



Programming Fundamentals

Week 13



Learning Outcomes:

After this lesson, students will be able to:

1. Make the practice of programming more entertaining by promoting the fun and spirit of challenge.
2. Putting the learner into a game atmosphere with clear and significant educational goals.
3. Effectiveness principles that verify to what degree serious games created is a good way of learning

Instructions

- Use proper indentation to make your programs readable.
- Use descriptive variables in your programs (Name of the variables should show their purposes)

Pacman

Pac-Man is a maze-based 2D game

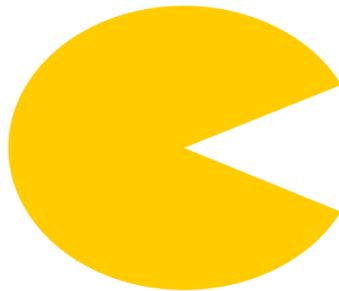
There are a total 5 characters in the Pac-Man Game.

1 Pac-Man.

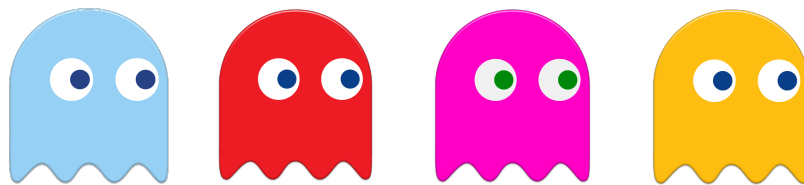
4 Ghosts.



A yellow, circular character named Pac-Man is controlled by the player with the help of arrow keys.



Each of the four ghosts is controlled by the computer.



Small white dots are called “Food Pellets”, whereas the large flashing white dots are called “Power Pellets” or “Energizers”.

Blue outline represents the walls of the maze.

Rules & Interactions

Pac-Man can eat food pellets that have been put across the maze.

Pac-Man loses a life if he collides with any of the ghosts.

If Pac-man eats Power Pallets then the ghosts will turn blue and then Pac-Man can touch the ghosts as well.

One (1) Score increases when the Pac-Man eats food pallets.

Goal:

The goal of the game is to eat all of the food pallets that have been put across the maze while avoiding the Ghosts.

Note:

This is a GUI-based Game. We will develop console based game for now.

The Logic behind both GUI and CLI game is the same which is the most important.

```
#####
| | .. ..... | |
| |  %%%%%%%%%% .. %%%%%%%%%% | |  %%%
| |  |%| |%| |%|... |%| |%| |%|.. |%|
| |  |%| |%| |%|... |%| |%| |%|.. |%|
| |  %%%%%%%%%% . . |%|... %%%%%%%%%% .. %%%.
| |  |%| . . |%|... ..... |%| .. .
| |  %%%%%%%%%% . . |%|... %%%%%%%%%% |%| .. %%%.
| |  |%| . |%| . P |%| ..... |%| .. |%|.
| |  |%| |%| %%% |%| . |%| . |%| .. |%| |%|.
| |  |%| |%| |%|.. %%%%%%%%%% ..... |%| . |%|.
| |  |%| |%| |%|.. ... |%| %%%%%%%%%% . |%|.
| |  |%| . ... |%| |%| .. |%|.
| |  |%| %%%%%%%%%% ... |%| %%%%%%%%%% |%| .. |%| %%%
| | ..... |%| .....
| | .. |%| |%| |%|.. %%%%%%%%%% ..... |%| |%|..
| | .. |%| |%| |%|.. ... |%| %%%%%%%%%% |%| .. |%|.
| | .. |%| . G ... |%| |%| .. |%|.
| | .. |%| %%%%%%%%%% ... |%| %%%%%%%%%% |%| .. |%| %%%
| | ..... |%| .....
| | .....
#####
```

Note: we will use different Character as

- i. %, # as wall
- ii. . as food
- iii. P as Pac-Man
- iv. G as Ghost

We will perform following steps

Step#1:

Main function of the program, code for arrow key movement, clear screen, sleep

System("CLS") for clear screen

Sleep(200) for some delay to minimize flickering of screen issue

```
#include <iostream>
#include <fstream>
#include <time.h>
#include <windows.h>
using namespace std;

// Global Variable
char maze[24][71];
int pacmanX = 9; // X Coordinate of Pacman
int pacmanY = 31; // Y Coordinate of Pacman
int score = 0;

int ghostX = 19; // X Coordinate of Ghost
int ghostY = 25; // Y Coordinate of Ghost
char previousItem = ' ';

// Main Function
main()
{
    loadMaze();
    bool gameRunning = true;
    while (gameRunning)
    {
        Sleep(100);
        system("CLS");
        printMaze();
        gameRunning = ghostMovement();
        printScore();
        if (GetAsyncKeyState(VK_LEFT))
        {
            movePacmanLeft(); // Function call to move Pacman towards left
        }
        if (GetAsyncKeyState(VK_RIGHT))
        {
            movePacmanRight(); // Function call to move Pacman towards right
        }
    }
}
```

```

    if (GetAsyncKeyState(VK_UP))
    {
        movePacmanUp(); // Function call to move Pacman towards up
    }
    if (GetAsyncKeyState(VK_DOWN))
    {
        movePacmanDown(); // Function call to move Pacman towards down
    }
    if (GetAsyncKeyState(VK_ESCAPE))
    {
        gameRunning = false; // Stop the game
    }
}

```

Step#2;

Initialize 2D array Character Array of 24x71

```

char maze[24][71] = {
{"#####"},
{"| |.. ....."},
{"| |"},
{"| |.. %%%%%%%%%% ... %%%%%%%%%% |%|.. %%%%",
{"| |"},
{"| |.. |%| |%| |%|... |%| |%| |%|.. |%|",
{"| |"},
{"| |.. |%| |%| |%|... |%| |%| |%|.. |%|",
{"| |"},
{"| |.. %%%%%%%%%% . . |%|... %%%%%%%%%% .. %%%%",
{"| |"},
{"| |.. |%| . . |%|... ..... |%| .. ."},
{"| |"},
{"| |.. %%%%%%%%%%. . |%|... %%%%%%%%%% |%| .. %%%%",
{"| |"},
{"| |.. |%|. |%|..... |%| .. |%|."},
{"| |"},
{"| |.. ..... |%|. P |%|.....|%| .. |%|."},
{"| |"},
{"| |..|%| |%|%%%|%|. |%|. |%| .....|%| ..|%| |%|."},
{"| |"},
{"| |..|%| |%| |%|.. %%%%%%%%%% .....|%| .|%|."},
{"| |"},
{"| |..|%| |%| |%|.. ...|%| %%%%%%%%%% . |%|."},
{"| |"},
{"| |..|%| . ...|%| |%| ..|%|."},
{"| |"},
{"| |..|%| %%%%%%%%%% ...|%|%%%%%%%%% |%| ..|%|%%%%%%%%%"},
{"| |"},
{"| |..... |%| ....."},
{"| |"},

```

```

{"||      .....
||"},
{"||..|%|  |%|  |%|..  %%%%%%%%%% .....|%|  |%|  ..|%|.
||"},
{"||..|%|  |%|  |%|..  ...|%|  %%%%%%%%%  |%|  ..|%|.
||"},
{"||..|%|      .      G      ...|%|  |%|  ..|%|.
||"},
{"||..|%|  %%%%%%%%%% .....|%|%%%%%%%%%  |%|  ..|%|%%%%%%%%
||"},
{"||.....  |%|  .....
||"},
{"||      .....
||"},
{"#####"}
;

```

Step#3

Printing Function () to print the maze as given below

```

void printMaze()
{
    for (int row = 0; row < 24; row = row + 1)
    {
        for (int col = 0; col < 71; col = col + 1)
        {
            cout << maze[row][col];
        }
        cout << endl;
    }
}

```

Step#4

Calculate Score

```

void calculateScore()
{
    score = score + 1;
}

```

Step 5:

Print Score

```

void printScore()
{
    cout << endl << "Score: " << score << endl;
}

```

Step#6

Update function of move left, right, up, down of Pac-Man by implementing condition of wall collision, score updation.

```
void movePacmanLeft()
{
    if (maze[pacmanX][pacmanY - 1] == ' ' || maze[pacmanX][pacmanY - 1] ==
    '.')
    {

        maze[pacmanX][pacmanY] = ' ';
        pacmanY = pacmanY - 1;
        if (maze[pacmanX][pacmanY] == '.')
        {
            calculateScore();
        }
        maze[pacmanX][pacmanY] = 'P';
    }
}
```

```
void movePacmanRight()
{
    if (maze[pacmanX][pacmanY + 1] == ' ' || maze[pacmanX][pacmanY + 1] == '.')
    {
        maze[pacmanX][pacmanY] = ' ';
        pacmanY = pacmanY + 1;
        if (maze[pacmanX][pacmanY] == '.')
        {
            calculateScore();
        }
        maze[pacmanX][pacmanY] = 'P';
    }
}
```

```
void movePacmanUp()
{
    if (maze[pacmanX - 1][pacmanY] == ' ' || maze[pacmanX - 1][pacmanY] == '.')
    {
        maze[pacmanX][pacmanY] = ' ';
        pacmanX = pacmanX - 1;
        if (maze[pacmanX][pacmanY] == '.')
        {
            calculateScore();
        }
        maze[pacmanX][pacmanY] = 'P';
    }
}
```

```
void movePacmanDown()
{
    if (maze[pacmanX + 1][pacmanY] == ' ' || maze[pacmanX + 1][pacmanY] == '.')
    {
        maze[pacmanX][pacmanY] = ' ';
        pacmanX = pacmanX + 1;
    }
}
```

```

        if (maze[pacmanX][pacmanY] == '.')
        {
            calculateScore();
        }
        maze[pacmanX][pacmanY] = 'P';
    }
}

```

Step#7

Movement of Ghost randomly

We will look at srand() function for random movement of Ghost at 2D array.

Note: Ghost will move only on blank spaces and the places with food pellets.

We will use rand() function to generate a random movement of the ghost for now.

The rand() function is used in C/C++ to generate random numbers in the range [0, RAND_MAX).

Note: If random numbers are generated with rand() without first calling srand(), your program will create the same sequence of numbers each time it runs.

Code to generate the random number is as follows: you have to include time.h or ctime library first.

```

#include<time.h>

int ghostDirection()
{
    srand(time(0));
    int result = 1 + (rand() % 4);
    return result;
}

```

Now what we will do is that we will assign 1 to move left, 2 to move right, 3 to move up and 4 to move down.

```

bool ghostMovement()
{
    int value = ghostDirection();
    if (value == 1)
    {
        if (maze[ghostX][ghostY - 1] == ' ' || maze[ghostX][ghostY - 1] == '.')
        || maze[ghostX][ghostY - 1] == 'P')
        {

```



```

        maze[ghostX][ghostY] = previousItem;
        ghostY = ghostY - 1;
        previousItem = maze[ghostX][ghostY];
        if (previousItem == 'P')
        {
            return 0;
        }
        maze[ghostX][ghostY] = 'G';
    }
}
if (value == 2)
{
    if (maze[ghostX][ghostY + 1] == ' ' || maze[ghostX][ghostY + 1] == '.')
    || maze[ghostX][ghostY + 1] == 'P')
    {
        maze[ghostX][ghostY] = previousItem;
        ghostY = ghostY + 1;
        previousItem = maze[ghostX][ghostY];
        if (previousItem == 'P')
        {
            return 0;
        }
        maze[ghostX][ghostY] = 'G';
    }
}
if (value == 3)
{
    if (maze[ghostX - 1][ghostY] == ' ' || maze[ghostX - 1][ghostY] == '.')
    || maze[ghostX - 1][ghostY] == 'P')
    {
        maze[ghostX][ghostY] = previousItem;
        ghostX = ghostX - 1;
        previousItem = maze[ghostX][ghostY];
        if (previousItem == 'P')
        {
            return 0;
        }
        maze[ghostX][ghostY] = 'G';
    }
}
if (value == 4)
{
    if (maze[ghostX + 1][ghostY] == ' ' || maze[ghostX + 1][ghostY] == '.')
    || maze[ghostX + 1][ghostY] == 'P')
    {

```

```
        maze[ghostX][ghostY] = previousItem;
        ghostX = ghostX + 1;
        previousItem = maze[ghostX][ghostY];
        if (previousItem == 'P')
        {
            return 0;
        }
        maze[ghostX][ghostY] = 'G';
    }
}
return 1;
}
```

Challenges:

Now, following are your tasks:

1. Load the maze configuration from the File
2. Store the maze configuration in a separate file
3. Move the ghost towards the Pacman and do not let the ghost stuck at the wall.
 - a. **Hint: You have pacman and ghost row and column location. Find the direction to move the ghost by subtracting the (ghostX – pacmanX) and (ghostY - pacmanY)**