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

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

default_random_engine e_generator;
uniform_int_distribution<int> place_distribution(1, 19);

bool hitplayer = false;

Enemy::Enemy() {
}

int Enemy::genPlacement(int p) {
    p = place_distribution(e_generator);
    return p;
}

void Enemy::setHitPlayer(bool x) {
    hitplayer = x;
}

bool Enemy::getHit() {
    return hitplayer;
}

int Enemy::getX_pos() {
    return x_pos;
}

int Enemy::getY_pos() {
    return y_pos;
}

int** Enemy::spawn(int **g) {
    active = true;
    int placement = 0;
    bool valid = false;
    while (!valid) {
        placement = genPlacement(placement);
        x_pos = placement;
        placement = genPlacement(placement);
        y_pos = placement;
```

```
54          if (g[x_pos][y_pos] == 0)
55          {
56              g[x_pos][y_pos] = 2;
57              valid = true;
58              cout << "Enemy x: " << x_pos << " y: " << y_pos << endl;
59          }
60      }
61      return g;
62  }
63
64  int** Enemy::move(int** g) {
65      uniform_int_distribution<int> dir_distribution(1, 4);
66      //1 = up 2 = right 3 = down 4 = left
67      if (g[x_pos - 1][y_pos] == 1
68          or g[x_pos][y_pos + 1] == 1
69          or g[x_pos + 1][y_pos] == 1
70          or g[x_pos][y_pos - 1] == 1)
71      {
72          setHitPlayer(true);
73      }
74      else if (g[x_pos - 1][y_pos] != 0
75          and g[x_pos][y_pos + 1] != 0
76          and g[x_pos + 1][y_pos] != 0
77          and g[x_pos][y_pos - 1] != 0) {
78          //wait to move
79      }
80      else {
81          int direction = 0;
82          bool valid = false;
83          while (!valid) {
84              direction = dir_distribution(e_generator);
85              switch (direction)
86              {
87              case 1:
88                  //up
89                  if (g[x_pos - 1][y_pos] == 0)
90                  {
91                      g[x_pos - 1][y_pos] = 2;
92                      g[x_pos][y_pos] = 0;
93                      x_pos = x_pos - 1;
94                      valid = true;
95                  }
96                  else if (g[x_pos - 1][y_pos] == 1) {
97                      cout << "Hit player!" << endl;
98                      setHitPlayer(true);
99                  }
100                 break;
101             case 2:
102                 //right
103                 if (g[x_pos][y_pos + 1] == 0)
104                 {
105                     g[x_pos][y_pos + 1] = 2;
106                     g[x_pos][y_pos] = 0;
```

```
107                     y_pos = y_pos + 1;
108                     valid = true;
109                 }
110                 else if (g[x_pos][y_pos + 1] == 1) {
111                     cout << "Hit player!" << endl;
112                     setHitPlayer(true);
113                 }
114                 break;
115             case 3:
116                 //down
117                 if (g[x_pos + 1][y_pos] == 0)
118                 {
119                     g[x_pos + 1][y_pos] = 2;
120                     g[x_pos][y_pos] = 0;
121                     x_pos = x_pos + 1;
122                     valid = true;
123                 }
124                 else if (g[x_pos + 1][y_pos] == 1) {
125                     cout << "Hit player!" << endl;
126                     setHitPlayer(true);
127                 }
128                 break;
129             case 4:
130                 //left
131                 if (g[x_pos][y_pos - 1] == 0)
132                 {
133                     g[x_pos][y_pos - 1] = 2;
134                     g[x_pos][y_pos] = 0;
135                     y_pos = y_pos - 1;
136                     valid = true;
137                 }
138                 else if (g[x_pos][y_pos - 1] == 1) {
139                     cout << "Hit player!" << endl;
140                     setHitPlayer(true);
141                 }
142                 break;
143             }
144         }
145     }
146     return g;
147 }
148
149 void Enemy::setActive(bool h) {
150     if (h)
151     {
152         active = false;
153     }
154 }
155
156 bool Enemy::getActive() {
157     return active;
158 }
159
```

```
160
161 Enemy::~Enemy()
162 {
163 }
164
```