COMPUTER SCIENCE PROJECT

*YOGA TUTOR*

****

**-DONE BY**

**S.S.SNEHA**

**XII A**

**ROLL NO : 18**

P S SENIOR SECONDARY SCHOOL

MYLAPORE, CHENNAI

PROJECT

“ YOGA TUTOR ”

IN THE SUBJECT OF COMPUTER SCIENCE (SUBJECT CODE 083)

SUBMITTED FOR AISSCE PRACTICAL EXAMINATION, 2017.

ROLL NUMBER OF THE STUDENTS : 18

NAMES OF THE STUDENTS : S.S.SNEHA

DATE OF SUBMISSION: 17 . 01 . 2017

SIGNATURE OF THE INTERNAL EXAMINER:

EXTERNAL EXAMINER NUMBER:

SIGNATURE OF THE EXTERNAL EXAMINER:

***P S SENIOR SECONDARY SCHOOL***

***MYLAPORE***

***CHENNAI***

***BONAFIDE CERTIFICATE***

***Certified that the project* "YOGA TUTOR"**

***in the subject of COMPUTER SCIENCE (subject code 083) is a bonafide work of S.S.SNEHA***

***I further attest that this work was carried out under my supervision...***

***ROLL NUMBER OF THE STUDENT:18***

***NAME OF THE STUDENT: S.S.SNEHA***

***DATE OF SUBMISSION:17.01.2017***

***SIGNATURE OF THE SUBJECT TEACHER:***

***HEAD OF THE DEPARTMENT OF COMPUTER SCIENCE***

***P.S.SENIOR SECONDARY SCHOOL***

***MYLAPORE, CHENNAI.***

P S SENIOR SECONDARY SCHOOL MYLAPORE, CHENNAI

ACKNOWLEDGEMENT

I acknowledge the support and help provided by the faculty and staff of the COMPUTER SCINCE DEPARTMENT in helping me to successfully complete and submit my project in COMPUTER

SCIENCE for the AISSCE PRACTICAL EXAMINATION, 2017.

ROLL NUMBER: 18

NAME: S.S.SNEHA

DATE: 17 .01.2017

|  |  |  |
| --- | --- | --- |
| S.NO | TITLE | PAGE NO. |
| 1. | ON OUR PROJECT | 6 |
| 2. | SOURCE CODE | 9 |
| 3. | INPUT /OUTPUT SCREENS | 22 |
| 4. | SUGGESITIONS FOR IMPROVEMENT | 26 |
| 5 | BIBILOGRAPHY | 27 |

INDEX:

ON OUR PROJECT :

This is an effort to bring out essential elements of the ancient art YOGA through animation.

YOGA is a group of [physical](https://en.wikipedia.org/wiki/Human_body), [mental](https://en.wikipedia.org/wiki/Mind), and [spiritual](https://en.wikipedia.org/wiki/Soul) practices or disciplines which originated in [ancient India](https://en.wikipedia.org/wiki/Ancient_India).

Among the most well-known types of yoga are Hatha yoga and Rāja yoga.

*Yoga* can take on meanings such as "connection", "contact", "union", "method", "application", "addition" and "performance". In simpler words, Yoga also means "**combined**".

Our project yoga tutor is a client friendly effort which has the following features.

The project broadly shows the important basic asanas in yoga one needs to know before starting official training in the field of yoga.

It also has an animated version of the SURYANAMASKAR which is an integral part of YOGA .

Also the slokas to be chanted during the practise of suryanamaskar have been included.

Our project also offers the client a daily routine course that they can follow in order that they have an equilibrium in mind body and spirit.

***Importance of SURYANAMASKAR:***

*Surya Namaskar* or *Sun Salutations* limbers up the whole body in preparation for the asanas.  
It is a graceful sequence of positions performed as one continuous exercise.

Each position counteracts the one before, stretching the body in a different way and alternately expanding and contracting the chest to regulate the breathing.

Practiced daily it will give great flexibility to your spine and joints and trims your waist.

One full round (12 positions) repeated continuously for 9 times (12×9=108 postures) will make you sweat heavily and burn down all excess fat in body.

Nerves and Muscles will be toned up, all body pains will be removed.

It should take about 30 minutes to 1 hour, depending on practice and speed to complete 9 rounds (108 postures) of Surya Namaskar.

Start practicing Surya Namaskar Postures with Mantras early in the morning with empty stomach (after clearing bowels movement) and just before sunrise.

Keep your hands in one place from positions 3 to 10 and try to coordinate your movements with your breathing.

Start by practicing four rounds and gradually build up to twelve rounds.

Our project is based on object oriented programming and uses the concept of inheritance to a large extent.

As we have also employed virtual functions in a class we have used Abstract class concepts.

***Virtual Functions:***

A **virtual function** is a member **function** that you expect to be redefined in derived classes. When you refer to a derived class object using a pointer or a reference to the base class, you can call a **virtual function** for that object and execute the derived class's version of the **function**.

***Abstract class:***

An **abstract class** can only be a base class for other classes. A class is abstract if it has any one pure virtual function (i.e. function that has no defined body) and therefore necessarily must be overridden by methods of ***inheritance*** in its subclasses. An abstract class cannot be instantiated.

SOURCE CODE :

#include<iostream.h>

#include<graphics.h>

#include<stdio.h>

#include<conio.h>

#include<dos.h>

#include<stdlib.h>

#include<string.h>

#define X getmaxx()

#define Y getmaxy()

#define grnd Y\*0.65

void loading();

void scroll();

void menu();

int option();

void slokas();

// function members of this class displays the various

// basic yoga postures one must know before going ahead

class basics

{

public:

void man(int x)

{

//ground

line(150,grnd,X-150,grnd);

//head

circle(x,grnd-70,10);

//body

line(x,grnd-60,x,grnd-30);

//hands

line(x,grnd-50,x-15,grnd-40);

line(x,grnd-50,x+15,grnd-40);

//legs

line(x,grnd-30,x-10,grnd);

line(x,grnd-30,x+10,grnd);

delay(1750);

cleardevice();

}

void tadasana(int x)

{

//HEAD

circle(x,grnd-60,10 );

//BODY

line(x,grnd-50,x,grnd-20);

//HAND

line(x,grnd-50,x-9,grnd-50);

line(x,grnd-50,x+9,grnd-50);

// UPPER HANDS

line(x-9,grnd-50,x-12,grnd-100);

line(x+9,grnd-50,x+12,grnd-100);

//LEGS

line(x,grnd-20,x+5,grnd);

delay(1750);

cleardevice();

}

void bhagiratasana(int x)

{

//HEAD

circle(x,grnd-60,10 );

//BODY

line(x,grnd-50,x,grnd);

//HAND

line(x,grnd-50,x-5,grnd-50);

line(x,grnd-50,x+5,grnd-50);

// UPPER HANDS

line(x-9,grnd-50,x-12,grnd-100);

line(x+9,grnd-50,x+12,grnd-100);

//LEGS

line(x,grnd-30,x-25,grnd-15);

line(x-25,grnd-15,x,grnd-20);

delay(1750);

cleardevice();

}

void navasana(int x)

{

//HEAD

circle(x-20,grnd-60,10);

//BODY

line(x-20,grnd-50,x-10,grnd);

line(x-10,grnd,x+20,grnd-50);

//HAND

line(x-20,grnd-50,x+20,grnd-50);

delay(1750);

cleardevice();

}

void urdravanukasana(int x)

{

//head

circle(x+15,grnd-25,10);

//body

line(x+15,grnd-10,x-15,grnd-10);

//legs

line(x-15,grnd-10,x-30,grnd-2);

//hands

line(x+15,grnd-15,x+15,grnd-2);

line(x+15,grnd-2,x+20,grnd-3);

delay(1750);

cleardevice();

}

void utkatasana(int x)

{

//head

circle(x,grnd-40,10);

//body

line(x,grnd-30,x-3,grnd-1);

//legs

line(x-3,grnd-1,x+10,grnd-20);

line(x+10,grnd-20,x+15,grnd-1);

//hands

line(x-1,grnd-20,x+15,grnd-35);

line(x-1,grnd-20,x-15,grnd-35);

delay(1750);

cleardevice();

}

void adhm(int x)

{

//HEAD

circle(x+20,grnd-10,10);

//BODY

line(x-20,grnd,x,grnd-60);

line(x,grnd-60,x+20,grnd-20);

//HAND

line(x+20,grnd,x+30,grnd);

line(x+30,grnd,x+40,grnd);

delay(1750);

cleardevice();

}

void upp(int x)

{

//HEAD

circle(x,grnd-12,10 );

//BODY

line(x,grnd-2,x+30,grnd-2);

//HAND

line(x-20,grnd-2,x,grnd-2);

//LEGS

line(x+30,grnd-2,x+30,grnd-50);

line(x+30,grnd-50,x+35,grnd-60);

delay(1750);

cleardevice();

}

void utt(int x)

{

//HEAD

circle(x+10,grnd-30,10 );

//BODY

line(x,grnd-50,x,grnd);

line(x,grnd-50,x+10,grnd-50);

line(x+10,grnd-50,x+10,grnd-40);

//HAND

line(x+10,grnd-20,x+10,grnd);

line(x+10,grnd,x+9,grnd);

//LEG

line(x,grnd,x+5,grnd);

delay(1750);

cleardevice();

}

void autt(int x)

{

//HEAD

circle(x+25,grnd-50,10 );

//BODY

line(x,grnd-50,x,grnd);

line(x,grnd-50,x+15,grnd-50);

//HAND

line(x+35,grnd-50,x+45,grnd-50);

//LEG

line(x,grnd,x+5,grnd);

delay(1750);

cleardevice();

}

void vas(int x)

{

//HEAD

circle(x+30,grnd-60,10);

//BODY

line(x-20,grnd,x,grnd-60);

line(x,grnd-60,x+20,grnd-60);

//HAND

line(x+20,grnd-120,x+20,grnd);

delay(1750);

cleardevice();

}

void apasana(int x)

{

//head

circle(x,grnd-12,10);

//body

line(x+1,grnd-2,x+31,grnd-2);

//legs

line(x+31,grnd-2,x+15,grnd-16);

line(x+15,grnd-16,x+24,grnd-18);

//hands

line(x+5,grnd-2,x+15,grnd-16);

delay(1750);

cleardevice();

}

void sarvagasana(int x)

{

//head

circle(x,grnd-12,10);

//body

line(x+3,grnd-2,x+20,grnd-5);

line(x+20,grnd-5,x+20,grnd-30);

//legs

line(x+20,grnd-30,x+8,grnd-50);

//hands

line(x+20,grnd-5,x+28,grnd-2);

line(x+28,grnd-2,x+21,grnd-16);

line(x+21,grnd-16,x+23,grnd-18);

delay(1750);

cleardevice();

}

void carkravahasana(int x)

{

//head

circle(x+39,grnd-24,10);

//body

line(x,grnd-18,x+30,grnd-18);

//legs

line(x,grnd-2,x,grnd-18);

line(x,grnd-2,x-22,grnd-2);

//hands

line(x+25,grnd-18,x+25,grnd-2);

line(x+25,grnd-2,x+29,grnd-2);

delay(1750);

cleardevice();

}

void parsena\_uttanasana(int x)

{

//head

circle(x+10,grnd-25,10);

//body

line(x,grnd-25,x-25,grnd-25);

//legs

line(x-25,grnd-25,x-35,grnd-3);

line(x-25,grnd-25,x-15,grnd-3);

line(x-35,grnd-3,x-32,grnd-3);

line(x-15,grnd-3,x-12,grnd-3);

//hands

line(x-2,grnd-25,x+8,grnd-3);

line(x+8,grnd-3,x+11,grnd-3);

delay(1750);

cleardevice();

}

};

// type coordinate

struct Point

{

int x, y;

};

const int MAX\_SEGMENTS = 10;

//for setting coordinates and drawing hands, legs, body

struct Chain

{

int num;

// number of segments used for each body part (hands, legs, body)

Point points[MAX\_SEGMENTS+1];

// an array of points (coordinates) for each body part

void draw(int);

};

void Chain :: draw(int color = WHITE)

//draws a segment of a part (hand or leg or body)

{

setcolor(color);

for(int i=0; i<num; i++)

line(points[i].x, points[i].y, points[i+1].x, points[i+1].y);

}

//for setting coordinates and drawing head

struct Circle

{

Point center;

int radius;

void draw(int color = WHITE)

{

setcolor(color);

circle(center.x, center.y, radius);

}

};

#define LEFT 0

#define RIGHT 1

//draws all the body parts for each asana and animates the asanas

class Asana

{

public:

Circle head;

Chain body, hand[2], leg[2];

// draws the human with the given coordinates

void draw(int color = WHITE)

{

head.draw(color);

body.draw(color);

hand[LEFT].draw(color);

hand[RIGHT].draw(color);

leg[LEFT].draw(color);

leg[RIGHT].draw(color);

}

void clear() // erases the human with the given coordinates

{

draw(BLACK);

}

int loop\_length;

//no of iterations needed for the animation to be completed

public:

virtual void initialize(){}

// gives the initial coordinates for each posture / asana

virtual void move(){}

// gives how the coordinates of different body parts should move

void animation();

};

void Asana :: animation()

{

initialize();// gives the coordinates

draw(); // draws the human

for(int t=0; t<loop\_length; t++)

{

clear();// erases the human with old coordinates

move();// changes body coordinates

draw();// draws human with new coordinates

delay(500);

if (kbhit()) break;

}

delay(1500);

clear();

}

// A separate class for each asana is created,

// each of which inherits from `Asana`

// initialize and move functions are overridden

class Man: public Asana

{

public:

void initialize()

{

//set no of segments for each body part

body.num=1;

hand[LEFT].num=1;

hand[RIGHT].num=1;

leg[LEFT].num=1;

leg[RIGHT].num=1;

//LEGS...

leg[LEFT].points[0].x=295;

leg[LEFT].points[0].y=grnd-20;

leg[RIGHT].points[0].x=295;

leg[RIGHT].points[0].y=grnd-20;

leg[LEFT].points[1].x=295;

leg[LEFT].points[1].y=grnd-2;

leg[RIGHT].points[1].x=295;

leg[RIGHT].points[1].y=grnd-2;

//HANDS...

hand[RIGHT].points[0].x=295;

hand[RIGHT].points[0].y=grnd-50;

hand[LEFT].points[0].x=295;

hand[LEFT].points[0].y=grnd-50;

hand[RIGHT].points[1].x=295;

hand[RIGHT].points[1].y=grnd-20;

hand[LEFT].points[1].x=295;

hand[LEFT].points[1].y=grnd-20;

//HEAD...

head.center.x=295;

head.center.y=grnd-60;

head.radius=10;

//BODY...

body.points[0].x=295;

body.points[0].y=grnd-50;

body.points[1].x=295;

body.points[1].y=grnd-20;

// set loop\_length to number of iterations for the animation to complete

loop\_length=5;

}

void move()

{

// move the person by incrementing the coordinates

hand[LEFT].points[1].x-=2;

hand[RIGHT].points[1].x+=2;

leg[LEFT].points[1].x-=1;

leg[LEFT].points[1].y=grnd-2;

leg[RIGHT].points[1].x+=1;

leg[RIGHT].points[1].y=grnd-2;

}

};

class TadAsana: public Asana

{

public:

void initialize()

{

//set no of segments for each body part

body.num=1;

hand[LEFT].num=1;

hand[RIGHT].num=0;

leg[RIGHT].num=0;

leg[LEFT].num=2;

//LEGS...

leg[LEFT].points[0].x=315;

leg[LEFT].points[0].y=grnd-2;

leg[LEFT].points[1].x=318;

leg[LEFT].points[1].y=grnd-2;

leg[LEFT].points[2].x=322;

leg[LEFT].points[2].y=grnd-2;

//HANDS...

hand[LEFT].points[0].x=315;

hand[LEFT].points[0].y=grnd-50;

hand[LEFT].points[1].x=324;

hand[LEFT].points[1].y=grnd-25;

//HEAD...

head.center.x=315;

head.center.y=grnd-60;

head.radius=10;

//BODY...

body.points[0].x=315;

body.points[0].y=grnd-50;

body.points[1].x=315;

body.points[1].y=grnd-2;

// set loop\_length to number of iterations for the animation to complete

loop\_length=10;

}

void move()

{

// move the person by incrementing the coordinates

hand[LEFT].points[0].y-=1;

hand[LEFT].points[1].x+=2;

hand[LEFT].points[1].y-=5;

body.points[0].y-=1;

body.points[1].y-=1;

head.center.y-=1;

leg[LEFT].points[0].y-=1;

leg[LEFT].points[1].x+=0.2;

}

};

class ulti :public Asana

{

public:

void initialize()

{

//set no of segments for each body part

body.num=1;

hand[LEFT].num=0;

hand[RIGHT].num=0;

leg[RIGHT].num=0;

leg[LEFT].num=1;

//HEAD...

head.center.x=325;

head.radius=10;

head.center.y=grnd-62;

//LEGS...

leg[LEFT].points[0].x=315;

leg[LEFT].points[0].y=grnd-50;

leg[LEFT].points[1].x=315;

leg[LEFT].points[1].y=grnd-60;

//BODY...

body.points[0].x=315;

body.points[1].x=315;

body.points[0].y=grnd-50;

body.points[1].y=grnd-2;

// set loop\_length to number of iterations for the animation to complete

loop\_length=9;

}

void move()

{

// move the person by incrementing the co-ordinates

if(head.center.x<=340)

{

head.center.x+=5;

leg[LEFT].points[1].x+=5;

}

else

{

leg[LEFT].points[1].y+=10;

head.center.y+=10;

head.radius=10;

}

}

};

class bhuj :public Asana

{

public:

void initialize()

{

//set no of segments for each body part

body.num=1;

leg[LEFT].num=1;

leg[RIGHT].num=0;

hand[LEFT].num=0;

hand[RIGHT].num=0;

//HEAD...

head.center.x=340;

head.center.y=grnd-12;

head.radius=10;

//LEGS...

leg[LEFT].points[0].x=315;

leg[LEFT].points[0].y=grnd-50;

leg[LEFT].points[1].x=330;

leg[LEFT].points[1].y=grnd-10;

//BODY...

body.points[0].x=315;

body.points[0].y=grnd-2;

body.points[1].x=315;

body.points[1].y=grnd-50;

// set loop\_length to number of iterations for the animation to complete

loop\_length=10;

}

void move()

{

// move the person by incrementing the co-ordinates

if(body.points[1].y<=grnd-6)

{

body.points[0].x-=8;

body.points[1].y+=8;

leg[LEFT].points[0].y+=8;

}

else

{

leg[LEFT].points[1].y-=8;

head.center.y-=8;

head.radius=10;

}

}

};

class carkra: public Asana

{

public:

void initialize()

{

//set no of segments for each body part

body.num=1;

leg[LEFT].num=2;

leg[RIGHT].num=0;

hand[LEFT].num=1;

hand[RIGHT].num=0;

//HEAD...

head.center.x=340;

head.center.y=grnd-34;

head.radius=10;

//BODY...

body.points[0].x=315;

body.points[0].y=grnd-2;

body.points[1].x=330;

body.points[1].y=grnd-34;

//LEGS...

leg[LEFT].points[0].x=267;

leg[LEFT].points[0].y=grnd-2;

leg[LEFT].points[1].x=295;

leg[LEFT].points[1].y=grnd-2;

leg[LEFT].points[2].x=315;

leg[LEFT].points[2].y=grnd-2;

//HANDS...

hand[LEFT].points[0].x=330;

hand[LEFT].points[0].y=grnd-34;

hand[LEFT].points[1].x=330;

hand[LEFT].points[1].y=grnd-34;

// set loop\_length to number of iterations for animation to get over

loop\_length=10;

}

void move()

{

// move the person by incrementing the co-ordinates

leg[LEFT].points[2].x-=2;

leg[LEFT].points[2].y-=2;

body.points[0].x-=2;

body.points[0].y-=2;

body.points[1].x-=0.5;

body.points[1].y+=1.2;

hand[LEFT].points[0].x-=0.5;

hand[LEFT].points[0].y+=1.2;

hand[LEFT].points[1].x-=0.5;

hand[LEFT].points[1].y+=3.2;

head.center.x-=0.5;

head.center.y+=1.2;

head.radius=10;

}

};

class ulti2 : public Asana

{

public:

void initialize()

{

//set no of segments for each body part

body.num=1;

leg[LEFT].num=2;

leg[RIGHT].num=0;

hand[LEFT].num=1;

hand[RIGHT].num=0;

//LEGS...

leg[LEFT].points[0].x=267;

leg[LEFT].points[0].y=grnd-2;

leg[LEFT].points[1].x=295;

leg[LEFT].points[1].y=grnd-2;

leg[LEFT].points[2].x=295;

leg[LEFT].points[2].y=grnd-22;

//BODY...

body.points[0].x=295;

body.points[0].y=grnd-22;

body.points[1].x=325;

body.points[1].y=grnd-22;

//HEAD...

head.center.x=335;

head.center.y=grnd-22;

head.radius=10;

//HANDS...

hand[LEFT].points[0].x=325;

hand[LEFT].points[0].y=grnd-22;

hand[LEFT].points[1].x=325;

hand[LEFT].points[1].y=grnd-2;

// set loop\_length to number of iterations for the animation to complete loop\_length=5;

}

void move()

{

// move the person by incrementing the co-ordinates

leg[LEFT].points[1].x-=2.4;

leg[LEFT].points[1].y-=4;

leg[LEFT].points[2].y-=4;

body.points[0].y-=4;

body.points[1].x-=3;

body.points[1].y+=2;

head.center.x-=3;

head.center.y+=2;

hand[LEFT].points[0].x-=3;

hand[LEFT].points[0].y+=2;

hand[LEFT].points[1].x-=3;

hand[LEFT].points[1].y-=2;

}

};

class tada2 : public Asana

{

public:

void initialize()

{

//set no of segments for each body part

body.num=2;

leg[LEFT].num=0;

leg[RIGHT].num=0;

hand[LEFT].num=0;

hand[RIGHT].num=0;

//BODY...

body.points[0].x=267;

body.points[0].y=grnd-2;

body.points[1].x=295;

body.points[1].y=grnd-42;

body.points[2].x=310;

body.points[2].y=grnd-12;

//HEAD...

head.center.x=320;

head.center.y=grnd-12;

head.radius=10;

// set loop\_length to number of iterations for the animation to complete loop\_length=4+3+5;

}

void move ()

{

//move the person by incrementing the coordinates

if(body.points[0].x<295)

body.points[0].x+=7;

else

{

if(head.center.y>grnd-42)

{

head.center.y-=10;

body.points[2].y-=10;

}

else

{

body.points[2].y-=2;

body.points[2].x-=3;

head.center.y-=4;

head.center.x-=5;

}

}

}

};

class ManAsana: public Asana

{

public:

void initialize()

{

body.num=1;

hand[LEFT].num=1;

hand[RIGHT].num=1;

leg[LEFT].num=1;

leg[RIGHT].num=1;

hand[RIGHT].points[0].x=315;

hand[LEFT].points[0].x=315;

hand[RIGHT].points[1].x=315;

hand[LEFT].points[1].x=315;

leg[LEFT].points[0].x=315;

leg[LEFT].points[1].x=315;

leg[RIGHT].points[0].x=315;

leg[RIGHT].points[1].x=315;

leg[LEFT].points[0].y=grnd-20;

leg[LEFT].points[1].y=grnd-2;

leg[RIGHT].points[0].y=grnd-20;

leg[RIGHT].points[1].y=grnd-2;

hand[LEFT].points[0].y=grnd-50;

hand[RIGHT].points[0].y=grnd-50;

hand[LEFT].points[1].x+=9;

hand[LEFT].points[1].y=grnd-25;

hand[RIGHT].points[1].x-=9;

hand[RIGHT].points[1].y=grnd-25;

//CIRCLE...

head.center.x=315;

head.center.y=grnd-60;

head.radius=10;

//BODY...

body.points[0].x=315;

body.points[0].y=grnd-50;

body.points[1].x=315;

body.points[1].y=grnd;

loop\_length=5;

}

void move()

{

// move the person by incrementing the co-ordinates

hand[LEFT].points[1].x-=2;

hand[LEFT].points[1].y=grnd-25;// left hand

hand[RIGHT].points[1].x+=2; // right hand

hand[RIGHT].points[1].y=grnd-25;

leg[LEFT].points[1].x=315;

leg[LEFT].points[1].y=grnd;

}

};

class BhagiratAsana: public Asana

{

public:

void initialize()

{

body.num=1;

hand[LEFT].num=1;

hand[RIGHT].num=1;

leg[LEFT].num=2;

leg[RIGHT].num=0;

leg[LEFT].points[0].x=315;

leg[LEFT].points[1].x=315;

leg[LEFT].points[1].x=315;

leg[LEFT].points[0].y=grnd-20;

leg[LEFT].points[1].y=grnd-10;

leg[LEFT].points[2].y=grnd;

body.points[0].x=315;

body.points[1].x=315;

body.points[1].y=grnd;

body.points[0].y=grnd-50;

//point 0 ini to 315

hand[RIGHT].points[0].x=315;

hand[LEFT].points[0].x=315;

//point 1 ini to 315

hand[RIGHT].points[1].x=315;

hand[LEFT].points[1].x=315;

hand[LEFT].points[0].y=grnd-50;

hand[RIGHT].points[0].y=grnd-50;

hand[LEFT].points[1].x+=9;

hand[LEFT].points[1].y=grnd-25;

hand[RIGHT].points[1].x-=9;

hand[RIGHT].points[1].y=grnd-25;

head.center.x=315;

head.center.y=grnd-60;

head.radius=10;

loop\_length=10;

}

void move()

{

hand[LEFT].points[1].x+=2;

hand[LEFT].points[1].y-=2;// left hand

hand[RIGHT].points[1].x-=2; // right hand

hand[RIGHT].points[1].y-=2;

leg[LEFT].points[1].y=grnd-14;

leg[LEFT].points[1].x-=2;

leg[LEFT].points[2].y=grnd-8;

leg[LEFT].points[2].x=315;

}

};

class SIRA: public Asana

{

public:

void initialize()

{

body.num=2;

leg[LEFT].num=0;

leg[RIGHT].num=0;

hand[LEFT].num=0;

hand[RIGHT].num=0;

body.points[0].x=267;

body.points[0].y=grnd-2;

body.points[1].x=295;

body.points[1].y=grnd-42;

body.points[2].x=310;

body.points[2].y=grnd-12;

head.center.x=320;

head.center.y=grnd-12;

head.radius=10;

loop\_length=5;

}

void move ()

{

body.points[1].x+=3;

body.points[0].y-=12;

body.points[0].x+=7;

body.points[2].x+=2;

body.points[2].y-=1;

}

};

class nav: public Asana

{

public:

void initialize()

{

body.num=1;

hand[LEFT].num=1;

hand[RIGHT].num=1;

leg[LEFT].num=1;

leg[RIGHT].num=1;

hand[RIGHT].points[0].x=315;

hand[LEFT].points[0].x=315;

hand[RIGHT].points[1].x=315;

hand[LEFT].points[1].x=315;

hand[LEFT].points[0].y=grnd-50;

hand[RIGHT].points[0].y=grnd-50;

hand[LEFT].points[1].x+=9;

hand[LEFT].points[1].y=grnd-25;

hand[RIGHT].points[1].x-=9;

hand[RIGHT].points[1].y=grnd-25;

leg[LEFT].points[0].x=315;

leg[LEFT].points[1].x=315;

leg[RIGHT].points[0].x=315;

leg[RIGHT].points[1].x=315;

leg[LEFT].points[0].y=grnd;

leg[LEFT].points[1].y=grnd-20;

leg[LEFT].points[1].x=315;

leg[RIGHT].points[0].y=grnd;

leg[RIGHT].points[1].y=grnd-20;

leg[RIGHT].points[1].x=315;

head.center.x=315;

head.center.y=grnd-60;

head.radius=10;

body.points[0].x=315;

body.points[0].y=grnd-50;

body.points[1].x=315;

body.points[1].y=grnd;

loop\_length=5;

}

void move()

{

if(leg[RIGHT].points[1].x<345)

{

leg[RIGHT].points[1].x+=5;

hand[LEFT].points[1].x+=3;

hand[LEFT].points[1].y-=5;// left hand

hand[RIGHT].points[1].x+=3; // right hand

hand[RIGHT].points[1].y-=5;

}

else

{

leg[RIGHT].points[1].x-=5;

hand[LEFT].points[1].x-=3;

hand[LEFT].points[1].y+=5;// left hand

hand[RIGHT].points[1].x-=3; // right hand

hand[RIGHT].points[1].y+=5;

}

}

};

class SRA: public Asana

{

public:

void initialize()

{

body.num=1;

leg[LEFT].num=0;

leg[RIGHT].num=0;

hand[LEFT].num=0;

hand[RIGHT].num=0;

body.points[0].x=318;

body.points[0].y=grnd-2;

body.points[1].x=290;

body.points[1].y=grnd-2;

body.points[2].x=300;

body.points[2].y=grnd-2;

head.center.x=320;

head.center.y=grnd-12;

head.radius=10;

loop\_length=8;

}

void move ()

{

body.points[2].y-=5;

body.points[2].x-=2;

}

};

class sitting: public Asana

{

public:

void initialize()

{

body.num=1;

hand[LEFT].num=1;

hand[RIGHT].num=1;

leg[LEFT].num=2;

leg[RIGHT].num=2;

hand[LEFT].points[0].x=315;

hand[LEFT].points[0].y=grnd-32;

hand[RIGHT].points[0].x=315;

hand[RIGHT].points[0].y=grnd-32;

hand[LEFT].points[1].x=315-8;

hand[LEFT].points[1].y=grnd-7;

hand[RIGHT].points[1].x=315+8;

hand[RIGHT].points[1].y=grnd-7;

leg[LEFT].points[0].x=315;

leg[LEFT].points[0].y=grnd-12;

leg[RIGHT].points[0].x=315;

leg[RIGHT].points[0].y=grnd-12;

leg[LEFT].points[1].x=304;

leg[LEFT].points[1].y=grnd-2;

leg[RIGHT].points[1].x=326;

leg[RIGHT].points[1].y=grnd-2;

leg[LEFT].points[2].x=314;

leg[LEFT].points[2].y=grnd-2;

leg[RIGHT].points[2].x=312;

leg[RIGHT].points[2].y=grnd;

//BODY...

body.points[0].x=315;

body.points[1].x=315;

body.points[0].y=grnd-32;

body.points[1].y=grnd-12;

//HEAD...

head.center.x=315;

head.center.y=grnd-42;

head.radius=10;

loop\_length=5;

}

void move()

{

leg[LEFT].points[2].x+=1;

leg[LEFT].points[2].y-=1;

leg[RIGHT].points[2].x-=1;

leg[RIGHT].points[2].y-=1;

}

};

int main()

{

/\* initialize graphics mode \*/

int g=DETECT, d, errorcode;

initgraph(&g, &d,"C:\\turboc3\\bgi");

/\* read result of initialization \*/

errorcode = graphresult();

if (errorcode != grOk) /\* an error occurred \*/

{

printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:");

getch();

exit(1); /\* terminate with an error code \*/

}

int a;

cleardevice();

loading();

do{

cleardevice();

menu();

scroll();

a=option();

}while(a);

cleardevice();

settextstyle(DEFAULT\_FONT, HORIZ\_DIR,3);

settextjustify(1,1);

setcolor(WHITE);

outtextxy(315,Y/2,"THANK YOU!!!");

delay(1000);

closegraph();

cout<<"PRESS ENTER TO QUIT";

getch();

return 0;

}

void loading()

{

int x,y,i;

x=getmaxx()/2;

y=getmaxy()/2;

settextstyle(TRIPLEX\_FONT,HORIZ\_DIR,3);

setbkcolor(rand());

setcolor(4);

int c=150;

for(int e=0 ; e<15 ; e++) //Change 15 to your desired value

{

for(i=50;i<100;i++)

{

setcolor(3);

settextstyle(SMALL\_FONT, HORIZ\_DIR, 10);

outtextxy(x,y,"Loading...");

settextstyle(SMALL\_FONT, HORIZ\_DIR, 5);

outtextxy(x+5,y+5,"Please Wait... ");

circle(x,y,i);

circle(x,y,c);

c--;

cleardevice();

}

}

}

void scroll()

{

char \*str, ch;

int i,j,location,n,c=10;

strcpy(str," Welcome to Yoga Tutor ");

for(i=0 ; !kbhit() ; i++)

{

setfillstyle(SOLID\_FILL,BLACK);

bar(50,50,600,90);

//two-color banner

if(c==3)

c=WHITE;

else

c=3;

setcolor(c);

settextstyle(GOTHIC\_FONT,HORIZ\_DIR,2);

outtextxy(50,50,str);

delay(350);

ch = str[0];

//substring logic

n=strlen(str);

location=0;

for(j=location;j<n;j++)

str[j] = str[j+1];

//appending char at end

str[n-1]=ch;

str[n]=NULL;

}

}

int option()

{

int a,x=315;

a=getch();

cleardevice();

if(a=='B'||a=='b')

{

basics b;

settextstyle(DEFAULT\_FONT, HORIZ\_DIR,2);

setcolor(3);

settextjustify(1,1);

outtextxy(315,30,"THIS IS APANASANA");

//ground

line(150,grnd,X-150,grnd);

setcolor(WHITE);

b.apasana(x);

setcolor(3);

settextjustify(1,1);

outtextxy(315,30,"THIS IS UTKATASANA");

//ground

line(150,grnd,X-150,grnd);

setcolor(WHITE);

b.utkatasana(x);

setcolor(3);

settextjustify(1,1);

outtextxy(315,30,"THIS IS SARVANGASANA");

//grnd

line(150,grnd,X-150,grnd);

setcolor(WHITE);

b.sarvagasana(x);

setcolor(3);

settextjustify(1,1);

outtextxy(315,30,"THIS IS CARKRAVAKASANA");

//ground

line(150,grnd,X-150,grnd);

setcolor(WHITE);

b.carkravahasana(x);

setcolor(3);

settextjustify(1,1);

outtextxy(315,30,"THIS IS URDVAMUKHASVANASANA");

//ground

line(150,grnd,X-150,grnd);

setcolor(WHITE);

b.urdravanukasana(x);

setcolor(3);

settextjustify(1,1);

outtextxy(315,30,"THIS IS PARSVA UTTANASANA");

//ground

line(150,grnd,X-150,grnd);

setcolor(WHITE);

b.parsena\_uttanasana(x);

setcolor(3);

settextjustify(1,1);

outtextxy(315,30,"THIS IS TADASANA");

//ground

line(150,grnd,X-150,grnd);

setcolor(WHITE);

b.tadasana(x);

setcolor(3);

settextjustify(1,1);

outtextxy(315,30,"THIS IS BHAGIRATASANA");

//ground

line(150,grnd,X-150,grnd);

setcolor(WHITE);

b.bhagiratasana(x);

setcolor(3);

settextjustify(1,1);

outtextxy(315,30,"THIS IS NAVASANA");

//ground

line(150,grnd,X-150,grnd);

setcolor(WHITE);

b.navasana(x);

setcolor(3);

settextjustify(1,1);

outtextxy(315,30,"THIS IS ADHOMUKHASVANASANA");

//ground

line(150,grnd,X-150,grnd);

setcolor(WHITE);

b.adhm(x);

setcolor(3);

settextjustify(1,1);

outtextxy(315,30,"THIS IS URDVA PRASITA PADASANA");

//ground

line(150,grnd,X-150,grnd);

setcolor(WHITE);

b.upp(x);

setcolor(3);

settextjustify(1,1);

outtextxy(315,30,"THIS IS UTTANASANA");

//ground

line(150,grnd,X-150,grnd);

setcolor(WHITE);

b.utt(x);

setcolor(3);

settextjustify(1,1);

outtextxy(315,30,"THIS IS ARDHA UTTANASANA");

//ground

line(150,grnd,X-150,grnd);

setcolor(WHITE);

b.autt(x);

setcolor(3);

settextjustify(1,1);

outtextxy(315,30,"THIS IS VASISTASANA");

//ground

line(150,grnd,X-150,grnd);

setcolor(WHITE);

b.vas(x);

cleardevice();

}

else if(a=='C' || a=='c')

{

settextstyle(DEFAULT\_FONT , HORIZ\_DIR , 2);

settextjustify(1,1);

setcolor(3);

outtextxy(315,50,"THIS IS SURYANAMASKARAM");

line(150,grnd,X-150,grnd);

setcolor(WHITE);

// Run the animation:

TadAsana s1;

s1.animation();

ulti s2;

s2.animation();

bhuj s3;

s3.animation();

carkra r;

r.animation();

ulti2 u;

u.animation();

tada2 t;

t.animation();

Man s;

s.animation();

cleardevice();

}

else if(a=='D' || a=='d')

{

settextstyle(DEFAULT\_FONT, HORIZ\_DIR , 2);

SETTEXTJUSTIFY(1,1);

setcolor(3);

outtextxy(315,50,”PRACTICE THESE ASANAS EVERYDAY”);

line(150,grnd,X-150,grnd);

setcolor(WHITE);

ManAsana s;

s.animation();

TadAsana t;

t.animation();

BhagiratAsana s1;

s1.animation();

SIRA s2;

s2.animation();

nav s3;

s3.animation();

bhuj b;

b.animation();

sitting s5;

s5.animation();

SRA s4;

s4.animation();

cleardevice();

}

else if(a=='s' || a=='S')

{

slokas();

cleardevice();

}

else if(a=='q' || a=='Q')

{

cleardevice();

return 0;

}

else

return 0;

setbkcolor(BLACK);

setcolor(WHITE);

settextstyle(DEFAULT\_FONT, HORIZ\_DIR, 2);

settextjustify(1,1);

outtextxy(getmaxx()/2 , getmaxy()/2 , "Press <enter> to go back to main menu");

if(getch())

return 1;

return 0;

}

void menu()

{

int x= getmaxx()/2, y= getmaxy()/2;

setcolor(WHITE);

rectangle(x-60,y-60,x+60,y-30);

rectangle(x-60,y-20,x+60,y+10);

rectangle(x-60,y+20,x+60,y+50);

rectangle(x-60,y+60,x+60,y+90);

rectangle(x-60,y+100,x+60,y+130);

setcolor(3);

settextstyle(DEFAULT\_FONT, HORIZ\_DIR, 2);

settextjustify(1,1);

outtextxy(x,y-120,"MENU");

settextstyle(DEFAULT\_FONT, HORIZ\_DIR,1);

outtextxy(x,y-45,"Basics");

outtextxy(x,y-5,"Courses");

outtextxy(x,y+75,"Daily practice");

outtextxy(x,y+35,"Slokas");

outtextxy(x,y+115,"Quit");

outtextxy(x,y+175,”Press ‘B’ for basics, ‘C’ for courses ”);

outtextxy(x,y+185,”’S’ for slokas, ‘D’ for daily practice ”);

outtextxy(x,y+195,”and ’Q’ for quit”);

settextjustify(0,2);

}

void slokas()

{

int x= getmaxx()/2, y= getmaxy()/2;

settextjustify(1,1);

settextstyle(TRIPLEX\_FONT,HORIZ\_DIR,2);

setbkcolor(BLUE);

outtextxy(x,y-190,"HERE ARE THE SLOKAS WHICH YOU CAN ");

outtextxy(x,y-170,"RECITE AS YOU PRACTISE SURYANAMASKAR ");

outtextxy(x,y-130,"STEP 1: Om Mitraaya Namaha");

outtextxy(x,y-100,"STEP 2: Om Ravaye Namaha");

outtextxy(x,y-70,"STEP 3:Om Suryaya Namaha ");

outtextxy(x,y-40,"STEP 4:Om Bhaanave Namaha");

outtextxy(x,y-10,"STEP 5: Om Khagaya Namaha ");

outtextxy(x,y+20,"STEP 6: Om Pooshne Namaha ");

outtextxy(x,y+50,"STEP 7 :Om Hiranyagarbhaaya Namaha");

outtextxy(x,y+80,"STEP 8: Om Mareechaye Namaha");

outtextxy(x,y+110,"STEP 9 :Om Aadityaaya Namaha");

outtextxy(x,y+140,"STEP 10: Om Savitre Namaha");

outtextxy(x,y+170,"STEP 11: Om Aarkaaya Namaha");

outtextxy(x,y+200,"STEP 12: Om Bhaaskaraya Namaha");

getch();

}

INPUT / OUTPUT SCREENS:

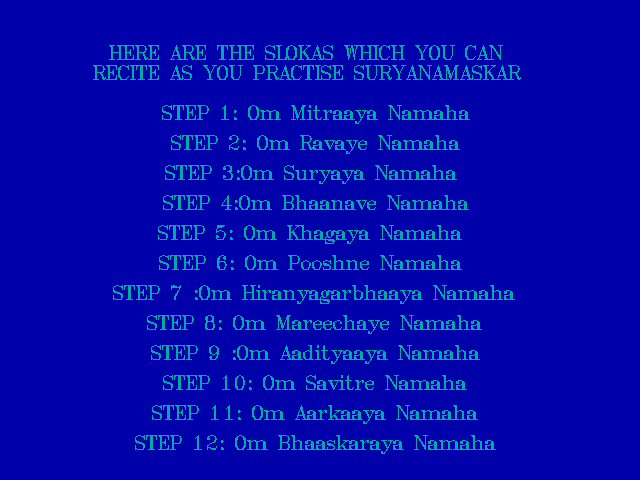


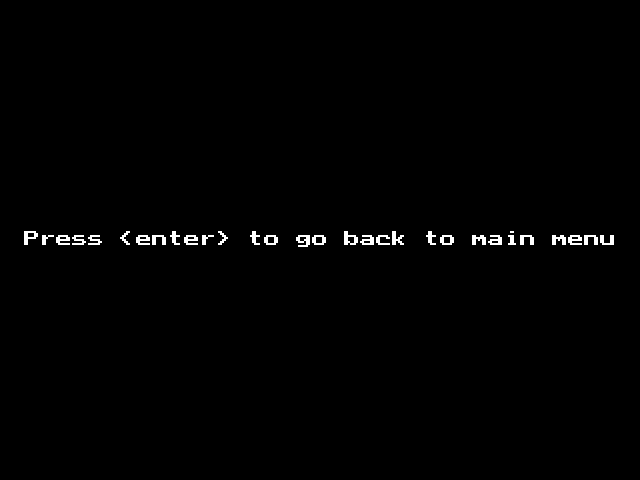














SUGGESIONS FOR IMPROVEMENT:

* Interactive interface .
* More details on history of poses(asanas).
* Client query based system can be applied .

BIBLIOGRAPHY:

* yoga postures and ideas : yoga raksanam
* virtual functions and abstract classes: class computer science notes
* animation ideas : youtube (https://www.youtube.com/watch?v=wvv4rz4xneu)

***other references:***

quora(*https://www.****quora****.com/*) ,

stack overflow(***stackoverflow****.com/*),

http://www.dailyfreecode.com