SUBMISSION DATE: 04-DECEMBER-2022

FINAL PROJECT

PROGRAMMING FUNDAMENTALS (CS1002) – FALL2022

INSTRUCTIONS

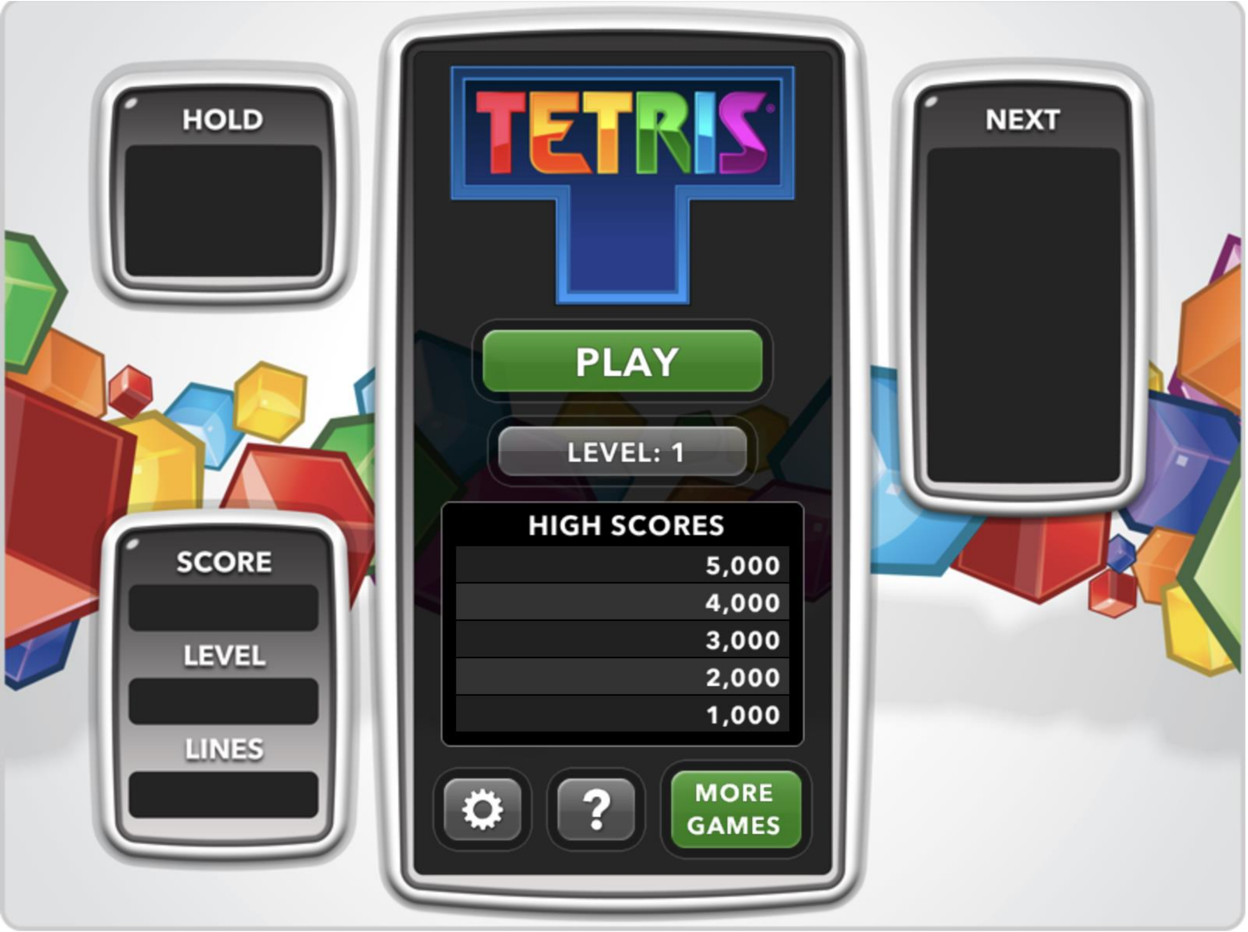
1. **Plagiarism in course project will result in F grade in the course.**
2. This is not a group project and each person will be working on the project individually.
3. Make sure you submit your project before the submission time. Late submissions won't be accepted even if they are late by just one minute.
4. You can earn bonus marks by implementing extra features in the project.
5. Use good programming practices (well commented and indented code; meaningful variable names, readable code etc.).
6. Each file that you submit must contain your name, student-id, and assignment # on top of the file in comments.
7. Combine all your work in one folder and compress it into a zip file. The folder must contain *.cpp* files (no binaries, no exe files etc.).
8. Submit the solutions via google classroom. Submissions via email will not be accepted.
9. Use proper naming convention to name the file containing source code. Failure to submit according to the above format would result in the deduction of 10% marks.

   E.g., *i21xxxx_project.cpp*, replace i21xxxx with your roll number.
10. Follow the given instructions to the letter, failing to do to so will result in a zero.
11. For timely completion of the project, start as early as possible.

DESCRIPTION

Tetris is a tile-matching puzzle game, originally designed and programmed by Russian game designer Alexey Pajitnov in 1984—the product of Alexey's computer programming experience and his love of puzzles.

The goal of Tetris is to score as many points as possible by clearing horizontal lines of blocks. The player must rotate, move, and drop the falling Tetriminos (Tetriminos are game pieces shaped like tetrominoes, geometric shapes composed of four-square blocks each) inside the Matrix (playing field). Lines are cleared when they are filled with blocks and have no empty spaces. Below is an example picture attached from the official Tetris game.



HIGH SCORES
5,000
4,000
3,000
2,000
1,000

GAMEPLAY SUMMARY

The game play of Tetris is simple; you bring down blocks from the top of the screen. You can move the blocks around, either left to right and/or you can rotate them. The blocks fall at a certain rate, but you can make them fall faster if you're sure of the positioning. The objective is to get all the blocks to fill all the empty space in a line at the bottom of the screen; whenever you do this, you'll find that the blocks vanish and you get awarded some points.

Fundamental Rules

Tetris has very simple rules:

- The blocks can be moved only in specific ways (left, right, rotate, drop).
- The blocks can only be removed from the screen by filling all the blank spaces in a line and clearing that line.
- The game is over if the blocks stack up in such a way that the lines are not cleared and reach the top of the screen.
- Each time a line is cleared in Tetris, you are awarded points.
- Each time the blocks reach the top of the screen, the game is over.

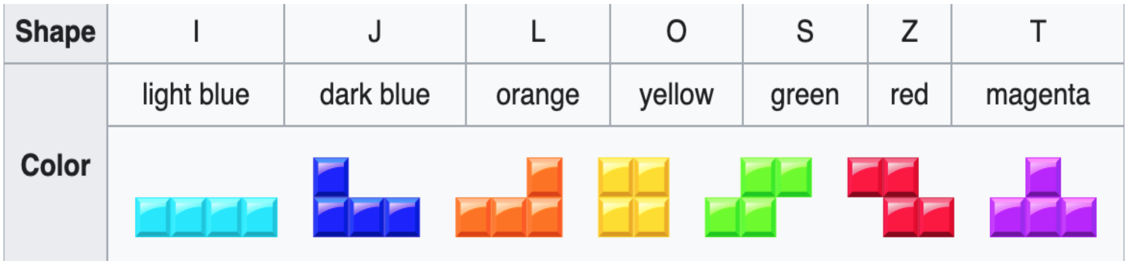
Features To Be Implemented

Game Interface:

The game consists of a matrix i.e., the playing field. All the blocks and lines are to be enclosed within the matrix. Initial code of the game interface is provided. You are required to build upon that interface.

Generation of random shape:

The shapes are composed of 4 squares arranged in 7 different patterns – you can use your creativity for implementing the logic of creating different shapes. There are seven different types of blocks: I, O, L, J, S, Z, and T. You are required to generate these blocks by defining separate array for each block or a single array for the all the shapes. The colors of the blocks are varied. This means that block J can be in all the given possible colors. At each step blocks are to be randomly introduced in the game, analyze the attached figure.

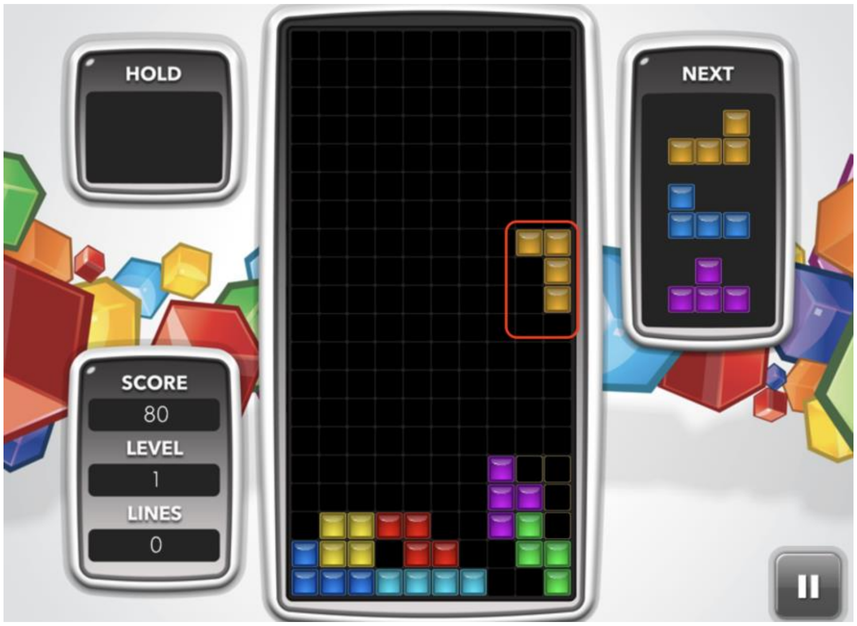


Shape	I	J	L	O	S	Z	T
Color	light blue	dark blue	orange	yellow	green	red	magenta

NOTE: The image for these pieces your will be using in the game is already in the provided folder.

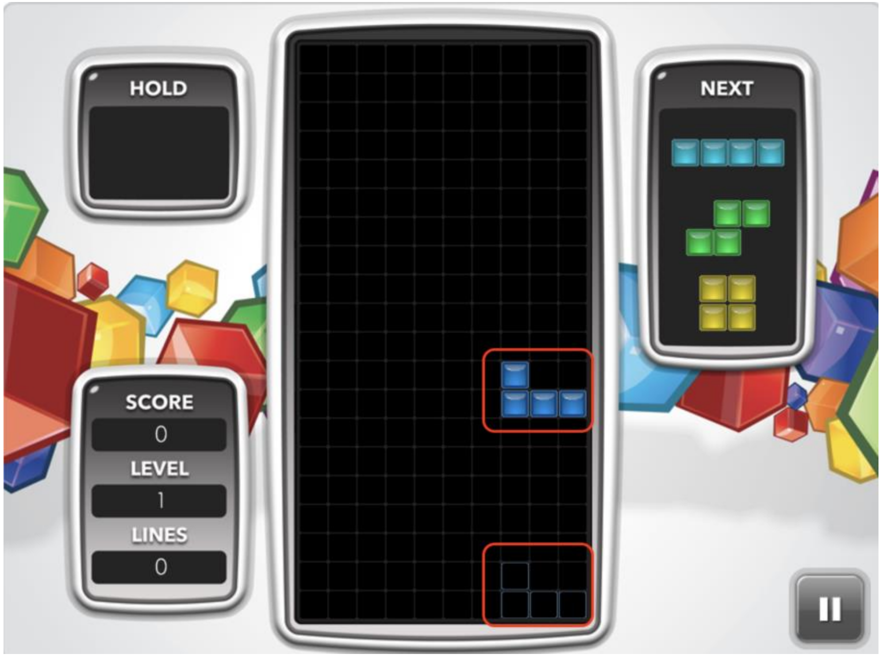
Movement, Rotation and Dropping of shapes:

The user can move the blocks around and rotate them. The movement of the block is either left or right. The movement of blocks is to be done using the arrow keys of the keyboard. The blocks are supposed to be dropped in the matrix automatically. Conventionally, the blocks fall at a certain rate, but you can make them fall faster by using the downward movement.



Shadow of the dropping shapes:

Before the shape is dropped a shadow of it should appear in the matrix. This feature will help the user in deciding if the block can be placed at that particular spot in the matrix of not.



Dropping the Bomb:

1. Bomb will fall randomly on a random spot and would not be a controlling piece like the others.
2. If the bomb color matches the block, then all blocks will be destroyed.
3. If the bomb drop does not match the block color; only 4 blocks will be destroyed
4. If the bomb is dropped in vacant areas, then nothing will happen.

NOTE: More than 10 blocks matching the same color should be destroyed

POINT SCORING:

a. Line completion:

If the user can completely fill one horizontal line with colored squares, that line disappears and any filled squares above move down. If the player is unable to fill lines completely, the blocks will stack up and eventually reach the top of the playing field. Following are the scores given in line completion:

LINES COMPLETED	SCORE
1 line	10 points * level
2 lines simultaneously	30 points * level
3 lines simultaneously	60 points * level
4 lines simultaneously	100 points * level

b. Same colored block matching:

GAME OVER:

Each time you clear a line in Tetris, you are awarded more points. Each time your pieces reach the top of the screen, your game is over.

GAME LEVEL:

The game is based on 2 levels. In the beginner's level the user is provided with the option of only 4 different blocks. In the advanced level all blocks are unlocked and the pace of dropping down is increased.

LEVELING UP THE DIFFICULTY:

The user spending more than five competing/struggling in filling game rows, the difficulty level should increase. When the specified interval runs short, the last row should become non-interactive for the game anymore. Continue the process each time five minutes are spent.

HIGH SCORES:

When the game starts, the high-scores are read from a file (named highscores.txt) and stored in an array of size 10. If the score of the current game makes it to the list of high-scores, the lowest score is removed from the array and the new contents of the array overwrite the file.

MENU:

The game should be menu-driven and the following options should be available to the user:

1. Start a new game
2. See high-scores
3. Help
4. Exit

During the execution of the game, the player can press the 'P' key to pause the game and see the following menu:

1. Start a new game
2. See high-scores

3. Help
4. Exit
5. Continue

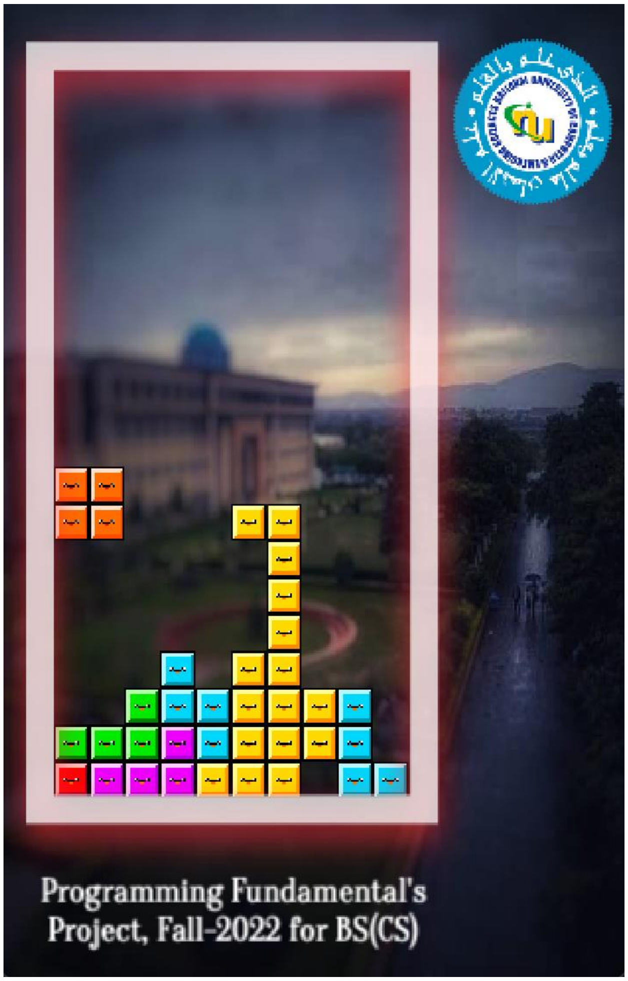
The names of the menu items are self-explanatory and should perform the relevant function upon pressing the respective number key. In the Help menu, you can show the rules/keys to play the game.

BONUS:

You may think of any interesting (and programmatically challenging) feature to implement to get bonus marks. The decision to give bonus marks will be with the teachers so you should discuss any idea with your respective teachers to know if it qualifies as a bonus or not.

STARTER CODE & INTERFACE

We have provided the basic skeleton code of the game in C++ and you are allowed to make changes to it, modify it and add functionality according to the project statement. Exploring and understanding the code is part of the project. After running the game, you will be able to see the following interface and a single block should be falling from top-left corner. Read the code comments to understand several parts of the project.



Best of Luck
Happy Coding ☺