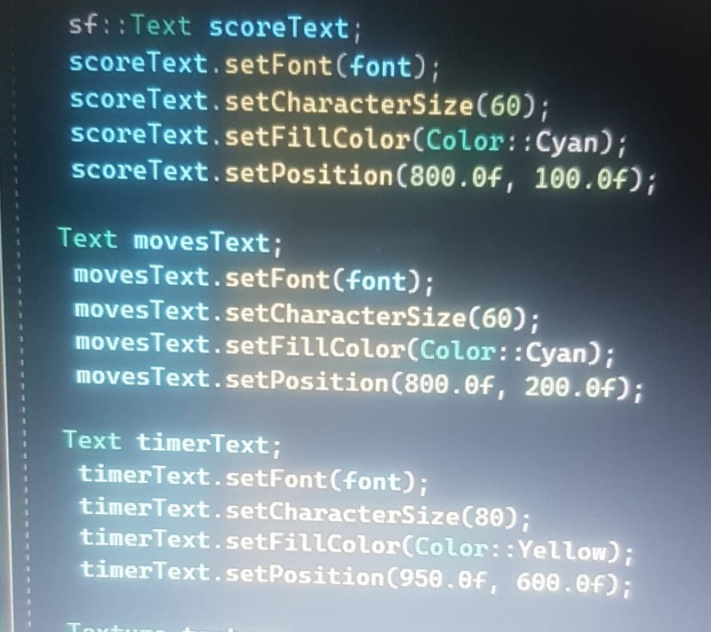
The function **nomoves** is used to shuffle.te board if there is no mpve possible. This functions calls many function in it .First checkformatches function is called in a if condition if it is True is run 2 for loops which assigns every index of 2d array with value 0 .Then print board function is called which generates new value to the every index and graphicboard another function is.used to show sprite on window presentation the board of the game .

MAIN

The code starts by declaring a **Clock** object named timer to measure time intervals and a float variable **padding** for space between candies on the board.

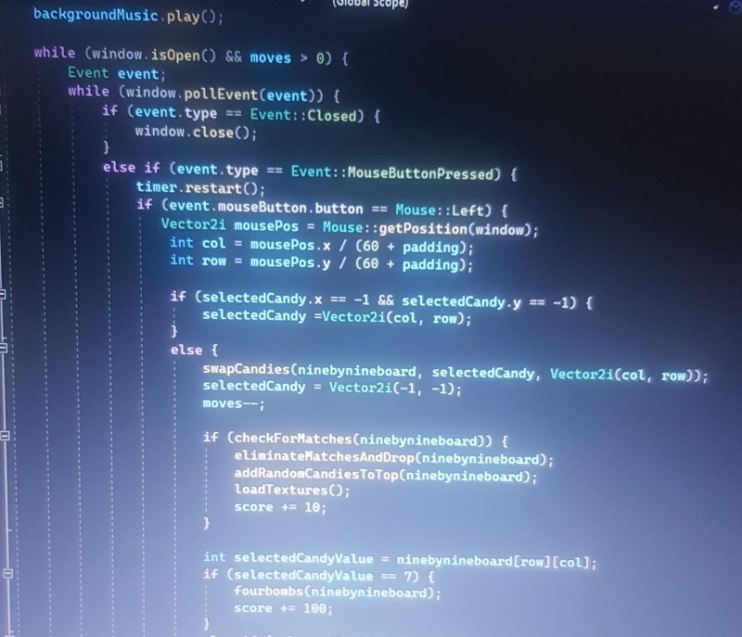
It loads a **font style** from the (.ttf) file for displaying text in the game. If the font loading causes error, the program returns tells that error has occurred.

Variables for the number of moves **moves** , the player's score **score** , and an array representing the game board (ninebynineboard) are initialized.

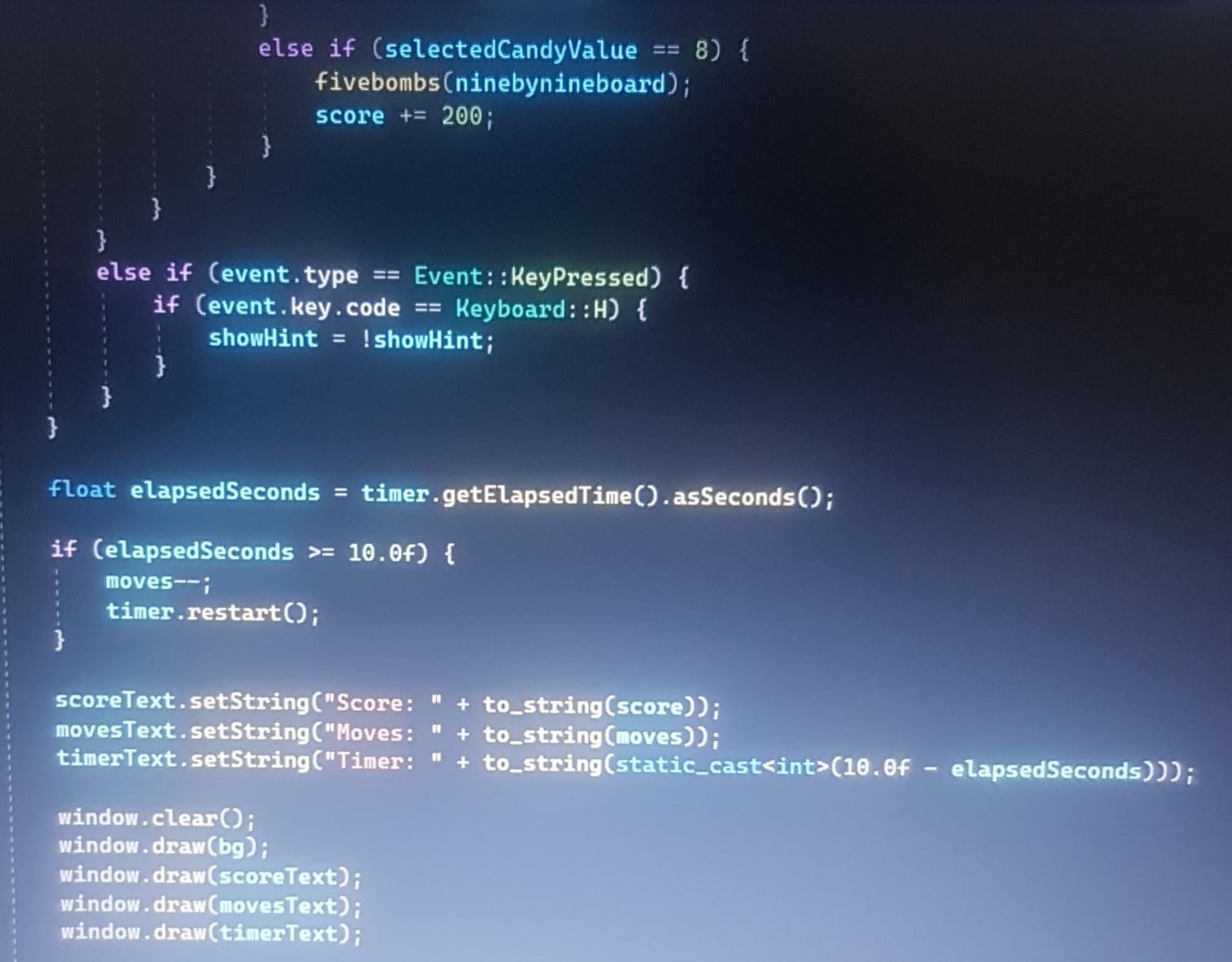


The code opens a file named "savegame.txt" to check if a **saved game** exists. If yes it closes the file, loads the saved game using the **loadGame** function, and continues the game from the saved data.If no saved game is found it prints the initial state of the candy board using the **printboard** function.The code loads textures for candies and sets the background image using an SFML Texture and Sprite. The background image is scaled to fit according to the window.

SFML Text are created for displaying the score, moves, and timer. These texts are positioned on the window. Background music is also loaded from the file ogg. If the music loading fails, an error message is displayed.



The main game loop, controlled by while **(window.isOpen() && moves > 0 )** handles user input and updates the game state according to the moves by the user.Player input is detected, including mouse clicks for candy **swapping**.The timer is used to decrement moves after a certain time interval, and the corresponding text displays are updated.The candies are also **highlighted** by colouring outer boundary of selected cell.



The code checks for candy matches, eliminates matched candies, drops candies down, adds random candies to the top, and updates the score. Special candies haver specific actions, such as explosions.

When the game loop exits (due to running out of moves or closing the window) it stops the background music, saves the game state using the saveGame function, and returns 0 to indicate successful program completion.

