Monte Carlos Simulation

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 pi:

Class object: Monto is an instance of pi class.It helps to encapsulate all the methods required for estimating pi using monte carlo method

- Private Members:
  - int lx
  - int ly
  - int ux
- Public Members:
  - void draw(int)
  - void disprandpoints()
  - double calcpi(int)
  - void driver(int,int)

FUNCTIONS USED:

- Member functions:

Pi class:

- void draw(int):

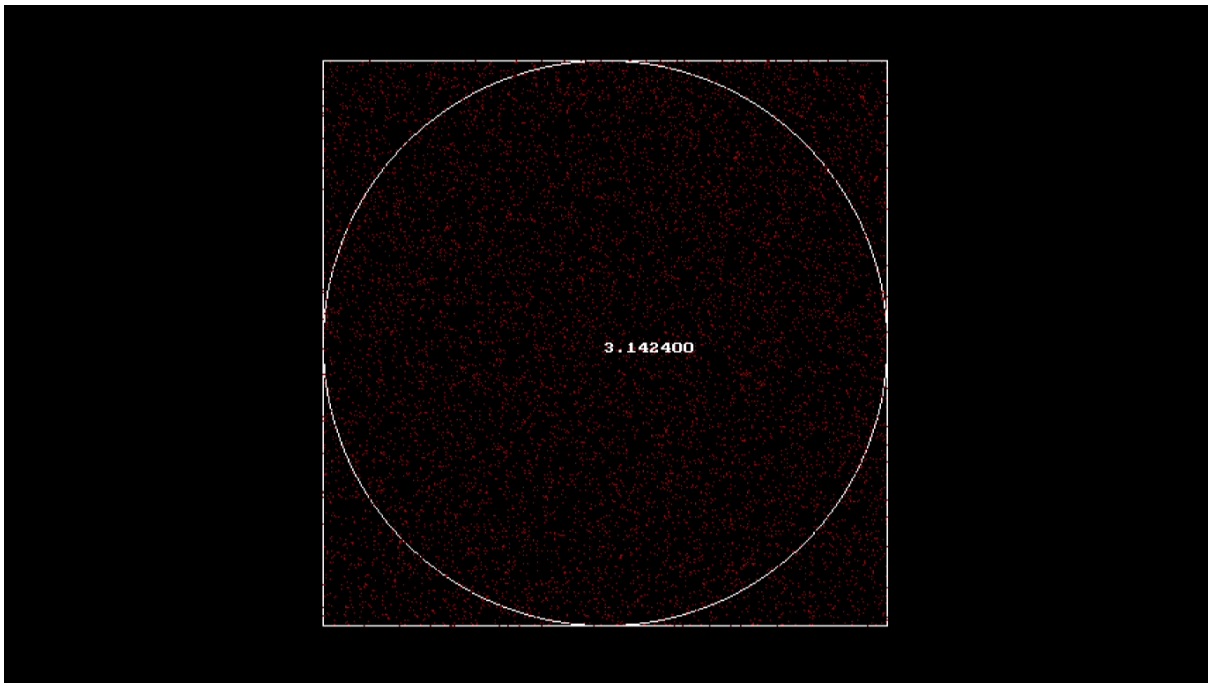
  Draws a circle in the given radius at the center of the screen.Also draws a square whose side length is determined by the radius of the circle.
- void disprandpoints():

  Displays a random point which lies inside the square.
- double calcpi(int):

  Calculates the approximate value of pi based on probability.It takes number of trials as its parameter.
- void driver(int,int):

  It integrates all the functions for calculating pi.It takes number of trials and radius as parameter.

OUTPUT:



Estimating the value of pi.

SHORTCOMINGS:

- Floating point overflow for larger values of number of trials.