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

class Variables
{
public:
    bool gameover, obstacles, fired, reload, quit;
    const int width = 15, height = 15;
    int x, y, gamespeed = 20, score, level, p = 0;
    int posX[4], posY[4];
    int bullX[4], bullY[4];
    bool enemy_inscene, enemy1_fired, enemy2_fired, paused;
    int enemyX[2], enemyY[2];
    int e_bullX[4], e_bullY[4];

    void Start()
    {
        gameover = false;
        obstacles = false;
        fired = false;
        reload = false;
        enemy_inscene = false;
        enemy1_fired = false;
        enemy2_fired = false;
        paused = false;
        quit = false;
        gamespeed = 60;
        score = 0;
        level = 1;
        x = width / 2;
        y = height - 2;
    }
    void Input()
    {
        if (_kbhit())
        {
            switch (_getch())
            {
            case 'a': x--;
                break;
            case 'd': x++;
                break;
            case 'w': y--;
                break;
            case 's': y++;
                break;
            case 'z': fired = true;
                break;
            case 'p': if (paused == false)
                    {
                        paused = true;
                    }
                    else
                    {
                      paused = false;
                    }
                    break;
            case 'q': quit = true;
```

```
        }
    }
}
void Update()
{
    system("cls");
    printf("* SCORE : %d *\n", score);
    for (int i = 0; i < height; i++)
    {
        printf("*");
        for (int j = 1; j < width; j++)
        {
            if (j == width - 1)
            {
                cout << "*";
            }
            else if (i == y && j == x)
            {
                cout << "#";
            }

            else if (i == posY[0] && j == posX[0] || i == posY[1] && j == posX[1] ||
                i == posY[2] && j == posX[2] || i == posY[3] && j == posX[3])
            {
                cout << "O";
            }

            else if (j == bullX[0] && i == bullY[0] || j == bullX[1] && i == bullY[1] ||
                j == bullX[2] && i == bullY[2] || j == bullX[3] && i == bullY[3])
            {
                cout << "|";
            }

            else if (j == enemyX[0] && i == enemyY[0] || j == enemyX[1] && i == enemyY[1])
            {
                cout << "X";
            }

            else if (j == e_bullX[0] && i == e_bullY[0] || j == e_bullX[1] && i == e_bullY[1] ||
                j == e_bullX[2] && i == e_bullY[2] || j == e_bullX[3] && i == e_bullY[3])
            {
                cout << "*";
            }

            else
            {
                cout << " ";
            }

            if (posX[0] == bullX[0] && posY[0] == bullY[0])
            {
                posY[0] = 14;
                bullY[0] = 0;
            }

            else if (posX[1] == bullX[0] && posY[1] == bullY[0])
            {
                posY[1] = 14;
                bullY[0] = 0;
            }
```

```
else if (posX[2] == bullX[0] && posY[2] == bullY[0])
{
    posY[2] = 14;
    bullY[0] = 0;
}

else if (posX[3] == bullX[0] && posY[3] == bullY[0])
{
    posY[3] = 14;
    bullY[0] = 0;
}

if (posX[0] == bullX[1] && posY[0] == bullY[1])
{
    posY[0] = 14;
    bullY[1] = 0;
}

else if (posX[1] == bullX[1] && posY[1] == bullY[1])
{
    posY[1] = 14;
    bullY[1] = 0;
}

else if (posX[2] == bullX[1] && posY[2] == bullY[1])
{
    posY[2] = 14;
    bullY[1] = 0;
}

else if (posX[3] == bullX[1] && posY[3] == bullY[1])
{
    posY[3] = 14;
    bullY[1] = 0;
}

if (posX[0] == bullX[2] && posY[0] == bullY[2])
{
    posY[0] = 14;
    bullY[2] = 0;
}

else if (posX[1] == bullX[2] && posY[1] == bullY[2])
{
    posY[1] = 14;
    bullY[2] = 0;
}

else if (posX[2] == bullX[2] && posY[2] == bullY[2])
{
    posY[2] = 14;
    bullY[2] = 0;

}

else if (posX[3] == bullX[2] && posY[3] == bullY[2])
{
    posY[3] = 14;
    bullY[2] = 0;
```

```
}

if (posX[0] == bullX[3] && posY[0] == bullY[3])
{
    posY[0] = 14;
    bullY[3] = 0;

}

else if (posX[1] == bullX[3] && posY[1] == bullY[3])
{
    posY[1] = 14;
    bullY[3] = 0;

}

else if (posX[2] == bullX[3] && posY[2] == bullY[3])
{
    posY[2] = 14;
    bullY[3] = 0;

}

else if (posX[3] == bullX[3] && posY[3] == bullY[3])
{
    posY[3] = 14;
    bullY[3] = -10;

}

if (e_bullX[0] == bullX[0] && e_bullY[0] == bullY[0])
{
    e_bullY[0] = 20;
    bullY[0] = -10;
}

else if (e_bullX[0] == bullX[1] && e_bullY[0] == bullY[1])
{
    e_bullY[0] = 20;
    bullY[1] = -10;
}

else if (e_bullX[0] == bullX[2] && e_bullY[0] == bullY[2])
{
    e_bullY[0] = 20;
    bullY[2] = -10;
}

else if (e_bullX[0] == bullX[3] && e_bullY[0] == bullY[3])
{
    e_bullY[0] = 20;
    bullY[3] = -10;
}

if (level == 1 && score >= 1000)
{
    level = 2;
}
```

```
            if (e_bullX[0] == x && e_bullY[0] == y || e_bullX[1] == x && e_bullY[1] == y)
            {
                gameover = true;
            }

            if (posX[0] == x && posY[0] == y || posX[1] == x && posY[1] == y ||
                posX[2] == x && posY[2] == y || posX[3] == x && posY[3] == y)
            {
                gameover = true;
            }
        }
        cout << endl;;
    }
  }
};

class Player: public Variables
{
public:

  void Fire()
  {
    if (fired && p < 4)
    {
      bullX[p] = x;
      bullY[p] = y - 1;
      p++;
      fired = false;

      for (int i = 0; i < p; i++)
      {
        bullY[i]--;
      }
    }
    else
    {
      for (int i = 0; i < p; i++)
      {
        bullY[i]--;
      }
    }

    if (p >= 4 && bullY[3] <= -1)
    {
      reload = true;
    }

    if (reload)
    {
      p = 0;
      reload = false;
    }
  }
};

class Enemies : public Player
{
public:

  void Enemy_fire()
```

```
{
    int n = rand() % 2;


    if (!enemy1_fired && enemyY[0] < 7)
    {
        for (int i = 0; i < n; i++)
        {
            e_bullX[i] = enemyX[i];
            e_bullY[i] = enemyY[i];

        }
        enemy1_fired = true;
    }

    if (enemy1_fired)
    {
        for (int i = 0; i < n; i++)
        {
            e_bullY[i]++;
            e_bullX[i] = x;
        }
    }

    if (e_bullY[n - 1] > height)
    {
        enemy1_fired = false;
    }

}
void Enemy()
{
    int range = rand() % 5;

    if (range == 1 && !enemy_inscene)
    {
        enemyX[0] = x;
        enemyY[0] = 1;

        enemyX[1] = rand() % width;
        enemyY[1] = 2;

        enemy_inscene = true;
    }

    if (enemy_inscene)
    {
        enemyY[0]++;
        enemyY[1]++;

        Enemy_fire();

    }

    if (enemyY[0] > height)
    {
        enemy_inscene = false;
    }
}
void Obstacles()
```

```
    {
        int count = 0;

        if (obstacles == false)
        {
            if (!enemy_inscene)
            {
                for (int i = 0; i < 4; i++)
                {
                    posX[i] = rand() % width;
                    posY[i] = i;
                }
            }

            obstacles = true;
        }

        for (int i = 0; i < 4; i++)
        {
            posY[i]++;
        }

        for (int i = 0; i < 4; i++)
        {
            if (posY[i] >= height - 2)
            {
                count++;
            }
        }

        if (count >= 4)
        {
            obstacles = false;
        }

        if (!gameover)
        {
            score += 10;
        }

    }
};

int main()
{
    Enemies e;
    e.Start();

    char choice;
    cout<<"ENTER 'c' TO START GAME ";
    cin >> choice;
    system("cls");

    if (choice == 'c')
    {
        while (!e.gameover)
        {
            if (!e.paused)
            {
                e.Input();
```

```
            e.Obstacles();
            e.Fire();

            if (e.level == 2)
                e.Enemy();

            e.Update();

            Sleep(e.gamespeed);
        }

        else
        {
            e.Input();
            e.Update();
        }

        if (e.quit)
        {
            break;
        }
    }
}
else
{
    cout<<"WRONG CHOICE !!!";
}

if (e.quit)
    cout << "YOU QUIT THE GAME";
else
    cout << "GAME OVER !!!";



return 0;
}
```