# Project Evaluation Members

## First Evaluator

_____

**(Signature)**

## Second Evaluator

_____

**(Signature)**

# Abstract

We are working in a project based on software that is DX-Ball. The game is basically a breaking out the player controls a paddle at the bottom and deflects a single ball, hitting different colored blocks on the top of the screen without having the ball fall below the screen. Clearing all the blocks results in completing the level and going to the next. Similarly, there is an inclusion of power-ups other than extra balls. Various power-ups appear when hitting random blocks, floating downwards towards the bottom and can be picked up by touching it with the paddle. If only a single block remains on a level and it continues to be untouched by bouncing ball for a minute or so, an electricity sound begins to build and eventually the block is blasted away by a lightning.

# Acknowledgements

A project is a golden opportunity for a student to learn and self-development. We consider ourselves very lucky and honored to have so many wonderful people who lead us through in completion of this project.

My grateful thanks to **Dr. Md. Mijanur Rahman** who in spite of being extraordinarily busy with his duties, took time out to hear, guide and keep us on the correct path. I do not know where I would have been without him.

Last but not the least there were so many who shared valuable information that helped in the successful completion of this project.

**Group Members:**

Monisha Dey

Roll: 17102032

Kanis Fatema

Roll: 17102010

# Organization of This Project

## Chapter 1

1.1 Introduction

1.2 Objective

1.3 Background Study

1.4 Software Development

## Chapter 2

## Implementation

2.1 Introduction

2.2 Algorithms

2.2.1 Methods

2.3 Methodological Steps

2.3.1 Block Diagram/objects/events

## Chapter-3

## Result and Discussion

3.1 Introduction

3.2 Process of Program Execution

3.3 Experimental Result

3.4 Limitations

# Chapter -1

## 1.1 Introduction

We are working on software base development project. Software is a set of instructions that makes the computer perform tasks. In other words, software tells the computer what to do. The term program refers to any piece of software.

In addition to, software is a general term for the various kinds of programs used to operate computers and related devices. Software can be thought of as the variable part of a computer and hardware the invariable part. Software is often divided into application software (programs that do work users are directly interested in) and software (which includes operating system and any program that supports application software). The term middleware is sometimes used to describe programming that mediates between application and system software or between two different kinds of application software (for example, sending a remote work request from an application in a computer that has one kind of operating system to an application in a computer with a different operating system).

## 1.2 Objective

We want to work on the topic that is DX-Ball. First of all the game will inquire the name of player. In this game the player controls a paddle at the bottom and deflects a single ball, hitting different colored blocks on the top of the screen without having the ball fall below the screen. When the ball fall down the computer will show the game over on the screen. Then it will show the score board on the screen.

## 1.3 Background Study

Our game is based on the programming language C. 'C' seems a strange name for a programming language. But this strange sounding language is one of the most popular computer languages because it is a structured, middle level and independent language. It allows software developers to develop programs without worrying about the hardware platforms.The language was invented and first implemented by Dennis Ritchie on a DEC PDP-11 using UNIX operating system. C is the result of a development process that started with an older language called BCPL (Basic Combined Programming Language), developed by Martin Richards. C was enveloped from ALGOL, BCPL and B by Dennis Ritchie at the Bell Laboratories in 1972.

The increasing popularity of C is probably due to its many desirable qualities. It is a robust language whose rich set of built-in functions and operators can be used to write any complex program. The C compiler combines the capabilities of an assembly language with the features of a high-level language and therefore it is well suited for writing both system software and business packages.

## 1.4 Software Development

The name our course is **Software Development Project-1**. Software development is a process by which stand alone or individual. Software is created using a specific programming language. It involves writing a series of interrelated programming code, which provides the functionality of the developed software. Software development may also be called application development and software design.

There are some steps for software development:

I. Analyzing the problem
II. Market research
III. Gathering requirements for the proposed business solution
IV. Devising a plan or design for the software-based solution
V. Implementation (coding) of the software
VI. Testing the software
VII. Deployment
VIII. Maintenance and bug fixing

These stages are often referred to collectively as the software development lifecycle, or SDLC.

# Chapter 2

# Implementation

## 2.1 Introduction

Here we implement the concept of bricks games. We make a feature of normal brick game. Then we add many function, structure, parameter to implement the DX-Ball project. We color the bricks using RGB concepts. We add text for understanding the project better. We do some graphical design in this project by visual studio.

We make background, brick ball, bricks of difference color & also we edit text option of entering name. Then we move the ball. If stick fail to hit or touch the ball the game will end and we have to restart the game again. Finally we have to see the score of all player in time sequence.

## 2.2 Algorithm

Step 1: Start

Step 2: We make width and height of the screen

Step 3: We make a ball and place it to middle

Step 3: We can move the ball left and right up to the end of the boarder

Step 4: Then we through the ball and it return.

Step 5: We make the bricks using RGB concepts.

```
iSetColor(red, green, blue);
```

```
brickk[i].red = rand() % 255;
brickk[i].green = rand() % 255;
brickk[i].blue = rand() % 255;
```

Step 6: We add text using iText() function.

```
iText(1200-40, 700-40,
p[index].sc_char,GLUT_BITMAP_HELVETICA_18);
```

Step 7: Show Higther Score.

Step 8: Stop.

## Methods

restart() → This function restart the ball again.

mainbounce() → This function is used to bounce the ball on the screen given.

thebricks() → In this function we make bricks.

iText() → In this function we can see text and enter our text.

save() → Here we save score and name in file.

read() → we read the score and name of scorer in time sequence.

iDraw() → This is use for custom shape, color, asset etc.

iclear() → this clear before system.

ishowBMP2() → this is used for showing picture.

isetColor() → By this we set the color of bricks.

## 2.3 Methodological Steps

I. First we make the screen for background.

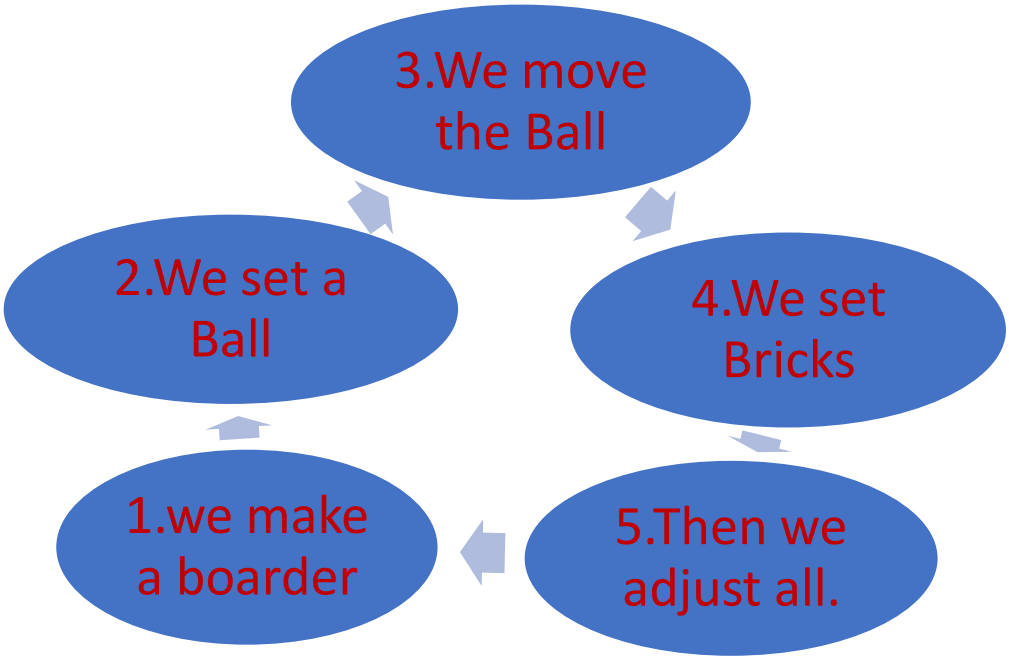
II. Then we make ball

III. Then we move the ball

IV. We set bricks by RGB function

V. Then we adjust all.

## Block diagram



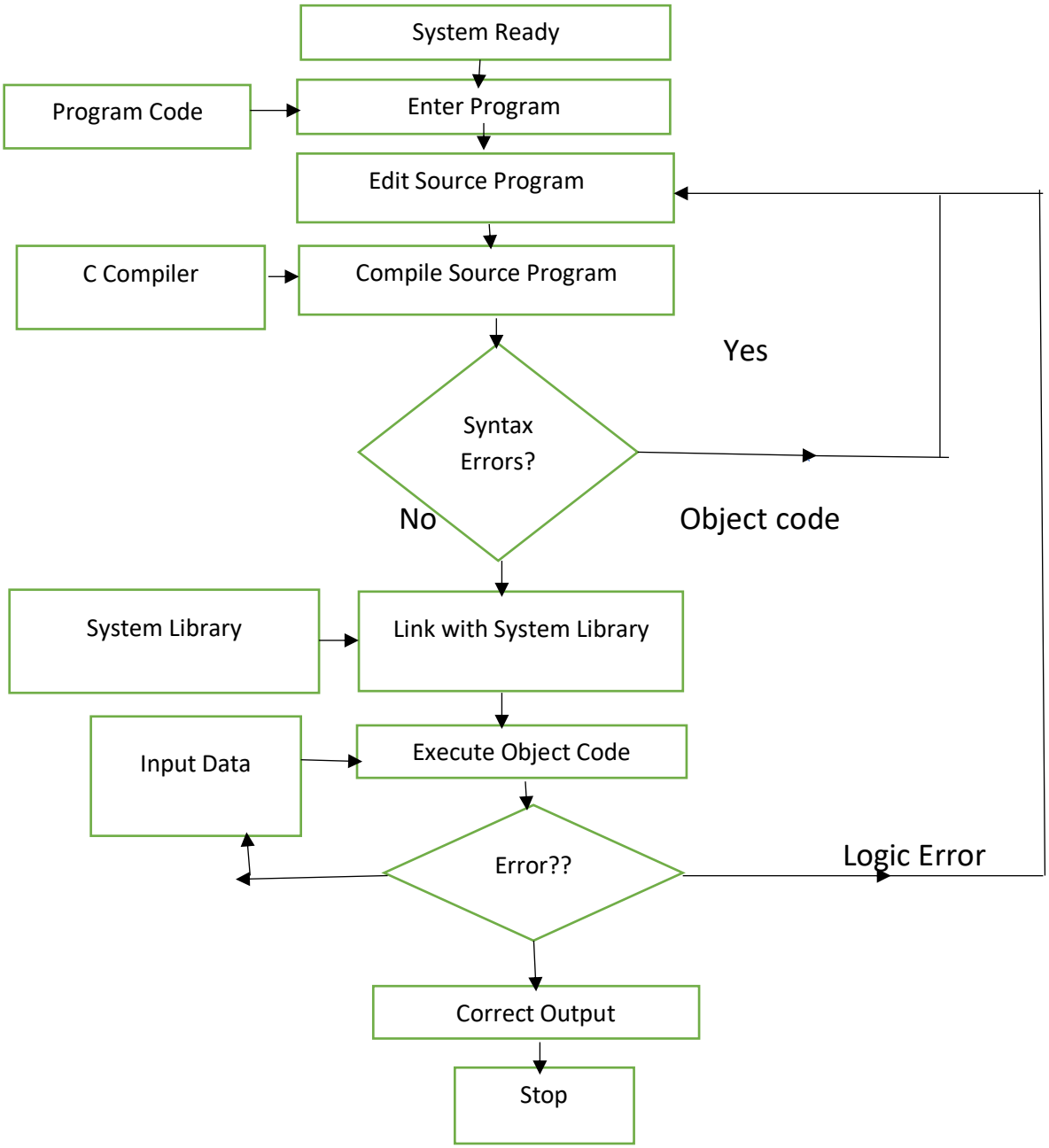
# Chapter-3

# Results and Discussion

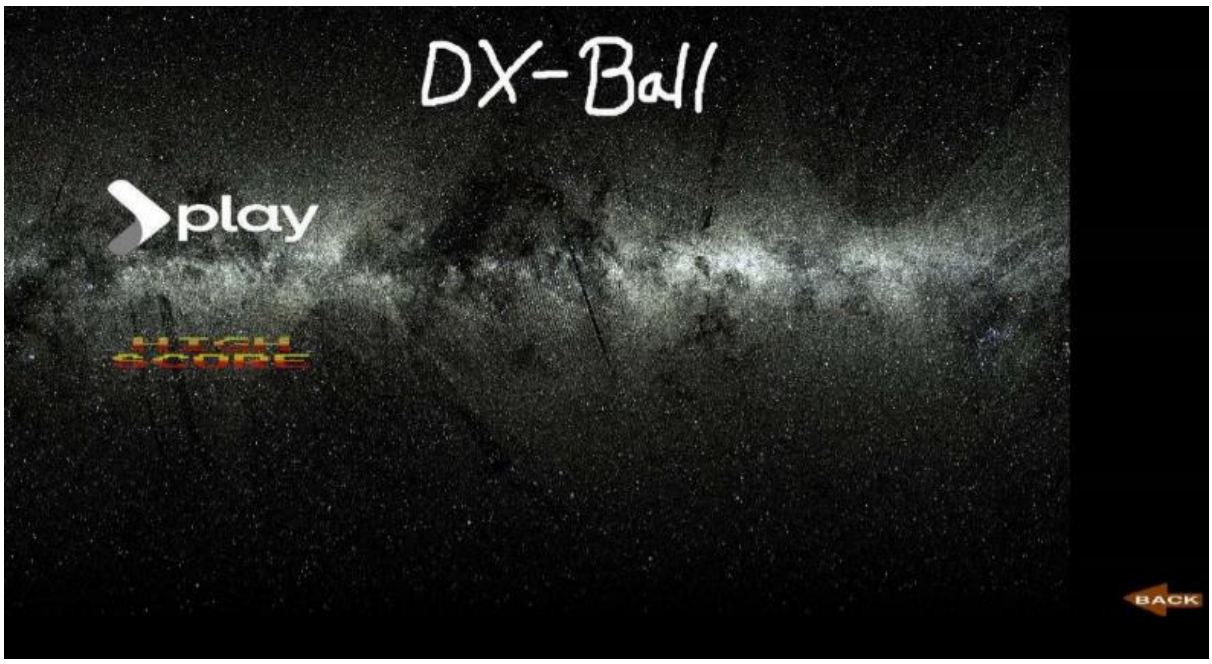
## 3.1 Introduction

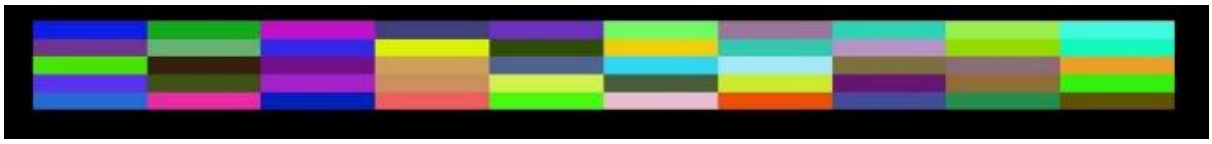
Now we come the result and discussion about our project. Our main object is to break the bricks with the ball which is in a base. When we move the ball it breaks the bricks and the bar maintains the balance. We use the 'igraphics.h' header file.In the program there is a function that is idraw(). In the function idraw() whatever we want to draw in the program can be implemented. After finishing all the procedures we are successful to finish our project. In the project the ball will hit the bars and also will break them. Hence our goal of running the program has succeed.

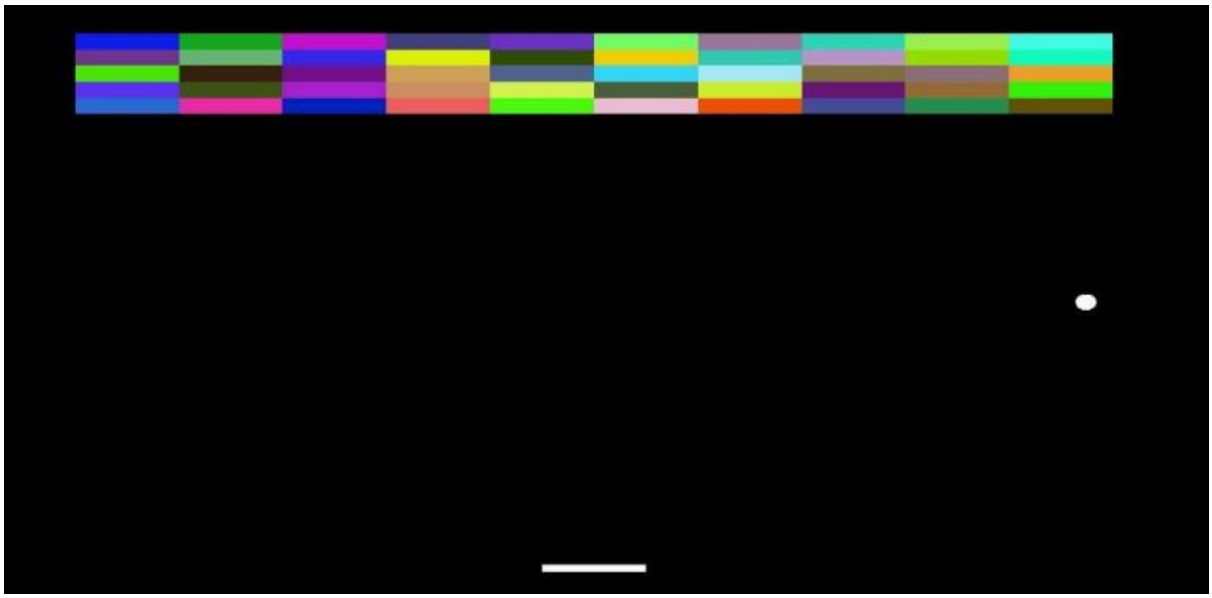
## 3.2 Process of Program Execution

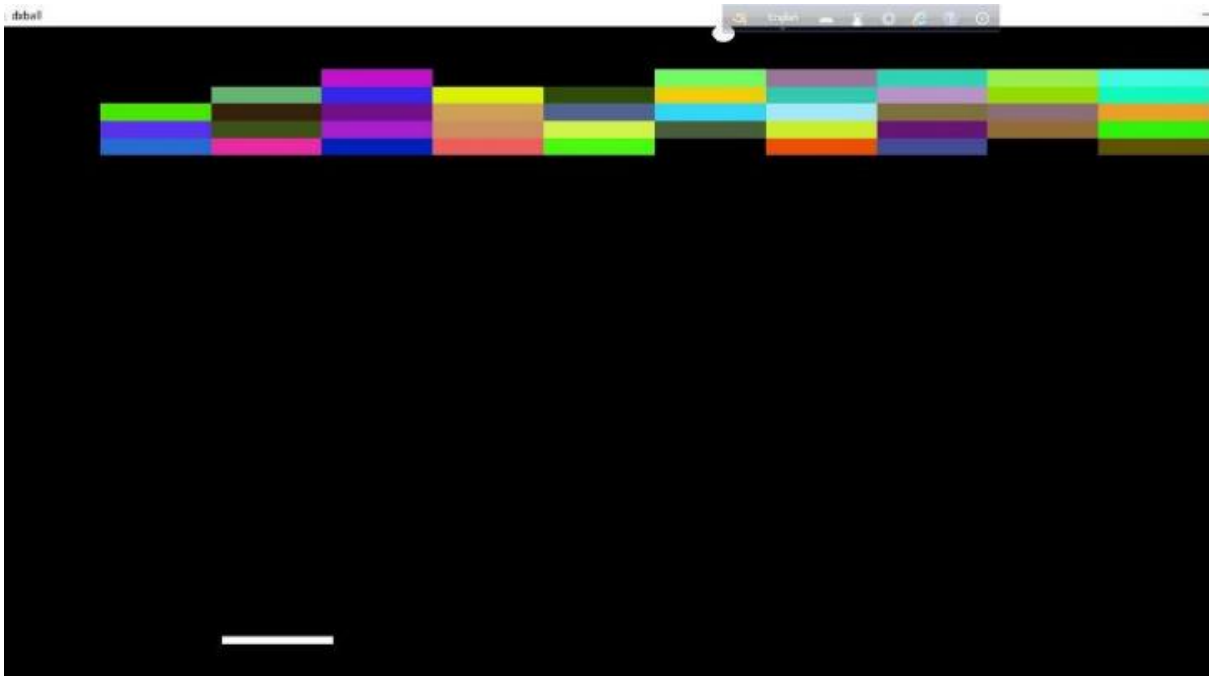


## 3.3 Experimental Output







## 3.4 Limitation of This Project

I. We can’t add the live chances here.

II. There is only one level exists.

III. Creating graphics with the graphics methods takes place in code, which means you have to run the application to see the effect of a graphics method.

IV. Graphics methods therefore don't work as well as graphical controls for creating simple design elements of an interface. Changing the appearance of graphical controls at design time is easier than modifying and testing the code for a graphics method.

## 3.5 Future Scope

Games are sometimes played purely for entertainment, sometimes for achievement or reward as well. They can be played alone, in teams, or online; by amateurs or by professionals. The players may have an audience of non-players, such as when people are entertained by watching a chees cheership. On the other hand, players in a game may constitute their own audience as they take their turn to play. Often, part of the entertainment for children playing a game is deciding who is part of their audience and who is a player. DX-ball is not different. It,s a game for enjoyment. We can play this game in our free time. Even in many other gaming contest it can be presented. Our future effort will be to develop this game more attractively

## 3.6 Conclusion

We want to pay a cordial thanks to our project supervisor **Dr. Md. Mijanur Rahman**. As a supervisor he tried hard to introduce us with software development but it is our bad luck that we can't utilize that knowledge. Again thanks a lot to our honorable supervisor **Dr. Md. Mijanur Rahman.**

# Reference/Bibliography

We took help from various books and websites. They are:


I. C Programming -By Md. Moktar Hossain
II. Teach Yourself C the Complete Refence- By Herbert Schieldt
III. https://www.google.com/aclk?sa=L&ai=DChcSEwiO9Ne60YraAhXZIysKHWsEAY4YABADGgJzZg&sig=AOD64_1ClmG-T83OTb78N459zFAl1UId2A&q=&ved=0ahUKEwjzy9O60YraAhUdSo8KHVJmAIQQ0QwIMA&adurl
IV. https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwjzy9O60YraAhUdSo8KHVJmAIQQFgg-MAA&url=http%3A%2F%2Fwww.techcrashcourse.com%2F2015%2F08%2Fc-graphics-programming-tutorial.html&usg=AOvVaw1FcNsTsOPseJ7B1HJRJAwe
V. https://www.youtube.com/watch?v=WKN1z8duQwY&list=PLKiZXxQe7OiDVNhkwgGZ6A6xW-zMbnSXb&index=2
VI. Programming in ANSI C -By E. Balagurusamy

# Appendix

## Program Code

### Border

```
# include <iostream>


# include "iGraphics.h"
using namespace std;
#define screenWidth 1200
#define screenHight 800


void iDraw()
{
	iClear();
}


/*
	function iMouseMove() is called when the user presses and drags the mouse.
	(mx, my) is the position where the mouse pointer is.
```

```
*/
void iMouseMove(int mx, int my)
{
	//place your codes here
}


/*
	function iMouse() is called when the user presses/releases the mouse.
	(mx, my) is the position where the mouse pointer is.
*/
void iMouse(int button, int state, int mx, int my)
{
	if(button == GLUT_LEFT_BUTTON && state == GLUT_DOWN)
	{


	}
	if(button == GLUT_RIGHT_BUTTON && state == GLUT_DOWN)
	{
		//place your codes here
```

```
        }
}

void iKeyboard(unsigned char key)
{

}

void iSpecialKeyboard(unsigned char key)
{

        if(key == GLUT_KEY_LEFT)
        {

        }
        else if(key == GLUT_KEY_RIGHT)
        {

        }
        else if(key == GLUT_KEY_UP)
        {
```

```
        }

    }
    int main()
    {
    iInitialize(screenWidth, screenHight, "dxball");

        return 0;
    }
```

## Brick with Ball

```
# include <iostream>

# include "iGraphics.h"
using namespace std;
#define screenWidth 1200
#define screenHight 800
#define bricks 50

struct bricksturcture
{
```

```
	int x;
	int y;
	int dx;
	int dy;
	bool show;
	int red;
	int green;
	int blue;
};

struct bricksturcture brickk[bricks];

int index=0;

int xmahin = screenWidth/2;
int ymahin = 100;
int radius = 10;
int dxmahin = 3;
int dymahin = 2;
void thebricks();
void iDraw()
```

```
{
	iClear();


	   int i = 0;
	   for(i=0 ; i < bricks ; i++)
	   {
		if(brickk[i].show)
		{
			iSetColor(brickk[i].red, brickk[i].green,
brickk[i].blue);
			iFilledRectangle(brickk[i].x , brickk[i].y , brickk[i].dx
, brickk[i].dy);
		}
	   }
}


/*
	function iMouseMove() is called when the user presses and
drags the mouse.
	(mx, my) is the position where the mouse pointer is.
*/
void iMouseMove(int mx, int my)
```

```
{
	//place your codes here
}


/*
	function iMouse() is called when the user presses/releases the mouse.
	(mx, my) is the position where the mouse pointer is.
*/
void iMouse(int button, int state, int mx, int my)
{
	if(button == GLUT_LEFT_BUTTON && state == GLUT_DOWN)
	{


	}
	if(button == GLUT_RIGHT_BUTTON && state == GLUT_DOWN)
	{
		//place your codes here
	}
}


void iKeyboard(unsigned char key)
```

```
{

}

void iSpecialKeyboard(unsigned char key)
{

	if(key == GLUT_KEY_LEFT)
	{

	}
	else if(key == GLUT_KEY_RIGHT)
	{

	}
	else if(key == GLUT_KEY_UP)
	{

	}

}
```

```
void thebricks()
{
	int sx = 100;
	int sy = 650;
	int i;
	for(i=0 ; i<bricks ; i++)
	{
		brickk[i].x = sx;
		brickk[i].y = sy;
		brickk[i].dx = 100;
		brickk[i].dy = 20;
		brickk[i].show = true;
		brickk[i].red = rand() % 255;
		brickk[i].green = rand() % 255;
		brickk[i].blue = rand() % 255;
		sx += 100;
		if(sx > 1000)
		{
			sx = 100;
			sy += 20;
```

```
            }
        }
}

int main()
{
        thebricks();
        iInitialize(screenWidth, screenHight, "dxball");

        return 0;
}
```

## Breaking the bricks

```
# include <iostream>
# include "iGraphics.h"
using namespace std;
#define screenWidth 1200
#define screenHight 800
#define bricks 50

struct bricksturcture
{
```

```
	int x;
	int y;
	int dx;
	int dy;
	bool show;
	int red;
	int green;
	int blue;
};

struct bricksturcture brickk[bricks];

struct balllocation
{
	int x;
	int y;
};
struct balllocation bl[2];
int index=0;

int red = 255;
```

```
int green = 255;
int blue = 255;

int xshanta  = screenWidth/2;
int yshanta  = 100;
int radius = 10;
int dxshanta  = 3;
int dyshanta  = 2;

int xstick = xshanta  - 50;
int ystick = yshanta  - 20;
int dxstick = 100;
int dystick = 10;

bool start = false;

void restartshanta ();
void shanta bounce();
void thebricks();

void iDraw()
```

```
{
	iClear();
	   iSetColor(red, green, blue);
	   iFilledCircle(xshanta , yshanta , radius, 100);
	   iFilledRectangle(xstick, ystick, dxstick, dystick);

	   int i = 0;
	   for(i=0 ; i < bricks ; i++)
	   {
		if(brickk[i].show)
		{
			iSetColor(brickk[i].red, brickk[i].green,
brickk[i].blue);
			iFilledRectangle(brickk[i].x , brickk[i].y , brickk[i].dx
, brickk[i].dy);
		}
	   }
}

/*
	function iMouseMove() is called when the user presses and
drags the mouse.
```

```
	(mx, my) is the position where the mouse pointer is.
*/
void iMouseMove(int mx, int my)
{
	//place your codes here
}

/*
	function iMouse() is called when the user presses/releases the mouse.
	(mx, my) is the position where the mouse pointer is.
*/
void iMouse(int button, int state, int mx, int my)
{
	if(button == GLUT_LEFT_BUTTON && state == GLUT_DOWN)
	{
	}
	if(button == GLUT_RIGHT_BUTTON && state == GLUT_DOWN)
	{
		//place your codes here
	}
}
```

```
void iKeyboard(unsigned char key)
{
	if(key == 'a')
	{
		if(xstick > 0)
		{
		 xstick -= 10;
		 if(!start)
		 {
			xshanta  -= 10;
		 }
		}
	}
	if(key == 'd')
	{
		if(xstick < screenWidth - dxstick)
		{
		 xstick += 10;
		 if(!start)
		 {
```

```
                xshanta  += 10;
            }
            }
        }
        if(key == 'w')
        {
            if(!start)
            {
                start = true;
            }
        }
        if(key == ' ')
        {
            restartshanta ();
    }
}

void iSpecialKeyboard(unsigned char key)
{

        if(key == GLUT_KEY_LEFT)
```

```
{
        if(xstick > 0)
        {
         xstick -= 10;
         if(!start)
         {
                xshanta  -= 10;
         }
        }
}
else if(key == GLUT_KEY_RIGHT)
{
        if(xstick < screenWidth - dxstick)
        {
         xstick += 10;
         if(!start)
         {
                xshanta  += 10;
         }
        }
}
```

```
        else if(key == GLUT_KEY_UP)
        {
                if(!start)
                {
                        start = true;
                }
        }

}

void restartshanta ()
{
        xshanta  = screenWidth/2;
    yshanta  = 100;
        dxshanta  = 3;
    dyshanta  = 2;
    xstick = xshanta  - 50;
    ystick = yshanta  - 20;

        start = false;
```

```
        int i;
        for(i=0 ; i<bricks ; i++)
        {
                brickk[i].show = true;
        }

}

void shantabounce()
{
        if(start)
        {
          xshanta  += dxshanta ;
          yshanta  += dyshanta ;
          if(xshanta  >= screenWidth || xshanta  <= 0)
         {
                dxshanta *=(-1);
         }
          if(yshanta  >= screenHight || yshanta  <= 0)
         {
                dyshanta *=(-1);
```

```
 }
}

if(xshanta >=xstick && xshanta <=xstick+dxstick && yshanta >=ystick && yshanta <=ystick+dystick)
{
        dyshanta *=(-1);
}
else
{
        if(yshanta  < ystick)
   {
          restartshanta ();
   }
}

//brickbreaker
int i;
for(i=0 ; i<bricks ; i++)
{
        if(brickk[i].show)
        {
```

```
                    if(xshanta >=brickk[i].x && (xshanta
<=brickk[i].x+brickk[i].dx))
                    {
                        if(yshanta >=brickk[i].y && (yshanta
<=brickk[i].y+brickk[i].dy))
                        {
                       dyshanta *=(-1);
                          brickk[i].show = false;
                          printf("\a");
                        }
                    }
                    else if(yshanta >=brickk[i].y && (yshanta
<=brickk[i].y+brickk[i].dy))
                    {
                       if(xshanta >=brickk[i].x && (xshanta
<=brickk[i].x+brickk[i].dx))
                       {
                            dxshanta *=(-1);
                            brickk[i].show = false;
                            printf("\a");
                       }
                  }
               }
```

```
        }

}

void thebricks()
{
        int sx = 100;
        int sy = 650;
        int i;
        for(i=0 ; i<bricks ; i++)
        {
                brickk[i].x = sx;
                brickk[i].y = sy;
                brickk[i].dx = 100;
                brickk[i].dy = 20;
                brickk[i].show = true;
                brickk[i].red = rand() % 255;
                brickk[i].green = rand() % 255;
                brickk[i].blue = rand() % 255;
                sx += 100;
                if(sx > 1000)
```

```
            {
                sx = 100;
                sy += 20;
            }
        }
}


int main()
{
        thebricks();
        iSetTimer(5,shanta bounce);
        iInitialize(screenWidth, screenHight, "dxball");

        return 0;
}
```