```
#include<
iostream>
            #include<conio.h>
            #include<dos.h>
            #include<stdlib.h>
            #include<string.h>
            #include <windows.h>
            #include <time.h>

            #define SCREEN_WIDTH 52
            #define SCREEN_HEIGHT 20

            #define MINX 2
            #define MINY 2
            #define MAXX 49
            #define MAXY 19

            using namespace std;

            HANDLE console = GetStdHandle(STD_OUTPUT_HANDLE);
            COORD CursorPosition;

            int bricks[24][2] = {

            {2,7},{2,12},{2,17},{2,22},{2,27},{2,32},{2,37},{2,42},

            {4,7},{4,12},{4,17},{4,22},{4,27},{4,32},{4,37},{4,42},

            {6,7},{6,12},{6,17},{6,22},{6,27},{6,32},{6,37},{6,42}
                                                    };

            int visibleBricks[24] = {1,1,1,1, 1,1,1,1, 1,1,1,1, 1,1,1,1,
            1,1,1,1, 1,1,1,1};
```

```
int sliderPos[2] = {18,22};
int ballPos[2] = {17,26};
int startBall = 0;
int dir = 1; // 1- TopRight, 2- TopLeft, 3- BottomLeft, 4-BottomRight
int bricksLeft = 24;
int win = 0;
int lose = 0;


void gotoxy(int x, int y)
{
	CursorPosition.X = x;
	CursorPosition.Y = y;
	SetConsoleCursorPosition(console, CursorPosition);
}


void setcursor(bool visible, DWORD size) // set bool visible = 0 - invisible, bool visible = 1 - visible
{
	if(size == 0)
	{
		size = 20;	// default cursor size Changing to numbers from 1 to 20, decreases cursor width
	}
	CONSOLE_CURSOR_INFO lpCursor;
	lpCursor.bVisible = visible;
	lpCursor.dwSize = size;
	SetConsoleCursorInfo(console,&lpCursor);
}


void drawBorder(){

gotoxy(0,0);cout<<"-----------------------------------------------";
```

```
gotoxy(0,SCREEN_HEIGHT);cout<<"----------------------------------
--------------------";

	for(int i=0; i<SCREEN_HEIGHT; i++){
		gotoxy(0,i); cout<<"|";
		gotoxy(SCREEN_WIDTH,i); cout<<"|";
	}
}

void drawBricks(){
	for( int i=0; i<24; i++){
		if( visibleBricks[i] == 1 ){
			gotoxy(bricks[i][1], bricks[i][0]);
			cout<<"±±±±";
		}
	}
}

void ballHitSlider(){
	if( ballPos[1]>=sliderPos[1] &&
ballPos[1]<=sliderPos[1]+8){
		if( ballPos[0] == sliderPos[0]-1 ){
			if( dir == 3 )
				dir = 2;
			else if( dir == 4 )
				dir = 1;
		}
	}
}

void ballHitBrick(){
	for( int i=0; i<24; i++){
		if(visibleBricks[i]==1){
```

```
                    if( ballPos[1]>=bricks[i][1] &&
ballPos[1]<=bricks[i][1]+8){
                        if( ballPos[0] == bricks[i][0] ){
                            visibleBricks[i] = 0;
                            bricksLeft--;
                        }
                    }
                }
        }
}


void play(){
        while(1){
                system("cls");
                drawBricks();
                drawBorder();


                gotoxy(sliderPos[1],sliderPos[0]);
                cout<<"±±±±±±±±±";


                gotoxy(ballPos[1],ballPos[0]);
                cout<<"0";


                if(kbhit()){
                    char ch = getch();
                    if( ch=='d'||ch=='D'|| ch==77 ){
                        if(sliderPos[1] < 44)
                            sliderPos[1] = sliderPos[1]+2;
                    }
                    if( ch=='a'||ch=='A'|| ch==75 ){
                        if(sliderPos[1] > 2)
                            sliderPos[1] = sliderPos[1]-2;
                    }
```

```
		if(ch==32){
			startBall = 1;
		}
		if(ch==27){
			break;
		}
	}

	if( startBall == 1 ){
		if( dir == 1) { // TOP RIGHT
			ballPos[0] = ballPos[0] - 1;
			ballPos[1] = ballPos[1] + 2;
			if( ballPos[1] > MAXX ){
				dir = 2;
			}
			else if( ballPos[0] < MINY ){
				dir = 4;
			}
		}
		else if( dir == 2) { // TOP LEFT
			ballPos[0] = ballPos[0] - 1;
			ballPos[1] = ballPos[1] - 2;
			if( ballPos[0] < MINY ){
				dir = 3;
			}
			else if( ballPos[1] < MINX  ){
				dir = 1;
			}
		}
		else if( dir == 3) { // BOTTOM LEFT
			ballPos[0] = ballPos[0] + 1;
			ballPos[1] = ballPos[1] - 2;
```

```
                        if( ballPos[0] > MAXY ){
                                lose = 1;
                                break;
                        }
                        else if( ballPos[1] < MINX  ){
                                dir = 4;
                        }
                }
                else if( dir == 4) { // BOTTOM RIGHT
                        ballPos[0] = ballPos[0] + 1;
                        ballPos[1] = ballPos[1] + 2;
                        if( ballPos[1] > MAXX ){
                                dir = 3;
                        }
                        else if( ballPos[0] > MAXY ){
                                lose = 1;
                                break;
                        }
                }


                ballHitSlider();
        }


        ballHitBrick();


        if( bricksLeft == 0){
                win = 1;
                break;
        }


        Sleep(30);
}
```

```
	if( lose == 1){
		system("cls");

		gotoxy(10,5); cout<<" --------------------- ";
		gotoxy(10,6); cout<<" |     YOU LOSE      | ";
		gotoxy(10,7); cout<<" --------------------- ";


		gotoxy(10,9); cout<<"Press any key to go back to
Menu.";
		getch();
	}

	if( win == 1){
		system("cls");
		gotoxy(10,5); cout<<" --------------------- ";
		gotoxy(10,6); cout<<" |     YOU WIN       | ";
		gotoxy(10,7); cout<<" --------------------- ";


		gotoxy(10,9); cout<<"Press any key to go back to
Menu.";
		getch();
	}
}

void instructions(){

	system("cls");
	cout<<"Instructions";
	cout<<"\n----------------";
	cout<<"\n1. Use 'a' key to move slider to left";
	cout<<"\n2. Use 'd' key to move slider to right";
	cout<<"\n3. Press spacebar to start game";
```

```
	cout<<"\n\nPress any key to go back to menu";
	getch();
}

int main()
{
	setcursor(0,0);

	do{
		system("cls");
		gotoxy(10,5); cout<<" -------------------------- ";
		gotoxy(10,6); cout<<" |     BRICK BREAKER      | ";
		gotoxy(10,7); cout<<" --------------------------";
		gotoxy(10,9); cout<<"1. Start Game";
		gotoxy(10,10); cout<<"2. Instructions";
		gotoxy(10,11); cout<<"3. Quit";
		gotoxy(10,13); cout<<"Select option: ";
		char op = getche();

		if( op=='1') play();
		else if( op=='2') instructions();
		else if( op=='3') exit(0);

	}while(1);

	play();

	cout<<endl<<endl;
	return 0;
}
```