```cpp
#include "pch.h"
#include "Player.h"
#include <iostream>
#include <random>
#include <cstdlib>
#include <time.h>
using namespace std;

#include <SFML/Graphics.hpp>
using namespace sf;

int x_coord, y_coord, x_move, y_move, random_int, bomb_x_coord, bomb_y_coord;
int lives = 3;

default_random_engine generator;
uniform_int_distribution<int> distribution(1, 20);


Player::Player() {
}

void Player::randGenerator() {
    random_int = distribution(generator);
}

int** Player::p_spawn(int **g) {
    bool valid = false;
    while (!valid) {
        randGenerator();
        x_coord = random_int;
        randGenerator();
        y_coord = random_int;
        cout << "Player x: " << x_coord << " y: " << y_coord << endl;
        if (g[x_coord][y_coord] == 0)
        {
            g[x_coord][y_coord] = 1;
            valid = true;
        }
    }
    return g;
}

int Player::getX() {
    return x_coord;
}

int Player::getY() {
    return y_coord;
}

int** Player::move(int x, int y, int **g) {
    x_move = x_coord + x;
    y_move = y_coord + y;

    if (g[x_coord + x][y_coord + y] == 0)
    {
        if (g[x_coord][y_coord] == 4) //bomb check
        {
            g[x_move][y_move] = 1;
        }
        else if (g[x_coord][y_coord] == 6) {
            g[x_move][y_move] = 1;
            g[x_coord][y_coord] = 4;
        }
        else {
            g[x_move][y_move] = 1;
            g[x_coord][y_coord] = 0;
        }
        x_coord = x_coord + x;
        y_coord = y_coord + y;
    }
    return g;
}

void Player::minusLife() {
    lives--;
}

int Player::getLives() {
    return lives;
}


Player::~Player()
{
}

```