**C++ 2019A - MTA - Exercises**

**Requirements and Guidelines**

The exercises in the course would require you to implement a game combining in some way two known games: [Pong](https://en.wikipedia.org/wiki/Pong) and [Tetris](https://en.wikipedia.org/wiki/Tetris).

Note: the exercise should be implemented in Visual Studio 2015 or later, with standard C++ libraries and run on Windows with Console screen of standard size (80\*25), using gotoxy for printing at specific location on screen (X, Y).

Submission is in MAMA, as single zip file containing only the code and the vcproj and sln files + readme.txt with IDs -- but without the DEBUG folder and without any compiled artifact.

**Exercise 2**

In this exercise you will implement the following additions to your pong game:

1. **BOMB**

There is a new trick in the game - turning the Ball into a Bomb.

Left player “Bomb” key is ‘s’ or ‘S’ key

Right player “Bomb” key is ‘k’ or ‘K’ key

Once pressed, the Ball turns into “becoming a Bomb” for 8 game iterations (square moves) and it changes its color to a color of your choice that indicates its threatening new state, then after 8 moves, the Ball becomes a “Bomb” and changes its color again to a color indicating that it is a Bomb. The rules for when a Bomb become back a regular Ball appears below.

After pressing the “Bomb” key, the game would ignore following “Bomb” key press requests from the same player, till the ball would visit 4 times the column x=40

The behavior of a Ball in “becoming a Bomb” and “Bomb” states are as following:

1. When the ball is in “becoming a Bomb” state - the player can hit it and the behavior is the same as regular Ball hit - the ball would become a “Regular Ball” again, immediately after the hit - without becoming a “Bomb”
2. When the ball is in “becoming a Bomb” state and the player did not hit it, it is not yet a miss, the ball continues on screen - it will NOT be a “miss” yet!
3. If the ball hits a “dead board” or the screen edge while being a “Bomb” - the entire column of “dead boards” (or first line in case it hit the screen edge) would be removed and the player would gain one column back - the Ball would appear again at X=40, moving to the direction of the other player.
4. In case the ball hits “dead board” or the screen edge when NOT being a Bomb - it is a miss.
5. In case the ball is in “Regular Ball” state and it misses the board - it’s an immediate miss, do not continue with the ball movement.
6. In case the ball is in “Bomb” state and it hits the board - it’s a BIG MISS and the player moves 3 columns ahead (+ falling as a dead board, as usual).
7. A ball in “Bomb” state counts the time of being a bomb - if the time reaches 4 game iterations when the board is in the area between boards - it becomes a “Regular Ball” again, if it reaches 4 game iterations count when not being between Boards - it stays a “Bomb” and rule 3 would apply.
8. **Game against the Computer**

Game shall allow now to play against the computer. The computer shall have 3 levels: (a) BEST (b) GOOD and (C) NOVICE

In all levels the computer will move only one square in each “game iteration” - so when the ball makes one square move, the computer may make one square move

BEST - should calculate the exact position for not missing the ball

GOOD - should miss the ball occasionally (randomly, once in 40 ball hits)

NOVICE - should miss the ball occasionally (randomly, once in 10 ball hits)

=> The BEST level would also use the Bomb trick wisely, the other levels would not

There shall be three options to play:  
2 human players

Human vs. Computer - Human will always be on the left

2 Computer players

**Menu**

The game shall have now the following entry menu:

(1) Start a new game - Human vs Human

(2) Start a new game - Human vs Computer

(3) Start a new game - Computer vs Computer

(4) Continue a paused game

(8) Present instructions and keys

(9) EXIT

**Note**

You should change your original code, to use new materials that we learned - where appropriate.

**Priorities**

1. Game Human vs Computer and Computer vs Computer - allows 80/100  
   (if written properly!!!)
2. Adding the “Bomb” story - only for Human Players - allows 100/100  
   (if written properly!!!)

Note: Human vs Computer - the Human can use Bomb and it is OK if the Computer doesn’t play right even in BEST level when hit by a Bomb.

1. Adding the “Bomb” story - also for Computer - allows 110/100  
   (if written properly!!!)  
   Note: Computer shall be able to turn Ball into Bomb occasionally (randomly, but in the proper position on screen!) + the Computer shall avoid getting hit by a bomb in all levels, GOOD and NOVICE would randomly ignore the Bomb and get hit by a bomb.