

BMS COLLEGE OF ENGINEERING

Department of Information Science and Engineering



“THE GAME OF TETRIS”

Guided By:

Dr. Shambhavi B R
Dr. H S Guruprasad

Presented By:

H Anarghya - 1BM17IS028
Irene komal P - 1BM17IS033
Lalitashree R hegde - 1BM17IS039
Pannaga Sharma M L - 1BM17IS053

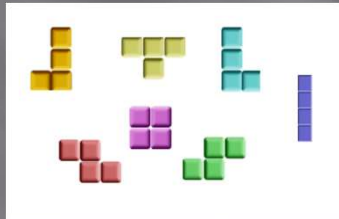


THE GAME OF TETRIS:

Tetris is a fall-matching puzzle video game originally designed and programmed by Alexey Pajitnov in the Soviet Union.

Problem statement:

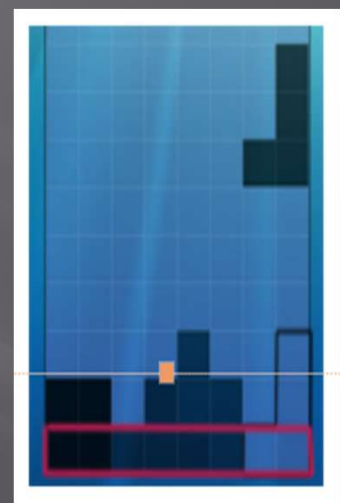
Tetris is played on 10 by 20 grid called the 'Matrix'. Shapes called tetrominoes are to be generated randomly. Player places them on the matrix. The primary way to score points in tetris is to clear lines by manipulating the pieces so that they fill horizontal row within the matrix.



7 tetrominoes

How do you play tetris?

- As I'm sure most of you know, in tetris *tetrominoes* falling from the top of the board. The player moves and rotates the blocks and stacks them up.
- Here the black outline is one of the places you can put the funny shaped block. And when a row is filled entirely with blocks x, you get a *clear*; that entire row is removed and the rest of the board is shifted down and increases your score

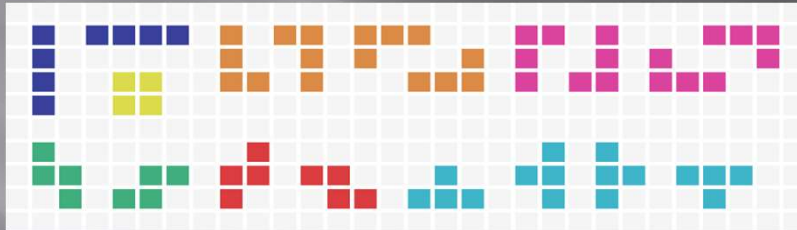


Tetris Outline:

Data structure used :

An array of structure with variables, i.e., an array to store the shapes of the tetrominoes and an integer variable to store points assigned to each variable.

A 3 dimensional array of dimension 4x4x4 in which the first index of the array holds the possible rotations the shapes could have and the rest hold the shapes.



Algorithm used:

Brute force, which is the basic algorithm, is used as our design tool for implementing the game. We make use of this algorithm to check the boundary conditions and collision detection.

The implementation of the game is divided into different modules. Different modules are:

▪ Void rotate_shape() :

The rotation of the tetrominoes is done by this function. It takes the shape and rotation number as argument.

Then as and when the player wants to do rotation for the shapes, the rotation number is incremented by one, as it is stored in the array and the shape corresponding to the incremented rotation number gets printed in the matrix.

Since the shapes have different number of rotations, it's justifiable that those that have 4 rotations will have the rotation numbers from 0 to 3 and those that have 2 rotations will have rotation numbers 0 to 2 in their array indices.

▪ Void print_shape_on_board():

This method is used to print the temporary shapes and their movements onto the draft matrix. It takes the shape number, rotation number and the co-ordinates of the shape as arguments. The matrix gets updated everytime the user makes movements to the shape in the game, i.e., either the rotations or the left, right movements.

- Void print_on_fboard():

This method is same as that given above ,the only difference is that this function makes changes to the fmatrix or the final matrix that has the final or permanent positions of the blocks .Initially it will be same as the matrix ,but as the user keeps on making movements and finally places in the matrix,those changes will be updated in this fmatrix through this function.

- Void checkposition():

This is the error and the collision detection function. The function checks for the matrix of the shapes that have 1 and then checks it with the boundary conditions.If the block encounters any of the boundaries of the framework,the function returns 0. If the block encounters another block,it returns 2. If there is a successful placing of the block, then the function returns 1.

- void score():

This method is used to display the score on the console ,everytime a block is placed and everytime a row gets cleared.Also displays the final score when the game gets over.Basically, it checks the rows of the fmatrix from the bottom,whether the rows have filled up or not. If the rows have filled up,the score of 100 points is added to the current score. Also when each block gets placed,its respective points gets added to the score. When there is no space available for any other blocks to come or get placed, game gets over and the final score will be displayed.

- Void initfmatrix(): This function is used to get the outer framework or the main matrix of the tetris in which the game is to be played.

- Void displayboard(): This function is used to print a matrix that acts like a draft matrix for printing the temporary movements of the shapes and gets reinitialized everytime its called and the screen is cleared for a new matrix to get printed.

- Main():

Here ,first the initfmatrix function is called to set the layout of the game.Then the game begins. First a random shape appears and the player is asked for rotations and other required movements.The left right movements are restricted within the boundaries of the layout using the checkposition function.When the player tries to move beyond the boundary conditions,its not allowed. Then the block is made to fall down by default,if the player is not interested to make any left or right movements.Finally the fmatrix gets printed with the permanent positions of the blocks inside it and the score is shown. This keeps continuing. If the player wants to quit the game in the middle,Ctrl+C is to be pressed.

THE END

Yes, we can play with the tetris now :D