```cpp
#include "pch.h"
#include "Bomb.h"
#include <iostream>
using namespace std;

int bomb_x, bomb_y;
int score = 0;
int activeenemies = 4;
bool playerhit = false;

int** enemyHitArray = new int*[4];


Bomb::Bomb()
{
    for (int row = 0; row < 4; row++)
    {
        enemyHitArray[row] = new int[2];
    }
}

void setBomb_Coord(int x, int y) {
    bomb_x = x;
    bomb_y = y;
}


int** Bomb::dropBomb(int** g, int x, int y) {
    setBomb_Coord(x, y);
    g[bomb_x][bomb_y] = 6;
    return g;
}

void enemyHit(int x, int y, int row) {
    enemyHitArray[row][0] = x;
    enemyHitArray[row][1] = y;
}

int** Bomb::explode(int** g) {
    int a = 0;
    if (g[bomb_x][bomb_y] == 6)
    {
        playerhit = true;
    }

    g[bomb_x][bomb_y] = 5;

    //explosion up
    for (int j = 1; j < 4; j++)
    {
        if (g[bomb_x - j][bomb_y] == 0) {
            g[bomb_x - j][bomb_y] = 5;
        }
```

```
54
55          if (g[bomb_x - j][bomb_y] == 2) {
56              enemyHit(bomb_x - j, bomb_y, a);
57              a++;
58              score++;
59              g[bomb_x - j][bomb_y] = 5;
60          }
61
62          if (g[bomb_x - j][bomb_y] == 1)
63          {
64              playerhit = true;
65              g[bomb_x - j][bomb_y] = 5;
66          }
67
68          if(g[bomb_x - j][bomb_y] == 3){
69              j = 4;
70          }
71      }
72
73      //explosion right
74      for (int j = 1; j < 4; j++)
75      {
76          if (g[bomb_x][bomb_y + j] == 0) {
77              g[bomb_x][bomb_y + j] = 5;
78          }
79
80          if (g[bomb_x][bomb_y + j] == 2) {
81              enemyHit(bomb_x, bomb_y + j, a);
82              a++;
83              score++;
84              g[bomb_x][bomb_y + j] = 5;
85          }
86
87          if (g[bomb_x][bomb_y + j] == 1)
88          {
89              playerhit = true;
90              g[bomb_x][bomb_y + j] = 5;
91          }
92
93          if (g[bomb_x][bomb_y + j] == 3)
94          {
95              j = 4;
96          }
97      }
98
99      //explosion down
100     for (int j = 1; j < 4; j++)
101     {
102         if (g[bomb_x + j][bomb_y] == 0) {
103             g[bomb_x + j][bomb_y] = 5;
104         }
105
106         if (g[bomb_x + j][bomb_y] == 2) {
```

```
107             enemyHit(bomb_x + j, bomb_y, a);
108             a++;
109             score++;
110             g[bomb_x + j][bomb_y] = 5;
111         }
112
113         if (g[bomb_x + j][bomb_y] == 1)
114         {
115             playerhit = true;
116             g[bomb_x + j][bomb_y] = 5;
117         }
118
119         if(g[bomb_x + j][bomb_y] == 3) {
120             j = 4;
121         }
122     }
123
124     //explosion left
125     for (int j = 1; j < 4; j++)
126     {
127         if (g[bomb_x][bomb_y - j] == 0) {
128             g[bomb_x][bomb_y - j] = 5;
129         }
130
131         if (g[bomb_x][bomb_y - j] == 2) {
132             enemyHit(bomb_x, bomb_y - j, a);
133             a++;
134             score++;
135             g[bomb_x][bomb_y - j] = 5;
136         }
137
138         if (g[bomb_x][bomb_y - j] == 1)
139         {
140             playerhit = true;
141             g[bomb_x][bomb_y - j] = 5;
142         }
143
144         if(g[bomb_x][bomb_y - j] == 3) {
145             j = 4;
146         }
147     }
148
149     //explosion up-right
150     for (int j = 1; j < 4; j++)
151     {
152         if (g[bomb_x - j][bomb_y + j] == 0) {
153             g[bomb_x - j][bomb_y + j] = 5;
154         }
155
156         if (g[bomb_x - j][bomb_y + j] == 2) {
157             cout << "Enemy Hit!" << endl;
158             enemyHit(bomb_x - j, bomb_y + j, a);
159             a++;
```

```
160             score++;
161             g[bomb_x - j][bomb_y + j] = 5;
162         }
163
164         if (g[bomb_x - j][bomb_y + j] == 1)
165         {
166             playerhit = true;
167             g[bomb_x - j][bomb_y + j] = 5;
168         }
169
170         if(g[bomb_x - j][bomb_y + j] == 3){
171             j = 4;
172         }
173     }
174
175     //explosion down-right
176     for (int j = 1; j < 4; j++)
177     {
178         if (g[bomb_x + j][bomb_y + j] == 0) {
179             g[bomb_x + j][bomb_y + j] = 5;
180         }
181
182         if (g[bomb_x + j][bomb_y + j] == 2) {
183             enemyHit(bomb_x + j, bomb_y + j, a);
184             a++;
185             score++;
186             g[bomb_x + j][bomb_y + j] = 5;
187         }
188
189         if (g[bomb_x + j][bomb_y + j] == 1)
190         {
191             playerhit = true;
192             g[bomb_x + j][bomb_y + j] = 5;
193         }
194
195         if(g[bomb_x + j][bomb_y + j] == 3) {
196             j = 4;
197         }
198     }
199
200     //explosion down-left
201     for (int j = 1; j < 4; j++)
202     {
203         if (g[bomb_x + j][bomb_y - j] == 0) {
204             g[bomb_x + j][bomb_y - j] = 5;
205         }
206
207         if (g[bomb_x + j][bomb_y - j] == 2) {
208             enemyHit(bomb_x + j, bomb_y - j, a);
209             a++;
210             score++;
211             g[bomb_x + j][bomb_y - j] = 5;
212         }
```

```
213
214         if (g[bomb_x + j][bomb_y - j] == 1)
215         {
216             playerhit = true;
217             g[bomb_x + j][bomb_y - j] = 5;
218         }
219
220         if(g[bomb_x + j][bomb_y - j] == 3){
221             j = 4;
222         }
223     }
224
225     //explosion up-left
226     for (int j = 1; j < 4; j++)
227     {
228         if (g[bomb_x - j][bomb_y - j] == 0) {
229             g[bomb_x - j][bomb_y - j] = 5;
230         }
231
232         if (g[bomb_x - j][bomb_y - j] == 2) {
233             enemyHit(bomb_x - j, bomb_y - j, a);
234             a++;
235             score++;
236             g[bomb_x - j][bomb_y - j] = 5;
237         }
238
239         if (g[bomb_x - j][bomb_y - j] == 1)
240         {
241             playerhit = true;
242             g[bomb_x - j][bomb_y - j] = 5;
243         }
244
245         if(g[bomb_x - j][bomb_y - j] == 3){
246             j = 4;
247         }
248     }
249
250     return g;
251 }
252
253 void Bomb::setHitPlayer(bool x) {
254     playerhit = x;
255 }
256
257 bool Bomb::getHitPlayer() {
258     return playerhit;
259 }
260
261 bool Bomb::checkHit(int x, int y) {
262     for (int i = 0; i < 4; i++)
263     {
264         if (enemyHitArray[i][0] == x and enemyHitArray[i][1] == y)
265         {
```

```
266             activeenemies--;
267             return true;
268         }
269     }
270     return false;
271 }
272
273 void Bomb::resetHitArray() {
274     for (int i = 0; i < 4; i++)
275     {
276         enemyHitArray[i][0] = NULL;
277         enemyHitArray[i][1] = NULL;
278     }
279 }
280
281 int Bomb::getScore() {
282     return score;
283 }
284
285 void Bomb::setScore(int x) {
286     score = x;
287 }
288
289 int Bomb::getActiveEnemies() {
290     return activeenemies;
291 }
292
293 void Bomb::setActiveEnemies(int count) {
294     activeenemies = count;
295 }
296
297 Bomb::~Bomb()
298 {
299 }
300
```