

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
#include<time.h>
```

```
#define MAX_BLOCKS 100
int game_over=0;
int score=0;
```

```
typedef struct {
    int x1, y1, x2, y2;
} Block;
```

```
Block blocks[MAX_BLOCKS];
int block_count = 0;
```

```
int iscollision(int block3_x1, int  
block3_y1, int block3_x2, int  
block3_y2) {  
    int i;  
    for (i = 0; i < block_count; i++) {  
        if (!(blocks[i].x2 < block3_x1 ||  
              block3_x2 < blocks[i].x1 || blocks[i].y2 <  
              block3_y1 || block3_y2 < blocks[i].y1)) {  
            return 1; // Collision detected  
        }  
    }  
    return 0;  
}
```

```
void board() {  
    int board_x1 = 2, board_y1 = 0;  
    int board_x2 = 639, board_y2 = 450;
```

```
rectangle(board_x1, board_y1,
board_x2, board_y2);
}

void draw_block(int x1, int y1, int x2,
int y2, int color) {
    setfillstyle(SOLID_FILL, color);
    rectangle(x1, y1, x2, y2);
    floodfill((x1 + x2) / 2, (y1 + y2) / 2,
WHITE);
}

void move_block(int *x1, int *y1, int
*x2, int *y2, int dx, int dy) {
    *x1 += dx;
    *y1 += dy;
    *x2 += dx;
}
```

```
*y2 += dy;  
}  
  
void game_ovr(){  
  
int board_x1 = 2, board_y1 = 0;  
int board_x2 = 639, board_y2 = 450;  
  
rectangle(board_x1, board_y1,  
board_x2, board_y2);  
setcolor(WHITE);  
settextstyle(3,HORIZ_DIR,4);  
outtextxy(200,300,"Game  
Over!!!");  
outtextxy(70,350,"press any key  
to go Main Menu");  
printf("SCORE=%d",score);
```

getch();

}

void handle_block(int block3_x1, int
block3_y1, int block3_x2, int
block3_y2, int color) {

int i,j;

int board_x1 = 2, board_y1 = 0;

int board_x2 = 639, board_y2 = 450;

char ch;

int m = 20;

int u = 30;

int n = 2;

int landing = 0;

char score_str[20];

board();

```
draw_block(block3_x1, block3_y1,  
block3_x2, block3_y2, color);  
  
if(iscollision(block3_x1,block3_y1,block  
3_x2,block3_y2)) {  
    game_over=1;  
  
}  
do {  
if (block3_y2 + n <= board_y2) {  
    if (!iscollision(block3_x1, block3_y1  
+ n, block3_x2, block3_y2 + n)) {  
        move_block(&block3_x1, &block3_y1,  
&block3_x2, &block3_y2, 0, n);  
    } else {  
        landing = 1;  
        score+=2;  
    }  
}
```

```
    }  
}  
} else {  
    landing = 1;  
    score+=2;  
}  
  
if (kbhit()) {  
    ch = getch();  
    if ((ch == 'a' || ch == 'A') &&  
        block3_x1 - m >= board_x1) {  
        if (!iscollision(block3_x1 - m,  
                          block3_y1, block3_x2 - m, block3_y2)) {  
            if (!game_over){  
                move_block(&block3_x1,  
                           &block3_y1, &block3_x2, &block3_y2,  
                           -m, 0);  
            }  
        }  
    }  
}
```

```
if ((ch == 'd' || ch == 'D') &&
block3_x2 + m <= board_x2) {
if (!iscollision(block3_x1 + m,
block3_y1, block3_x2 + m, block3_y2)) {
if(!game_over) {
move_block(&block3_x1,
&block3_y1, &block3_x2, &block3_y2,
m, 0);
}}
}
if(ch=='q'||ch=='Q'){
game_over=1;
return;
}
if ((ch == 'l' || ch == 'L') &&
block3_y2 + u <= board_y2) {
if (!iscollision(block3_x1, block3_y1 +
u, block3_x2, block3_y2 + u)) {
```

```
if(!game_over){  
    move_block(&block3_x1,  
&block3_y1, &block3_x2, &block3_y2,  
0, u);  
}  
  
}  
  
}  
  
}  
  
cleardevice();  
board();
```

```
setbkcolor(BLACK);
```

```
for ( i=0; i<block_count; i++ ) {  
    draw_block(blocks[i].x1, blocks[i].y1,  
blocks[i].x2, blocks[i].y2, BLUE);  
}  
draw_block(block3_x1, block3_y1,  
block3_x2, block3_y2, color);  
setcolor(WHITE);  
settextstyle(1,HORIZ_DIR,2);  
  
sprintf(score_str,"Score : %d",score);  
outtextxy(10,10,score_str);  
delay(500);  
}  
} while (!landing && ch != 27 && ch!
```

```
= 'g' && !game_over);  
    if (block_count < MAX_BLOCKS) {  
        blocks[block_count].x1 = block3_x1;  
        blocks[block_count].y1 = block3_y1;  
        blocks[block_count].x2 = block3_x2;  
        blocks[block_count].y2 = block3_y2;  
        block_count++;  
    }  
}
```

```
void b33() {  
    int offset = rand() %  
(getmaxx() - 575);  
  
    handle_block(515 + offset, 3,  
575 + offset, 63, RED);
```

}

```
void b11() {  
    int offset=rand()%getmaxx()-  
340);  
  
    handle_block(250+offset, 3,  
340+offset, 33, GREEN);  
}
```

```
void b55() {  
    int offset=rand()%getmaxx()-  
130);  
  
    handle_block(100+offset,  
3,130+offset,33, MAGENTA);  
}  
void b22() {
```

```
int offset=rand()%getmaxx()-  
240);
```

```
    handle_block(210+offset, 3,  
240+offset, 110,YELLOW);  
}
```

```
void info(){
```

```
setbkcolor(BLACK);
```

```
setcolor(WHITE);
```

```
settextstyle(3,HORIZ_DIR,6);
```

```
outtextxy(80,30,"PROGRAMMER  
INFO.");
```

```
settextstyle(3,HORIZ_DIR,4);
```

```
setcolor(WHITE);
```

```
outtextxy(10,130,"Name:-Sanju");
```

```
settextstyle(3,HORIZ_DIR,4);
```

```
setcolor(WHITE);
outtextxy(10,170,"Age:-19");
settextstyle(3,HORIZ_DIR,4);
```

```
setcolor(WHITE);
outtextxy(10,210,"Phone
no:-9877583155");
settextstyle(3,HORIZ_DIR,4);
```

```
setcolor(WHITE);
outtextxy(10,250,"Insta ID:-
majin.sanju");
settextstyle(3,HORIZ_DIR,4);
```

```
setcolor(WHITE);
outtextxy(10,295,"snap ID:-
sanjusingh8071");
settextstyle(3,HORIZ_DIR,3);
```

```
setcolor(WHITE);
outtextxy(10,350,"(Follow me on
instagram for more info!!!)");
settextstyle(3,HORIZ_DIR,3);
```

```
setcolor(WHITE);
outtextxy(10,385,"Disclaimer :- Data is
according to Year 2024");
```

```
settextstyle(8,HORIZ_DIR,1);
setcolor(WHITE);
outtextxy(5,450,"Note :- For any
enquiries, please contact the
programmer");
```

```
}
```

```
void cont() {  
setbkcolor(BLACK);  
  
settextstyle(1,HORIZ_DIR,5);  
setcolor(WHITE);  
outtextxy(30,30,"CONTROLS OF  
GAME");  
  
settextstyle(1,HORIZ_DIR,2);  
setcolor(WHITE);  
outtextxy(10,150,"Press 'a' for left  
movement");  
  
settextstyle(1,HORIZ_DIR,2);  
setcolor(WHITE);  
outtextxy(10,190,"Press 'd' for right  
movement");  
  
settextstyle(1,HORIZ_DIR,2);  
setcolor(WHITE);  
outtextxy(10,230,"press 'L' for speed
```

```
downward movement");  
    settextstyle(1,HORIZ_DIR,2);  
    setcolor(WHITE);  
    outtextxy(10,265,"press 'Q' for exit");  
    settextstyle(1,HORIZ_DIR,2);  
    setcolor(WHITE);  
    outtextxy(10,300,"Enjoy !!!");  
    settextstyle(1,HORIZ_DIR,2);  
    setcolor(WHITE);  
    outtextxy(10,340,"Note:-Game over!!  
when block overlaps top of Game  
board");  
}  
  
void start() {  
    setbkcolor(BLACK);  
    setfillstyle(SOLID_FILL,YELLOW);  
    rectangle(80,30,550,100);
```

```
floodfill((80+550)/2,(30+100)/  
2,WHITE);  
setfillstyle(SOLID_FILL,GREEN);  
rectangle(0,180,250,230);  
floodfill((0+250)/2,(180+230)/  
2,WHITE);  
setfillstyle(SOLID_FILL,GREEN);  
  
rectangle(0,250,250,300);  
floodfill((0+250)/2,(250+300)/  
2,WHITE);  
  
setfillstyle(SOLID_FILL,GREEN);  
  
rectangle(0,320,330,370);  
floodfill((0+250)/2,(320+370)/  
2,WHITE);
```

```
setfillstyle(SOLID_FILL, GREEN);
rectangle(0,390,250,440);
floodfill((0+250)/2, (390+440)/
2, WHITE);
setfillstyle(SOLID_FILL, MAGENTA);

rectangle(490,200,520,330);
floodfill((490+520)/2, (200+330)/
2, WHITE);
setfillstyle(SOLID_FILL, GREEN);

rectangle(400,330,530,360);
floodfill((400+530)/2, (330+360)/
2, WHITE);
setfillstyle(SOLID_FILL, RED);

rectangle(450,250,380,330);
```

```
floodfill((450+380)/2,(250+330)/  
2,WHITE);  
setcolor(BLACK);  
settextstyle(1,HORIZ_DIR,7);  
  
outtextxy(110,25,"TETRIS");  
setcolor(WHITE);  
settextstyle(1,HORIZ_DIR,3);  
outtextxy(10,190,"Press 1 for Start");  
setcolor(WHITE);  
settextstyle(1,HORIZ_DIR,2);  
outtextxy(10,260,"Press 2 for  
CONTROLS");  
setcolor(WHITE);  
settextstyle(1,HORIZ_DIR,2);  
outtextxy(10,330,"Press 3 for  
PROGRAMMER INFO");  
setcolor(WHITE);
```

```
settextstyle(1,HORIZ_DIR,2);
outtextxy(10,400,"Press 4 for Exit");
setcolor(WHITE);
settextstyle(1,HORIZ_DIR,3);
outtextxy(420,370,"Tetris *-*");
} void main() {
int gd = DETECT, gm;
int choice;
initgraph(&gd, &gm, "C:\\Turboc3\\
\\BGI");
srand(time(NULL));
while (1) { // Main game loop
cleardevice();
start(); // Display the main menu
scanf("%d", &choice);
cleardevice();
```

```
switch (choice) {  
    case 1: // Start the game  
        cleardevice();  
        game_over = 0;  
        score = 0;  
        block_count=0;  
  
    while (!game_over) {  
        int random_block = rand() % 4;  
        switch (random_block) {  
            case 0: b11(); break;  
            case 1: b33(); break;  
            case 2: b55(); break;  
            case 3: b22(); break;  
        }  
  
        if (kbhit() && getch() == 'g') {  
            game_over = 1; // Exit to the  
        }  
    }  
}
```

menu

```
    }  
}  
}
```

```
cleardevice();  
game_ovr(); // Display Game  
Over screen
```

```
break;
```

case 2: // Show controls

```
cleardevice();  
cont();  
getch();  
break;
```

case 3: // Show programmer info

```
cleardevice();  
info();
```

```
getch();
break;

case 4: // Exit
closegraph();
exit(0);

default:
printf("Invalid choice! Try again.
\n");
break;

}

}

}

}
```