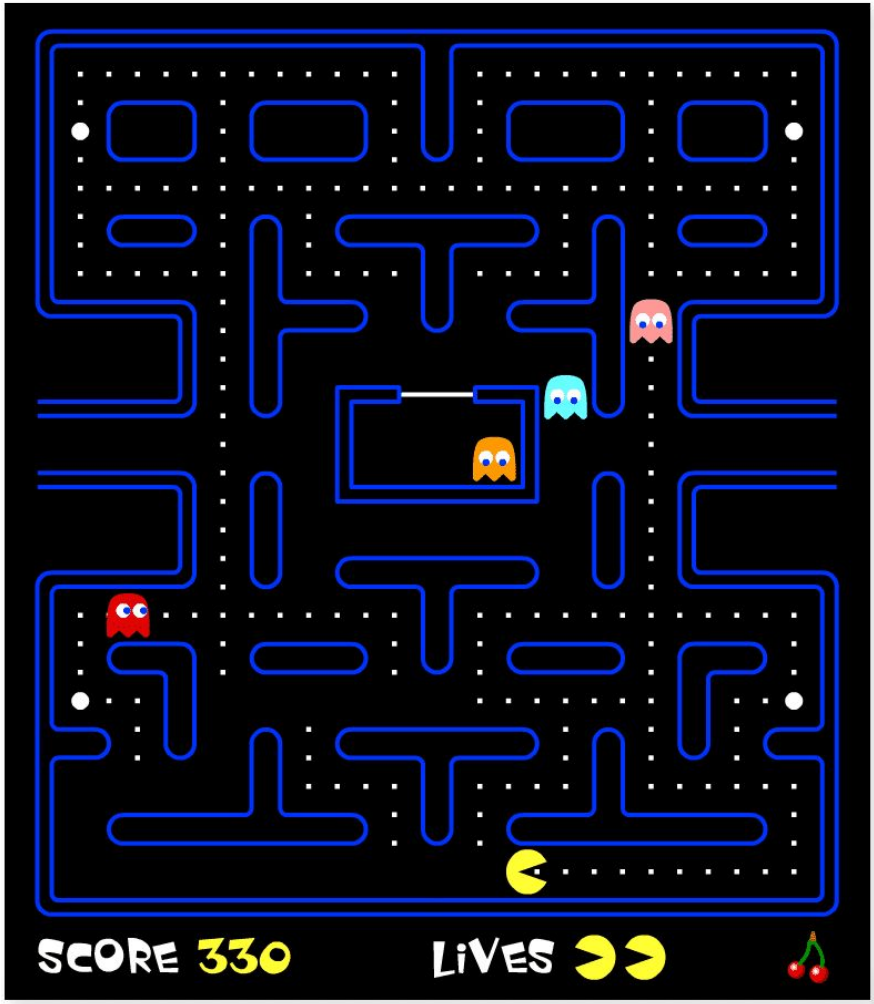
Game Development

(Display on Console at a Specific Location)

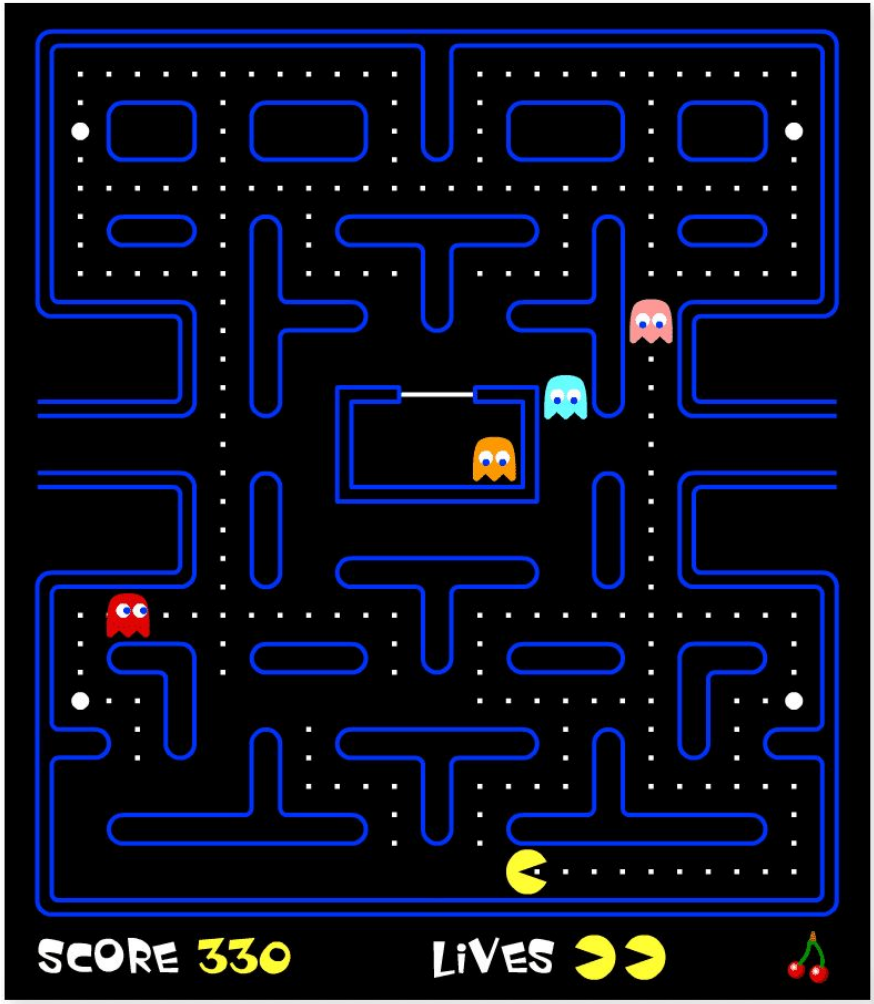
Pac-Man

Now, we have covered all the concepts related to implement any Game, except one.



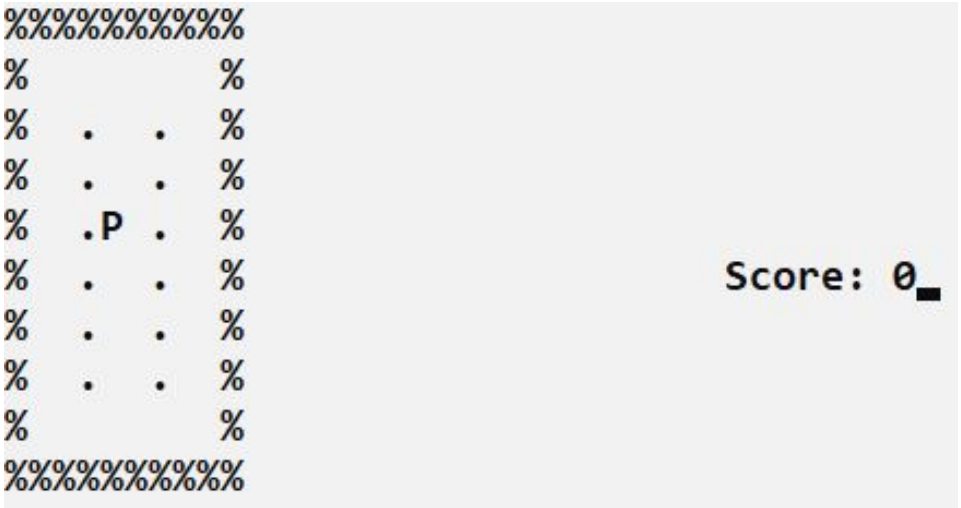
Pac-Man

We have not seen how to display the game statistics on a specific location on the console.



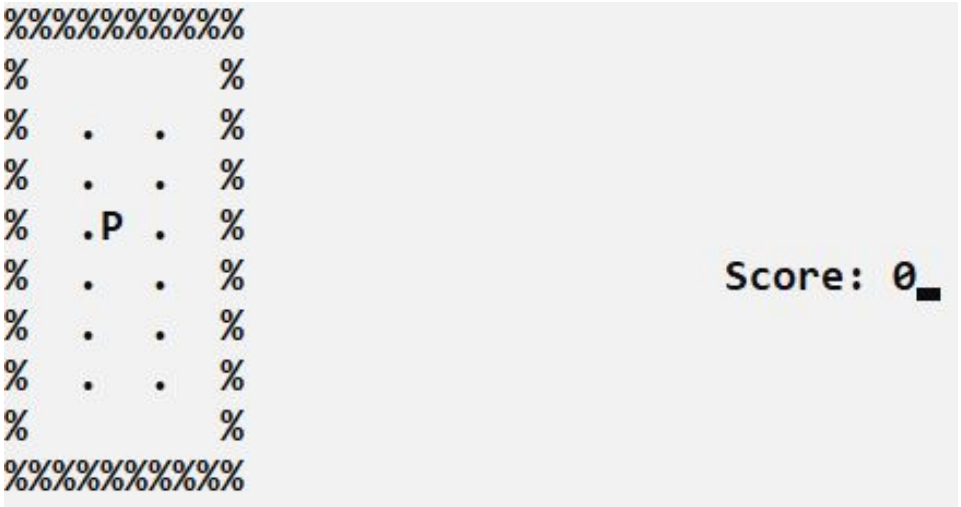
Pac-Man

We have not seen how to display the game statistics on a specific location on the console.



Pac-Man

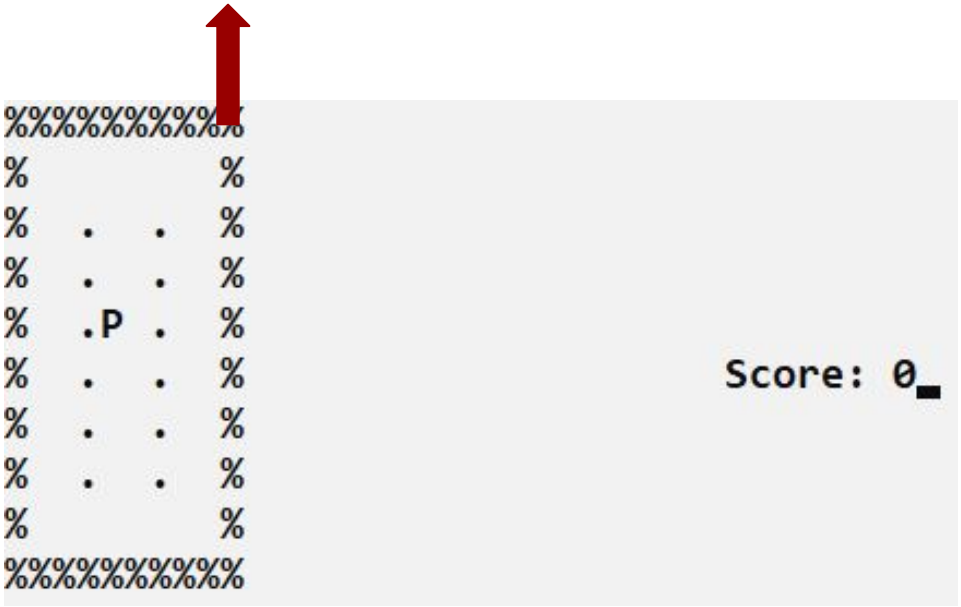
We will use a function gotoxy() and we will pass it the x coordinates of the console and y coordinates of the console and it will place the cursor on that specific location on the console.



Pac-Man: X and Y Coordinates of Console

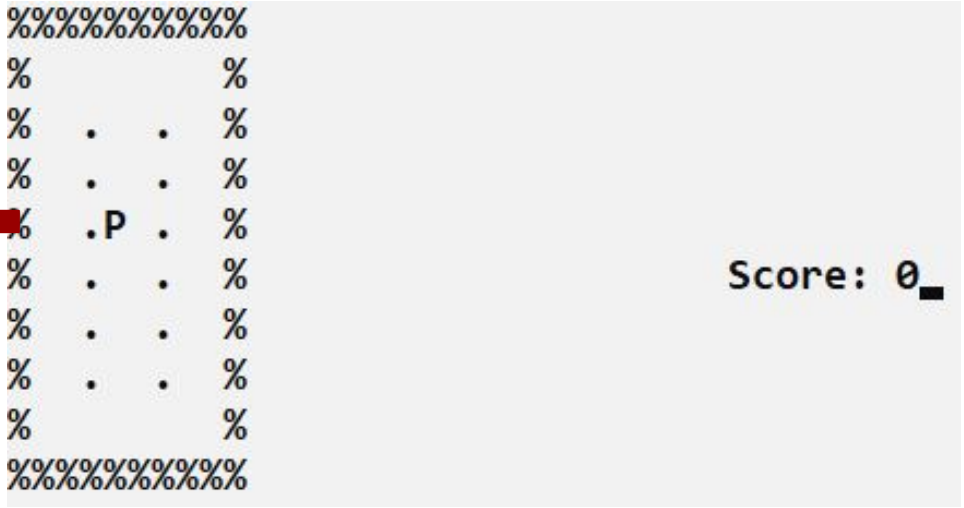
Pac-Man: X and Y Coordinates of Console

(9,0)



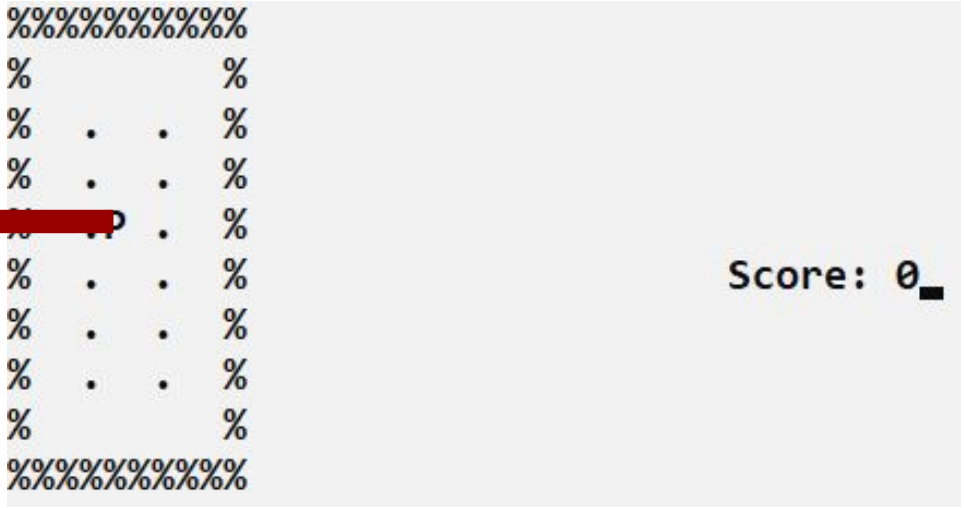
Pac-Man: X and Y Coordinates of Console

(0,4)



Pac-Man: X and Y Coordinates of Console

(4,4)



Pac-Man: gotoxy() Function

To use gotoxy() function we have to include windows.h file.

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

Pac-Man: gotoxy() Function

The definition of gotoxy() function is given by:

```
void gotoxy(int x, int y)
{
 COORD coordinates;
 coordinates.X = x;
 coordinates.Y = y;
 SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
```

Pac-Man: gotoxy() Function

We will not go into the functionality of this function, we will just copy this function in our project and use it.

```
void gotoxy(int x, int y)
{
 COORD coordinates;
 coordinates.X = x;
 coordinates.Y = y;
 SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
```

Activity 01

Let's write "Welcome to C++ Programming" at the (0,0) coordinates of console

Activity 01

Let's write "Welcome to C++ Programming" at the (0,0) coordinates of console

Activity 01: Solution

```
#include <iostream>
#include<windows.h>
using namespace std;
void gotoxy(int x, int y);    // Function Prototype
main()       // Main Function
{
     // Write your Code here
}

void gotoxy(int x, int y)     // Function Definition
{
 COORD coordinates;
 coordinates.X = x;
 coordinates.Y = y;
 SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
```

Activity 01: Solution

```
#include <iostream>
#include<windows.h>
using namespace std;
void gotoxy(int x, int y);   // Function Prototype
main()       // Main Function
{   system("CLS");
    gotoxy(0, 0);
    cout << "Welcome to C++ Programming";
}
void gotoxy(int x, int y)     // Function Definition
{
 COORD coordinates;
 coordinates.X = x;
 coordinates.Y = y;
 SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
```

Activity 01: Solution

First of all everything will be cleared from the console and then "Welcome to C++ programming" will be written at (0,0) coordinates of the console.

Activity 02

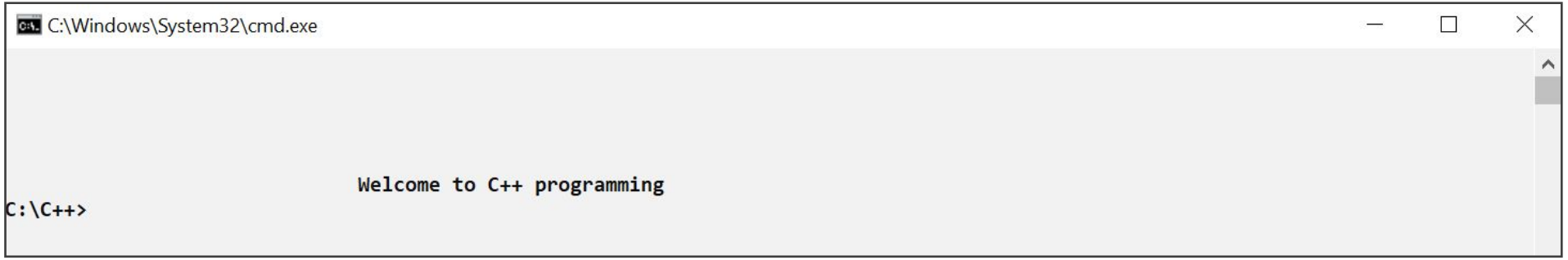
Let's write "Welcome to C++ Programming" at the (5,0) coordinates of console.

Activity 02: Solution

```
#include <iostream>
#include<windows.h>
using namespace std;
void gotoxy(int x, int y);   // Function Prototype
main()       // Main Function
{   system("CLS");
    gotoxy(5, 0);
    cout << "Welcome to C++ Programming";
}
void gotoxy(int x, int y)     // Function Definition
{
 COORD coordinates;
 coordinates.X = x;
 coordinates.Y = y;
 SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
```

Activity 03

Let's write "Welcome to C++ Programming" at the 30th x and 5th y (30,5) coordinates of console

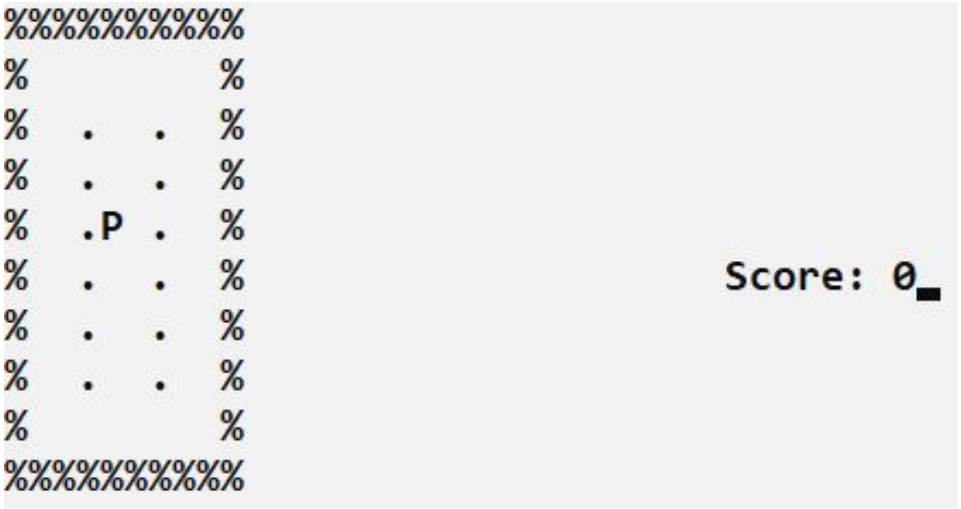


Activity 03: Solution

```
#include <iostream>
#include<windows.h>
using namespace std;
void gotoxy(int x, int y);   // Function Prototype
main()       // Main Function
{   system("CLS");
    gotoxy(30, 5);
    cout << "Welcome to C++ Programming";
}
void gotoxy(int x, int y)     // Function Definition
{
 COORD coordinates;
 coordinates.X = x;
 coordinates.Y = y;
 SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
```

Activity 04

Make a function that prints the score of the game at 30th x and 5th y coordinate on the console.



Activity 04

Make a function that prints the score of the game at 30th x and 5th y coordinate on the console.

```
int score = 0;
main(){
    bool gameRunning = true;
    while (gameRunning){
        Sleep(100);
        system("CLS");
        printMaze();
        printScore();
        if (GetAsyncKeyState(VK_LEFT)){
            movePacmanLeft();
       }
        if (GetAsyncKeyState(VK_RIGHT)){
            movePacmanRight();
        }
        if (GetAsyncKeyState(VK_UP)){
            movePacmanUP();
        }
        if (GetAsyncKeyState(VK_DOWN)){
            movePacmanDown();
        }
        if (GetAsyncKeyState(VK_ESCAPE)){
            gameRunning = false; }}
}
```

Activity 04

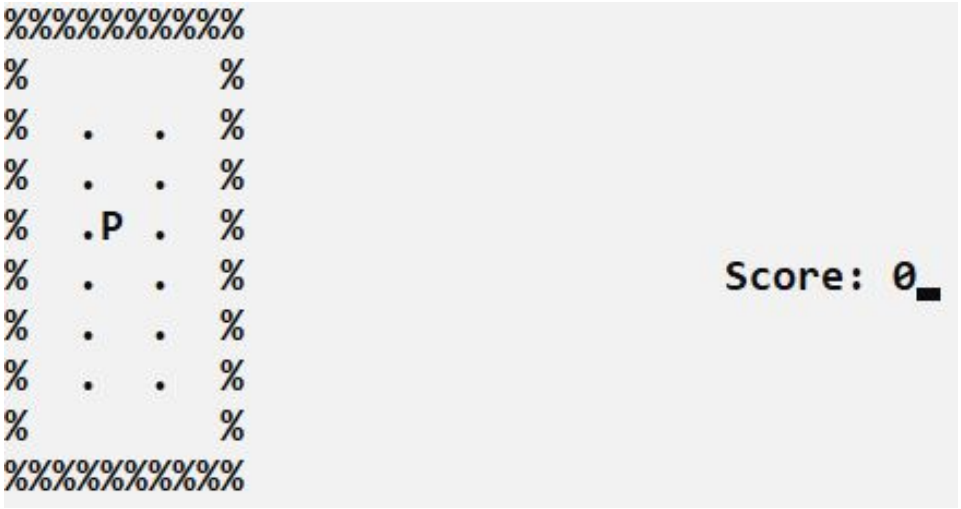
Make a function that prints the score of the game at 30th x and 5th y coordinate on the console.

```
void printScore()
{
    gotoxy(30, 5);
    cout << "Score: " << score;
}
```

```
int score = 0;
main(){
    bool gameRunning = true;
    while (gameRunning){
        Sleep(100);
        system("CLS");
        printMaze();
        printScore();
        if (GetAsyncKeyState(VK_LEFT)){
            movePacmanLeft();
        }
        if (GetAsyncKeyState(VK_RIGHT)){
            movePacmanRight();
        }
        if (GetAsyncKeyState(VK_UP)){
            movePacmanUP();
        }
        if (GetAsyncKeyState(VK_DOWN)){
            movePacmanDown();
        }
        if (GetAsyncKeyState(VK_ESCAPE)){
            gameRunning = false; } }
}
```

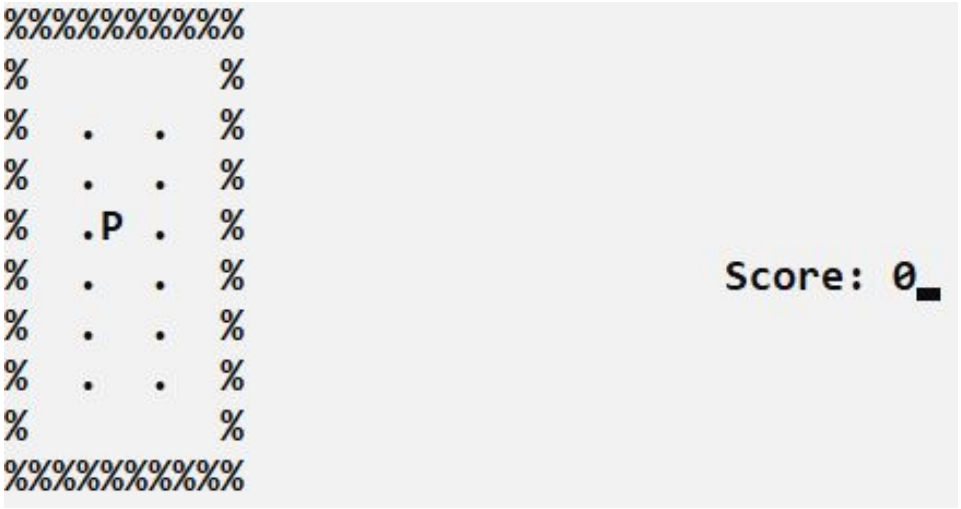
Pac-Man: Screen Flickering

Have you noticed the Screen Flickering while running the game?



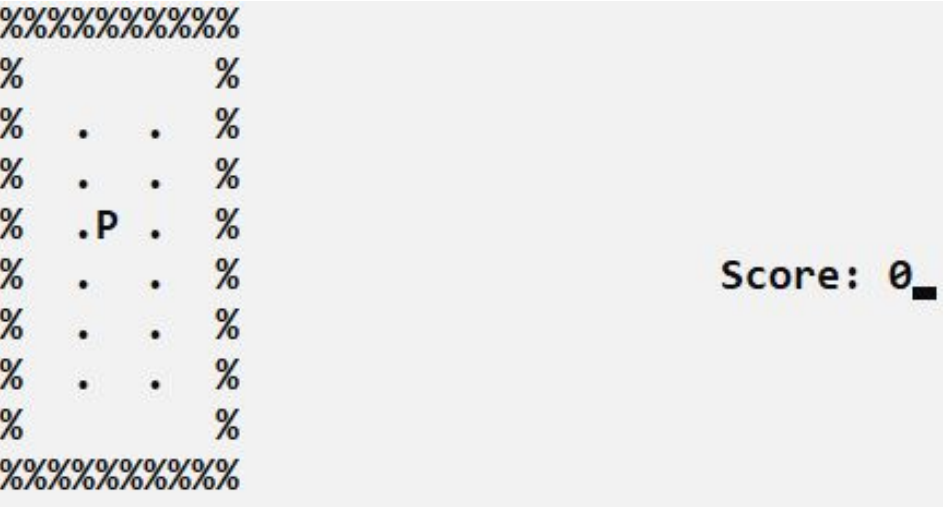
Pac-Man: Screen Flickering

Even Though we have used the Sleep(100) function.



Screen Flickering

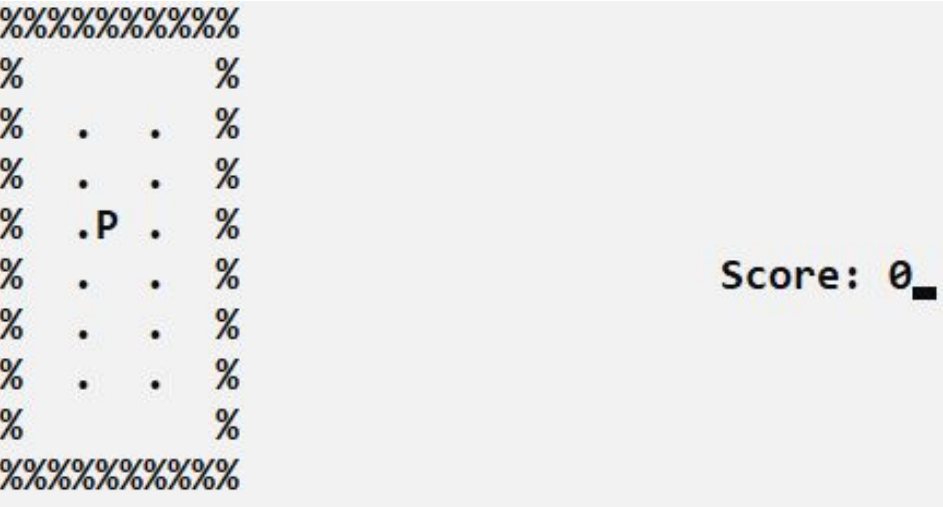
This is because we are printing the maze after every 100 seconds.




```
int score = 0;
main(){
    bool gameRunning = true;
    while (gameRunning){
        Sleep(100);
        system("CLS");
        printMaze();
        printScore();
        if (GetAsyncKeyState(VK_LEFT)){
            movePacmanLeft();
        }
        if (GetAsyncKeyState(VK_RIGHT)){
            movePacmanRight();
        }
        if (GetAsyncKeyState(VK_UP)){
            movePacmanUP();
        }
        if (GetAsyncKeyState(VK_DOWN)){
            movePacmanDown();
        }
        if (GetAsyncKeyState(VK_ESCAPE)){
            gameRunning = false; } }
}
```

Screen Flickering

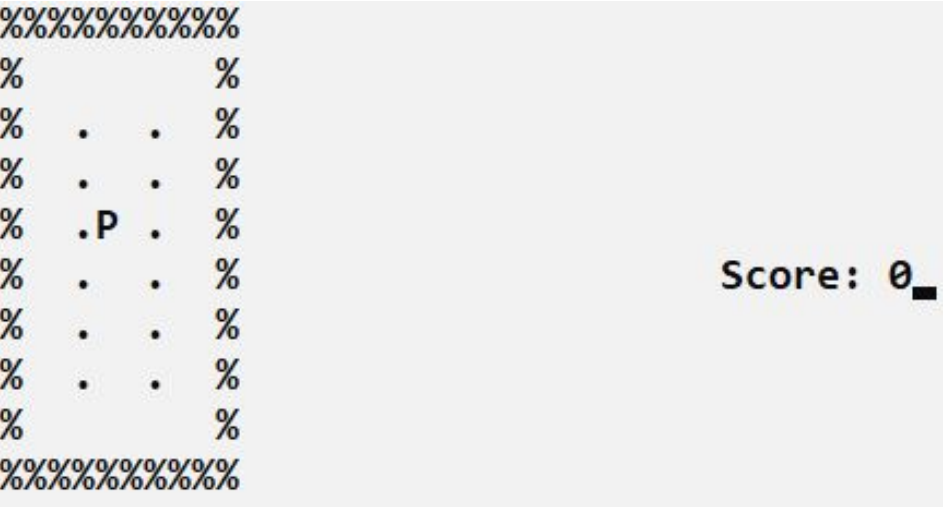
This is because we are printing the maze after every 100 seconds.




```
int score = 0;
main(){
    bool gameRunning = true;
    while (gameRunning){
        Sleep(100);
        system("CLS");
        printMaze();
        printScore();
        if (GetAsyncKeyState(VK_LEFT)){
            movePacmanLeft();
        }
        if (GetAsyncKeyState(VK_RIGHT)){
            movePacmanRight();
        }
        if (GetAsyncKeyState(VK_UP)){
            movePacmanUP();
        }
        if (GetAsyncKeyState(VK_DOWN)){
            movePacmanDown();
        }
        if (GetAsyncKeyState(VK_ESCAPE)){
            gameRunning = false; } }
}
```

Screen Flickering

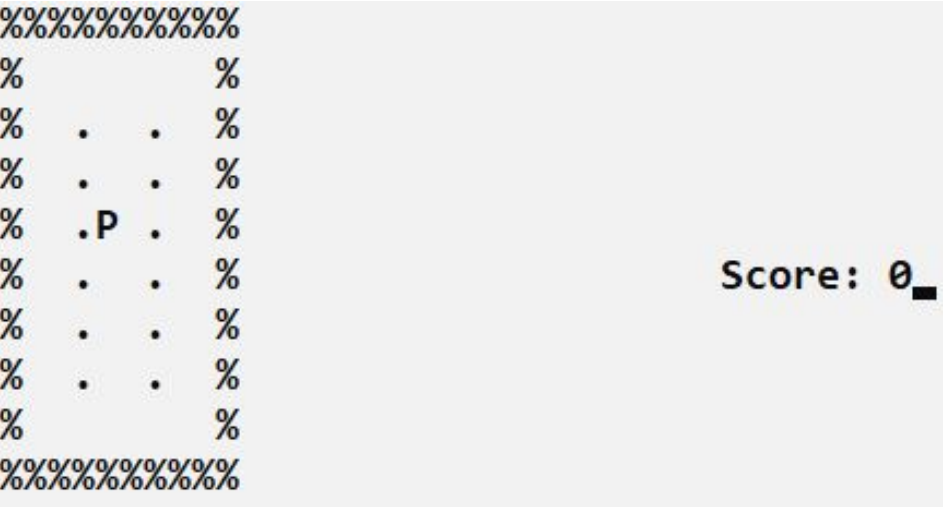
It takes some considerable time to print 10 rows and 10 columns again and again.




```
int score = 0;
main(){
    bool gameRunning = true;
    while (gameRunning){
        Sleep(100);
        system("CLS");
        printMaze();
        printScore();
        if (GetAsyncKeyState(VK_LEFT)){
            movePacmanLeft();
        }
        if (GetAsyncKeyState(VK_RIGHT)){
            movePacmanRight();
        }
        if (GetAsyncKeyState(VK_UP)){
            movePacmanUP();
        }
        if (GetAsyncKeyState(VK_DOWN)){
            movePacmanDown();
        }
        if (GetAsyncKeyState(VK_ESCAPE)){
            gameRunning = false; } }
}
```

Screen Flickering

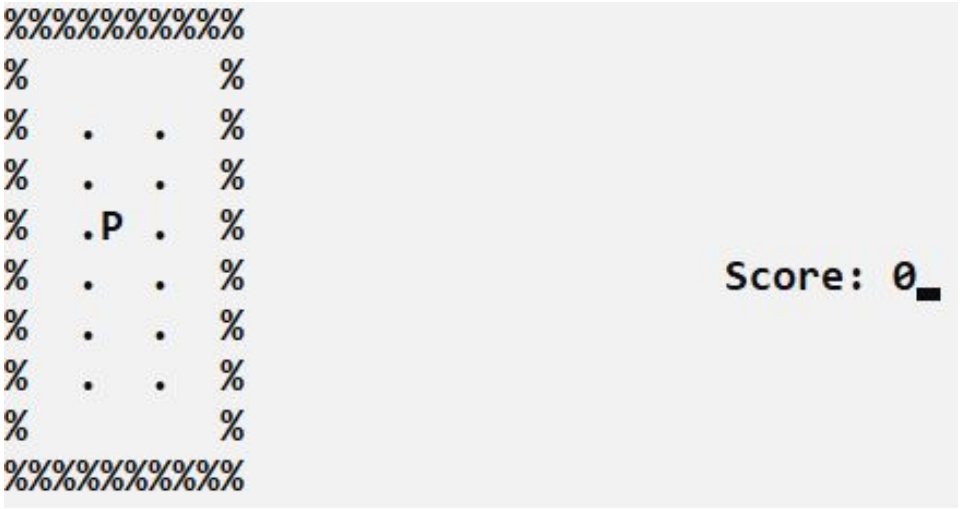
Therefore, the screen flickers.




```
int score = 0;
main(){
    bool gameRunning = true;
    while (gameRunning){
        Sleep(100);
        system("CLS");
        printMaze();
        printScore();
        if (GetAsyncKeyState(VK_LEFT)){
            movePacmanLeft();
        }
        if (GetAsyncKeyState(VK_RIGHT)){
            movePacmanRight();
        }
        if (GetAsyncKeyState(VK_UP)){
            movePacmanUP();
        }
        if (GetAsyncKeyState(VK_DOWN)){
            movePacmanDown();
        }
        if (GetAsyncKeyState(VK_ESCAPE)){
            gameRunning = false; } }
}
```

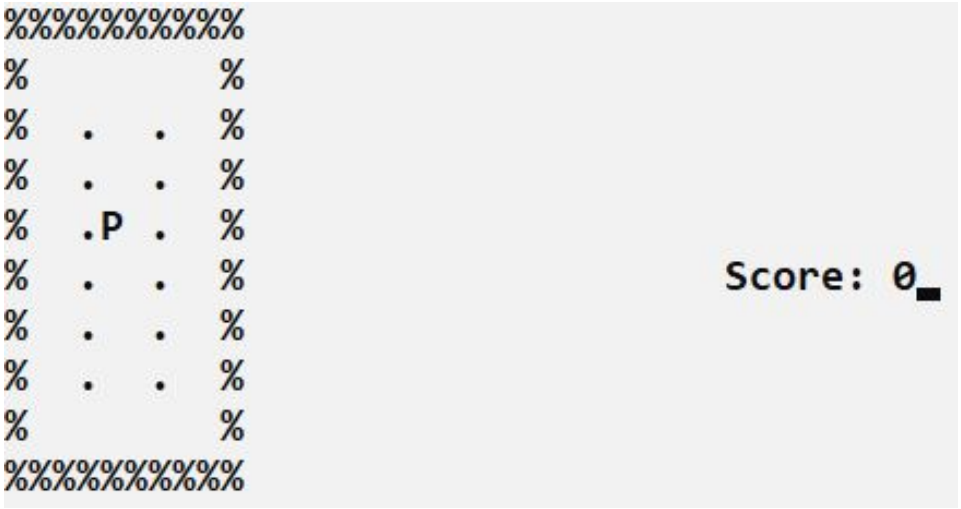
Pac-Man: Screen Flickering

Can we stop the screen flickering? To make a better User Experience?



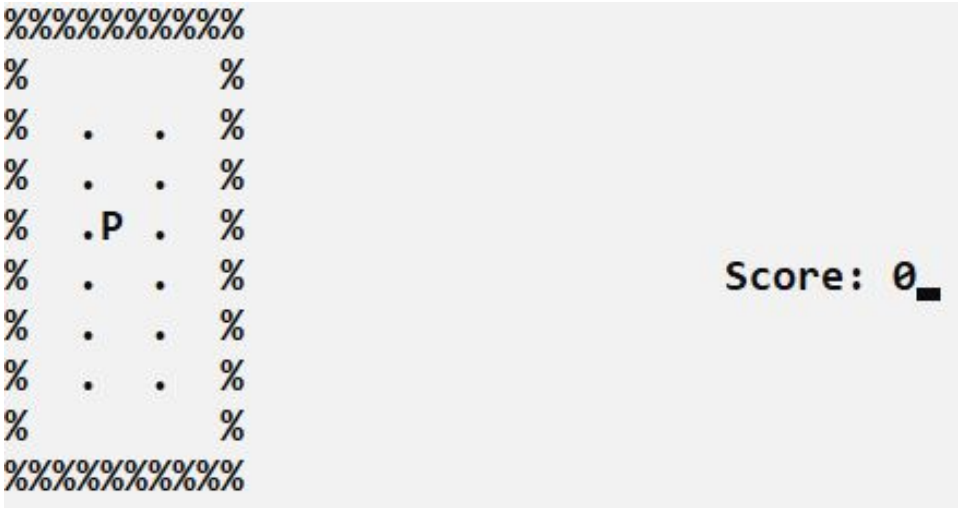
Solution: Screen Flickering

We can print the maze just once without the Pac-Man. And we only print the updated Pac-Man using gotoxy() function again and again.



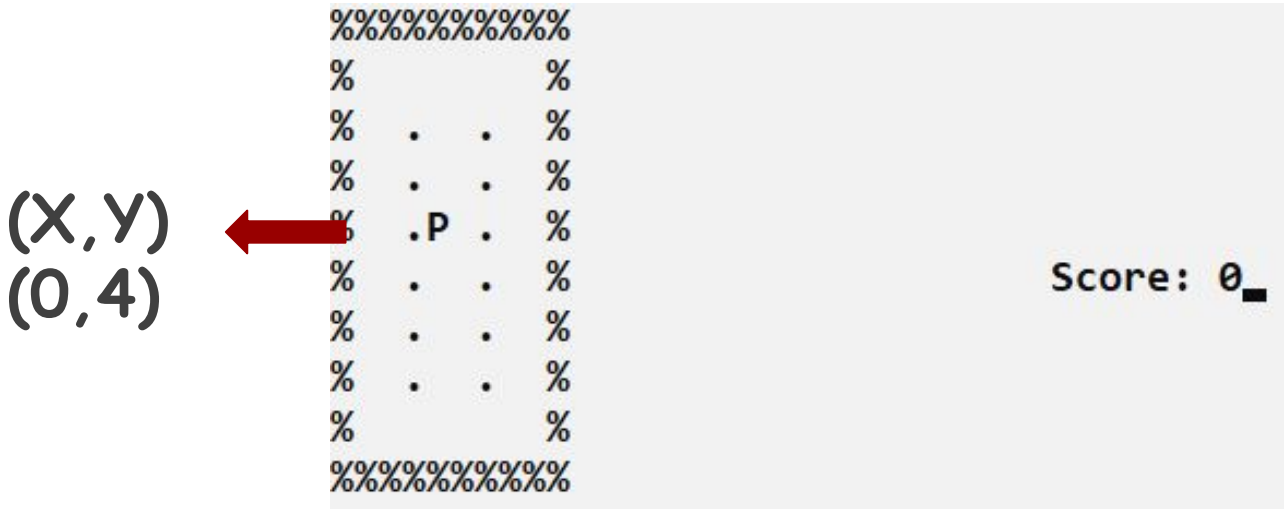
Solution: Screen Flickering

Before moving towards the solution, lets compare the x and y coordinates of the console and rows and columns of the 2D array.



Solution: Screen Flickering

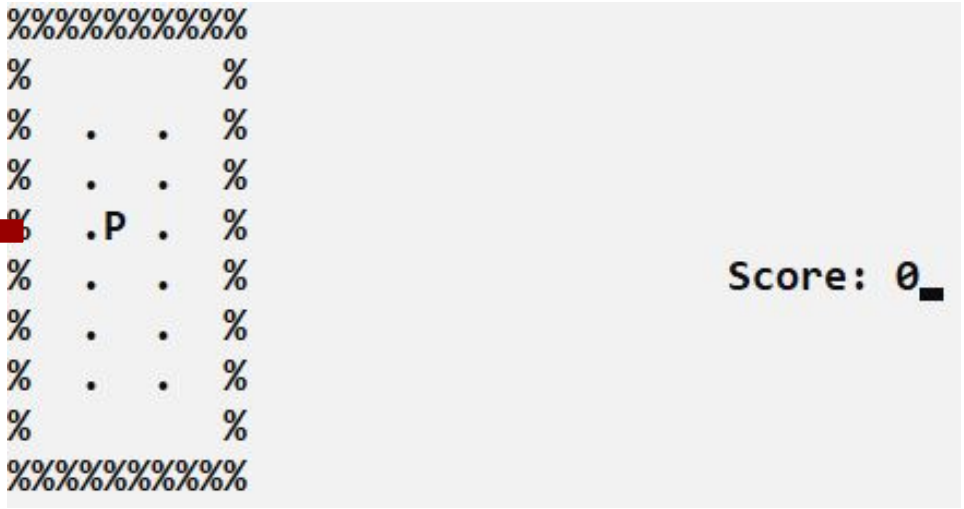
X and Y coordinates on console VS rows and columns of 2D array.



Solution: Screen Flickering

X and Y coordinates on console VS rows and columns of 2D array.

(X,Y)
(0,4)

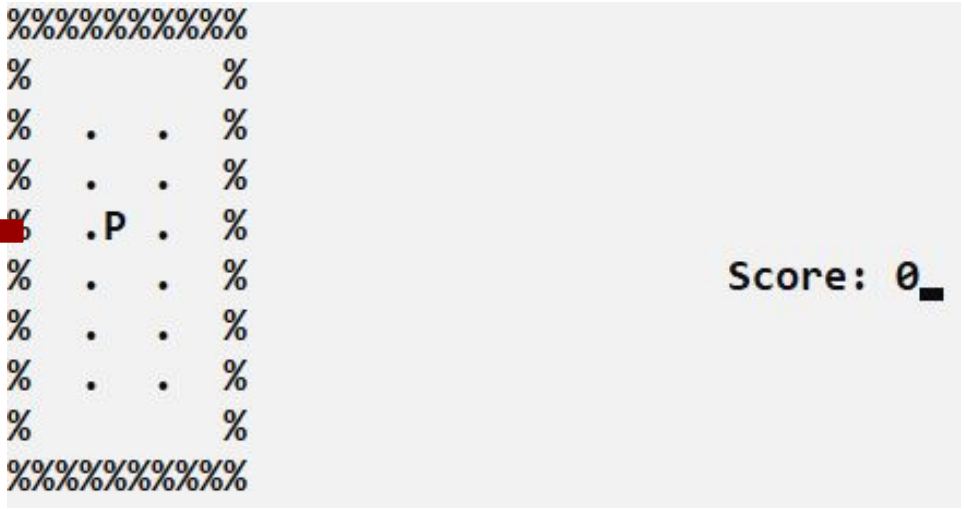


row 4,
column 0

Solution: Screen Flickering

It means if we want to translate the array position on the console then row number should be given as y coordinate and column number should be given as x coordinate.

(X,Y)
(0,4)



row 4,
column 0

```cpp
#include <iostream>
#include <windows.h>
using namespace std;
            // Function Prototype
void printMaze();
void movePacmanLeft();
void movePacmanRight();
void movePacmanUP();
void movePacmanDown();
            // Global Parameters
char maze[10][10] = {
    {'%', '%', '%', '%', '%', '%', '%', '%', '%', '%'},
    {'%', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '%'},
    {'%', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '%'},
    {'%', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '%'},
    {'%', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '%'},
    {'%', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '%'},
    {'%', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '%'},
    {'%', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '%'},
    {'%', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '%'},
    {'%', '%', '%', '%', '%', '%', '%', '%', '%', '%'}
};

int pacmanX = 4; // X Coordinate of Pacman
int pacmanY = 4; // Y Coordinate of Pacman
```

```cpp
#include <iostream>
#include <windows.h>
using namespace std;
            // Function Prototype
void printMaze();
void movePacmanLeft();
void movePacmanRight();
void movePacmanUP();
void movePacmanDown();
            // Global Parameters
char maze[10][10] = {
    {'%', '%', '%', '%', '%', '%', '%', '%', '%', '%'},
    {'%', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '%'},
    {'%', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '%'},
    {'%', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '%'},
    {'%', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '%'},
    {'%', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '%'},
    {'%', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '%'},
    {'%', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '%'},
    {'%', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '%'},
    {'%', '%', '%', '%', '%', '%', '%', '%', '%', '%'}
};

int pacmanX = 4; // X Coordinate of Pacman
int pacmanY = 4; // Y Coordinate of Pacman
```

```cpp
int score = 0;
main(){
    bool gameRunning = true;
    system("CLS");
    printMaze();
    gotoxy(pacmanY, pacmanX);
    cout << "P";
    while (gameRunning){
        Sleep(100);
        printScore();
        if (GetAsyncKeyState(VK_LEFT)){
            movePacmanLeft(); }
        if (GetAsyncKeyState(VK_RIGHT)){
            movePacmanRight(); }
        if (GetAsyncKeyState(VK_UP)){
            movePacmanUP(); }
        if (GetAsyncKeyState(VK_DOWN)){
            movePacmanDown(); }
        if (GetAsyncKeyState(VK_ESCAPE)){
            gameRunning = false; }
    }
}
```

Pac-Man: movePacmanLeft()

```
void movePacmanLeft()
{
    if (maze[pacmanX][pacmanY - 1] == ' ' || maze[pacmanX][pacmanY - 1] == '.')
    {
        maze[pacmanX][pacmanY] = ' ';
        gotoxy(pacmanY, pacmanX);
        cout << " ";
        pacmanY = pacmanY - 1;
        gotoxy(pacmanY, pacmanX);
        cout << "P";
    }
}
```

Pac-Man: movePacmanRight()

```
void movePacmanRight()
{
    if (maze[pacmanX][pacmanY + 1] == ' ' || maze[pacmanX][pacmanY + 1] == '.')
    {
        maze[pacmanX][pacmanY] = ' ';
        gotoxy(pacmanY, pacmanX);
        cout << " ";
        pacmanY = pacmanY + 1;
        gotoxy(pacmanY, pacmanX);
        cout << "P";
    }
}
```

Pac-Man: movePacmanUp()

```
void movePacmanUp()
{
    if (maze[pacmanX - 1][pacmanY] == ' ' || maze[pacmanX - 1][pacmanY] == '.')
    {
        maze[pacmanX][pacmanY] = ' ';
        gotoxy(pacmanY, pacmanX);
        cout << " ";
        pacmanX = pacmanX - 1;
        gotoxy(pacmanY, pacmanX);
        cout << "P";
    }
}
```

Pac-Man: movePacmanDown()

```
void movePacmanDown()
{
    if (maze[pacmanX + 1][pacmanY] == ' ' || maze[pacmanX + 1][pacmanY] == '.')
    {
        maze[pacmanX][pacmanY] = ' ';
        gotoxy(pacmanY, pacmanX);
        cout << " ";
        pacmanX = pacmanX + 1;
        gotoxy(pacmanY, pacmanX);
        cout << "P";
    }
}
```

Learning Objective

Write a C++ program to display output on the console at a specific location using gotoxy() function.

Self Assessment: (Video Profile Activity)

1. Now your task is to print the maze only once and then show the updated locations of pacman and ghosts using the gotoxy() function.
2. Move Ghost after the Pac-man and do not let the Ghost Stuck at the walls.

Self Assessment: (Project Requirements)

About your Game Project.

Following are the requirements for fighting games

1. Your game should have 1 player
2. Your game should have at least 3 enemies
3. Proper Scoring System
4. Proper 3 Life system
5. Movement implementation with gotoxy function
6. 2D array must be used
7. Health decreasing system (i.e. player health will decrease after getting hit by the fire and after 3 or 4 hits he will die)

Self Assessment: (Video Profile Activity)

Here are some videos for Project Ideas.

1. https://www.youtube.com/watch?v=ucRYLobga0g
2. https://youtu.be/rfJLLdKcbww