POINT BLANK - 2 PLAYER SHOOT GAME

REVIEW REPORT

Submitted by

Mohd Umar (18BCE0196)

G. Srinivasa Rao (19BCE0119)

Vedant Vikas Naik (19BCE0797)

Penke Lohith Sasi Anvesh (19BCE2069)

Prepared For

MICROPROCESSOR AND INTERFACING (CSE2006) – PROJECT COMPONENT

Submitted To

Prof. Antony Xavier Glittas X

School of Computer Science and Engineering

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

1. CERTIFICATE

This is to certify that the project work entitled "**Point Blank-2 Shooting Game**" that is being submitted by us for **Microprocessor and Interfacing (CSE2006)** is a record of bonafide work done under my supervision. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted for any other CALCourse.

Place: Vellore

Date: 202`1

Signature of Students

Mohd Umar	**(18BCE0196)**
G. Srinivasa Rao	**(19BCE0119)**
Vedant Vikas Naik	**(19BCE0797)**
Penke Lohith Sasi Anvesh	**(19BCE2069)**

2. ACKNOWLEDGEMENT

We take immense pleasure in thanking **Dr. G. Viswanathan**, our beloved Chancellor, VIT University and respected Dean, Dr. R. Saravanan, for having permitted us to carry out the project.

We express gratitude to our guide, **Prof. Antony Xavier Glittas** , for guidance and suggestions that helped us to complete the project on time. Words are inadequate to express our gratitude to the faculty and staff members who encouraged and supported us during the project. Finally, we would like to thank our ever-loving parents for their blessings and our friends for their timely help and support.

Signature of Students

Mohd Umar	**(18BCE0196)**
G. Srinivasa Rao	**(19BCE0119)**
Vedant Vikas Naik	**(19BCE0797)**
Penke Lohith Sasi Anvesh	**(19BCE2069)**

Contents

3. ABSTRACT

Computer games have always been seen in awe in all stages of the computing world, it has its roots emerging from the developmental aftermath of Artificial Intelligence research, and since then digital games have taken over globally and as the processors become more and more powerful the gaming industry is in a boom as well. Hence with this project we wanted to understand how this vast and attractive industry came to be existence and try to unravel its roots.

With this project we tend to understand the working of the first ever interactive games created by the Point Blank-2 Player Shoot Game. We are also trying to understand the maximum potential in which the 8086 microprocessor could be used and what better way could be the development of games as it exhausts all possibilities for any processors as they are an enigma in itself an epitome of testimony to testify the strength and tenure of any microprocessor.

4. INTRODUCTION

This paper provides an algorithm for the realization of the popular computer games incorporated in a Gaming Console . The basic idea of this work was the development of a gaming console using DOSBox programming environment and libraries used for easier implementation of the game in assembly programming language. Assembler (assembly) language is a low-level programming language, and it is specific to a particular processor (microcontroller) architecture . As the oldest programming language, and of all languages, it bears the closest resemblance to a native machine language. It provides direct access to computer hardware, requiring you to understand much about your computer's architecture and operating system . A large number of modern microcontrollers are based on a similar architecture as the microcontroller 8086 and have similar abbreviations for the names of assembly instructions. So, the study of microcontroller 8086 is reckoned not only for pedagogical reasons but also for usability .

Through numerous examples of research being studied through lectures, students are led to a correct conclusion that when designing microcontrollers it is necessary to know the algorithm which the hardware will use and that it is necessary to know the hardware on which the software will be executed to write an optimal algorithm. So, both parts of the design should be projected in parallel. Today's software for digital signal processing is based on some high-level programming language, but optimization of code execution is achieved by writing individual functions in assembly . We can say that the assembly is used when we need fast and efficient processing of the signal. The MASM library makes programming in assembly easier and provides libraries for work with strings and support for DOSBox features. Some of the main supporting features are graphical environment functions. In this paper, we will describe designing a game box using the x86 architecture with MASM libraries in the programming environment DOSBox.

5. OBJECTIVE

The objective of this project is to create a popular multiplayer arcade game using EMU8086. The arcade game implemented in this project is a successful compilation of multiple electronic games, which are called Point Blank-2 Player Shoot game, these were some of the first digital games ever created. The main objective of a shooting game is to strike your opponent and score will be updated with every hit you make. The player surviving at the end would be declared as the winner.

Apart from these this project also has aim to:-

- To further understand the interrupt 10h , 13h and 16h.
- To understand and make the subject more dynamic and interesting.
- To understand the implementation of image processing using early 8086 Assembly and showing the power of the assembly language and why it is important till dat

7. COMPONENTS INVOLVED

Hardware Requirements are as follows:-

- Operating System: Windows XP and above
- Processor: Pentium 4 or higher
- RAM: 128 MB or more
- Hard Drive: 1 GB or more

Software Requirements are as follows:-

- Assembler - MASM
- DOS-Box

7.1 COMPONENT DESCRIPTION

MASM

MASM - is the Microsoft Macro Assembler. It is an assembler. It takes your code, pre-processes it and converts it to binary. The links it to a runnable executable or an Object file.
All Intel processors 32bit and em64t processors (80386 and up) support the 8086 compatibility mode called "real mode". Which means that all PCs to this day are backward compatible with say MS-DOS and all the games that used to run on IBM XT. Those will run on modern machines but really fast so those will be unplayable :-)
All PCs to this day are booting with their processor in the real mode and modern operating systems switch the processor to the 32bit/64bit "protected mode".
Basically what happens in real mode is that the CPU knows it is working as 8086. E.g : all operations are on 16bit registers and the memory is addressed by a segment: offset pair. The memory addresses are physical memory addresses and you have access to the first 1MB of RAM. Physical address is calculated by segment shifted left by 4 bits + offset. Thus 8000h:100h is the same address as 8010h:0h , physical address is 80100h in the memory.

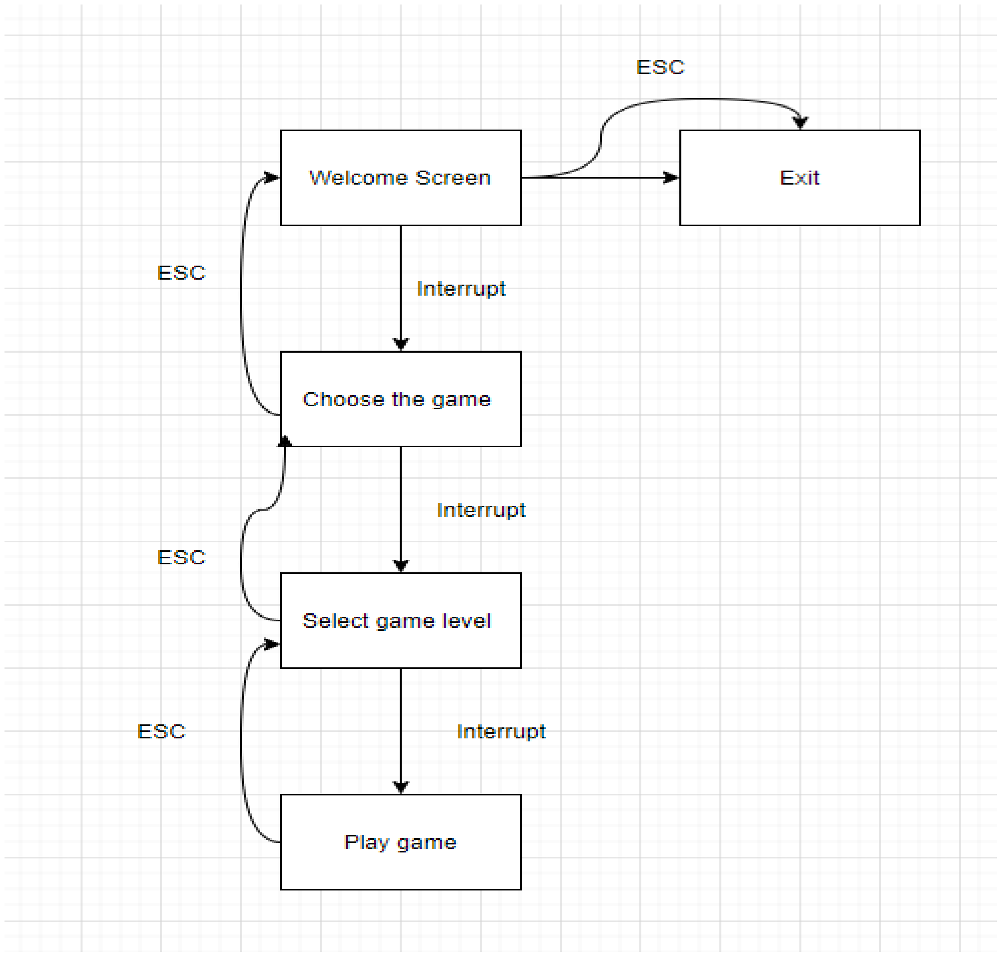
MS DOS-Box

It is a free open-source emulator. The main application of this software is designing games using ALP and running those games (playing) on this emulator. The software allows the players to take screenshots of the game and even record gameplay footage, which is almost the replica of the actual gaming program.
MS DOS-Box is a command-line program which can be configured by a set of command-line arguments or by editing text configuration files. The software allows us to use in-built commands and libraries and import commands to make assembly language programming easier.

8. WORKING AND IMPLEMENTATION

The games we made are run using the DOS-Box solely due to the fact that modern processors work as a 64 bit or a 32 bit opcode or instruction but the 8086 known for it's 16 bit address bus requires an external application which could virtually allow a 64-bit machine to allow use of 16 bit interrupts. Hence we will be using DOS-Box as the external factor.

8.1 *FLOW CHART FOR WORK FLOW*



8.2 WORKING WITH MS DOS-BOX

STEP-1: DOWNLOAD MS DOS- BOX:-

DOSBox is open source and free.

Get it here: http://www.dosbox.com/download.php?main=1 and use the installer.

STEP -2: CREATING THE GAME FOLDER

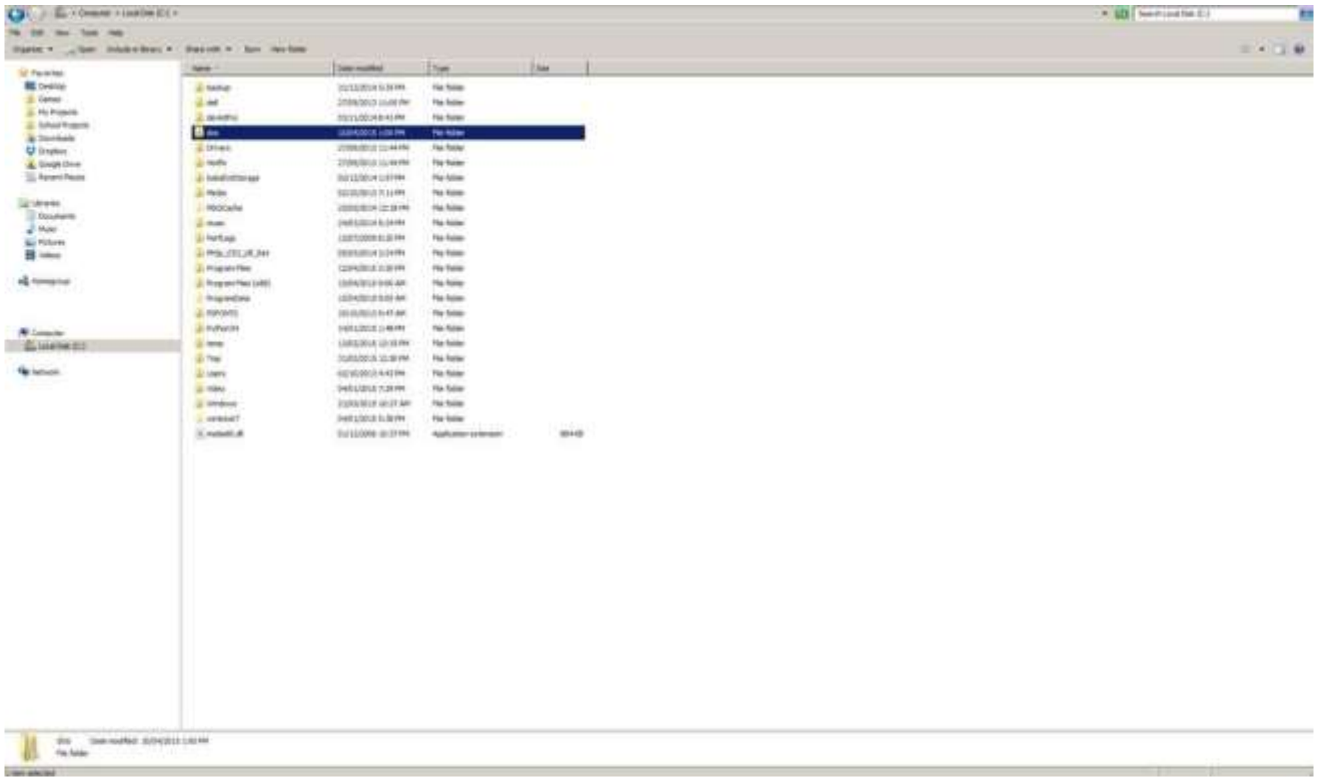
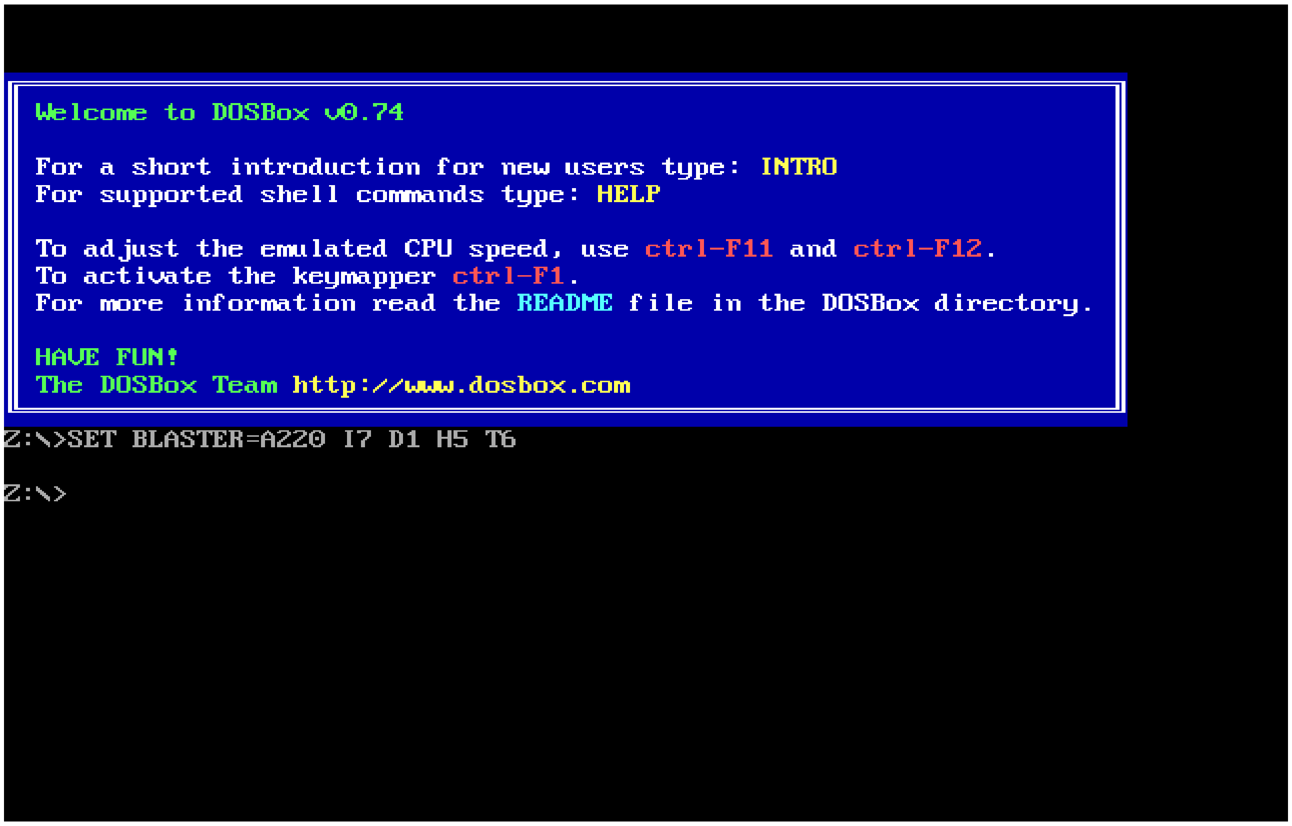


Figure 1Go to the root of your main hard drive(Mine is C:\) and create a folder. Here, in our case we named it "dos"

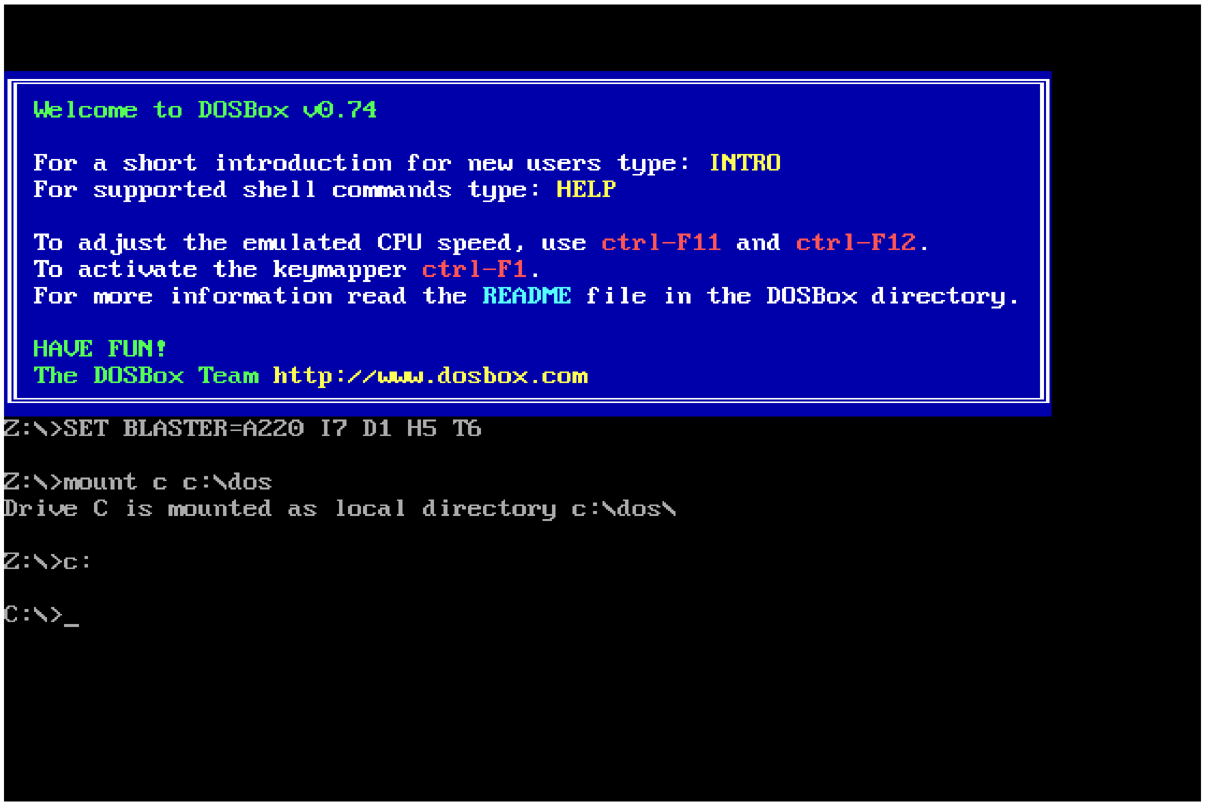
STEP-3: START DOS-BOX EMULATOR



```
Welcome to DOSBox v0.74

For a short introduction for new users type: INTRO
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>
```

Figure 2 Starting DOSBox isn't really that hard. All you need to do is click the icon. When DOSBox is started, a window will pop up. Press Alt +Enter at any time to make it full screen or exit full screen

STEP-4: MOUNT THE C:/ DOS DIRECTORY

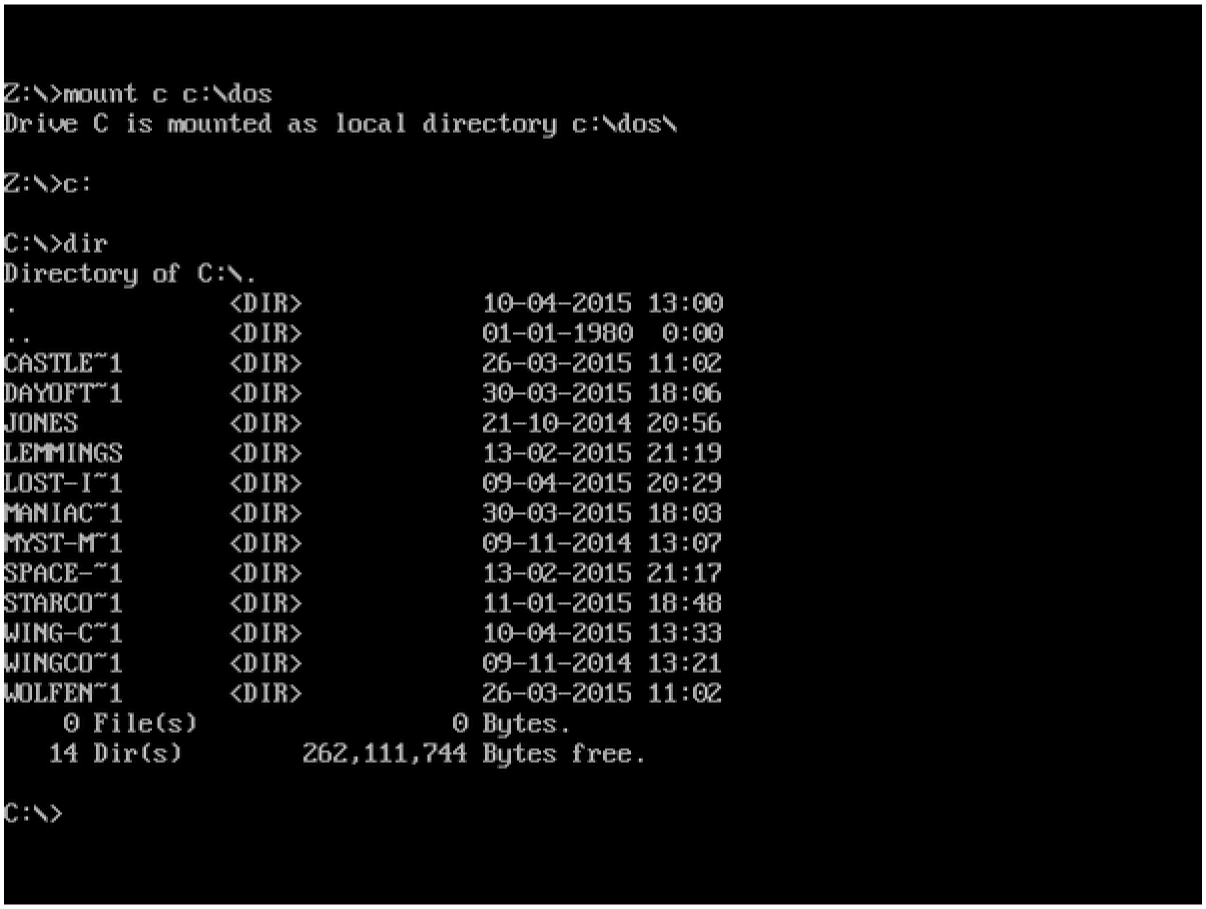


```
Welcome to DOSBox v0.74

For a short introduction for new users type: INTRO
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c c:\dos
Drive C is mounted as local directory c:\dos\

Z:\>c:

C:\>_
```

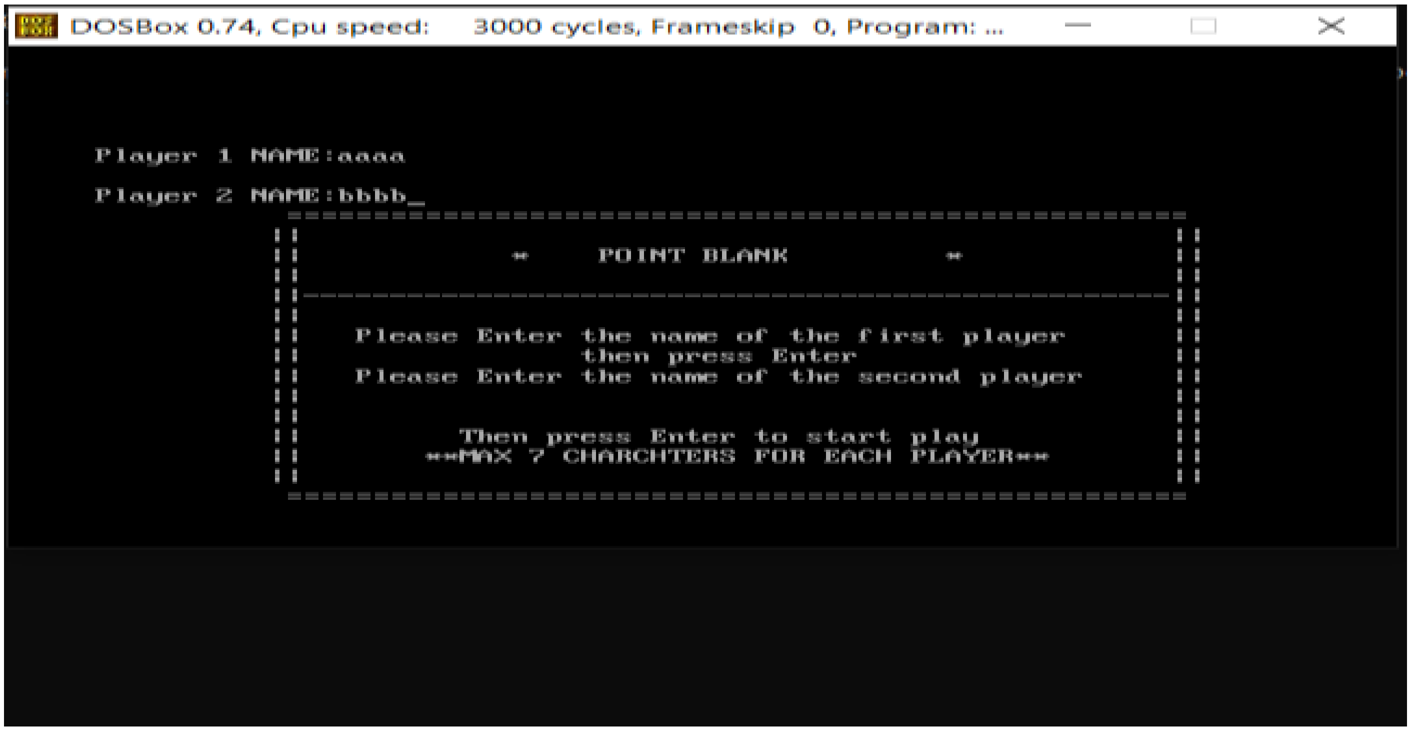
Figure 3 Type this into DOSBox: mount c c:\dos

Then press Enter or Return. It will display the message "Drive C is mounted as a local directory c:\dos\." After that, type c: and hit Enter to enter the c:\dos directory.

STEP-5: ENTER THE DIRECTORY CONTAINING THE GAME

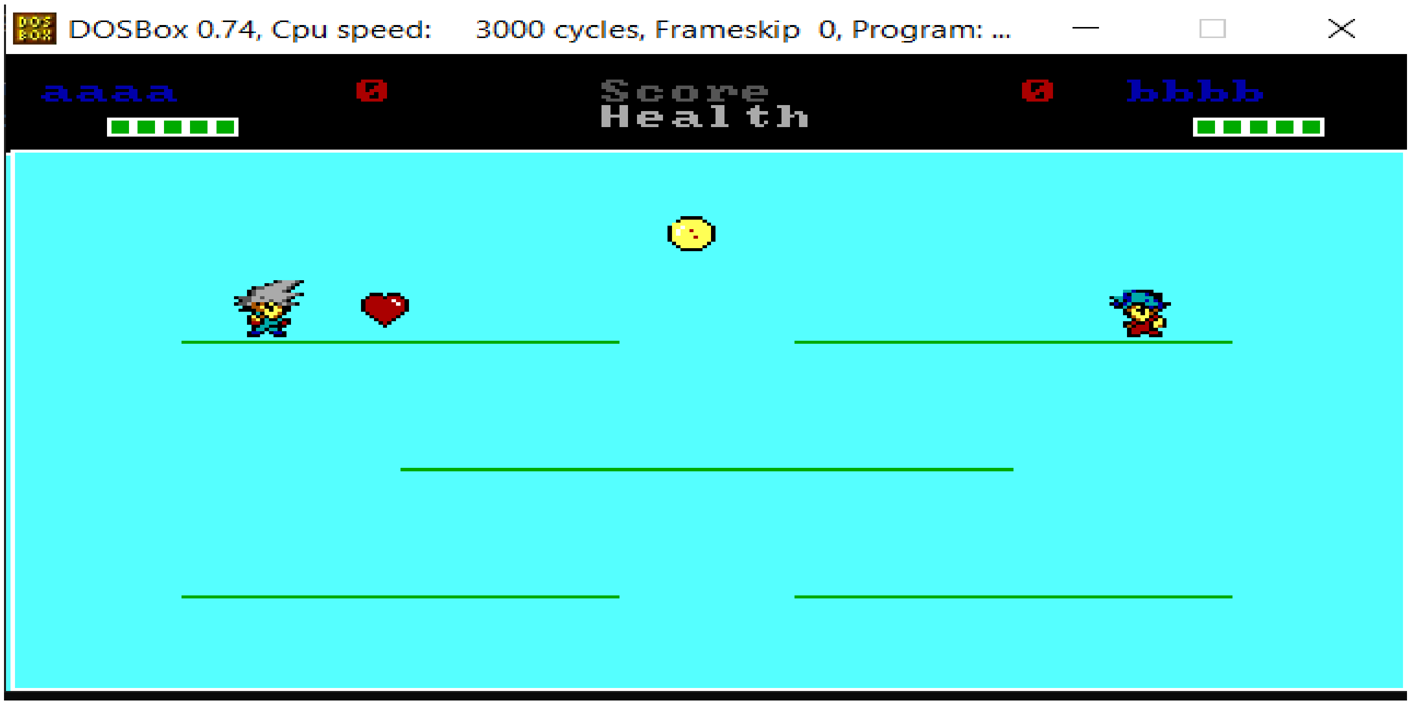


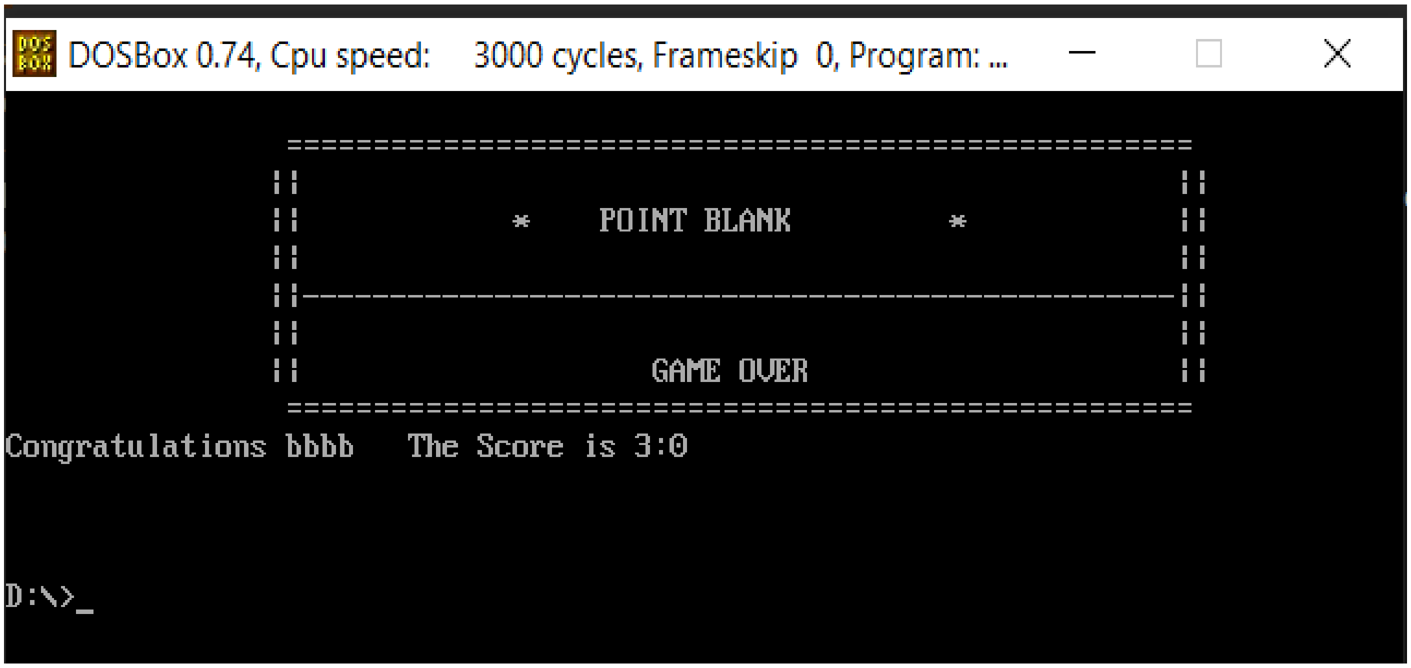
```
Z:\>mount c c:\dos
Drive C is mounted as local directory c:\dos\

Z:\>c:

C:\>dir
Directory of C:\.
.              <DIR>            10-04-2015 13:00
..             <DIR>            01-01-1980  0:00
CASTLE~1       <DIR>            26-03-2015 11:02
DAYOFT~1       <DIR>            30-03-2015 18:06
JONES          <DIR>            21-10-2014 20:56
LEMMINGS       <DIR>            13-02-2015 21:19
LOST-I~1       <DIR>            09-04-2015 20:29
MANIAC~1       <DIR>            30-03-2015 18:03
MYST-M~1       <DIR>            09-11-2014 13:07
SPACE-~1       <DIR>            13-02-2015 21:17
STARCO~1       <DIR>            11-01-2015 18:48
WING-C~1       <DIR>            10-04-2015 13:33
WINGCO~1       <DIR>            09-11-2014 13:21
WOLFEN~1       <DIR>            26-03-2015 11:02
    0 File(s)                 0 Bytes.
   14 Dir(s)        262,111,744 Bytes free.

C:\>
```



DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ...

```
Player 1 NAME:aaaa

Player 2 NAME:bbbb_
            ================================================
           ||                                              ||
           ||            *     POINT BLANK        *        ||
           ||                                              ||
           ||----------------------------------------------||
           ||                                              ||
           ||   Please Enter the name of the first player  ||
           ||              then press Enter                ||
           ||   Please Enter the name of the second player ||
           ||                                              ||
           ||                                              ||
           ||         Then press Enter to start play       ||
           ||     **MAX 7 CHARCHTERS FOR EACH PLAYER**     ||
           ||                                              ||
            ================================================
```


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ...
aaaa
0
Score
Health
0
bbbb


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ...

```
    ============================================================
   ||                                                          ||
   ||              *     POINT BLANK          *                ||
   ||                                                          ||
   ||----------------------------------------------------------||
   ||                                                          ||
   ||                        GAME OVER                         ||
    ============================================================
Congratulations bbbb   The Score is 3:0


D:\>_
```

11. ADVANTAGES AND DISADVANTAGES

11.1 ADVANTAGES

- Very useful to learn and understand the use of 8086 microprocessor interrupts
- Makes learning 8086 more interactive and interesting
- Makes the subject more fundamental and easy to understand while making the process more dynamic.
- Easy syntax for assembly language programming
- Simple design and easy to replicate
- Doesn't require many software to run and uses minimal memory
- Doesn't require additional hardware and can be run on almost any PC or laptop
- Smaller instruction set of 8086 microprocessor
- Gives an insight into how the gaming industry started

11.2 DISADVANTAGES

- Does not include complex features in the games
- The games don't have high-end graphics
- Not many applications besides learning the use of interrupts in ALP
- MS DOS-Box doesn't allow use of long filenames
- Not up-to-date with the current technology and hardware specifications
- Has only educational features and is not fundamentally an actual gaming box
- Does not include any hardware and is simply an interrupt and image-based project.

12. APPLICATIONS

- Learn the use of interrupts in assembly language programming
- Teaching ALP and practical microprocessor and interfacing classes to students
- Understanding code for systems with older processors that have limited high-level language options such as the Atari 2600, Commodore 64, and graphing calculators.
- Understand Programs that create vectorized functions for programs in higher-level languages such as C. In the higher-level language this is sometimes aided by compiler intrinsic functions which map directly to SIMD mnemonics, but nevertheless result in a one-to-one assembly conversion specific for the given vector processor.

13. CONCLUSION

we have to make some changes in the game play .We have to add the new movement feature for player. we make player can move in x-axis (Horizontally) and axis(Vertically).By this the game become more interesting .We can also make new powerups which give boosting to the player. The outcome of this project is to a full fledged 2 player game where in the both the users have separate buttons to control their respective characters. The score would then be tallied declaring any one of them as the winner of the game.