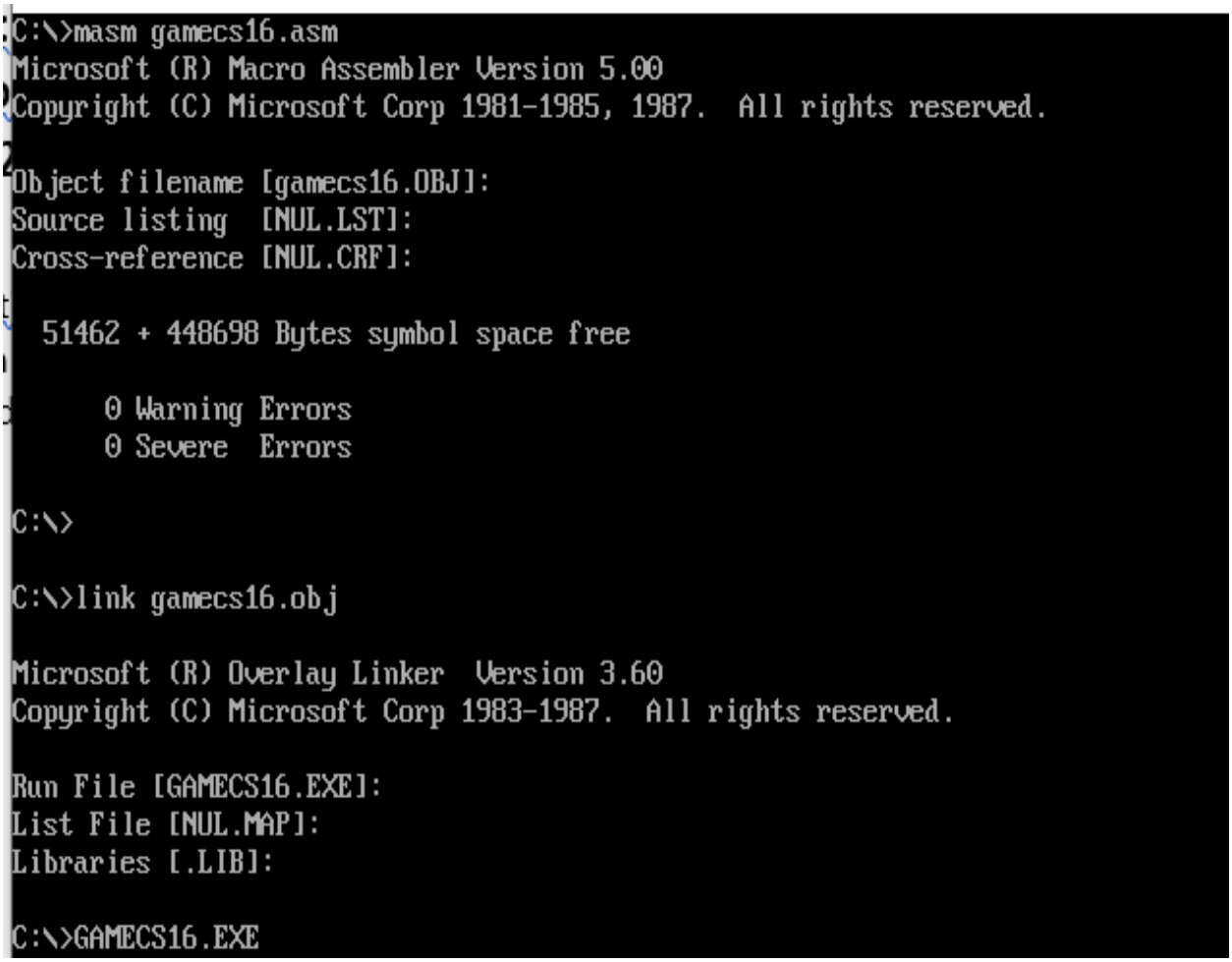
CS322 Computer Architecture Lab 4 Report

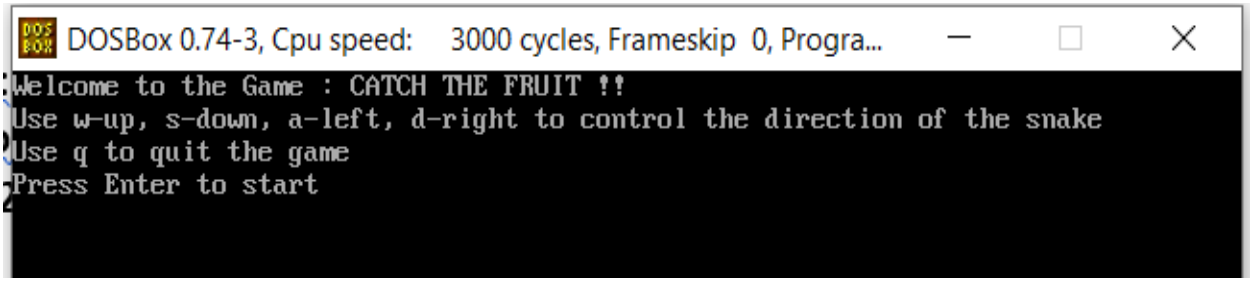
Name : Chandrawanshi Mangesh Shivaji
Roll No. : 1801CS16

Introduction : I have created a simple game that reads input from the keyboard and based on the given input and written code, the output is displayed on the screen. It is a snake game where initially we would have a snake of length 1 (+ Head) and its size will grow as it accumulates the fruit(denoted by P)

Instructions to start : Place the ASM file attached in the zip folder in the mount folder for the DOS BOX and follow the procedure in the following screenshot and press enter. You will see some messages on the screen.



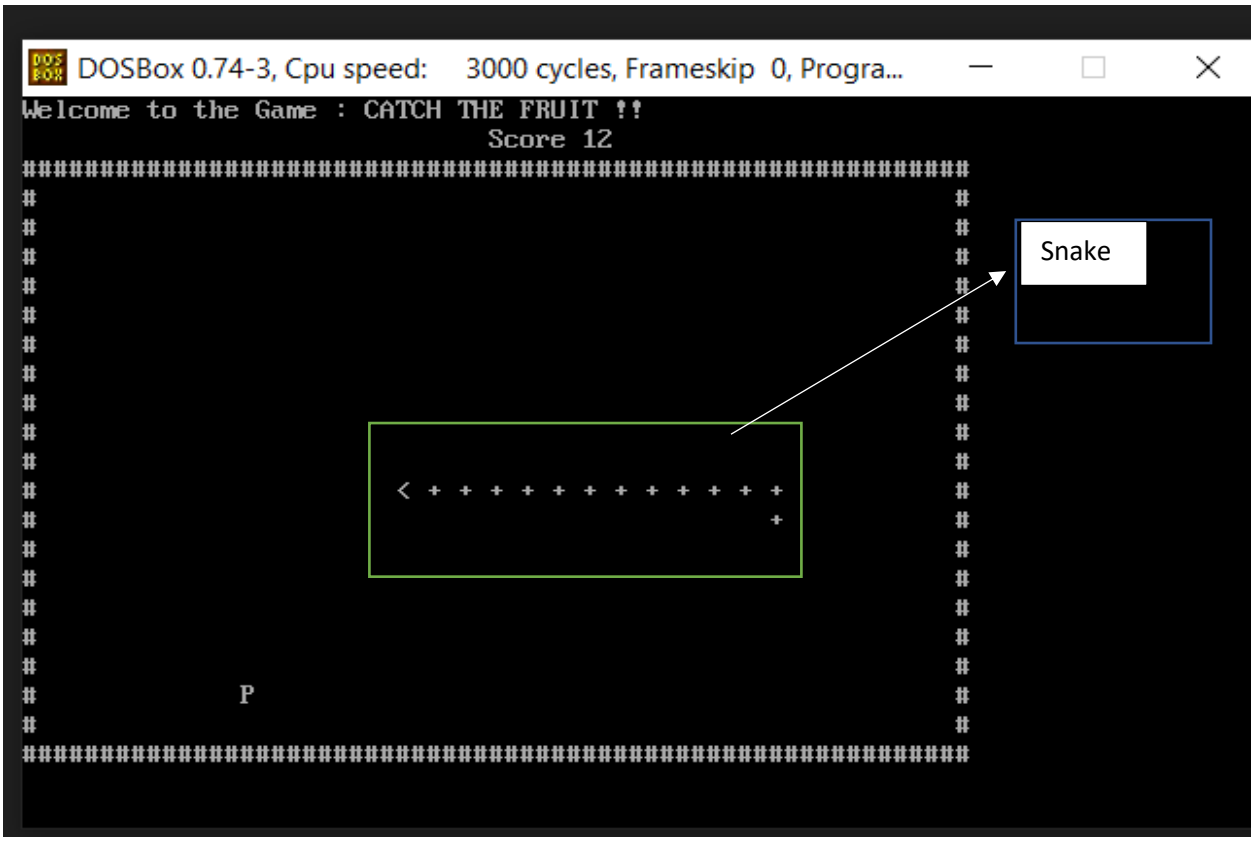
How to Play : So, this will pop up after pressing enter. Press Enter to start the game.



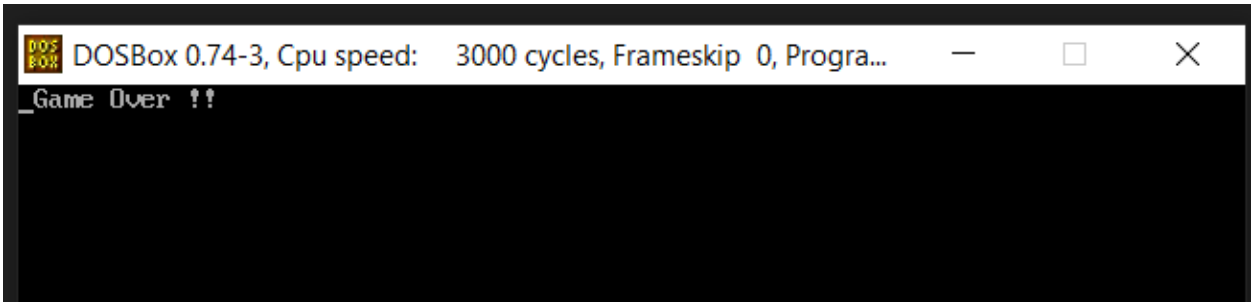
Controlling the Snake :

w – upward movement
s – downward movement
a – leftward movement
d – rightward movement

The Snake will automatically eat the Fruit when it touches it. Score will increment by one each such occurrence. Press q to quit the game at any time. If the Snake touches the Borders or Itself, it will die and game will finish.



Game Over : When You press q or your snake dies this screen instance will pop up



Overview of the Code :

Data : It contains all the variables declared which are to be used in Code Segment. Mainly it includes display messages, positions of the current fruit, length of the snake, Flags for ending the game, etc.

Code : It comprises of various functions including the MAIN. Here, are all the details of some majorly used functions in short

1) DISPLAY_STRING : To display string on the screen

2) WRITE_CHAR : To write a single character on screen at desired location

3) READ_CHAR : To read a single char from the screen

4) PRINT_BORDERS : To print borders of the box in which the snake will be constrained

5) DELAY : To manage the speed of the snake using Delaytime

6) SNAKE : To Determine the size and direction of the Snake

7) KEYBOARD_FUNCTIONS : read the input using READ_KEY function and accordingly act as per the pressed key

8) READ_KEY : To determine which key is pressed by user (0 if no input is provided)

9) CREATE_FRUIT : To generate the Fruit (Checks if it is already there at (Fruitx,Fruity) using READ_CHAR)

10) DRAW : To write current score (using segment count), draw the snake and the fruit

11) SET_CURSORPOS : To set cursor at required location (used in DRAW function)

12) DISPLAY_NUM and DISPLAY_DIG : To display score on the screen (used in DRAW function)

Some Other Important Keywords

MAINLOOP : Base Code Area for the whole Program

GAME_OVER_MAINLOOP : Code area to quit the game

EXIT_MAINLOOP : clear the screen and exit the game

Interrupts Used :

INT 10h / **AH = 0** - set video mode.
input: **AL** = desired video mode.
used in code: **03h** - text mode. 80x25. 16 colors. 8 pages.

INT 21h / **AH=9** - output of a string at **DS:DX**.
String must be terminated by '**$**'.

INT 21h / **AH=7** - character input without echo to AL.
if there is no character in the keyboard buffer, the function waits until any key is pressed.

INT 21h / **AH=4Ch** - return control to the operating system (stop program).

INT 1Ah / **AH = 00h** - get system time.

INT 21h / **AH=2** - write character to standard output.
entry: **DL** = character to write, after execution **AL = DL**.

INT 10h / **AH = 2** - set cursor position.
input:
DH = row.
DL = column.
BH = page number (0..7).

INT 16h / **AH = 01h** - check for keystroke in the keyboard buffer.
return:
ZF = 1 if keystroke is not available.
ZF = 0 if keystroke available.
AH = BIOS scan code.
AL = ASCII character.

INT 16h / **AH = 00h** - get keystroke from keyboard (no echo).
return:
AH = BIOS scan code.
AL = ASCII character.
(if a keystroke is present, it is removed from the keyboard buffer).

// EOF DATE : 21/09/2020