Classic Tetris Project Report

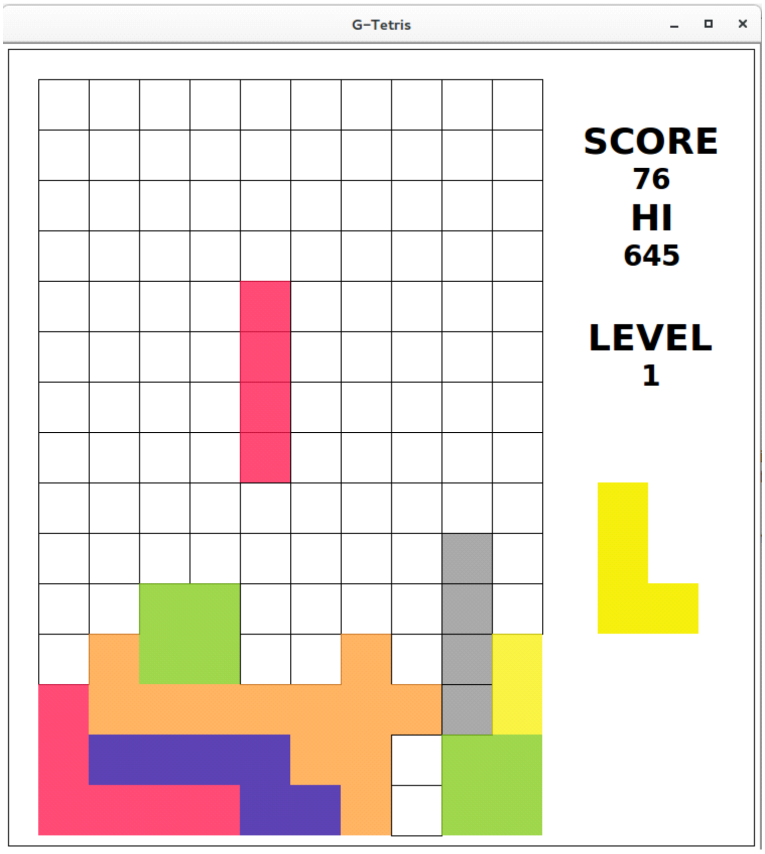
CS 154 - Abstractions and Paradigms in Programming

Project by the following students:-

Yash Gupta	Mayank Singhal	Piyush Onkar
(160050037)	(160050039)	(160050012)

DESCRIPTION:

We have implemented a famous Russian game Tetris in our project with an AI which tells the best position for the falling tetromino to place. Each Tetromino consists of four square pieces joint orthogonally in all the possible ways. The aim of this project was to learn how to use abstractions and higher order functions. We have used syntactical abstractions (list-comprehension and while loop), state, structures and graphics from universe.rkt. This is a very good example of the same.



SCORE
76
HI
645

LEVEL
1

Fig 1

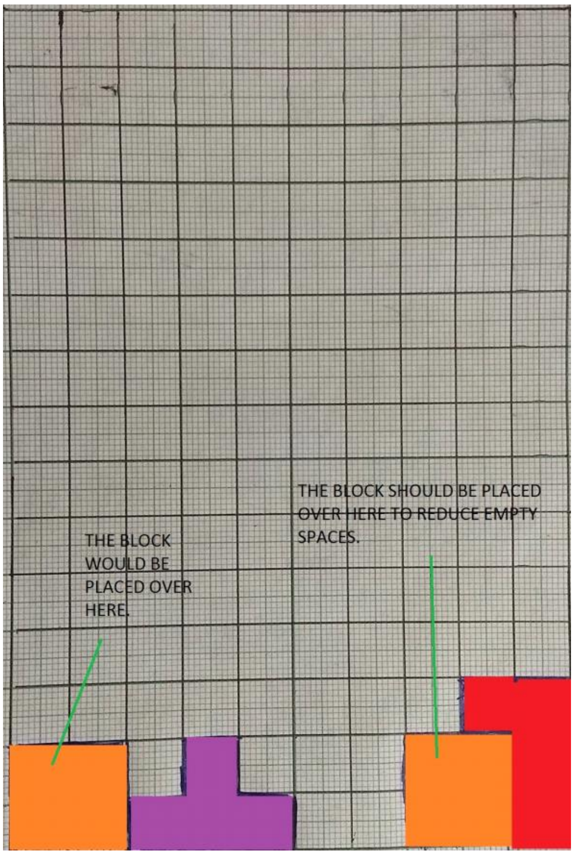

THE BLOCK SHOULD BE PLACED OVER HERE TO REDUCE EMPTY SPACES.
THE BLOCK WOULD BE PLACED OVER HERE.

Fig 2

DESIGN OF THE PROGRAM:

We represent the grid of the game as a vector of 150 elements where each element consists of position and a value. The value of the element tells whether that position is filled or not, if filled then with which color. Each tetromino is a structure named block which consists of its type, a list of four numbers telling its position and the state of the block (related to rotation).

We have used all the possible combinations of Tetrominos.

There is a structure world which represents the current state of the game and consists of the grid, current-block, next-block, level, score, time, best position of block. We have used brute-force algorithm for developing the AI of the game.

We have used htdp/universe library for management of events and htdp/image library for drawing the various visuals in the game. The event handling in the pogram is done by a function called big-bang, which is basically a higher-order function that uses a set of mathematical functions to manipulate the state in response to events like ticks(of an internal clock), key presses etc.

To maintain between subsequent executions of the program, the highscore is written to and read from an external text file.

Sample Input and Output:

Input has to be given in the form of arrow buttons to rotate a block and to move it.

The UP arrow key rotates the block in clockwise direction. The LEFT and RIGHT arrow key is used to move the block horizontally and the DOWN arrow key moves the block downwards. Pressing the ‘c’ key clears the highscore.

The output consists of the playing area, to the right of which the score, highscore, level and next incoming tetromino are displayed. *(Fig 1)*

NOTE: This being a graphical program we have not provided sample input output cases.

Limitations and Bugs:-

1) The AI algorithm only checks possibility of falling block about how block should be placed so that empty spaces remain minimum. The AI fails to check the possibility of sliding and placing the falling block in accumulated blocks when the falling block is just about to get accumulated. *(Fig 2)*
2) The AI algorithm uses some arbitrary coefficients to calculate a ‘score’ for each possible placement of the tetromino. These coefficients have been given their values by repeated trials and error so while it is possible that they are near a local maximum, but a method to find the global maximum remains elusive. So this is clearly not the optimal solution.