**MAHARAJA SURAJMAL INSTITUTE OF TECHNOLOGY**

****

**Bullet Escape**

**Submitted To Submitted By-**

Ms. Vimal Gaur Enrol. No. Name

05715002719 Ritik Ramuka

05415002719 Anshika

**INDEX**

* Introduction
* Instructions
* Source code
* Outputs
* Drawbacks
* Future Scope
* References

**INTRODUCTION**

The following is a single player 2D game programmed in C++. This game follows the concept of survival by escaping or dodging the bullets. The bullets are fired from the top of the screen which progress towards the bottom where the player is present. The player can escape the bullets by moving sideways i.e. left or right. On the output screen, the scores, levels and remaining lives of the player are also displayed simultaneously. The player has limited lives i.e. limited immunity in the game. The level and score of the player get upgraded as and when he/she progresses through the game. The speed of the incoming bullets increases with subsequent levels making it comparatively difficult to dodge them. The player who completes five levels wins the game.

**INSTRUCTIONS**

1. There are five levels in this game.
2. The player has been provided with three lives at the start of the game.
3. Use left (<-) and right (->) arrow keys to move the person.
4. The player loses a life each time it gets hit by a bullet.
5. The goal is to dodge the bullets until all the levels are completed.

**SOURCE CODE**

#include <iostream.h>

#include <conio.h>

#include <graphics.h>

#include <dos.h>

#include <process.h>

#include <stdlib.h>

void logo()

{

for (int i = 100; i < 500; i = i + 5)

{

circle(300, 300, 50);

line(i, 100, 300, 300);

line(i, 400, 300, 300);

delay(30);

}

for (int j = 0; j < 500; j = j + 10)

{

line(50 + j, 300, 80 + j, 300);

delay(20);

}

}

void bullet(int p, int q)

{

rectangle(p + 1, q, p + 49, q + 100);

arc(p + 25, q + 55, 180, 0, 10);

line(p + 15, q + 55, p + 15, q + 20); //left

line(p + 35, q + 55, p + 35, q + 20); //right

line(p + 15, q + 20, p + 35, q + 20); //top

setfillstyle(SOLID\_FILL, 8);

floodfill((p + p + 50) / 2, (q + q + 100) / 2, 15);

}

void man(int x, int y)

{

rectangle(x + 1, y, x + 49, y + 100);

circle(x + 25, y + 35, 10);

line(x + 25, y + 50, x + 25, y + 65); //body

line(x + 25, y + 50, x + 15, y + 50); //arm

line(x + 25, y + 50, x + 35, y + 50); //arm

line(x + 25, y + 65, x + 15, y + 77); //leg

line(x + 25, y + 65, x + 35, y + 77); //leg

setfillstyle(SOLID\_FILL, 8);

floodfill((x + x + 50) / 2, (y + y + 100) / 2, 15);

}

void clear2(int p, int q)

{

setcolor(8);

rectangle(p + 1, q, p + 49, q + 100);

arc(p + 25, q + 55, 180, 0, 10);

line(p + 15, q + 55, p + 15, q + 20); //left

line(p + 35, q + 55, p + 35, q + 20); //right

line(p + 15, q + 20, p + 35, q + 20); //top

setfillstyle(SOLID\_FILL, 8);

floodfill((p + p + 50) / 2, (q + q + 100) / 2, 8);

setcolor(15);

}

void clear(int x, int y)

{

setcolor(8);

rectangle(x + 1, y, x + 49, y + 100);

circle(x + 25, y + 35, 10);

line(x + 25, y + 50, x + 25, y + 65); //body

line(x + 25, y + 50, x + 15, y + 50); //arm

line(x + 25, y + 50, x + 35, y + 50); //arm

line(x + 25, y + 65, x + 15, y + 77); //leg

line(x + 25, y + 65, x + 35, y + 77); //leg

setfillstyle(SOLID\_FILL, 8);

floodfill((x + x + 50) / 2, (y + y + 100) / 2, 8);

setcolor(15);

}

void main()

{

int driver, mode;

driver = DETECT;

initgraph(&driver, &mode, "c:\\tc\\bgi");

float snd[7] = {130, 143.81, 197, 200, 148.54, 246.7, 220.89};

logo();

cleardevice();

for (int k = 0; k < 100; k++)

{

setbkcolor(MAGENTA);

settextstyle(10, HORIZ\_DIR, 6);

outtextxy(50, 120, "BULLET");

outtextxy(250, 200, "ESCAPE");

arc(170, 270, 270, 90, 40);

line(170, 230, 80, 230);

line(170, 310, 80, 310);

line(80, 230, 80, 310);

settextstyle(0, HORIZ\_DIR, 1);

outtextxy(350, 350, "Press any key to continue..");

delay(9);

}

while (!kbhit())

{

sound(snd[random(7)] \* 4);

delay(290);

}

nosound();

cleardevice();

int x = 300, y = 350, ch, life = 3, score = 0;

char choice;

setbkcolor(BLUE);

settextstyle(1, HORIZ\_DIR, 5);

outtextxy(40, 90, "..INSTRUCTIONS..");

settextstyle(0, HORIZ\_DIR, 1);

outtextxy(40, 160, "1. There are five levels in this game.");

outtextxy(40, 190, "2. The player has been provided with three lives at the start of the game.");

outtextxy(40, 210, "3. Use left ( <-) and right ( -> ) arrow keys to move the person.");

outtextxy(40, 240, "4. The player loses a life each time it gets hit by a bullet.");

outtextxy(40, 270, "5. The goal is to dodge the bullets until all the levels are completed.");

delay(3500);

for (int m = 1; m < 25; m++)

for (int n = 1; n < 80; n++)

{

gotoxy(n, m);

cout << " ";

}

cleardevice();

setbkcolor(BLACK);

setcolor(RED);

rectangle(20, 60, 200, 120);

rectangle(20, 300, 200, 420);

gotoxy(5, 21);

cout << "HELLO ";

gotoxy(5, 23);

cout << "Good luck!";

gotoxy(5, 24);

rectangle(249, 0, 401, getmaxy());

setfillstyle(SOLID\_FILL, 8);

floodfill(325, getmaxy() / 2, RED);

setcolor(RED);

rectangle(20, 200, 200, 250);

gotoxy(5, 15);

cout << "Press <Esc> to Exit";

for (int level = 1; (level <= 5) && (life > 0); level++)

{

if (level == 1)

{

gotoxy(4, 5);

cout << "!READY SET MOVE!";

gotoxy(5, 7);

cout << "All the best";

delay(5000);

gotoxy(5, 5);

cout << " ";

gotoxy(5, 7);

cout << " ";

}

else

{

gotoxy(5, 5);

cout << "Next level.";

delay(5000);

gotoxy(5, 5);

cout << " ";

}

for (int i = 0; (i < 15) && (life > 0); i++)

{

if ((level == 5) && (i == 14))

{

gotoxy(5, 5);

cout << "You have won" << endl;

cout << "Final score: " << score;

gotoxy(5, 7);

cout << "Wanna continue <y/n>";

cin >> choice;

if ((choice == 'y') || (choice == 'Y'))

main();

else

exit(0);

}

setcolor(RED);

rectangle(420, 125, 600, 175);

gotoxy(55, 10);

cout << "Level = " << level;

rectangle(420, 250, 600, 300);

gotoxy(55, 18);

cout << "Lives = " << life;

rectangle(420, 350, 600, 400);

gotoxy(55, 24);

cout << "Score = " << score;

int accident = 0;

int y1 = 1, x1 = 250 + ((rand() % 3) \* 50);

int y2 = 1, x2 = 250 + ((rand() % 3) \* 50);

score += 10;

while (y1 < getmaxy() - 1)

{

clear2(x1, y1);

clear2(x2, y2);

y1++;

y2++;

bullet(x1, y1);

bullet(x2, y2);

man(x, y);

delay(5 - level);

if (kbhit())

{

man(x, y);

ch = getch();

switch (ch)

{

case 27:

exit(0);

break;

case 75:

clear(x, y);

if (x > 250)

x = x - 50;

man(x, y);

break;

case 77:

clear(x, y);

if (x < 350)

x = x + 50;

man(x, y);

break;

}

}

if ((x == x1) || (x == x2))

if ((((y - y1) < 100) && ((y - y1) > 0)) || (((y - y2) < 100) && ((y - y2) > 0)))

accident = 1;

}

if (accident == 1)

{

life = life - 1;

score -= 10;

if (life == 0)

{

gotoxy(5, 5);

cout << "GAME OVER ";

gotoxy(5, 6);

cout << "You lose.";

break;

}

gotoxy(5, 5);

cout << "You have lost 1 life";

delay(3000);

gotoxy(5, 5);

cout << " ";

}

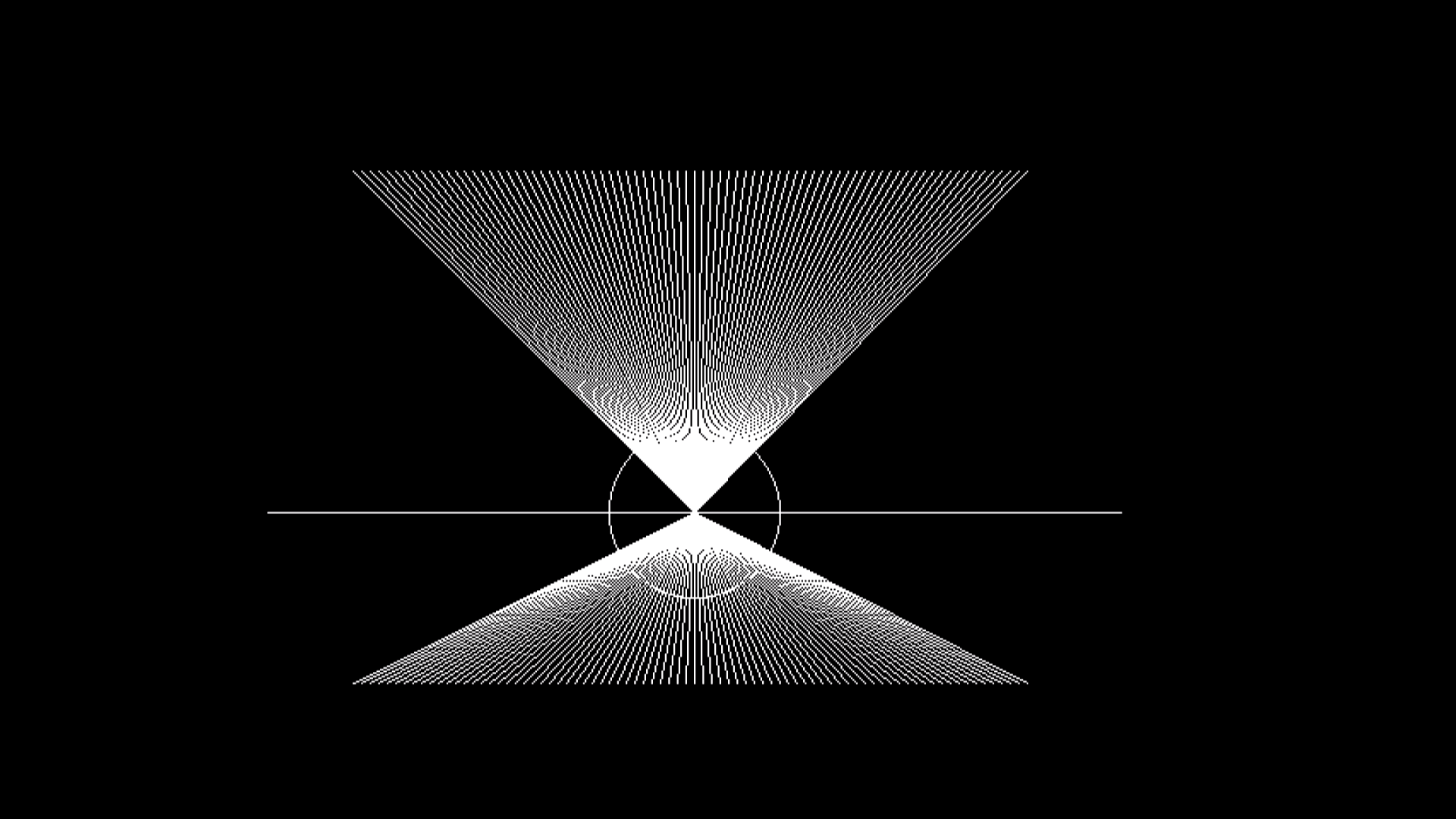
}

}

getch();

}

**SCREENSHOTS**











**DRAWBACKS**

* The player can move only in horizontal direction.
* Only one player can play at a time.
* The player is confined within the three columns only.
* The game cannot be paused in between.
* The interface is not so attractive.

**FUTURE SCOPE**

* We further wish to add various graphics features like animations, 3d graphics, voice control etc.
* This game can be extended to multiplayer so that many users can play at the same time and can be ranked at the end of the game.
* Further under the proper guidance of an expert, we wish to include various features like voice control and hand gesture.
* Since this game is a prototype; it can be further used to program various large and complex games.
* Moreover, using better and the latest versions of various software, we can improve the interface of the game.

**REFERENCES**

* https://www.includehelp.com/c-programs/sound-and-nosound-functions-of-dos-h.aspx
* <https://www.programmingsimplified.com/c/graphics.h>
* <http://see-programming.blogspot.com/p/c-graphics.html>
* https://www.quora.com/How-do-I-make-a-game-using-C