Article

Two image viewers: A projector and a screen saver

WenJun Zhang

School of Life Sciences, Sun Yat-sen University, Guangzhou, China

E-mail: zhwj@mail.sysu.edu.cn, wjzhang@iaees.org

Abstract

Computer software will help biological researchers view and handle bio-graphics and images. Two standalone executable image viewers, Projector 1.0.exe and scrSaver 1.0.scr, which can be used to play JPG, BMP or ICO image files automatically and continually in Windows operating systems, were developed in present study. Projector 1.0 is an image projector and scrSaver 1.0 is a screen saver. In the Projector 1.0, users can set the time duration for displaying every image, choose random or sequential displaying and choose to stretch images or remain their original sizes on screen. After downloading and decompressing the package, copy and put them in any local folder containing multiple images. Double-click to run them. For scrSaver 1.0, the screen will enter the screen saver pattern, and randomly play the images in the folder. For the two software, they will exit the playing when there is a mouse or keyboard action. Both software and demo image files were given in a free package.

Keywords software; projector; screen saver; image viewer.

1 Introduction

Computer software, including standalone executable ones and Java ones, will help biological researchers view and handle bio-graphics and images. In the past years, I have developed various software for biological uses (Zhang, 2007, 2012, 2021, 2023, 2024a-e). In present study, I will futher present two standalone executable image viewers, Projector 1.0.exe and scrSaver 1.0.scr. They can be used to play JPG, BMP or ICO image files automatically and continually in Windows operating systems. The software are free for downloading.

2 Software

2.1 Projector

The standalone executable software, Projector 1.0.exe, was developed using Delphi (Fig. 1). The following are the main Delphi codes of the software:

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, jpeg, ExtDlgs, StdCtrls, StrUtils;

type
  TForm1 = class(TForm)
    Timer1: TTimer;
    Image1: TImage;
    GroupBox1: TGroupBox;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    GroupBox2: TGroupBox;
    RadioButton3: TRadioButton;
    RadioButton4: TRadioButton;
    Edit1: TEdit;
    Label1: TLabel;
    Button1: TButton;
    Button2: TButton;
    procedure Timer1Timer(Sender: TObject);
    procedure Image1MouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Image1Click(Sender: TObject);
    procedure FormKeyDown(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure FormCreate(Sender: TObject);
    procedure allClose(myList: TStringList);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  gx,gy,sig,nfiles,ros,sop,tm,start,i: Integer;
  imgList: TStringList;
```

implementation

{$R *.dfm}

```
procedure TForm1.allClose(myList: TStringList);
begin
Form1.Caption:='Copyright: WJ Zhang (wjzhang@iaees.org)';
sleep(500);
myList.free;
Application.terminate;
end;


function SearchFiles(path: String; filter: String): TStringList;
var
    SearchRec: TSearchRec;
    found: Integer;
begin
    Result:=TStringList.Create;
    found:=FindFirst(path+filter,faAnyFile,SearchRec);
    while (found=0) do
      begin
          if (SearchRec.Name<>'.') and (SearchRec.Name<>'..') and (SearchRec.Attr<>faDirectory) then
              Result.Add(SearchRec.Name);
          found:=FindNext(SearchRec);
      end;
    FindClose(SearchRec);
end;


procedure TForm1.Timer1Timer(Sender: TObject);
var
    sign: Integer;
begin
if (start=1) then
if (ros=0) then
begin
Randomize;
sign:=random(nfiles);
while ((sign=sig) and (nfiles>1)) do
sign:=random(nfiles);
image1.picture.loadfromfile(imgList[sign]);
sig:=sign;
end
else
begin
```

```
i:=i+1;
image1.picture.loadfromfile(imgList[i]);
if (i>=nfiles) then i:=0;
end;
end;


procedure TForm1.Image1MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
allClose(imgList);
end;


procedure TForm1.Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
begin
if ((gx=0) and (gy=0)) then
begin
gx:=x;
gy:=y;
end;
if ((gx<>x) or (gy<>y)) then
allClose(imgList);
end;


procedure TForm1.Image1Click(Sender: TObject);
begin
allClose(imgList);
end;


procedure TForm1.FormKeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
allClose(imgList);
end;


procedure TForm1.FormCreate(Sender: TObject);
var
    i: Integer;
    str: String;
    fileList: TStringList;
begin
fileList:=SearchFiles(GetCurrentDir+'\','*.*');
if (fileList.count=0) then
allClose(fileList)
else
```

```
begin
imgList:=TStringList.Create;
for i:=0 to fileList.count-1 do
begin
str:=ExtractFileExt(fileList[i]);
if ((str='.jpeg') or (str='.JPEG') or (str='.jpg') or (str='.JPG') or (str='.bmp') or (str='.BMP') or (str='.ico') or (str='.ICO')) then
imgList.add(fileList[i]);
end;
fileList.free;
nfiles:=imgList.count;
if (nfiles=0) then
allClose(imgList);
gx:=0;
gy:=0;
sig:=-1;
end;
button1.taborder:=0;
start:=0;
end;


procedure TForm1.Button1Click(Sender: TObject);
begin
if RadioButton1.Checked then
ros:=0;
if RadioButton2.Checked then
ros:=1;
if RadioButton3.Checked then
sop:=0;
if RadioButton4.Checked then
sop:=1;
tm:=strtoint(Edit1.text);
form1.BorderStyle:=bsnone;
form1.Color:=clblack;
form1.WindowState:=wsMaximized;
if (sop=0) then
image1.Stretch:=true
else image1.Stretch:=false;
timer1.interval:=tm;
image1.align:=alclient;
groupbox1.visible:=false;
groupbox2.visible:=false;
label1.visible:=false;
edit1.visible:=false;
button1.visible:=false;
button2.visible:=false;
```

```
start:=1;
i:=0;
end;


procedure TForm1.Button2Click(Sender: TObject);
begin
Application.terminate;
end;


end.
```

Projector 1.0 is an image projector. In the Projector 1.0, users can set the time duration for displaying every image, choose random or sequential displaying and choose to stretch images or remain their original sizes on screen (Fig. 2). After downloading and decompressing the package, copy and put it in any local folder containing multiple images. Double-click to run it, and enter parameters or make choices, and it will play JPG, BMP or ICO image files automatically and continually. It will exit the playing when there is a mouse or keyboard action.

2.2 scrSaver

The standalone executable software, scrSaver 1.0.scr, was developed using Delphi. The following are the main Delphi codes of the software:

```
unit Unit1;

interface

uses
   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
   Dialogs, ExtCtrls, jpeg, ExtDlgs, StdCtrls, StrUtils;

type
   TForm1 = class(TForm)
     Timer1: TTimer;
     Image1: TImage;
     procedure Timer1Timer(Sender: TObject);
     procedure Image1MouseDown(Sender: TObject; Button: TMouseButton;
       Shift: TShiftState; X, Y: Integer);
     procedure Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
       Y: Integer);
     procedure Image1Click(Sender: TObject);
     procedure FormKeyDown(Sender: TObject; var Key: Word;
       Shift: TShiftState);
     procedure FormCreate(Sender: TObject);
     procedure allClose(myList: TStringList);

   private
```

```
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  gx,gy,sig,nfiles: Integer;
  imgList: TStringList;

implementation

{$R *.dfm}

procedure TForm1.allClose(myList: TStringList);
begin
Form1.Caption:='Copyright: WJ Zhang (wjzhang@iaees.org)';
sleep(500);
myList.free;
Application.terminate;
end;

function SearchFiles(path: String; filter: String): TStringList;
var
    SearchRec: TSearchRec;
    found: Integer;
begin
    Result:=TStringList.Create;
    found:=FindFirst(path+filter,faAnyFile,SearchRec);
    while (found=0) do
      begin
          if (SearchRec.Name<>'.') and (SearchRec.Name<>'..') and (SearchRec.Attr<>faDirectory) then
                Result.Add(SearchRec.Name);
          found:=FindNext(SearchRec);
      end;
    FindClose(SearchRec);
end;

procedure TForm1.Timer1Timer(Sender: TObject);
var
    sign: Integer;
begin
Randomize;
sign:=random(nfiles);
while ((sign=sig) and (nfiles>1)) do
```

```
sign:=random(nfiles);
image1.picture.loadfromfile(imgList[sign]);
sig:=sign;
end;


procedure TForm1.Image1MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
allClose(imgList);
end;


procedure TForm1.Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
begin
if ((gx=0) and (gy=0)) then
begin
gx:=x;
gy:=y;
end;
if ((gx<>x) or (gy<>y)) then
allClose(imgList);
end;


procedure TForm1.Image1Click(Sender: TObject);
begin
allClose(imgList);
end;


procedure TForm1.FormKeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
allClose(imgList);
end;


procedure TForm1.FormCreate(Sender: TObject);
var
    i: Integer;
    str: String;
    fileList: TStringList;
begin
fileList:=SearchFiles(GetCurrentDir+'\','*.*');
if (fileList.count=0) then
allClose(fileList)
else
begin
```

```
imgList:=TStringList.Create;

for i:=0 to fileList.count-1 do

begin

str:=ExtractFileExt(fileList[i]);

if ((str='.jpeg') or (str='.JPEG') or (str='.jpg') or (str='.JPG') or (str='.bmp') or (str='.BMP') or (str='.ico') or (str='.ICO')) then

imgList.add(fileList[i]);

end;

fileList.free;

nfiles:=imgList.count;

if (nfiles=0) then

allClose(imgList);

gx:=0;

gy:=0;

sig:=-1;

end;

end;


end.
```

scrSaver 1.0 is a screen saver. After downloading and decompressing the package, copy and put it in any local folder containing multiple images. Double-click to run it and the screen will enter the screen saver pattern, and randomly play the images in the folder. It will exit the playing when there is a mouse or keyboard action.

The software and demo image files are included in the package: http://www.iaees.org/publications/journals/np/articles/2024-9(3-4)/e-suppl/Zhang-Supplementary-Material.rar

Some of the images (some of them are downloaded from internet, subject to copyrights) attached in the package are shown in Fig. 3.
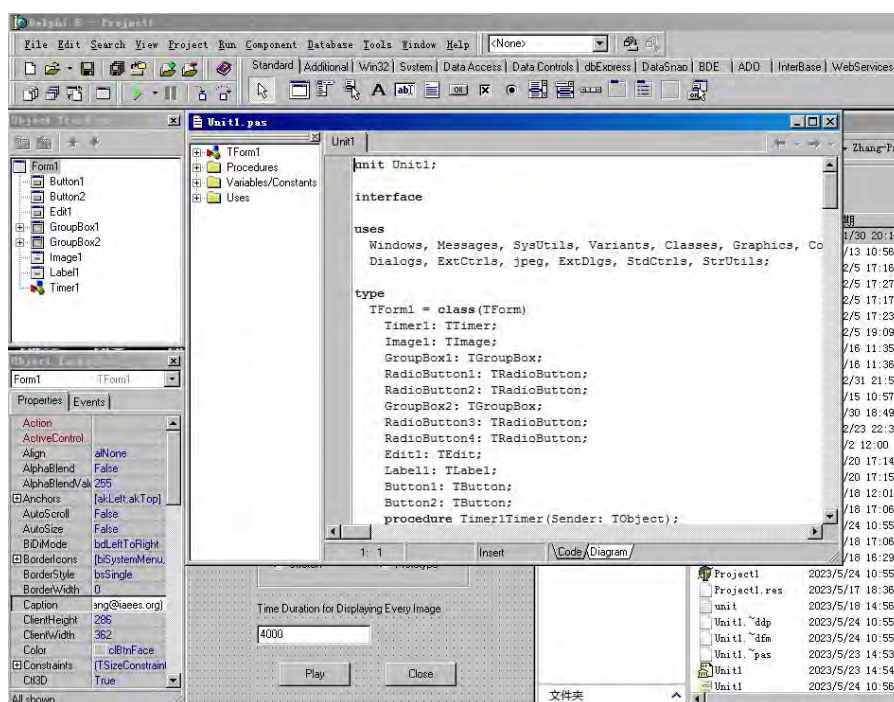

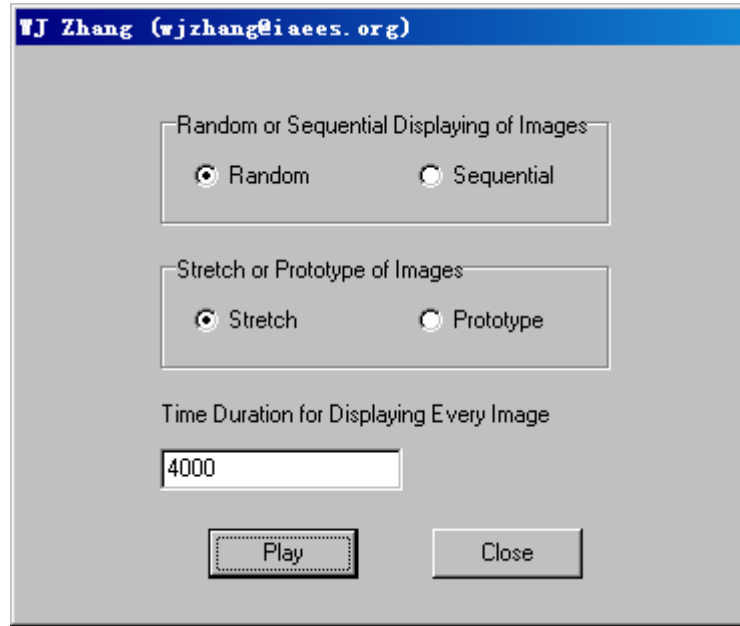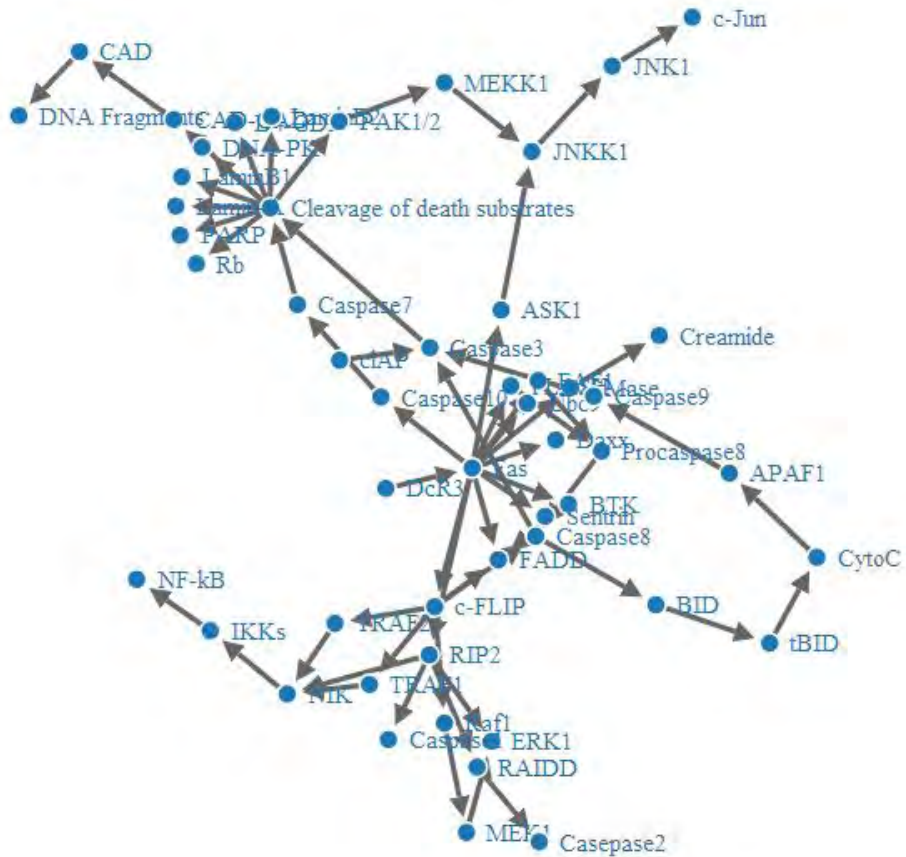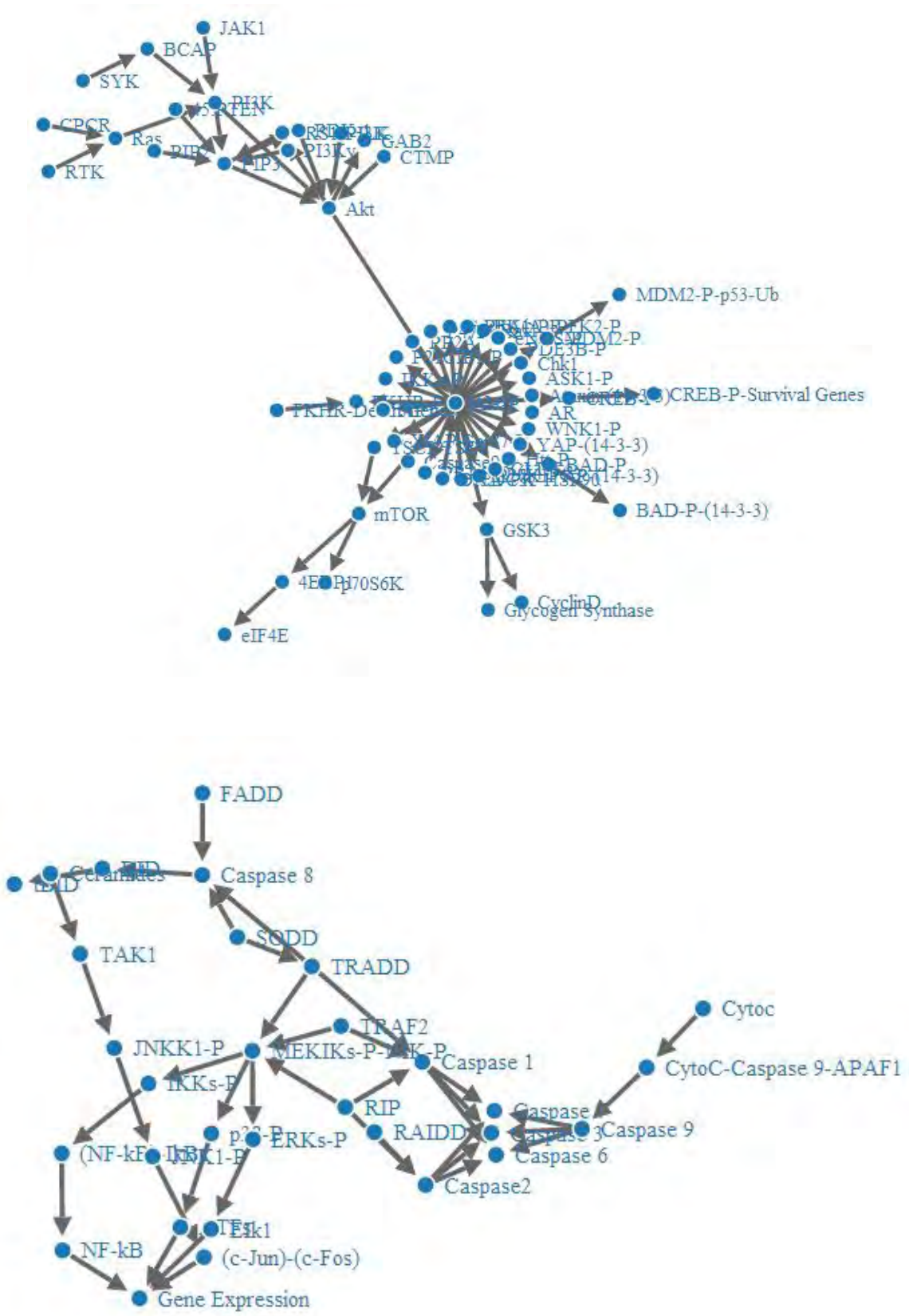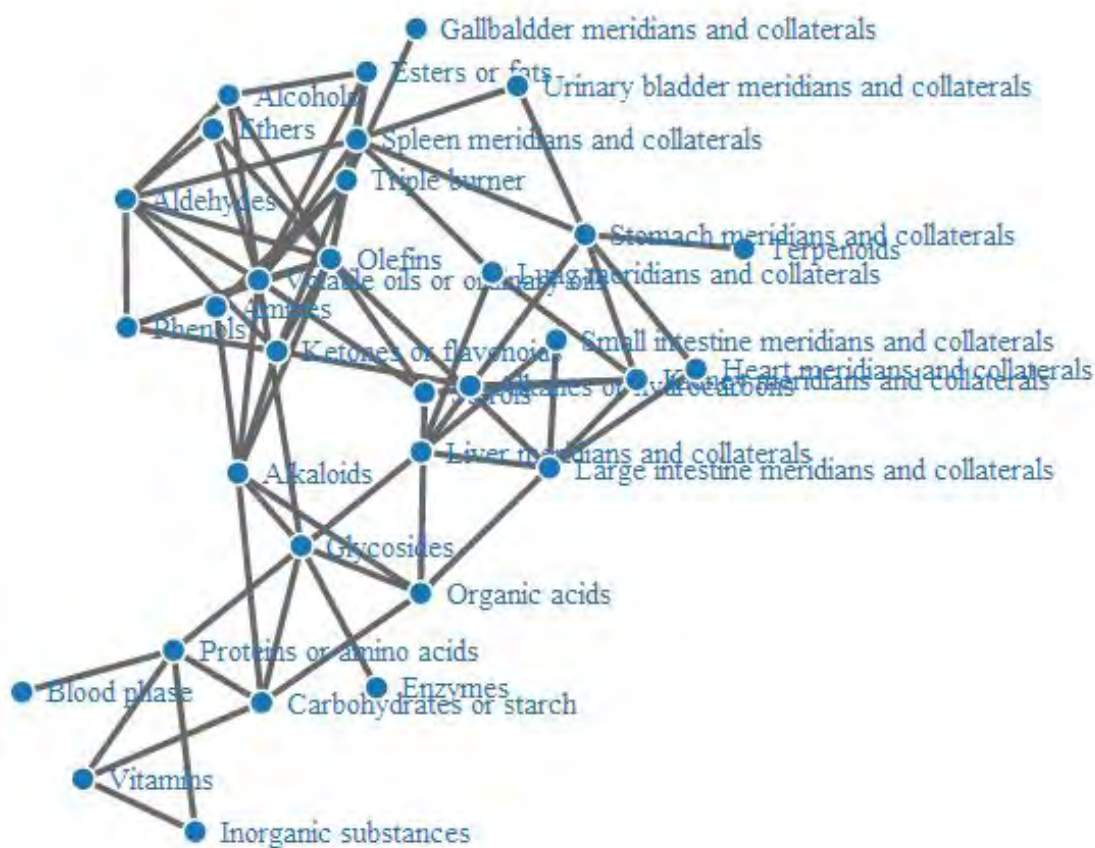**Fig. 1** The Delphi development environment of the software, Projector 1.0.exe.
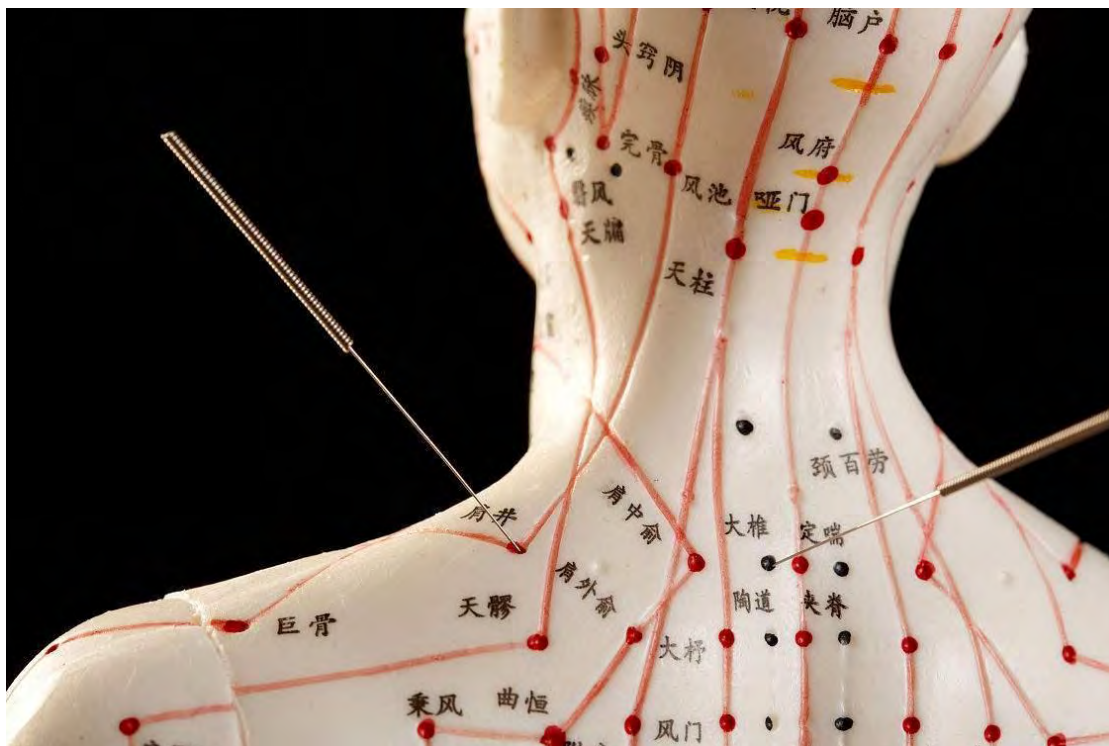
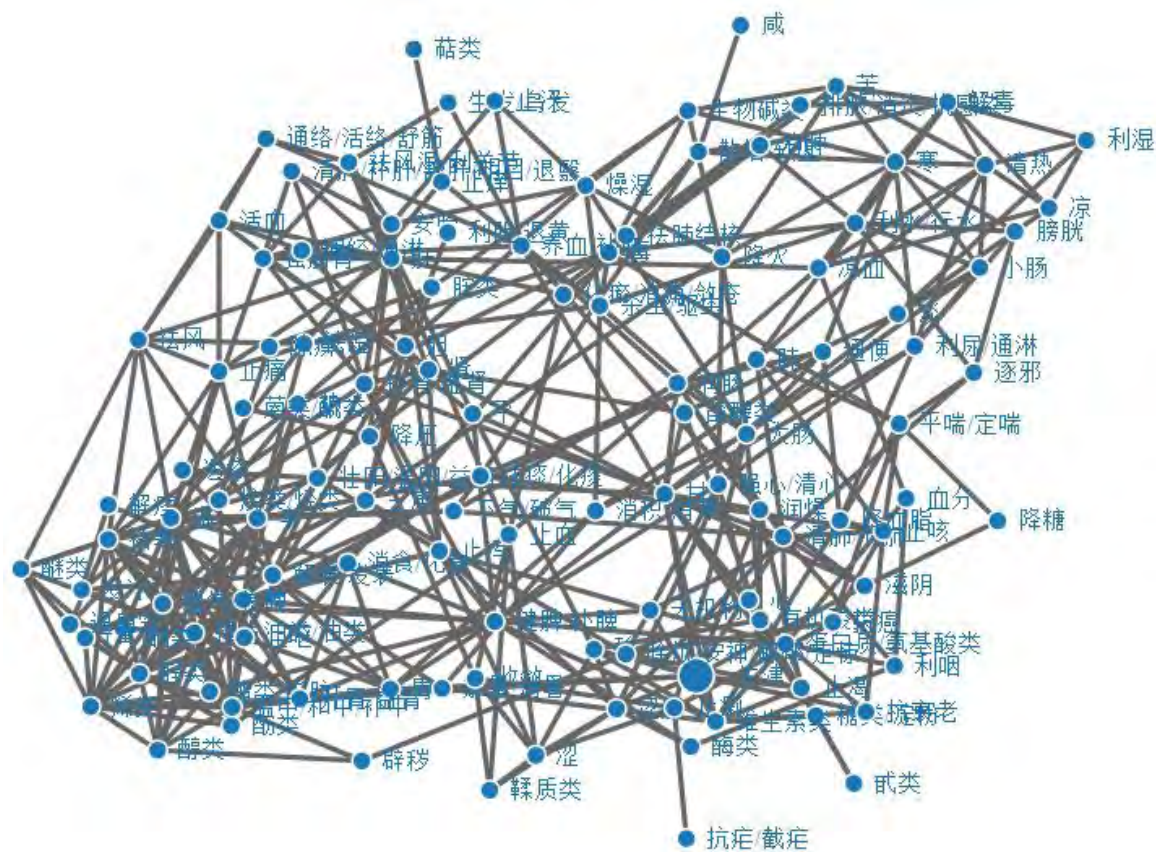**Fig. 2** The parameter input interface of the software, Projector 1.0.exe.

Fig. 3 Some pharmacological network related images attached in the package.

References

Zhang WJ. 2012. A Java software for drawing graphs. Network Biology, 2(1): 38-44

Zhang WJ. 2017. Network pharmacology of medicinal attributes and functions of Chinese herbal medicines: (II) Relational networks and pharmacological mechanisms of medicinal attributes and functions of Chinese herbal medicines. Network Pharmacology, 2(2): 38-66

Zhang WJ. 2021. A web tool for generating user-interface interactive networks. Network Biology, 11(4): 247-262

Zhang WJ. 2023. 3D visualization of objects and molecules: An integrative Java software. Computational Ecology and Software, 13(4): 81-108

Zhang WJ. 2024a. A Matlab software for visualizing user-interface interactive networks. Network Biology, 14(1): 13-19

Zhang WJ. 2024b. A standalone executable software for network visualization. Network Pharmacology, 9(1-2): 1-10

Zhang WJ. 2024c. An executable Java software for visualizing networks. Network Biology, 14(1): 1-12

Zhang WJ. 2024d. netGen 3.0: The executable Java software for network visualization. Selforganizology, 11(1-2): 1-27

Zhang WJ. 2024e. Several digital timers and clocks for desktop computers. Computational Ecology and Software, 14(1): 1-13