

Clear IT



Session: 2021 – 2025

Submitted by:

Khadim Hussain 2021-CS-204

Supervised by:

Ma'am Maida Shahid

Department of Computer Science
University of Engineering and Technology
Lahore, Pakistan

CONTENTS

Project Description:	3
Users of Application:	3
Screenshots	4
Data Structures:	5
Functions Prototypes:	5
Complete Code:	6

Description:

I developed a game named as Clear IT. In this game mission of player is to kill the attackers by moving his character left or right. Lives of player and Score has been mentioned. When lives of player ends then displays game over and when player kills all the attackers then displays level completed. There are four levels in my game.

Game Character:

My game has following characters:

- One player
- Simple Attackers
- Fast Moving Attackers
- Boss Enemy

Rules & Regulations:

1. When enemy reaches boundary one life decreases.
2. Kill Enemies before they reaches boundary.
3. Kill all enemies to play next level.

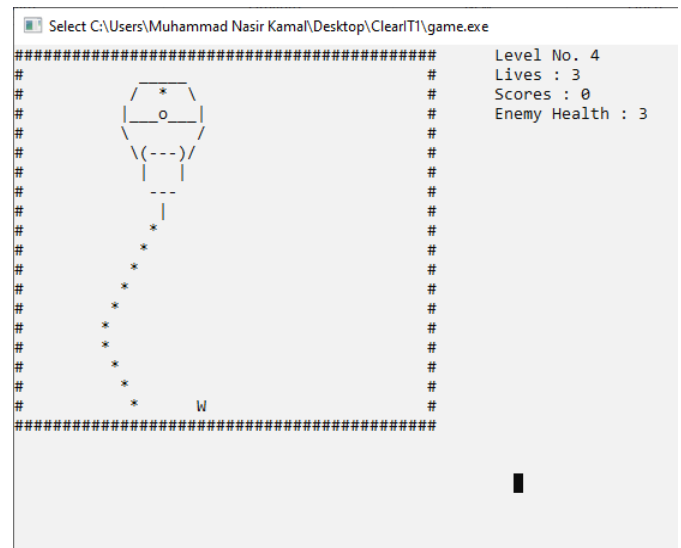
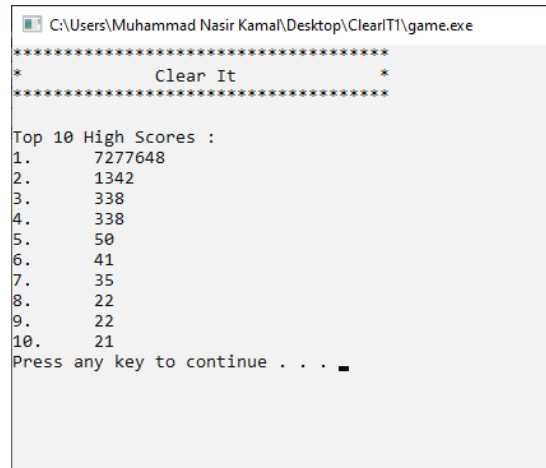
Keys to Move:

1. “Left Key” for left movement.
2. “Right Key” for right movement.
3. “Space” for fire.

Goal of the game:

The goal of my game is to kill all attackers and increase the score.

C:\Users\Muhammad Nasir Kamal\Desktop\CleartIt\game.exe



DATA STRUCTURES:

```
int bossX = 18;
int bossY = 20;
int enemyX = 1;
int enemyY = 19;
char level1[20][45];
char level2[20][45];
char level3[20][45];
char level4[20][45];
int score = 0;
int level = 1;
int highScore[100];
int scoreCount = 0;
bool loading = true, over = false;
int count = 0, count1 = 0, count2 = -1;
int lives = 3;
int lines = 0;
int x, y;
char ch;
int enemyMove = 13;
int enemyLife = 10;
```

FUNCTION PROTOTYPES:

```
void showLevel1();
void showLevel2();
void showLevel3();
void showLevel4();
void loadLevel1();
void loadLevel2();
void loadLevel3();
void loadLevel4();
void movebulletup1();
void movebulletup2();
void movebulletup3();
void movebulletup4();
void bullet1();
void bullet2();
void bullet3();
void bullet4();
void moveLeft1();
void moveLeft2();
void moveLeft3();
void moveLeft4();
void moveRight1();
void moveRight2();
void moveRight3();
void moveRight4();
void gotoxy(int x, int y);
char menu();
void playGame();
void chooseLevel();
void playLevel1();
void playLevel2();
void playLevel3();
void playLevel4();
```

Clear IT

```
void printHighScores();
void addHighScore(int hs);
void loadScores();
void storeScores();
void header();
void calculateScore();
void printScore();
bool completedLevel1();
bool completedLevel2();
bool completedLevel3();
bool completedLevel4();
void load();
void levelCompleted();
void moveLineDown();
void moveLineDown1();
void fallbomb();
void randomBomb();
void gameOver();
void printEnemy();
void moveEnemy();
void moveEnemyLeft();
void moveEnemyRight();
void creatEnemy();
```

Complete CODE:

```
#include <conio.h>
#include <fstream>
#include <iostream>
#include <time.h>
#include <windows.h>

using namespace std;
//.....Functions
// Prototypes.....
void showLevel1();
void showLevel2();
void showLevel3();
void showLevel4();
void loadLevel1();
void loadLevel2();
void loadLevel3();
void loadLevel4();
void movebulletup1();
void movebulletup2();
void movebulletup3();
void movebulletup4();
void bullet1();
void bullet2();
void bullet3();
void bullet4();
void moveLeft1();
void moveLeft2();
void moveLeft3();
void moveLeft4();
void moveRight1();
void moveRight2();
```

Clear IT

```
void moveRight3();
void moveRight4();
void gotoxy(int x, int y);
char menu();
void playGame();
void chooseLevel();
void playLevel1();
void playLevel2();
void playLevel3();
void playLevel4();
void printHighScores();
void addHighScore(int hs);
void loadScores();
void storeScores();
void header();
void calculateScore();
void printScore();
bool completedLevel1();
bool completedLevel2();
bool completedLevel3();
bool completedLevel4();
void load();
void levelCompleted();
void moveLineDown();
void moveLineDown1();
void fallbomb();
void randomBomb();
void gameOver();
void printEnemy();
void moveEnemy();
void moveEnemyLeft();
void moveEnemyRight();
void creatEnemy();
//.....Variables
// Declaration.....
int bossX = 18;
int bossY = 20;
int enemyX = 1;
int enemyY = 19;
char level1[20][45];
char level2[20][45];
char level3[20][45];
char level4[20][45];
int score = 0;
int level = 1;
int highScore[100];
int scoreCount = 0;
bool loading = true, over = false;
int count = 0, count1 = 0, count2 = -1;
int lives = 3;
int lines = 0;
int x, y;
char ch;
int enemyMove = 13;
int enemyLife = 10;
//.....Main
// Function.....
int main() {
```

Clear IT

```
lives = 3;
enemyLife = 3;
if (loading) {
    loadScores();
    loading = false;
}
load();
playGame();
}
//.....Show
// level1.....
void showLevel1() {
    for (int i = 0; i < 20; i++) {
        for (int j = 0; j < 45; j++) {
            cout << level1[i][j];
        }
        cout << "\n";
    }
}
void showLevel2() {
    for (int i = 0; i < 20; i++) {
        for (int j = 0; j < 45; j++) {
            cout << level2[i][j];
        }
        cout << "\n";
    }
}
void showLevel3() {
    for (int i = 0; i < 20; i++) {
        for (int j = 0; j < 45; j++) {
            cout << level3[i][j];
        }
        cout << "\n";
    }
}
void showLevel4() {
    for (int i = 0; i < 20; i++) {
        for (int j = 0; j < 45; j++) {
            cout << level4[i][j];
        }
        cout << "\n";
    }
}
//.....Move Boss
// Right.....
void moveRight1() {
    if (level1[bossX][bossY + 1] == '' || level1[bossX][bossY + 1] == '*') {

        level1[bossX][bossY] = ' ';
        gotoxy(bossY, bossX);
        cout << " ";
        bossY++;
        level1[bossX][bossY] = 'W';
        gotoxy(bossY, bossX);
        cout << "W";
    }
}
void moveRight2() {
```


Clear IT

```
if (level2[bossX][bossY + 1] == '' || level2[bossX][bossY + 1] == '*') {

    level2[bossX][bossY] = '';
    gotoxy(bossY, bossX);
    cout << " ";
    bossY++;
    level2[bossX][bossY] = 'W';
    gotoxy(bossY, bossX);
    cout << "W";
}
}

void moveRight3() {
    if (level3[bossX][bossY + 1] == '' || level3[bossX][bossY + 1] == '*') {

        level3[bossX][bossY] = '';
        gotoxy(bossY, bossX);
        cout << " ";
        bossY++;
        level3[bossX][bossY] = 'W';
        gotoxy(bossY, bossX);
        cout << "W";
    }
}

void moveRight4() {
    if (level4[bossX][bossY + 1] == '' || level4[bossX][bossY + 1] == '*') {

        level4[bossX][bossY] = '';
        gotoxy(bossY, bossX);
        cout << " ";
        bossY++;
        level4[bossX][bossY] = 'W';
        gotoxy(bossY, bossX);
        cout << "W";
    }
}

//.....Move Boss
// Left.....
void moveLeft1() {
    if (level1[bossX][bossY - 1] == '' || level1[bossX][bossY - 1] == '*') {
        level1[bossX][bossY] = '';
        gotoxy(bossY, bossX);
        cout << " ";
        bossY--;
        level1[bossX][bossY] = 'W';
        gotoxy(bossY, bossX);
        cout << "W";
    }
}

void moveLeft2() {
    if (level2[bossX][bossY - 1] == '' || level2[bossX][bossY - 1] == '*') {
        level2[bossX][bossY] = '';
        gotoxy(bossY, bossX);
        cout << " ";
        bossY--;
        level2[bossX][bossY] = 'W';
        gotoxy(bossY, bossX);
        cout << "W";
    }
}
```

Clear IT

```
}
void moveLeft3() {
    if (level3[bossX][bossY - 1] == '' || level3[bossX][bossY - 1] == '*') {
        level3[bossX][bossY] = ' ';
        gotoxy(bossY, bossX);
        cout << " ";
        bossY--;
        level3[bossX][bossY] = 'W';
        gotoxy(bossY, bossX);
        cout << "W";
    }
}
void moveLeft4() {
    if (level4[bossX][bossY - 1] == '' || level4[bossX][bossY - 1] == '*') {
        level4[bossX][bossY] = ' ';
        gotoxy(bossY, bossX);
        cout << " ";
        bossY--;
        level4[bossX][bossY] = 'W';
        gotoxy(bossY, bossX);
        cout << "W";
    }
}
//.....Bullet.....
void bullet1() {

    level1[bossX - 1][bossY] = '^';
    gotoxy(bossY, bossX - 1);
    cout << "^";
}
void bullet2() {

    level2[bossX - 1][bossY] = '^';
    gotoxy(bossY, bossX - 1);
    cout << "^";
}
void bullet3() {

    level3[bossX - 1][bossY] = '^';
    gotoxy(bossY, bossX - 1);
    cout << "^";
}
void bullet4() {

    level4[bossX - 1][bossY] = '^';
    gotoxy(bossY, bossX - 1);
    cout << "^";
}
//.....Gotoxy
// Function.....
void gotoxy(int x, int y) {
    COORD coordinates;
    coordinates.X = x;
    coordinates.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinates);
}
//.....Load
// level1.....
```

Clear IT

```
void loadLevel1() {
    string line;
    fstream file;
    file.open("Level1.txt", ios::in);
    for (int i = 0; i < 20; i++) {
        getline(file, line);

        for (int j = 0; j < 45; j++) {
            level1[i][j] = line[j];
        }
    }
    file.close();
}

void loadLevel2() {
    string line;
    fstream file;
    file.open("Level2.txt", ios::in);
    for (int i = 0; i < 20; i++) {
        getline(file, line);

        for (int j = 0; j < 45; j++) {
            level2[i][j] = line[j];
        }
    }
    file.close();
}

void loadLevel3() {
    string line;
    fstream file;
    file.open("Level3.txt", ios::in);
    for (int i = 0; i < 20; i++) {
        getline(file, line);

        for (int j = 0; j < 45; j++) {
            level3[i][j] = line[j];
        }
    }
    file.close();
}

void loadLevel4() {
    string line;
    fstream file;
    file.open("Level4.txt", ios::in);
    for (int i = 0; i < 20; i++) {
        getline(file, line);

        for (int j = 0; j < 45; j++) {
            level4[i][j] = line[j];
        }
    }
    file.close();
}

//.....MoveBulletUp.....
void movebulletup1() {
    for (int i = 0; i < 20; i++) {

        for (int j = 0; j < 45; j++) {
            if (level1[i][j] == '^') {
```

```

    level1[i][j] = ' ';
    gotoxy(j, i);
    cout << " ";
    if (level1[i - 1][j] == ' ') {
        level1[i - 1][j] = '^';
        gotoxy(j, i - 1);
        cout << "^";
    }
    if (level1[i - 1][j] == '*') {
        level1[i - 1][j] = ' ';
        gotoxy(j, i - 1);
        cout << " ";
        calculateScore();
    }
}
}
}
}

void movebulletup2() {
    for (int i = 0; i < 20; i++) {

        for (int j = 0; j < 45; j++) {
            if (level2[i][j] == '^') {
                level2[i][j] = ' ';
                gotoxy(j, i);
                cout << " ";
            }
            if (level2[i - 1][j] == ' ') {
                level2[i - 1][j] = '^';
                gotoxy(j, i - 1);
                cout << "^";
            }
            if (level2[i - 1][j] == '*') {
                level2[i - 1][j] = ' ';
                gotoxy(j, i - 1);
                cout << " ";
                calculateScore();
            }
        }
    }
}

void movebulletup3() {
    for (int i = 0; i < 20; i++) {

        for (int j = 0; j < 45; j++) {
            if (level3[i][j] == '^') {
                level3[i][j] = ' ';
                gotoxy(j, i);
                cout << " ";
            }
            if (level3[i - 1][j] == ' ') {
                level3[i - 1][j] = '^';
                gotoxy(j, i - 1);
                cout << "^";
            }
            if (level3[i - 1][j] == '*' || level3[i - 1][j] == '@') {
                if (level3[i - 1][j] == '@') {
                    randomBomb();
                }
            }
        }
    }
}

```

Clear IT

```
        level3[i - 1][j] = ' ';
        gotoxy(j, i - 1);
        cout << " ";
        calculateScore();
    }
}
}
}
}
void movebulletup4() {
    for (int i = 0; i < 20; i++) {

        for (int j = 0; j < 45; j++) {
            if (level4[i][j] == '^') {
                level4[i][j] = ' ';
                gotoxy(j, i);
                cout << " ";
                if (level4[i - 1][j] == ' ') {
                    level4[i - 1][j] = '^';
                    gotoxy(j, i - 1);
                    cout << "^";
                }
                if (level4[i - 1][j] == '*') {

                    level4[i - 1][j] = ' ';
                    gotoxy(j, i - 1);
                    cout << " ";
                    calculateScore();
                }
                if (level4[i - 1][j] == '%') {
                    enemyLife--;
                    calculateScore();
                }
            }
        }
    }
}
for (int i = 20; i > 0; i--) {

    for (int j = 45; j > 0; j--) {
        if (level4[i][j] == '*') {
            level4[i][j] = ' ';
            gotoxy(j, i);
            cout << " ";
            if (level4[i + 1][j] == ' ') {
                level4[i + 1][j] = '*';
                gotoxy(j, i + 1);
                cout << "*";
            }
            if (level4[i + 1][j] == 'W') {
                gameOver();
            }
        }
    }
}
}
char menu() {
    header();
    cout << endl;
```

Clear IT

```
cout << "OPTIONS:" << endl;
cout << "1.Play Game from Start." << endl;
cout << "2.Play Specific Level." << endl;
cout << "3.View High Scores." << endl;
cout << "4.Exit." << endl;
cout << "Choose Any option ...";
char option;
cin >> option;
return option;
}
void playGame() {
while (true) {

    system("cls");
    char option = menu();
    if (option == '1') {
        playLevel1();
    } else if (option == '2') {
        chooseLevel();
    } else if (option == '3') {
        printHighScores();
    } else if (option == '4') {
        storeScores();
        exit(0);
    } else {
        cout << "Invalid Option.\nTry Again.\n";
        playGame();
    }
    system("pause");
}
}
void chooseLevel() {
cout << "Choose Level (1-3) : ";
char op;
cin >> op;
if (op == '1') {
    level = 1;
    playLevel1();
} else if (op == '2') {
    level = 2;
    playLevel2();
} else if (op == '3') {
    level = 3;
    playLevel3();
} else if (op == '4') {
    level = 4;
    playLevel4();
}

else {
    cout << "Invalid Level." << endl;
    cout << "Try Again." << endl;
    chooseLevel();
}
}
void playLevel1() {
if (level == 1) {
    system("CLS");
```

Clear IT

```
showLevel1();
gotoxy(bossY, bossX);
cout << "W";
//.....While
// Loop.....
while (true) {
    Sleep(50);
    movebulletup1();
    printScore();
    if (GetAsyncKeyState(VK_RIGHT)) {

        moveRight1();
    }
    if (GetAsyncKeyState(VK_LEFT)) {

        moveLeft1();
    }
    if (GetAsyncKeyState(VK_SPACE)) {

        bullet1();
    }
}
}
void playLevel2() {
    system("CLS");
    showLevel2();
    gotoxy(bossY, bossX);
    cout << "W";
    //.....While
    // Loop.....
    while (true) {
        Sleep(50);
        movebulletup2();
        moveLineDown();
        printScore();
        if (GetAsyncKeyState(VK_RIGHT)) {

            moveRight2();
        }
        if (GetAsyncKeyState(VK_LEFT)) {

            moveLeft2();
        }
        if (GetAsyncKeyState(VK_SPACE)) {

            bullet2();
        }
    }
}
void playLevel3() {
    system("CLS");
    showLevel3();
    gotoxy(bossY, bossX);
    cout << "W";
    //.....While
    // Loop.....
    while (true) {
```

Clear IT

```
Sleep(50);
movebulletup3();
moveLineDown1();
fallbomb();
printScore();
if (GetAsyncKeyState(VK_RIGHT)) {

    moveRight3();
}
if (GetAsyncKeyState(VK_LEFT)) {

    moveLeft3();
}
if (GetAsyncKeyState(VK_SPACE)) {

    bullet3();
}
}
}
void playLevel4() {
    system("CLS");
    showLevel4();
    gotoxy(bossY, bossX);
    cout << "W";
    //.....While
    // Loop.....
    while (true) {
        Sleep(50);
        movebulletup4();
        // moveLineDown1();
        // fallbomb();
        creatEnemy();
        printEnemy();
        moveEnemy();
        printScore();
        if (GetAsyncKeyState(VK_RIGHT)) {

            moveRight4();
        }
        if (GetAsyncKeyState(VK_LEFT)) {

            moveLeft4();
        }
        if (GetAsyncKeyState(VK_SPACE)) {

            bullet4();
        }
    }
}
void printHighScores() {
    system("cls");
    header();
    cout << "Top 10 High Scores : " << endl;
    for (int i = 0; i < scoreCount; i++) {
        for (int j = i + 1; j < scoreCount; j++) {
            if (highScore[i] < highScore[j]) {
                swap(highScore[i], highScore[j]);
            }
        }
    }
}
```


Clear IT

```
    }
}
for (int i = 0; i < 10; i++) {
    cout << i + 1 << ".\t" << highScore[i] << endl;
}
}

void header() {
    cout << "*****" << endl;
    cout << "*          Clear It          *" << endl;
    cout << "*****" << endl;
    cout << endl;
}

void levelCompleted() {
    if (completedLevel1() && level == 1) {
        system("cls");
        showLevel1();
        gotoxy(50, 0);
        cout << "Level No. " << level;
        gotoxy(50, 1);
        cout << "Scores : " << score;
        gotoxy(50, 2);
        cout << "Congratulations ! Level " << level << " completed.";
        gotoxy(50, 3);
        cout << "Press Any Key To Play Next Level.";
        system("pause");
        level++;
        playLevel2();
    }
    if (completedLevel2() && level == 2) {

        system("cls");
        showLevel2();
        gotoxy(50, 0);
        cout << "Level No. " << level;
        gotoxy(50, 1);
        cout << "Scores : " << score;
        gotoxy(50, 2);
        cout << "Congratulations ! Level " << level << " completed.";
        gotoxy(50, 3);
        cout << "Press Any Key To Play Next Level.";
        system("pause");
        level++;
        playLevel3();
    }
    if (completedLevel3() && level == 3) {

        system("cls");
        for (int i = 0; i < 20; i++) {

            for (int j = 0; j < 45; j++) {
                if (level3[i][j] == '^' || level3[i][j] == '*') {
                    level3[i][j] = ' ';
                }
            }
        }
        showLevel3();
        gotoxy(50, 0);
        cout << "Level No. " << level;
```

Clear IT

```
gotoxy(50, 1);
cout << "Scores : " << score;
gotoxy(50, 2);
cout << "Congratulations ! Level " << level << " completed.";
gotoxy(50, 3);
cout << "Press Any Key To Play Next Level.";
system("pause");
level++;
playLevel4();
}
if (completedLevel4() && level == 4) {
    system("cls");
    for (int i = 0; i < 20; i++) {

        for (int j = 0; j < 45; j++) {
            if (level4[i][j] == '^' || level4[i][j] == '*') {
                level4[i][j] = ' ';
            }
        }
    }
    showLevel4();
    gotoxy(50, 0);
    cout << "Level No. " << level;
    gotoxy(50, 1);
    cout << "Scores : " << score;
    gotoxy(50, 2);
    cout << "Congratulations ! Level " << level << " completed.";
    gotoxy(50, 3);
    cout << "Press Any Key To Play Next Level.";
    system("pause");
    level++;
    gotoxy(50, 4);
    cout << "You Have Passed All Levels.";
    gotoxy(50, 5);
    addHighScore(score);
    system("pause");
    storeScores();
    level = 1;
    score = 0;

    main();
}
if (over) {
    over = false;
    gotoxy(50, 0);
    cout << "Level No. " << level;
    gotoxy(50, 1);
    cout << "Lives : " << lives;
    gotoxy(50, 2);
    cout << "Scores : " << score;
    gotoxy(50, 3);
    cout << "OOPS ! Game Over.";
    gotoxy(50, 4);
    addHighScore(score);
    system("pause");
    storeScores();
    level = 1;
    score = 0;
}
```

Clear IT

```
    main();
}
}
void calculateScore() { score = score + 1; }
void printScore() {
    gotoxy(50, 0);
    cout << "Level No. " << level;
    gotoxy(50, 1);
    cout << "Lives : " << lives;
    gotoxy(50, 2);
    cout << "Scores : " << score;
    if (level == 4) {
        gotoxy(50, 3);
        cout << "Enemy Health : " << enemyLife << " ";
    }
    levelCompleted();
}
bool completedLevel1() {
    for (int i = 0; i < 20; i++) {
        for (int j = 0; j < 45; j++) {
            if (level1[i][j] == '*') {
                return 0;
            }
        }
    }
    return 1;
}
bool completedLevel2() {
    for (int i = 0; i < 20; i++) {
        for (int j = 0; j < 45; j++) {
            if (level2[i][j] == '*') {
                return 0;
            }
        }
    }
    return 1;
}
bool completedLevel3() {
    for (int i = 0; i < 20; i++) {
        for (int j = 0; j < 45; j++) {
            if (level3[i][j] == '*') {
                return 0;
            }
        }
    }
    return 1;
}
bool completedLevel4() {
    if (enemyLife > 0) {
        return false;
    }
    return true;
}
void load() {
    loadLevel1();
    loadLevel2();
    loadLevel3();
    loadLevel4();
}
```

Clear IT

```
}  
void loadScores() {  
    fstream file;  
    int hs;  
    file.open("HighScores.txt", ios::in);  
    while (!file.eof()) {  
        if (file.eof()) {  
            break;  
        }  
        file >> hs;  
        addHighScore(hs);  
    }  
    file.close();  
}  
void storeScores() {  
    fstream file;  
    file.open("HighScores.txt", ios::out);  
    for (int i = 0; i < scoreCount; i++) {  
        file << highScore[i] << " ";  
    }  
  
    file.close();  
}  
void addHighScore(int hs) {  
    highScore[scoreCount] = hs;  
    scoreCount++;  
}  
void moveLineDown() {  
    if (count == 100) {  
        count = 0;  
  
        for (int i = 19; i >= 0; i--) {  
  
            for (int j = 44; j >= 0; j--) {  
  
                if (level2[i][j] == '*') {  
                    if (level2[i + 2][j] == '#') {  
                        level2[i][j] = ' ';  
                        gotoxy(j, i + 1);  
                        cout << "*";  
                        Sleep(100);  
                        gotoxy(j, i + 1);  
                        cout << " ";  
                        gameOver();  
                    } else {  
                        level2[i + 1][j] = '*';  
                        gotoxy(j, i + 1);  
                        cout << "*";  
                    }  
                }  
            }  
        }  
    }  
    for (int i = 1; i < 43; i++) {  
        level2[1][i] = '*';  
        gotoxy(i, 1);  
        cout << "*";  
    }  
}
```

Clear IT

```
count++;
}
void moveLineDown1() {
    if (count == 120) {
        count = 0;

        for (int i = 19; i >= 0; i--) {

            for (int j = 44; j >= 0; j--) {

                if (level3[i][j] == '*') {
                    if (level3[i + 2][j] == '#') {
                        level3[i][j] = ' ';
                        gotoxy(j, i + 1);
                        cout << "*";
                        Sleep(100);
                        gotoxy(j, i + 1);
                        cout << " ";
                        gameOver();
                    } else {
                        level3[i + 1][j] = '*';
                        gotoxy(j, i + 1);
                        cout << "*";
                    }
                }
            }
        }
    }
    for (int i = 1; i < 43; i++) {
        level3[1][i] = '*';
        gotoxy(i, 1);
        cout << "*";
    }
    count++;
}
void gameOver() {
    if (lives > 0) {
        lives--;
    }

    if (lives == 0) {
        over = true;
    }
}
void fallbomb() {

    if (count1 == 1000 || count2 == -1) {
        randomBomb();
    }
    if (count2 == 7) {
        count2 = 0;
        level3[x][y] = ch;
        gotoxy(y, x);
        cout << level3[x][y];
        ch = level3[x + 1][y];
        if (level3[x + 1][y] == '#') {
            level3[x][y] = ' ';
            gotoxy(y, x);
        }
    }
}
```

Clear IT

```
    cout << " ";
    gameOver();
    randomBomb();
} else if (level3[x + 1][y] == '^') {
    level3[x][y] = ' ';
    gotoxy(y, x);
    cout << " ";
    randomBomb();
} else {
    level3[x + 1][y] = '@';
    gotoxy(y, x + 1);
    cout << "@";

    x++;
}
}
count2++;
count1++;
}
void randomBomb() {
    count1 = 0;
    srand(time(0));
    x = 1 + (rand() % 10);
    srand(time(0));
    y = 1 + (rand() % 40);
    ch = level3[x][y];
    level3[x][y] = '@';
    gotoxy(y, x);
    cout << "@";
}
void printEnemy() {
    gotoxy(enemyY, enemyX);
    cout << "_____";
    level4[enemyX][enemyY] = level4[enemyX][enemyY + 1] =
        level4[enemyX][enemyY + 2] = level4[enemyX][enemyY + 3] =
            level4[enemyX][enemyY + 4] = '_';
    gotoxy(enemyY - 1, enemyX + 1);
    cout << "/ * \\";
    level4[enemyX + 1][enemyY - 1] = '/';
    level4[enemyX + 1][enemyY + 2] = '*';
    level4[enemyX + 1][enemyY + 5] = '/';
    gotoxy(enemyY - 2, enemyX + 2);
    cout << "|__o__|";
    level4[enemyX + 2][enemyY - 2] = '|';
    level4[enemyX + 2][enemyY - 1] = level4[enemyX + 2][enemyY + 1] =
        level4[enemyX + 2][enemyY] = '_';
    level4[enemyX + 2][enemyY + 2] = 'o';
    level4[enemyX + 2][enemyY + 3] = level4[enemyX + 2][enemyY + 4] =
        level4[enemyX + 2][enemyY + 5] = '_';
    level4[enemyX + 2][enemyY + 6] = '|';
    gotoxy(enemyY - 2, enemyX + 3);
    cout << "\\    /";
    level4[enemyX + 3][enemyY - 2] = '/';
    level4[enemyX + 3][enemyY + 6] = '/';
    gotoxy(enemyY - 1, enemyX + 4);
    cout << "\\(---)";
    level4[enemyX + 4][enemyY - 1] = '/';
    level4[enemyX + 4][enemyY] = '(';
```

Clear IT

```
level4[enemyX + 4][enemyY + 1] = '-';
level4[enemyX + 4][enemyY + 2] = '-';
level4[enemyX + 4][enemyY + 3] = '-';
level4[enemyX + 4][enemyY + 4] = ')';
level4[enemyX + 4][enemyY + 5] = '/';
gotoxy(enemyY, enemyX + 5);
cout << "|  |";
level4[enemyX + 5][enemyY] = '|';
level4[enemyX + 5][enemyY + 4] = '|';
gotoxy(enemyY + 1, enemyX + 6);
cout << "---";
level4[enemyX + 6][enemyY + 1] = '%';
level4[enemyX + 6][enemyY + 2] = '%';
level4[enemyX + 6][enemyY + 3] = '%';
gotoxy(enemyY + 2, enemyX + 7);
cout << "|";
level4[enemyX + 7][enemyY + 2] = '%';
}
void moveEnemy() {
    if (enemyMove > 0) {
        moveEnemyLeft();
        enemyMove--;
        if (enemyMove == 0) {
            enemyMove = -26;
        }
    }
    if (enemyMove <= 0) {
        moveEnemyRight();
        enemyMove++;
        if (enemyMove == 0) {
            enemyMove = 26;
        }
    }
}
void moveEnemyLeft() {
    gotoxy(enemyY, enemyX);
    cout << "  ";
    level4[enemyX][enemyY] = level4[enemyX][enemyY + 1] =
        level4[enemyX][enemyY + 2] = level4[enemyX][enemyY + 3] =
        level4[enemyX][enemyY + 4] = ' ';
    gotoxy(enemyY - 1, enemyX + 1);
    cout << "  ";
    level4[enemyX + 1][enemyY - 1] = ' ';
    level4[enemyX + 1][enemyY + 2] = '*';
    level4[enemyX + 1][enemyY + 5] = ' ';
    gotoxy(enemyY - 2, enemyX + 2);
    cout << "    ";
    level4[enemyX + 2][enemyY - 2] = ' ';
    level4[enemyX + 2][enemyY - 1] = level4[enemyX + 2][enemyY + 1] =
        level4[enemyX + 2][enemyY] = ' ';
    level4[enemyX + 2][enemyY + 2] = ' ';
    level4[enemyX + 2][enemyY + 3] = level4[enemyX + 2][enemyY + 4] =
        level4[enemyX + 2][enemyY + 5] = ' _';
    level4[enemyX + 2][enemyY + 6] = ' ';
    gotoxy(enemyY - 2, enemyX + 3);
    cout << "    ";
    level4[enemyX + 3][enemyY - 2] = ' ';
```

```

level4[enemyX + 3][enemyY + 6] = ' ';
gotoxy(enemyY - 1, enemyX + 4);
cout << "      ";
level4[enemyX + 4][enemyY - 1] = ' ';
level4[enemyX + 4][enemyY] = ' ';
level4[enemyX + 4][enemyY + 1] = ' ';
level4[enemyX + 4][enemyY + 2] = ' ';
level4[enemyX + 4][enemyY + 3] = ' ';
level4[enemyX + 4][enemyY + 4] = ' ';
level4[enemyX + 4][enemyY + 5] = ' ';
gotoxy(enemyY, enemyX + 5);
cout << "      ";
level4[enemyX + 5][enemyY] = ' ';
level4[enemyX + 5][enemyY + 4] = ' ';
gotoxy(enemyY + 1, enemyX + 6);
cout << "      ";
level4[enemyX + 6][enemyY + 1] = ' ';
level4[enemyX + 6][enemyY + 2] = ' ';
level4[enemyX + 6][enemyY + 3] = ' ';
gotoxy(enemyY + 2, enemyX + 7);
cout << "      ";
level4[enemyX + 7][enemyY + 2] = ' ';
enemyY--;
gotoxy(enemyY, enemyX);
cout << "_____";
level4[enemyX][enemyY] = level4[enemyX][enemyY + 1] =
    level4[enemyX][enemyY + 2] = level4[enemyX][enemyY + 3] =
    level4[enemyX][enemyY + 4] = '_';
gotoxy(enemyY - 1, enemyX + 1);
cout << "/" * "\\";
level4[enemyX + 1][enemyY - 1] = '/';
level4[enemyX + 1][enemyY + 2] = '*';
level4[enemyX + 1][enemyY + 5] = '/';
gotoxy(enemyY - 2, enemyX + 2);
cout << "|__o__|";
level4[enemyX + 2][enemyY - 2] = '|';
level4[enemyX + 2][enemyY - 1] = level4[enemyX + 2][enemyY + 1] =
    level4[enemyX + 2][enemyY] = '_';
level4[enemyX + 2][enemyY + 2] = 'o';
level4[enemyX + 2][enemyY + 3] = level4[enemyX + 2][enemyY + 4] =
    level4[enemyX + 2][enemyY + 5] = '_';
level4[enemyX + 2][enemyY + 6] = '|';
gotoxy(enemyY - 2, enemyX + 3);
cout << "\\      /";
level4[enemyX + 3][enemyY - 2] = '/';
level4[enemyX + 3][enemyY + 6] = '/';
gotoxy(enemyY - 1, enemyX + 4);
cout << "\\(---)";
level4[enemyX + 4][enemyY - 1] = '/';
level4[enemyX + 4][enemyY] = '(';
level4[enemyX + 4][enemyY + 1] = '-';
level4[enemyX + 4][enemyY + 2] = '-';
level4[enemyX + 4][enemyY + 3] = '-';
level4[enemyX + 4][enemyY + 4] = ')';
level4[enemyX + 4][enemyY + 5] = '/';
gotoxy(enemyY, enemyX + 5);
cout << "|  |";
level4[enemyX + 5][enemyY] = '|';

```


Clear IT

```
level4[enemyX + 5][enemyY + 4] = '|';
gotoxy(enemyY + 1, enemyX + 6);
cout << "---";
level4[enemyX + 6][enemyY + 1] = '%';
level4[enemyX + 6][enemyY + 2] = '%';
level4[enemyX + 6][enemyY + 3] = '%';
gotoxy(enemyY + 2, enemyX + 7);
cout << "|";
level4[enemyX + 7][enemyY + 2] = '%';
}
void moveEnemyRight() {

gotoxy(enemyY, enemyX);
cout << "  ";
level4[enemyX][enemyY] = level4[enemyX][enemyY + 1] =
    level4[enemyX][enemyY + 2] = level4[enemyX][enemyY + 3] =
        level4[enemyX][enemyY + 4] = ' ';
gotoxy(enemyY - 1, enemyX + 1);
cout << "  ";
level4[enemyX + 1][enemyY - 1] = ' ';
level4[enemyX + 1][enemyY + 2] = '*';
level4[enemyX + 1][enemyY + 5] = ' ';
gotoxy(enemyY - 2, enemyX + 2);
cout << "    ";
level4[enemyX + 2][enemyY - 2] = ' ';
level4[enemyX + 2][enemyY - 1] = level4[enemyX + 2][enemyY + 1] =
    level4[enemyX + 2][enemyY] = ' ';
level4[enemyX + 2][enemyY + 2] = ' ';
level4[enemyX + 2][enemyY + 3] = level4[enemyX + 2][enemyY + 4] =
    level4[enemyX + 2][enemyY + 5] = '_';
level4[enemyX + 2][enemyY + 6] = ' ';
gotoxy(enemyY - 2, enemyX + 3);
cout << "    ";
level4[enemyX + 3][enemyY - 2] = ' ';
level4[enemyX + 3][enemyY + 6] = ' ';
gotoxy(enemyY - 1, enemyX + 4);
cout << "    ";
level4[enemyX + 4][enemyY - 1] = ' ';
level4[enemyX + 4][enemyY] = ' ';
level4[enemyX + 4][enemyY + 1] = ' ';
level4[enemyX + 4][enemyY + 2] = ' ';
level4[enemyX + 4][enemyY + 3] = ' ';
level4[enemyX + 4][enemyY + 4] = ' ';
level4[enemyX + 4][enemyY + 5] = ' ';
gotoxy(enemyY, enemyX + 5);
cout << "    ";
level4[enemyX + 5][enemyY] = ' ';
level4[enemyX + 5][enemyY + 4] = ' ';
gotoxy(enemyY + 1, enemyX + 6);
cout << "  ";
level4[enemyX + 6][enemyY + 1] = ' ';
level4[enemyX + 6][enemyY + 2] = ' ';
level4[enemyX + 6][enemyY + 3] = ' ';
gotoxy(enemyY + 2, enemyX + 7);
cout << "    ";
level4[enemyX + 7][enemyY + 2] = ' ';
enemyY++;
gotoxy(enemyY, enemyX);
```

Clear IT

```
cout << " _____";
level4[enemyX][enemyY] = level4[enemyX][enemyY + 1] =
    level4[enemyX][enemyY + 2] = level4[enemyX][enemyY + 3] =
    level4[enemyX][enemyY + 4] = '_';
gotoxy(enemyY - 1, enemyX + 1);
cout << "/" * "\\";
level4[enemyX + 1][enemyY - 1] = '/';
level4[enemyX + 1][enemyY + 2] = '*';
level4[enemyX + 1][enemyY + 5] = '/';
gotoxy(enemyY - 2, enemyX + 2);
cout << "|__o__|";
level4[enemyX + 2][enemyY - 2] = '|';
level4[enemyX + 2][enemyY - 1] = level4[enemyX + 2][enemyY + 1] =
    level4[enemyX + 2][enemyY] = '_';
level4[enemyX + 2][enemyY + 2] = 'o';
level4[enemyX + 2][enemyY + 3] = level4[enemyX + 2][enemyY + 4] =
    level4[enemyX + 2][enemyY + 5] = '_';
level4[enemyX + 2][enemyY + 6] = '|';
gotoxy(enemyY - 2, enemyX + 3);
cout << "\\    /";
level4[enemyX + 3][enemyY - 2] = '/';
level4[enemyX + 3][enemyY + 6] = '/';
gotoxy(enemyY - 1, enemyX + 4);
cout << "\\(---)";
level4[enemyX + 4][enemyY - 1] = '/';
level4[enemyX + 4][enemyY] = '(';
level4[enemyX + 4][enemyY + 1] = '-';
level4[enemyX + 4][enemyY + 2] = '-';
level4[enemyX + 4][enemyY + 3] = '-';
level4[enemyX + 4][enemyY + 4] = ')';
level4[enemyX + 4][enemyY + 5] = '/';
gotoxy(enemyY, enemyX + 5);
cout << "| |";
level4[enemyX + 5][enemyY] = '|';
level4[enemyX + 5][enemyY + 4] = '|';
gotoxy(enemyY + 1, enemyX + 6);
cout << "---";
level4[enemyX + 6][enemyY + 1] = '%';
level4[enemyX + 6][enemyY + 2] = '%';
level4[enemyX + 6][enemyY + 3] = '%';
gotoxy(enemyY + 2, enemyX + 7);
cout << "|";
level4[enemyX + 7][enemyY + 2] = '%';
}
void creatEnemy() {
    level4[enemyX + 8][enemyY + 2] = '*';
    gotoxy(enemyY + 2, enemyX + 8);
    cout << "*";
}
```