REQUIREMENTS:

- It requires the MS-DOS version 2.0 or greater

HEADER FILES USED:

❖ #include<graphics.h>

- ♦ initgraph(int*,int*,char*) – It initializes graphics window.
- ♦ settextstyle(int,int,int) – It sets the text font,direction in which text to be displayed and size of the text in graphics window.
- ♦ settextjustify() – It justifies the text in the graphics window.
- ♦ outtextxy(int,int,char*) – It prints the given text in given co-ordinates in graphics window.
- ♦ getmaxx() – It returns the maximum abscissa.
- ♦ getmaxy() – It returns the maximum ordinate.
- ♦ setcolor(int) – Make the succeeding outputs in the given colour.
- ♦ setfillstyle(int,int) – It sets the current fill colour and fill-pattern.
- ♦ pieslice(int,int,int,int,int) – It draws and fills a pie slice with the given radius.
- ♦ cleardevice() – It erases the entire graphics screen and moves the current position to (0,0).

- closegraph() – It closes the graphics window.

❖ #include<conio.h>

- getch () – Reads a character directly from the console without buffer, and without echo.
- getche () – Reads a character directly from the console without buffer, but with echo.
- kbhit() – It returns 1 if any key is pressed else returns 0.
- clrscr () – Clears the screen.
- gotoxy() – Set the current position ,in default c++ output scrren, to the given co-ordinates.

❖ #include<dos.h>

- delay(int) – It is used to suspend execution for a given time.
- sound(int) – It play a sound in the given frequency.
- nosound() – It stops the sound previously played by sound function.

❖ #include<string.h>

- strcpy(char*,char*) – It copys the second string to the second string.

❖ #include<stdlib.h>

- ♦ randomize() – It seeds rand function with the system time.
- ♦ exit(int) – It terminates the program..

❖ #include<stdio.h>

- ♦ sprintf(char*,…) – It writes the given string in the given buffer.

CLASS and OBJECT:

Class snake:

This is a child class of score class.

Class object: Jack is a array of objects of snake.Each element of jack is a snake of a player.The object is used to invoke the member functions of the class snake to create snake, control snake, etc.
This class implements encapsulation.This allows to integrate the properties of snake and use them as a single entity.

- Private Members:
  - ➢ int pos[2][50]
  - ➢ int size
  - ➢ int score
- Public Members:
  - ➢ char keys[4]
  - ➢ char in
  - ➢ void eating(int)

- void move()
- void display_snake(int,int)
- void init_obj(int)
- int out(int)
- int xye(int,int,int)
- void ai()
- void aii()
- void aik()
- void ail()
- void aij()
- int check(char)

Class score:

There is no objects for this class.It serves as a base class for snake class.This implements inheritance.

- Protected Members:
  - int s
  - int top_five[6]
  - int high_score
  - int total

- Public Members:
  - void reset_s()
  - void tot(int)
  - void change(int)

- int scor()
- void sort()
- int high_s()
- void highs_init()
- void h_a()
- void dis5()

FUNCTIONS USED:

- Member functions:

Snake class:

- void eating(int):

  This function checks if the snake co-ordinate is equal to food co-ordinate, if equal increases score,size and assigns random position to the food co-ordinate.

- void move():

  This function moves the snake in the current direction.

- void display_snake(int, int):

  This function display the snake, that is the displays its body parts in the co-ordinates from the position array.

- void init_obj(int):

  This function initializes the size, score, and head position of the snake.

- int xye(int, int, int):

  This function returns 1 if the given co-ordinates match the co-ordinates of the given part of the snake, else return 0.

- void ai():

  This function calls the corresponding ai function, determined by its current direction.

- void aii():

  This function will be called by the ai function, if the snake was moving up. It tries to match the co-ordinates of its head with the food.

- void aik():

  This function will be called by the ai function, if the snake was moving down. It tries to match the co-ordinates of its head with the food.

- void ail():

  This function will be called by the ai function, if the snake was moving right. It tries to match the co-ordinates of its head with the food.

- void aij():

  This function will be called by the ai function, if the snake was moving left. It tries to match the co-ordinates of its head with the food.

- int check(char):

  This function checks if the snake moves in the current co-ordinate will it eat itself. If yes returns 0 else returns 1.

- int out(int):

  This function checks if the snake is eating itself.

Score class:

- void reset_s():

  Resets the score,makes s as zero.

- void tot(int):

  Adds the given number to total(total score).

- void change(int):

  Adds the given number to s(score).

- int scor():

  Returns the score.

- void sort():

  If high score is not on the top five array,replaces the last score with high score and sorts the array.

- int high_s():

  Returns the high score.

- void highs_init():

  Initializes top_five and high score as zero.

- void h_a():

  Updates the high score.

- void dis5():

  Displays top five scores.

• Global Functions:

- void food(void):

  This function display the food in the co-ordinates,stored in fx and fy.

- void display_border(int,int):

This functions displays border in color specified in one of its parameter .

- int input(snake[],int,int):

  This functions gets the inputs for the snakes from their respective player. If the player was computer then is calls the ai function.

- int snake_menu():

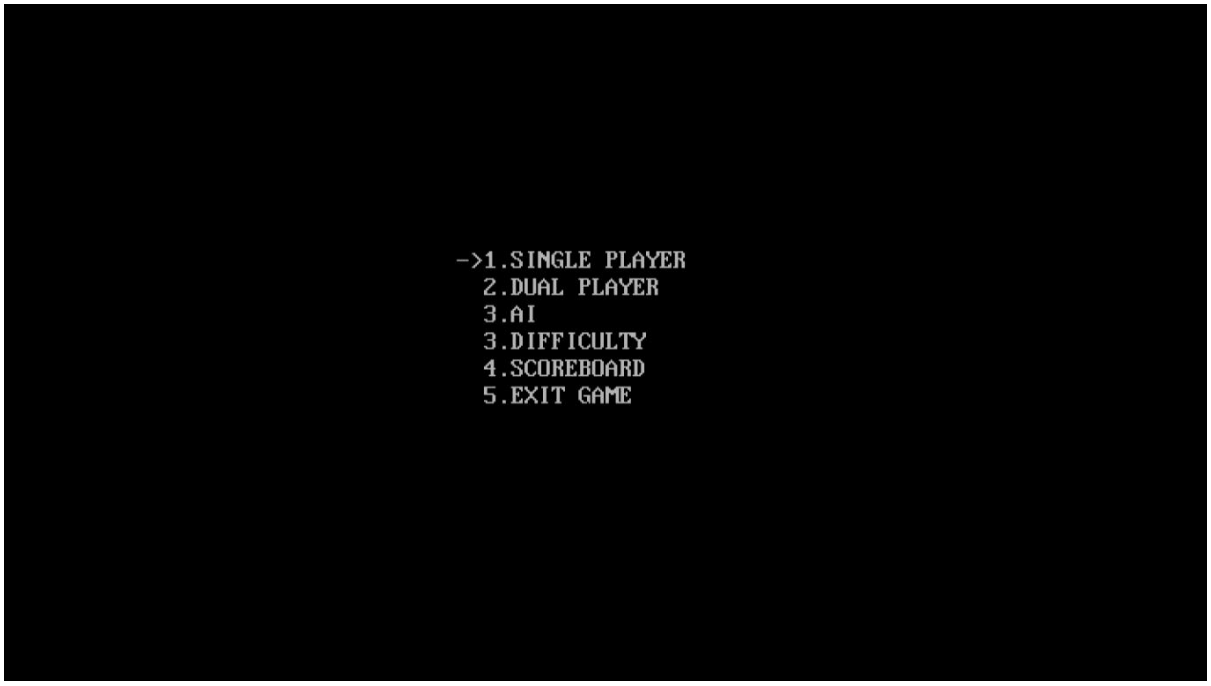
  This functions displays the menu and return the option selected by the player.

- void text3D(int,int,int,int,int,int,char):

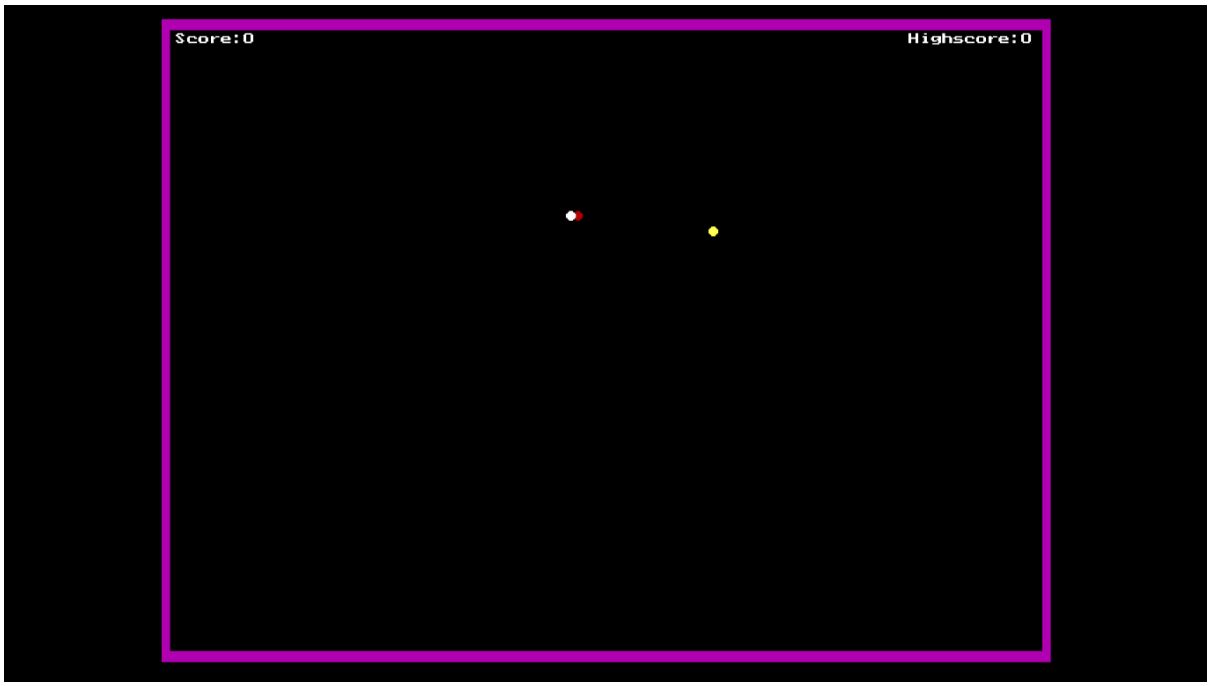
  Display the text given to the function in the given location.

OUTPUT:

Snake game opening screen.

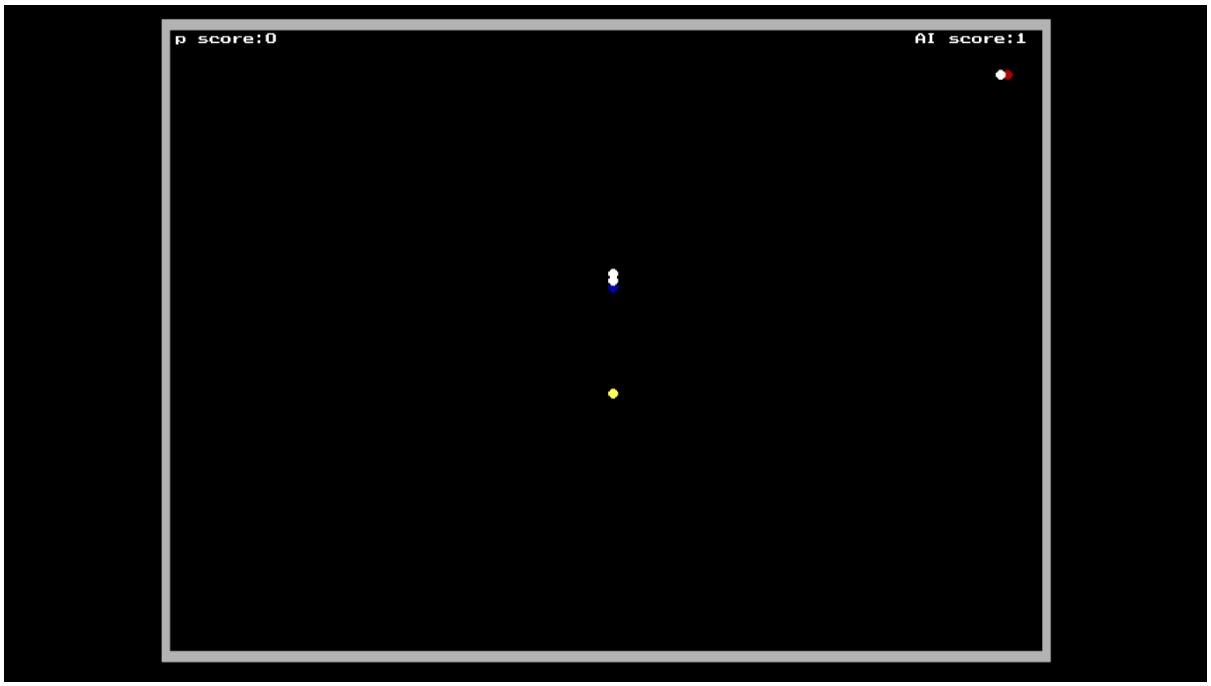


Snake game menu.

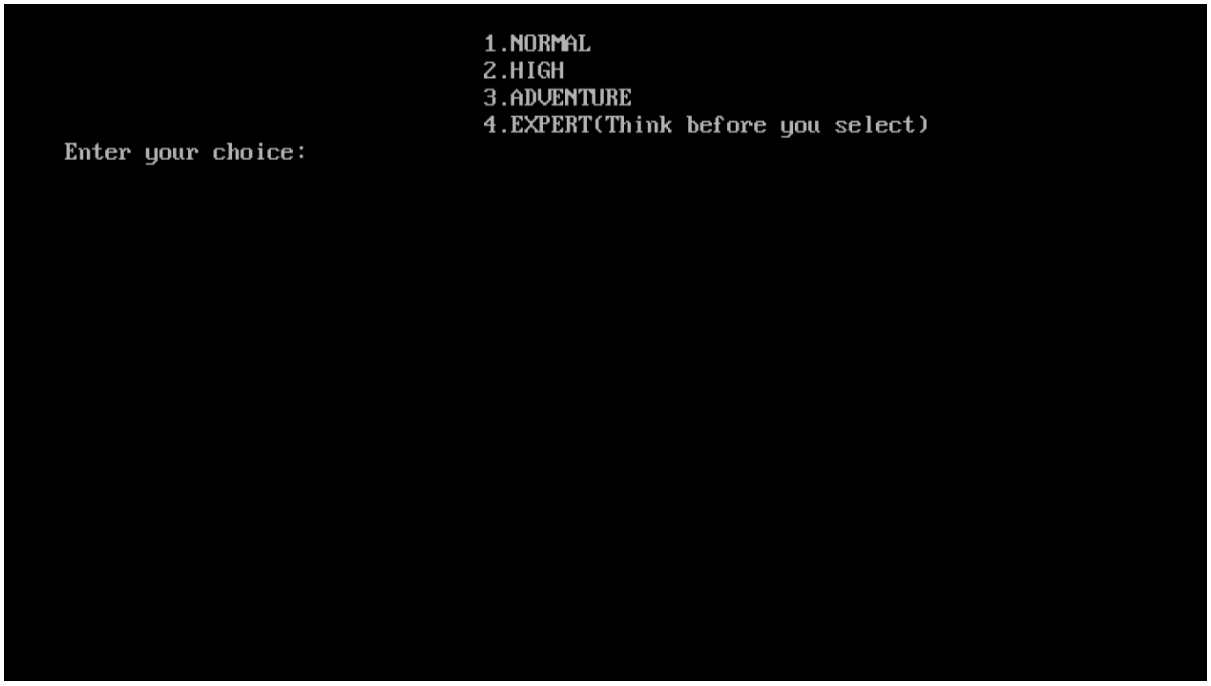


Snake game with single player.

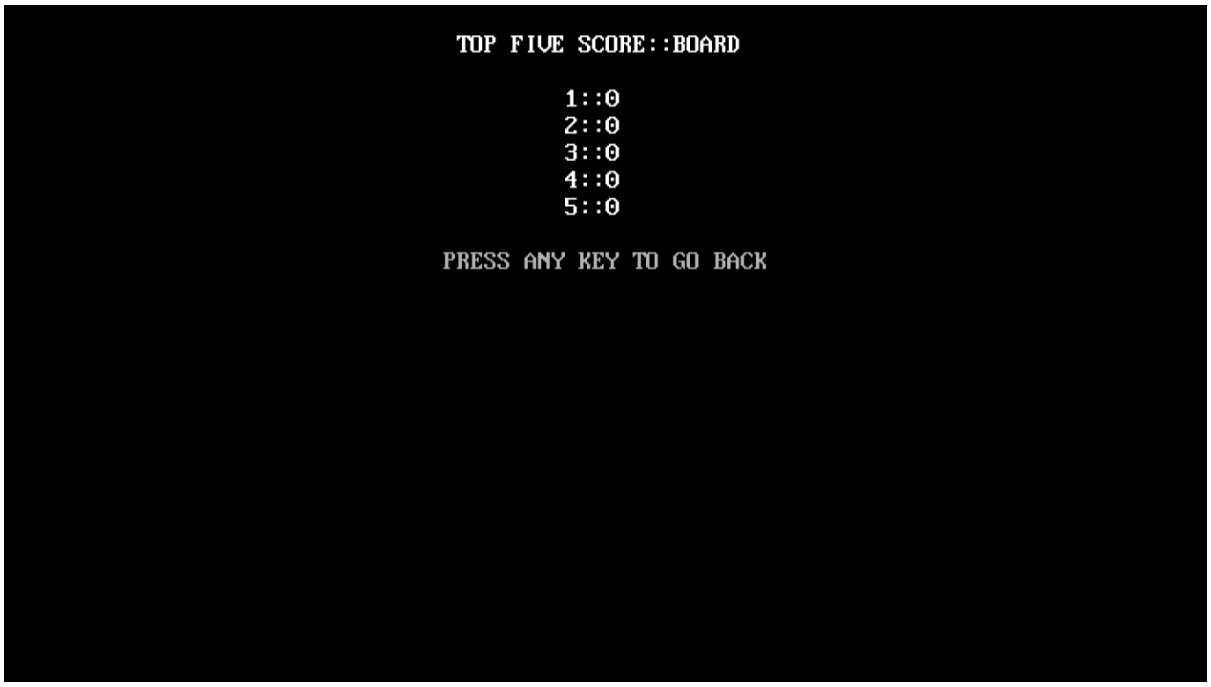
Snake game with two players.



Snake game with computer.



Menu for chosing the difficulty level.



Scorecard for the snake game

SHORTCOMINGS:

- As the snake becomes larger the chances of the computerized snake eating itself becomes more.
- As the snake becomes larger, the display flickers.