

C++ 2022-2023 B - MTA - Exercises

Requirements and Guidelines

The exercises in the course would require you to implement a [pacman game](#) as a console application.

Note: the exercise should be implemented in Visual Studio 2019 or later (e.g. VS 2021), with standard C++ libraries and run on Windows with Console screen of standard size (80*25), using gotoxy for printing at a specific location on screen (X, Y), using _kbhit and _getch for getting input from the user without blocking, and Sleep for controlling the pace of the game.

Submission is in MAMA, as a 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 1

In this exercise you will implement the basic pacman game for a human player.

Read the pacman rules in the wiki link above, but you should support only the required features listed below.

The screen should be used as follows:

- The first 24 rows should be used to present the board.
- The last row (row number 25 on the screen) should be used to present the number of points and “remaining lives” information.

When the game starts the pacman is positioned at his start position (you should decide where it is) without any movement. Once the user selects move direction (using the keys, as listed below) the pacman will continue to move in this direction even if the user doesn't press any key, as long as the pacman doesn't hit a wall and the “STAY” key was not pressed.

Keys:

LEFT	a or A
RIGHT	d or D
UP	w or W
Down	x or X
STAY	s or S

Menu

The game shall have the following entry menu:

- (1) Start a new game
- (8) Present instructions and keys
- (9) EXIT

Pausing a game

Pressing the ESC key during a game pauses the game. It *can be good* to present a message on screen saying: "Game paused, press ESC again to continue". But it is not mandatory to present such a message.

When the game is at pause state, pressing ESC would continue the game, with the pacman continuing his movement exactly as it was before pausing, as if the game hadn't paused.

The screen is consisted of:

- walls (you shall use any reasonable char to draw them)
- pacman (use @ or any other reasonable char)
- 2 ghosts (use \$ or any other reasonable char)
- "breadcrumbs" on all unvisited positions (you shall use any reasonable char to draw them)

The pacman travels on screen and "eats the breadcrumbs". Each eaten breadcrumb equals a point to be earned. Once all breadcrumbs on screen are eaten the game ends with a happy message of your choice. Pressing *any key* should get back to the main menu.

In case a ghost eats the pacman, the player loses one "life". If all "lives" are gone, there should be a message announcing "Game Over" then after pressing *any key* the main menu shall be presented.

The ghosts hover above the breadcrumbs without eating them. You can decide how they go (no need to make them "smart"). Their pace is half the pace of the pacman (for any two steps of the pacman they make 1 step). You may decide if a ghost can go over the other ghost if they meet (i.e. both may share the same position) or not.

The pacman can cross the screen in an invisible tunnel from the rightmost position to the leftmost position of the same row and vice versa, and from the topmost position to the downmost position of the same column and vice versa, if there are no walls at both sides. Ghosts cannot cross in those invisible tunnels!

Note that there is one screen in Ex1, which looks the same for all games.

Pacman has 3 lives. After losing a life all creatures start from the exact initial position, but without the breadcrumbs that were eaten.

Bonus points: colors. Other nice features.

Note: there will not be any bonus for music or any feature that requires additional binary files to be part of your submission! Adding large required binary files to your submissions may subtract points!

Notes on Ex1

1. If you decided to add colors (as a bonus feature) please add an option in the menu to run your game with or without colors (the default can be to use colors, but the menu shall allow a switch between Colors / No Colors) - to allow proper check of your exercise in case your color selection would not be convenient for our eyes. The game **MUST** work properly in the No Colors selection.
2. Please indicate inside your readme.txt file the bonus additions that you implemented.

הערות נוספות:

1. במצב ללא צבעים לא להשתמש בצבעים בכלל (גם לא כדי לצבוע את הטקסט חזרה בלבן) - כלומר במצב זה יהיו הדפסות רגילות עם COUT ללא שום COLOR CODES ושיטות אחרות.
2. במצב צבעים לעשות שימוש בשיטה שמתוארת בקוד שמצורף כאן, כדי לוודא שהצבעים יוצגו כמו שצריך (יש עוד שיטה עם COLOR CODES שנוטה לא לעבוד).

```
#include <windows.h>
#include <iostream>
```

```
int main()
{
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 1);
    std::cout << "blue line" << std::endl;
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 2);
    std::cout << "green line" << std::endl;
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 14);
    std::cout << "yellow line" << std::endl;

    return 0;
}
```