**“To develop a 2D platform game using SDL library and C”**

***A***

***Project Report***

*Submitted in partial fulfillment of the*

*Requirements for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE & ENGINEERING**

**By**

|  |  |
| --- | --- |
| **Name** | **Roll No.** |
| **Siddhant Kapoor** | **R970214068** |
| **Siddhant Srivastava** | **R970214069** |
| **Shikhar Shukla** | **R970214057** |

***Under the guidance of***

**Mrs. Richa Choudhary**

****

**Department of Computer Science & Engineering**

**Centre for Information Technology**

**University of Petroleum & Energy Studies**

**Bidholi, Via Prem Nagar, Dehradun, Uttarakhand**

**December – 2016**

**DECLARATION**

I/We hereby certify that the project work entitled **“To develop a 2D platform game using graphic library and C”** in partial fulfilment of the requirements for the award of the Degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING with specialization in Oil and Gas and submitted to the Department of Computer Science & Engineering at Center for Information Technology, University of Petroleum & Energy Studies, Dehradun, is an authentic record of my/ our work carried out during a period from **August**, **2016** to **November**, **2016** under the supervision of **Mrs. Richa Choudhary.**

The matter presented in this project has not been submitted by me/ us for the award of any other degree of this or any other University.

**Siddhant Kapoor R970214068**

**Siddhant Srivastava R970214069**

**Shikhar Shukla R970214057**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: December 9, 2016 **Mrs. Richa Choudhary**

Project Guide

**ACKNOWLEDGEMENT**

We wish to express our deep gratitude to our mentor Mrs. Richa Choudhary, for all advice, encouragement and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.

We sincerely thank to our respected Program Head of the Department, Mr. Venkatadri, for his great support in doing our project in Area at CIT.

We are also grateful to Mr. Kamal Bansal, Dean, CoES and Dr. Manish Prateek, Dean CoES, UPES for giving us the necessary facilities to carry out our project work successfully.

We would like to thank all our friends for their help and constructive criticism during our project work. Finally we have no words to express our sincere gratitude to our parents who have shown us this world and for every support they have given us.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Siddhant Kapoor** | **Siddhant Srivastava** | **Shikhar Shukla** |
| **Roll No.** | **R970214068** | **R970214069** | **R970214057** |

**ABSTRACT**

Our Aim is to develop a platform game using Graphics library and C programming language. The game is developed in C and Graphics Library! The Game will demonstrate hardware acceleration which is the use of computer hardware to perform some functions faster than is possible in software running on a more general-purpose CPU.

OpenGL or DirectX (available only on Windows) API, which in turn are using your video hardware. The source code of the game is very simple to understand. We have used comments in each line of code to help you understand each of them better. After testing and debugging, the project will be made available on various open source project discussion websites as well as on the largest home for open source projects GitHub, where the beginner and enthusiastic coders could study this project for learning purposes and various developers could access the code and further improve it.

**TABLE OF CONTENT**

**S.No. Contents Page No**

1. **Introduction 1**
   1. History 2
   2. Requirement Analysis 3
   3. Main Objective 4
   4. Sub Objectives 4
   5. Pert Chart Legend 5
2. **System Analysis 6**
   1. Existing System 6
   2. Motivations 7
   3. Proposed System 7
   4. Modules
3. **Design 10**
   1. Modelling in game 10

3.1.2. Use Case model for Requirement Analysis 11

3.1.3. The design model 12

3.1.4. State Transition 14

3.1.5. Activity Diagram 17

**4. Game**

**HISTORY**

Platform games were the most popular genre of video game. At the peak of their popularity, it is estimated that between one-quarter and one-third of console games were platformers. No genre either before or since has been able to achieve a similar market share. As of 2006, the genre had become far less dominant, representing a two percent market share as compared to fifteen percent in 1998.

We intend to bring back the nostalgia and glory of Platform games. Our Game will incorporate Hardware Acceleration for smooth and fluid Gameplay. We will use texture rendering which provides fast, flexible hardware based rendering. We will use SDL Library which is lightweight with minimal information hiding. It takes care of all the low-level nuts and bolts across various platforms, but leaves most of the heavy lifting to us so that we can just focus on Game Development.

**REQUIREMENT ANALYSIS**

HARDWARE REQUIREMENTS:

* Processor: 500 MHz
* Video Card: 64 MB OpenGL Compatible 3D Accelerated Graphics Card
* RAM: 256MB
* 800x600 Screen Resolution
* 16 Bit Color Display
* Keyboard + Mouse

SOFTWARE REQUREMENTS:

* Operating System: - Windows XP or onwards or Linux**.**
* Code Blocks IDE with SDL library linked.
* Inkscape for development of assets.

**OBJECTIVES**

MAIN OBJECTIVE:

* The first and the foremost objective is to design and create 2D cross platform game with a very Fluid Gameplay for the better experience of the user.

SUB OBJECTIVES:

* Providing hardware accelerated textures based 2D rendering.
* Testing and Debugging.
* Create a repository of the project GitHub.

**PERT CHART**

|  |  |  |  |
| --- | --- | --- | --- |
| Topic | Start Date | End Date | Duration |
| Understanding the concepts of graphics library, and implementation of search algorithms. | 11-08-2016 | 25-08-2016 | 2 Weeks |
| Developing the code. | 25-08-2016 | 06-10-2016 | 6 Weeks |
| Merging the graphics with Array concepts | 06-10-2016 | 27-10-2016 | 3 Weeks |
| Testing phase | 27-10-2016 | 10-11-2016 | 2 Weeks |

Designing a Basic Structure

Requirement Analysis and Resource Allocation

Background Study and Detailed Analysis

Coding and Testing

Analysis of Algorithm

Comparative Analysis

**SYSTEM ANALYSIS**

**EXISTING SYSTEM**

The system that the users use before the management decides to upgrade it is known as the existing system or “legacy” system.

* In the context of the project the existing system is regarding a 2D platform game.
* The game involves a ball and bombs in which we have to dodge the bombs to score higher.
* The existing game system provides a time lapse for dodging the bombs and the score achieved is based on the more the bombs are dodged.

It uses Inkscape as a vector graphics editor to achieve graphical images and objects manipulation.

**MOTIVATIONS**

In this particular game, there were several motivational factors responsible for game development. Featuring one of these factors was an urge to do programming in C Language. Programming practice is our passion and dedication. The current generation is most interested in playing or watching games. But on playing games we get into the depths and get curious to know about its design and development. Playing games is one of the best ways to gather our interest in programming practices for game development.This was the major factor that motivated us to use this approach.

**PROPOSED SYSTEM**

SET UP AND SHUT DOWN

Every operating system makes the programmer write some amount of code to find out what the OS will let you do and what the hardware can do. Then, you have to write more code to actually set things up the way you want them. You have to set the graphics modes, initialize the sound system check for a mouse and a joystick, and on and on.

EVENT BASED INPUT

SDL provides input from the keyboard, mouse, and joystick, using an event based model that will be familiar to every programmer who has written code under X11, Windows, or Mac OS. The difference is that SDL will give you the same events no matter which OS you are using and SDL doesn't make you deal with all the OS specific events that just complicate the job of writing a game.

TIME AND TIMERS

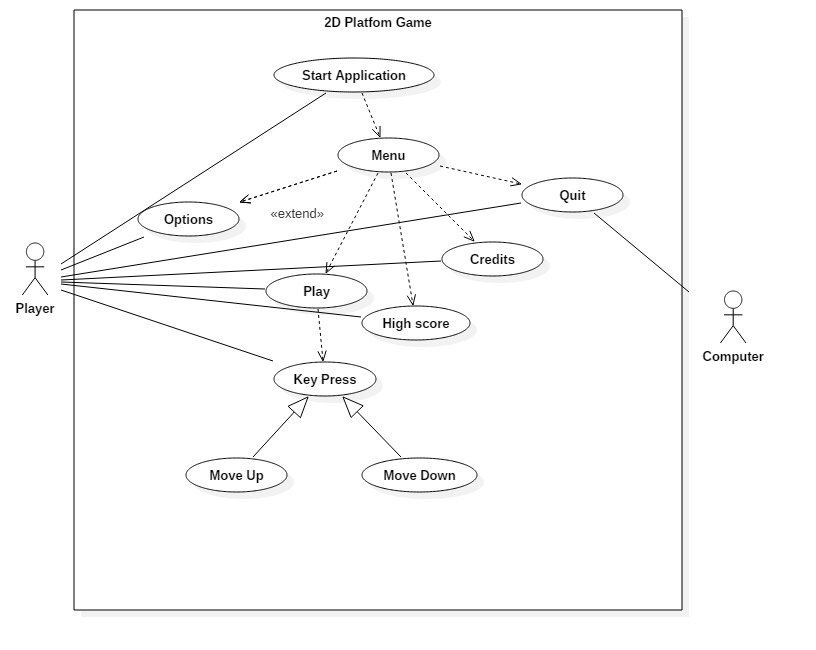
Every game program needs to know what time it is and needs to be notified when an action needs to be performed. SDL provides a clean, simple, and reliable time and timer API that is both machine and OS independent. The SDL timer APIs allow the creation of thousands of timers and can be used without the worries and limitations of other timer APIs I've used in the past.

GRAPHICS

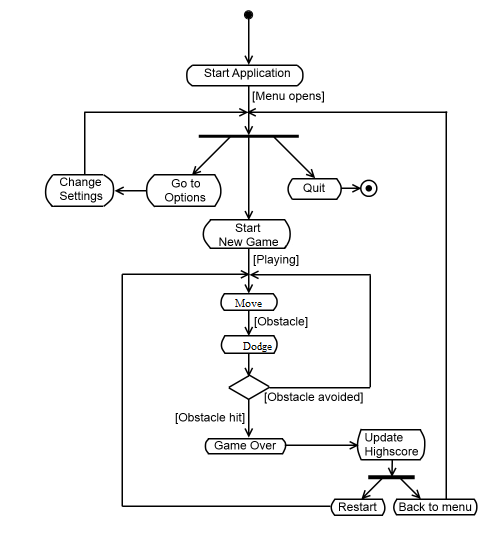
SDL gives you a choice of how you want to do graphics. You can get at the raw pixels and do pretty much anything you want with them, or you can use OpenGL to do hardware accelerated 2D and 3D graphics. The first approach works on every device out there, and since OpenGL is available for all 3D video cards it is a safe and portable way of doing high performance 3D graphics in a device and OS independent way. SDL takes care of setting the video modes, getting access to the frame buffer, if you use it, initializing OpenGL for you. SDL takes the pain out of setting up full screen graphics modes so commonly used in games, but it can set up windowed games too.

**DESIGN**

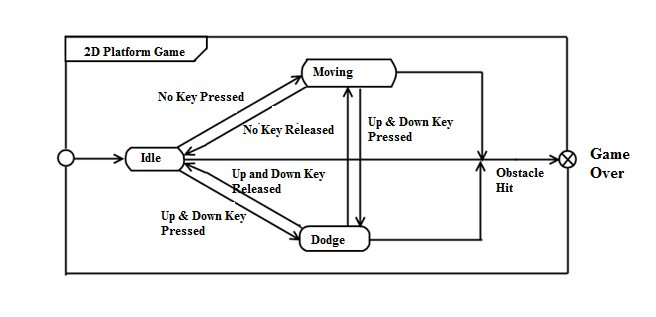
**USE CASE MODEL**



**ACTIVITY MODEL**



**STATE ACTIVITY**



**THEORY**

**Code:**

**#include <stdio.h>**

**#include <conio.h>**

**#include <graphics.h>**

**#include <dos.h>**

**#include <stdlib.h>**

**int c[10],v[10]; //bombs coordinates**

**int x,y; //ball coordinates**

**int a=1,b=0; //a=no. of bombs,b=counter**

**char buf[80];**

**int score=0; //score variable**

**void set\_coordinates(); //to set the bomb coordinates.**

**void ask(); //To implement Menu-driven programming.**

**void game\_over(); //Executes after collision of bombs and ball**

**void collide(); //Checks for collision between ball and bombs**

**void greet(); //Welcome Screen**

**void display\_bombs(); //Display bombs**

**void display\_ball(); //Display ball**

**void set\_bombs(); //Set bomb coordinates**

**void game(); //Game code**

**int main()**

**{**

**int gd = DETECT, gm;**

**initgraph(&gd, &gm, "C:\\TurboC3\\BGI");**

**setbkcolor(LIGHTGRAY);**

**greet();**

**ask();**

**closegraph();**

**return 0;**

**}**

**void display\_score()**

**{**

**char buf[80];**

**sprintf(buf,"Score=%d",score);**

**outtextxy(getmaxx()-50,getmaxy(),buf);**

**}**

**void game()**

**{**

**int col=0;**

**int i, flag=0,speed=30;**

**char ch;**

**char buf1[80];**

**x = getmaxx()/2;**

**y = getmaxy()/2;**

**while (1)**

**{**

**if(kbhit())**

**{**

**ch=getch();**

**if(ch==72) //move upwards**

**{**

**if(y<30)**

**{**

**}**

**else**

**{**

**y=y-10;**

**}**

**}**

**if(ch==80) //move downwards**

**{**

**if(y>getmaxy()-30)**

**{**

**}**

**else**

**{**

**y=y+10;**

**}**

**}**

**if(ch==27)**

**exit(0);**

**}**

**if(x >= getmaxx()-30 || x <= 30)**

**{**

**flag = !flag;**

**if(col>0)**

**collide();**

**col++;**

**set\_bombs();**

**score++;**

**}**

**display\_ball();**

**display\_bombs();**

**display\_score();**

**delay(speed);**

**cleardevice();**

**if(speed>10)**

**speed=speed-3;**

**if(flag){**

**x = x - 5;**

**}**

**else**

**{**

**x = x + 5;**

**}**

**}**

**}**

**void game\_over()**

**{**

**sprintf(buf,"Score=%d",score);**

**outtextxy(200,100,"GAME OVER");**

**outtextxy(200,150,buf);**

**outtextxy(200,200,"HIGH SCORE: 49");**

**outtextxy(200,250,"WELL PLAYED:D");**

**delay(2000);**

**exit(0);**

**}**

**void collide()**

**{**

**int i;**

**for(i=0;i<a;i++)**

**{**

**if(y<v[i])**

**{**

**if((v[i]-19)<(y+23))**

**{**

**clrscr();**

**game\_over();**

**}**

**}**

**else**

**{**

**if((v[i]+19)>(y-23))**

**{**

**clrscr();**

**game\_over();**

**}**

**}**

**}**

**}**

**void greet()**

**{**

**int i;**

**settextstyle(0,HORIZ\_DIR,4);**

**setcolor(RED);**

**outtextxy(150,60,"WELCOME TO");**

**outtextxy(80,200,"DODGE THE BOMBS");**

**delay(3000);**

**cleardevice();**

**}**

**void ask()**

**{**

**int i;**

**char cho;**

**settextstyle(0,HORIZ\_DIR,2);**

**setcolor(RED);**

**outtextxy(50,20,"DODGE THE BOMBS");**

**outtextxy(50,50,"1. PLAY GAME");**

**outtextxy(50,100,"2. HOW TO PLAY");**

**outtextxy(50,150,"3. CREDITS");**

**outtextxy(50,200,"4. EXIT");**

**cho=getch();**

**if(cho==49)**

**{**

**game();**

**}**

**else if(cho==50)**

**{**

**clrscr();**

**settextstyle(0,HORIZ\_DIR,1);**

**outtextxy(50,20,"HOW TO PLAY?");**

**outtextxy(50,50, "1. USE ARROW KEYS UP AND DOWN TO MOVE THE BALL.");**

**outtextxy(50,100,"2. TRY TO AVOID THE BOMBS AS MUCH AS POSSIBLE.");**

**outtextxy(50,150,"3. THE MORE YOU DODGE, HIGHER YOU SCORE.");**

**delay(5000);**

**clrscr();**

**ask();**

**}**

**else if(cho==51)**

**{**

**clrscr();**

**outtextxy(50,20,"Made by:");**

**outtextxy(50,50, "Shikhar Shukla");**

**outtextxy(50,100, "Siddhant Kapoor");**

**outtextxy(50,150, "Siddhant Srivastava");**

**outtextxy(300,20,"Mentor:");**

**outtextxy(300,50,"Ms. Richa Choudhary");**

**settextstyle(0,HORIZ\_DIR,4);**

**outtextxy(getmaxx()/2,350,"Thank You");**

**delay(3000);**

**clrscr();**

**ask();**

**}**

**else if(cho==27 || cho==52)**

**exit(0);**

**else**

**{**

**clrscr();**

**outtextxy(225,getmaxy()/2,"Wrong Input");**

**delay(2500);**

**cleardevice();**

**ask();**

**}**

**}**

**void display\_ball()**

**{**

**setcolor(RED);**

**setfillstyle(SOLID\_FILL, RED);**

**circle(x, y, 25);**

**floodfill(x, y, RED);**

**}**

**void display\_bombs()**

**{**

**int i;**

**setcolor(BLUE);**

**setfillstyle(SLASH\_FILL,BLUE);**

**for(i=0;i<a;i++)**

**{**

**circle(c[i],v[i],20);**

**floodfill(c[i],v[i],BLUE);**

**}**

**}**

**void set\_bombs()**

**{**

**if(a<=8)**

**{**

**if(b==0)**

**{**

**a++;**

**b++;**

**}**

**else if(b>0)**

**{**

**b=0;**

**}**

**}**

**set\_coordinates();**

**}**

**void set\_coordinates()**

**{**

**int i;**

**randomize();**

**for(i=0;i<a;i++)**

**{**

**if(x>=getmaxx()-30)**

**{**

**c[i]=30;**

**v[i]=random(getmaxy());**

**}**

**else if (x<=30)**

**{**

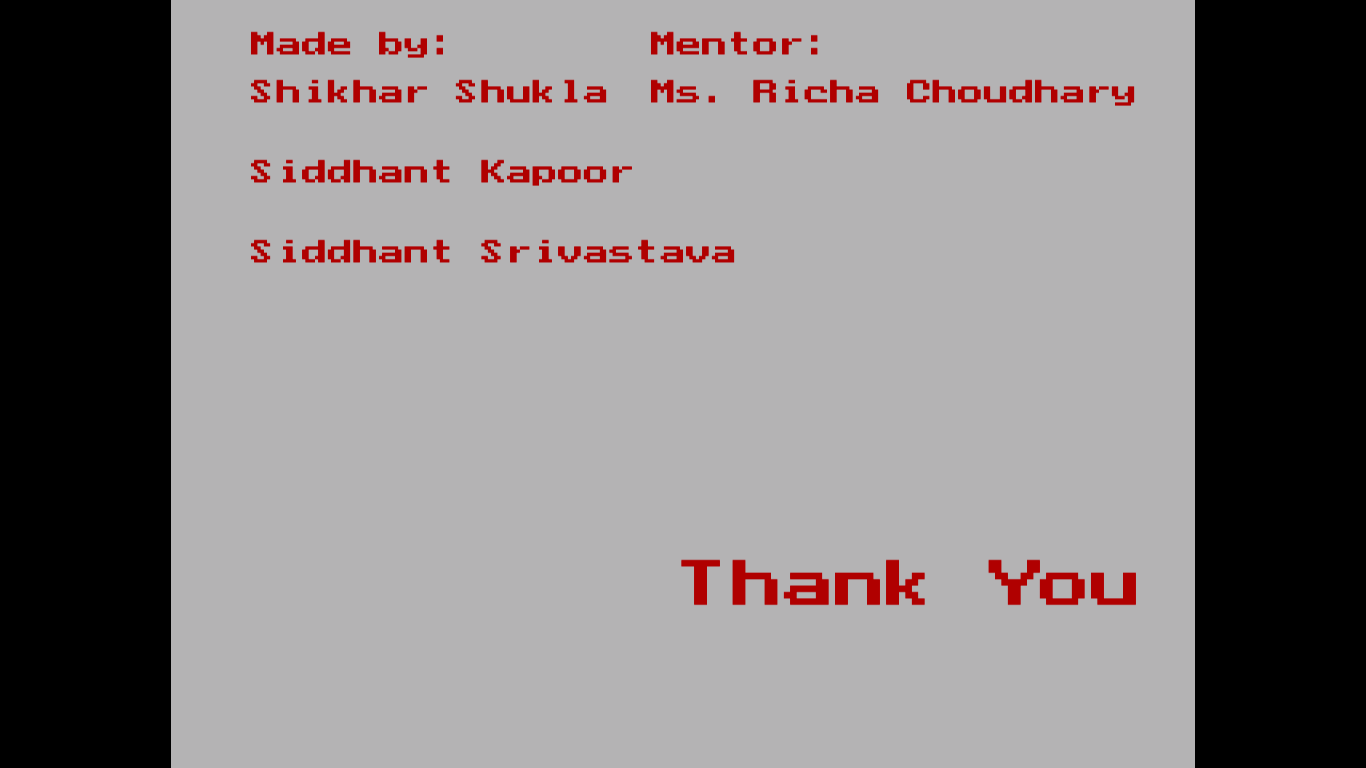
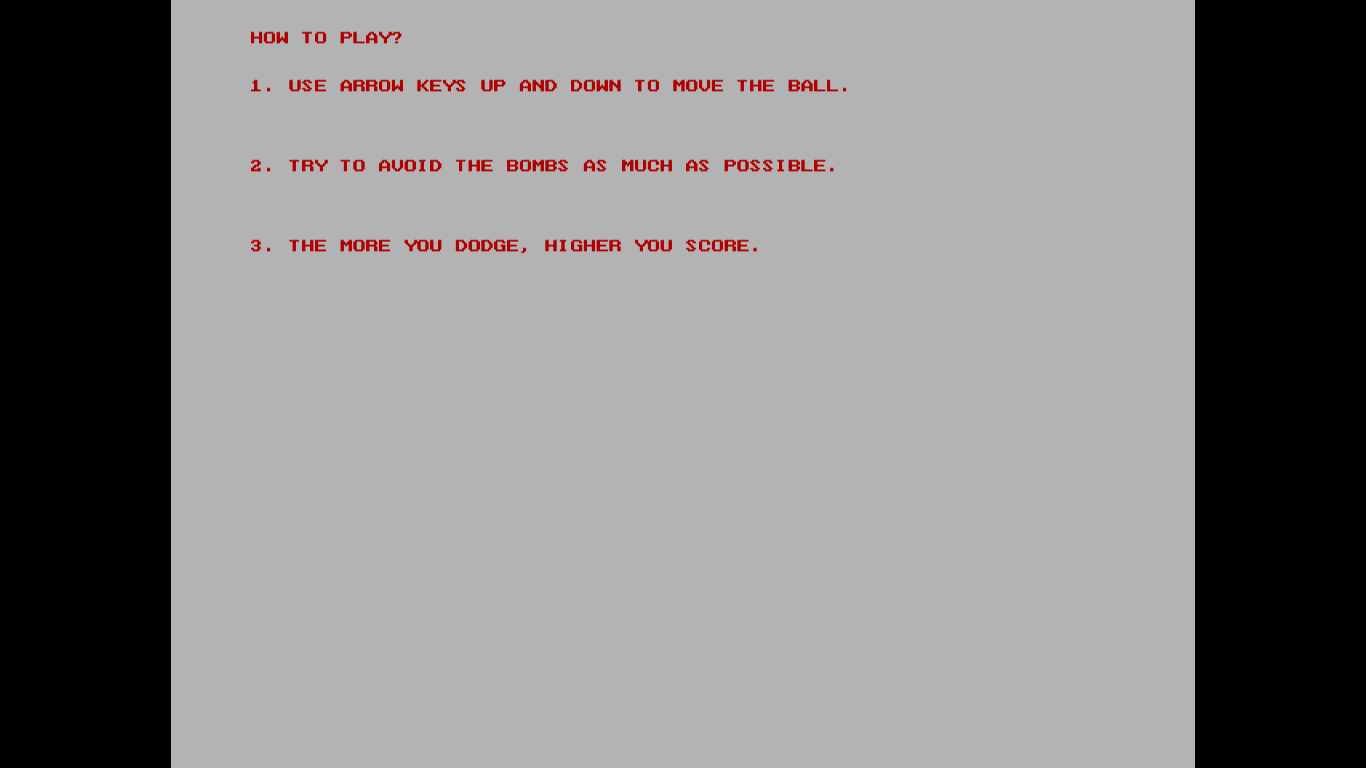
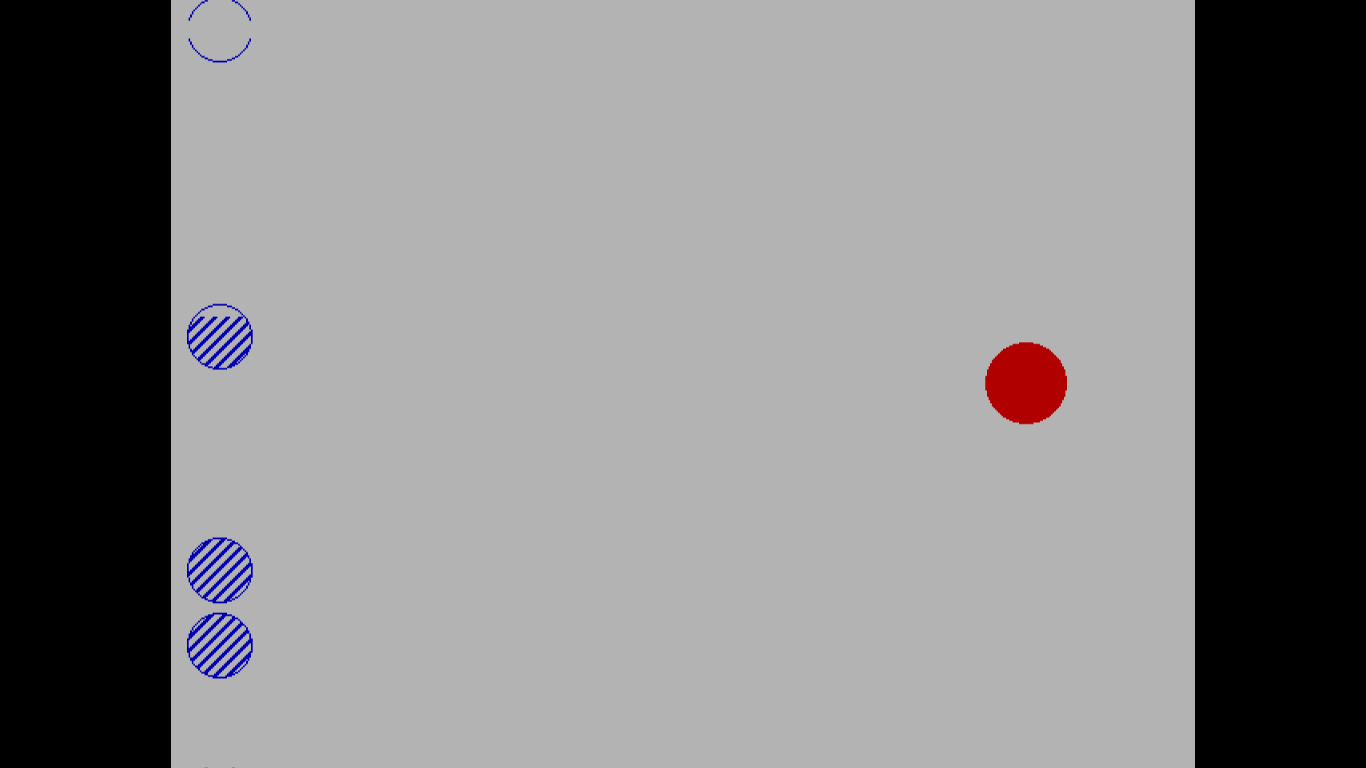
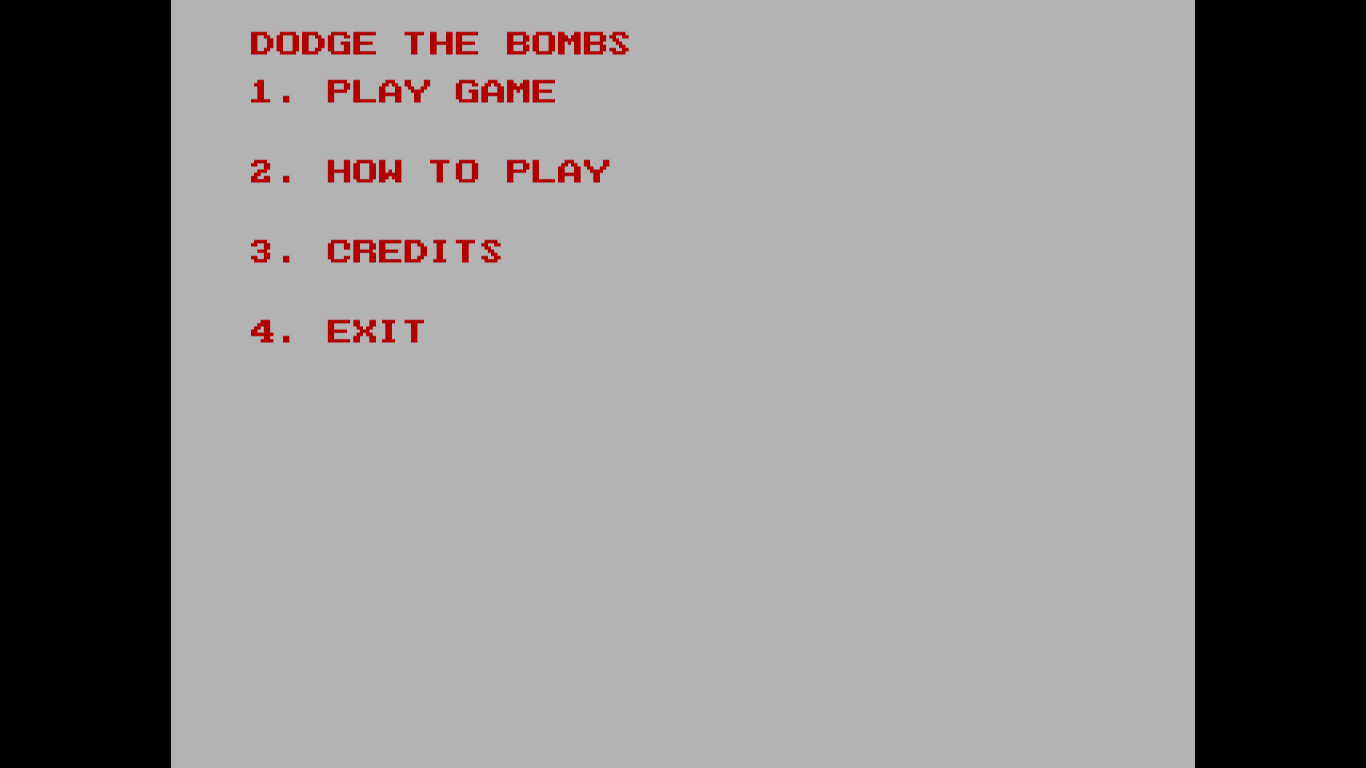
**c[i]=getmaxx()-30;**

**v[i]=random(getmaxy());**

**}**

**}**

**}**



**CONCLUSION**

Our conclusion is that we have successfully created a 2D platformer game using graphics library in C language.

The user can play the game by moving the object through an obstacle course. Hitting the Bombs will mean that the game is over.

Also the game is tested for various bugs.

Thus we have created a good game which will provide user with candid graphics of a 2D game.

**FUTURE ENHANCEMENTS**

* We will try to bring this game on mobile devices in future.

**RFERENCES**

1. How to write a game design: <http://www.pagtech.com/2006/07/04/how-to-Write-a-game-design-doc/>
2. Flappy Bird: [http://www.flappybird.com](http://www.flappybird.com/)
3. Guidelines:  
    [http://www.codeproject.com](http://www.codeproject.com/)
4. [http://www.codeincodeblock.com](http://www.codeincodeblock.com/)
5. C++ Game Programming tutorial:
6. <http://www.cppgameprogramming.com/cgi/nav.cgi?page=index>
7. Introduction to Algorithms, Second Edition, by Thomas H. Cormen, Charles E.  Leiserson, Ronald L. Rivest, Clifford Stei

MIT PRESS, 2001, ISBN 81­203­2141­3