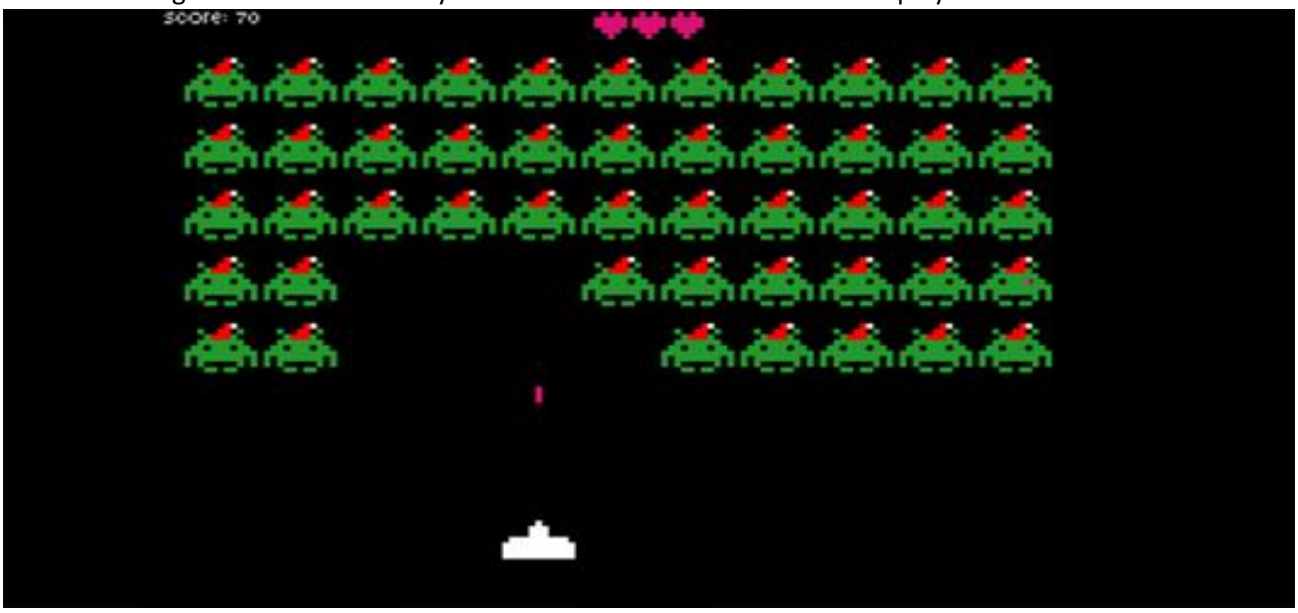
National University of Computer and Emerging Sciences, Lahore Campus				
	Course Name:	**Programming Fundamentals**	**Submission date**	**10-Dec-2023**
	Project	**Space Invader Game**	**Total Marks**	
	Section	**BCS-1D**	**Weight**	

SPACE INVADERS GAME

This is a screenshot of the original game to give you a rough idea. *Your* game will look very different with simple shapes instead of images.

Space Invaders is the classic retro shooting game in which a player has to out-maneuver and shoot down a hoard of incoming aliens before they take over the earth. You can play it at this link



http://www.freeinvaders.org/ (requires Flash player). **You must design and implement an updated version of this game with the functionalities described below.**

GAME FEATURES:

- **Scores** must be maintained and displayed throughout the game. The score of killing an alien is higher at the start of the game and reduced as the game progresses until it reaches a minimum score for killing an alien (you can decide that threshold yourself, after that threshold is reached the score will not be further reduced). So, the more time you take to kill an alien the lesser the score you will get for killing it.
- **Player lives** must be maintained and displayed throughout the game. A Player will have 3 lives at the start of the level.
- A **grid of aliens** (at least 5 X 10) must be present that **moves** side to side, and gradually makes its way to the bottom of the screen
- The aliens from the grid should **shoot the player** after random intervals of time.
- A **player ship** must always be displayed that can move UP, DOWN, LEFT, or RIGHT according to the key pressed. It can also **shoot a bullet** at the aliens.
- You must include a **pause** feature (e.g., by pressing ESC or pressing P) which will pause the game until the key is pressed again. A **pause menu** will be displayed after pausing the game with at least two options, quit the game and save the game.
- You must include a **save state** feature so you can close the game and resume it later. This will involve saving data into a text file and then later retrieving it.
- You must include a **high-score** feature. Your game must maintain a record of the 5 highest scores. These records must not be lost when the game is closed.

- You must have at least **two levels**, medium and hard. The hard level should have increased alien movement speed and increased alien bullet frequency.

Additional Notes:

- The player must be able to move in all **4 directions** (UP, DOWN, LEFT, RIGHT). Not just side to side.
- The player's **life must decrease** if it collides with an alien or encounters an alien bullet. If the lives drop to zero, the game is over.
- Alternatively, if even one alien reaches the bottom of the screen the game is over.
- The aliens must be destroyed if they collide with the bullet. The bullet must also disappear if it collides or reaches the top of the screen.

- If all the aliens are killed, the game is over, and the player has won.
- You can choose to have only one bullet on the screen at a given time.
- The objects on the screen (player, aliens, bullet) can be simple shapes (squares, rectangles, triangles), but try to differentiate between these objects by using different colors or shapes.
- You can have your own choice of key bindings for controlling the game (e.g., using WASD or using arrow keys)

Any additional features you implement to improve the gameplay or the game's appearance will be considered positively during your project evaluation. So, strive to make your game unique and enjoyable.

Submission Instructions:

- You must use the graphics library "help.h" you have been provided for rendering graphics and handling events. You can include this library by using **#include "help.h"** at the top of your program and putting the header file in the same folder as your source code. The help library includes functions like myRect, myEllipse, drawLine, drawText, isKeyPressed, etc., which you can use to implement graphics and input. You can add your own functions to the library too and read the existing functions in it. You are also provided with a **sample code** that can give you a basic idea of the workings of functions of the library.
- Plagiarism of any kind will not be tolerated.
- In case of any confusion, you can contact your TA or Course Instructor.
- You must submit a working, executable along with a source code. Also include a README file that details how to launch the game.
- You must follow good programming practices. Your code should be properly indented, with meaningful variable and function names, and with comments where necessary.