

โปรแกรม Emotion ขึ้นมา เป็นโปรแรมที่บันทึกอารมณ์ในแต่ละวันของผู้ใช้ เพื่อให้ผู้ใช้งานได้สำรวจและทบทวนอารมณ์ของตนเองในแต่ละวัน เราได้แบ่งอารมณ์คร่าวๆออกมาเป็น 4 อารมณ์ คือ สุข เศร้า โกรธ และเครียด โดยใน 1 วันจะให้ผู้ใช้งานเลือกอารมณ์ที่รู้สึกมากที่สุดวันนั้น และให้ระดับของอารมณ์นั้นเป็นเปอร์เซ็น และเก็บอารมณ์ในวันนั้นไว้ในระบบ และสามารถดูย้อนหลังได้ว่า วันไหนเรารู้สึกอย่างไร

**การทำงาน**

โปรแกรมจะแสดงหน้า menu ให้เลือก มี 6 หัวข้อ คือ

1. HOW DO YOU FEEL TODAY?

* สร้าง node และบันทึกข้อมูล คือ emotion และ rate
* ใช้ function **void feeltoday(Emotion\* head,Emotion\* tail,Emotion\*hh)** ในการสร้าง node และบันทึกข้อมูล
* หลังจากกรอกเสร็จแล้วพิมพ์ H เพื่อกลับหน้า menu

1. SHOW PREVIOUS DAY
   * ให้เลือกวันที่ต้องการจะดูมา 1 วัน
   * โปรแกรมจะแสดงข้อมูลของวันที่เลือกมา
   * ใช้ function **int checkday(Emotion\* head,Emotion\* tail,Emotion\*hh)** เพื่อตรวจสอบว่า โปรแกรมเก็บข้อมูลเอาไว้กี่วัน
   * ใช้ function **void selectday(Emotion\* head,Emotion\* tail,Emotion\*hh)** ในการหาวัน เลือกวันที่ต้องการจะรู้ และแสดงข้อมูลที่เก็บเอาไว้
   * ถ้าหากยังไม่เคยมีการเป็บข้อมูลมาก่อน โปรแกรมจะแสดงว่า DON'T HAVE PREVIOUS DAY. PLEASE DO YOUR FIRST DAY.
   * หลังจากดูเสร็จแล้วพิมพ์ H เพื่อกลับหน้า menu
2. MAXIMUM OR MINIMUM FEELS
   * เลือกแสดงวันที่ rate มากที่สุดหรือน้อยที่สุด
   * ใช้ function **void Emotion::sorting(Emotion\* head,Emotion\* tail,Emotion\* hh)** ในการ sorting ข้อมูลโดยการเทียบ rate และให้ผู้ใช้เลือกว่าจะดู max หรือ min
   * ถ้าหากยังไม่เคยมีการเป็บข้อมูลมาก่อน โปรแกรมจะแสดงว่า DON'T HAVE PREVIOUS DAY. PLEASE DO YOUR FIRST DAY.
   * หลังจากดูเสร็จแล้วพิมพ์ H เพื่อกลับหน้า menu
3. LOAD YOUR SAVE
   * Load save ที่เคยกรอกก่อนหน้าไปแล้ว
   * ใช้ function **void load\_save(Emotion\* head,Emotion\* tail,Emotion\*hh)** ในการโหลดเซฟที่เคยเขียนเอาไว้ในการเปิดโปรแกรมครั้งก่อน
   * หลังจากโหลดเสร็จแล้วพิมพ์ H เพื่อกลับหน้า menu
4. DELETE YOUR SAVE
   * ลบเซฟที่เคยกรอกก่อนหน้า
   * ใช้ function **void delfile(Emotion\* head,Emotion\* tail,Emotion\*hh)**
   * หลังจากลบเสร็จแล้วพิมพ์ H เพื่อกลับหน้า menu
5. QUIT ปิดโปรแกรม

**Class ที่ใช้**

* **class Emotion**

{

protected:

int rate;

Emotion \*next;

public:

virtual void name(){}

virtual void quote(){}

static void sorting(Emotion\*,Emotion\*,Emotion\*);

Emotion(int a=0){

rate=a;

next=NULL;

}

void append(Emotion \*a){

next=a;

}

Emotion\* movenext(){

return next;

}

};

* **class happy :public Emotion{**

private:

string happya="You are absolutely delighted";

string happyb="You are over the moon";

string happyc="You could not be happier";

public:

void name()

{

char bufferi[30];

sprintf(bufferi,"HAPPY %d %%",rate);

outtextxy(420,190,bufferi);

}

/\*void quote()

{

cout<<"happy ";//ได้แล้ว

}\*/

happy(int a):Emotion(a)

{

if(a>60)

{

char \*bufferm = new char(happya.length()+1);

strcpy(bufferm, happya.c\_str());

outtextxy(400, 170, bufferm);

//quote();

}

else if(a>30&&a<=60)

{

char \*bufferm = new char(happyb.length()+1);

strcpy(bufferm, happyb.c\_str());

outtextxy(400, 170, bufferm);

}

else if(a>=0&&a<=30)

{

char \*bufferm = new char(happyc.length()+1);

strcpy(bufferm, happyc.c\_str());

outtextxy(400, 170, bufferm);

}

}

};

* **class sad :public Emotion{**

private:

string sada="You feel blue";

string sadb="You are down in dumps";

string sadc="Your heart sink";

public:

void name()

{

char bufferi[30];

sprintf(bufferi,"SAD %d %%",rate);

outtextxy(420,190,bufferi);

}

/\*void quote()

{

cout<<"sad ";

}\*/

sad(int a):Emotion(a)

{

if(a>60)

{

char \*bufferm = new char(sada.length()+1);

strcpy(bufferm, sada.c\_str());

outtextxy(400, 170, bufferm);

//quote();

}

else if(a>30&&a<=60)

{

char \*bufferm = new char(sadb.length()+1);

strcpy(bufferm, sada.c\_str());

outtextxy(400, 170, bufferm);

}

else if(a>=0&&a<=30)

{

char \*bufferm = new char(sadc.length()+1);

strcpy(bufferm, sada.c\_str());

outtextxy(420, 170, bufferm);

}

}

};

* **class angry :public Emotion{**

private:

string angrya="You are angry";

string angryb="You are pissed";

string angryc="You are enraged";

public:

void name()

{

char bufferi[30];

sprintf(bufferi,"ANGRY %d %%",rate);

outtextxy(420,190,bufferi);

}

/\*void quote()

{

cout<<"angry ";

}\*/

angry(int a):Emotion(a)

{

if(a>60)

{

char \*bufferm = new char(angrya.length()+1);

strcpy(bufferm, angrya.c\_str());

outtextxy(400, 170, bufferm);

//quote();

}

else if(a>30&&a<=60)

{

char \*bufferm = new char(angryb.length()+1);

strcpy(bufferm, angryb.c\_str());

outtextxy(400, 170, bufferm);

}

else if(a>=0&&a<=30)

{

char \*bufferm = new char(angryc.length()+1);

strcpy(bufferm, angryc.c\_str());

outtextxy(400, 170, bufferm);

}

}

};

* **class stress :public Emotion{**

private:

string stressa="You are anxiety";

string stressb="You are worry";

string stressc="You are stressed";

public:

void name()

{

char bufferi[30];

sprintf(bufferi,"STRESS %d %%",rate);

outtextxy(420,190,bufferi);

}

/\*void quote()

{

cout<<"stress ";

}\*/

stress(int a):Emotion(a)

{

if(a>60)

{

char \*bufferm = new char(stressa.length()+1);

strcpy(bufferm, stressa.c\_str());

outtextxy(400, 170, bufferm);

//quote();

}

else if(a>30&&a<=60)

{

char \*bufferm = new char(stressb.length()+1);

strcpy(bufferm, stressb.c\_str());

outtextxy(400, 170, bufferm);

}

else if(a>=0&&a<=30)

{

char \*bufferm = new char(stressc.length()+1);

strcpy(bufferm, stressc.c\_str());

outtextxy(400, 170, bufferm);

}

}

};

**Main**

* **int main()**

{

Emotion \*head=NULL,\*tail=NULL,\*hh;

system("color F9");

menu(head,tail,hh);

}

**Function**

* **char menuchoice(Emotion\* head,Emotion\* tail,Emotion\*hh)**

{

char n;

initwindow(1000,450,"");

settextstyle(COMPLEX\_FONT, HORIZ\_DIR, 12);

readimagefile("emotion.jpg",85,50,300,390);

cout<<z;

outtextxy(400, 130, "1.>> HOW DO YOU FEEL TODAY? <<");

outtextxy(400, 150, "2.>> SHOW PREVIOUS DAY <<");

outtextxy(400, 170, "3.>> MAXIMUM OR MINIMUM FEELS <<");

outtextxy(400, 190, "4.>> LOAD YOUR SAVE <<");

outtextxy(400, 210, "5.>> DELETE YOUR SAVE <<");

outtextxy(400, 230, "6.>> QUIT <<");

n=getch();

return n;

}

* **void menu(Emotion\* head,Emotion\* tail,Emotion\*hh)**

{

char choice;

do

{

choice=menuchoice(head,tail,hh);

switch(choice)

{

case '1':

feeltoday(head,tail,hh);

break;

case '2':

selectday(head,tail,hh);

break;

case '3':

Emotion::sorting(head,tail,hh);

break;

case '4':

load\_save(head,tail,hh);

break;

case '5':

delfile(head,tail,hh);

break;

case '6':

closegraph();

exit(0);

break;

default:

closegraph();

break;

}

}

while(choice!='6');

}

* **void gohome(Emotion\* head,Emotion\* tail,Emotion\*hh)**

{

char m;

do

{

outtextxy(400, 230, "PRESS --> H <-- TO GO BACK TO MENU");

m=getch();

}

while(m!='H'&&m!='h');

closegraph();

menu(head,tail,hh);

}

* **void feeltoday(Emotion\* head,Emotion\* tail,Emotion\*hh)**

{

cleardevice();

int rate=0;

string mood;

std::fstream file;//start file save

file.open("detail.txt",std::ios::in | std::ios::out | std::ios::app);

readimagefile("how do you feel.jpg",85,50,300,390);

outtextxy(400, 130, "HAPPY/SAD/ANGRY/STRESS? : ");

cin>>mood;

z++;

char \*bufferm = new char(mood.length()+1);

strcpy(bufferm, mood.c\_str());

outtextxy(690, 130, bufferm);

for(int i=0; i<mood.size(); i++)

{

char c=tolower(mood[i]);

mood[i]=c;

}

while(mood.compare("happy")!=0&&mood.compare("sad")!=0&&mood.compare("angry")&&mood.compare("stress")!=0)

{

cleardevice();

readimagefile("how do you feel.jpg",85,50,300,390);

outtextxy(400, 130, "HAPPY/SAD/ANGRY/STRESS? : ");

cin>>mood;

char \*bufferm = new char(mood.length()+1);

strcpy(bufferm, mood.c\_str());

outtextxy(690, 130, bufferm);

for(int i=0; i<mood.size(); i++)

{

char c=tolower(mood[i]);

mood[i]=c;

}

}

file<<mood<<" ";//save mood

outtextxy(400, 150,"RATE (0-100%) : ");

cin>>rate;

char bufferi[10];

sprintf(bufferi,"%d",rate);

outtextxy(690, 150,bufferi);

while(!cin>>rate)

{

cleardevice();

readimagefile("how do you feel.jpg",85,50,300,390);

cin.clear();

cin.ignore(50,'\n');

outtextxy(400, 150,"TRY AGAIN! (0-100%) : ");

cin>>rate;

char bufferi[10];

sprintf(bufferi,"%d",rate);

outtextxy(690, 150,bufferi);

}

while(rate<0||rate>100)

{

cleardevice();

readimagefile("how do you feel.jpg",85,50,300,390);

outtextxy(400, 150,"TRY AGAIN! (0-100%) : ");

cin>>rate;

char bufferi[10];

sprintf(bufferi,"%d",rate);

outtextxy(690, 150,bufferi);

}

file<<rate<<endl;

if(!head)

{

if(mood.compare("happy")==0)

{

head=new happy(rate);

hh=head;

tail=head;

}

else if(mood.compare("sad")==0)

{

head=new sad(rate);

hh=head;

tail=head;

}

else if(mood.compare("angry")==0)

{

head=new angry(rate);

hh=head;

tail=head;

}

else if(mood.compare("stress")==0)

{

head=new stress(rate);

hh=head;

tail=head;

}

}

else

{

if(mood.compare("happy")==0)

{

tail->append(new happy(rate));

tail=tail->movenext();

}

else if(mood.compare("sad")==0)

{

tail->append(new sad(rate));

tail=tail->movenext();

}

else if(mood.compare("angry")==0)

{

tail->append(new angry(rate));

tail=tail->movenext();

}

else if(mood.compare("stress")==0)

{

tail->append(new stress(rate));

tail=tail->movenext();

}

}

outtextxy(500,200,"Your detail will be auto save.");

file.close();

gohome(head,tail,hh);

}

* **int checkday(Emotion\* head,Emotion\* tail,Emotion\*hh)**

{

int n=0;

hh=head;

if(head==NULL)

{

return n=0;//no linklist

}

else

{

while(hh!=NULL)

{

n++;

hh=hh->movenext();

}

}

return n;

}

* **void selectday(Emotion\* head,Emotion\* tail,Emotion\*hh)**

{

cleardevice();

int day,allday,i=1;

hh=head;

allday=checkday(head,tail,hh);

readimagefile("SHOW PRE DAY.jpg",85,50,300,390);

outtextxy(400, 130,">> WHICH DAY YOU WANT TO KNOW ? <<");

if(allday==0)

{

outtextxy(400,170,"DON'T HAVE PREVIOUS DAY");

outtextxy(400,190,"PLEASE DO YOUR FIRST DAY");

gohome(head,tail,hh);

}

else

{

char bufferi[30];

sprintf(bufferi,"SELECT DAY (1-%d) --> ",allday);

outtextxy(400,150,bufferi);

cin>>day;

while(!cin>>day)

{

cleardevice();

cin.clear();

readimagefile("SHOW PRE DAY.jpg",85,50,300,390);

outtextxy(400, 130,">> WHICH DAY YOU WANT TO KNOW ? <<");

cin.ignore(50,'\n');

char bufferi[30];

sprintf(bufferi,"SELECT DAY (1-%d) --> ",allday);

outtextxy(400,150,bufferi);

cin>>day;

}

while(day<1||day>allday)

{

cleardevice();

readimagefile("SHOW PRE DAY.jpg",85,50,300,390);

outtextxy(400, 130,">> WHICH DAY YOU WANT TO KNOW ? <<");

char bufferi[30];

sprintf(bufferi,"SELECT DAY (1-%d) --> ",allday);

outtextxy(400,150,bufferi);

cin>>day;

}

while(day!=i)

{

hh=hh->movenext();

i++;

}

char bufferj[5];

sprintf(bufferj,"%d",day);

outtextxy(400,190,bufferj);

hh->name(); //broken here

gohome(head,tail,hh);

}

}

* **void Emotion::sorting(Emotion\* head,Emotion\* tail,Emotion\* hh)**

{

cleardevice();

int allday,maxi=-1,mini=101;

string maxmin;

hh=head;

readimagefile("max min.jpg",85,50,300,390);

outtextxy(400, 130,">> MAXIMUM OR MINIMUM FEELS <<");

allday=checkday(head,tail,hh);

if(allday==0)

{

outtextxy(400,170,"DON'T HAVE PREVIOUS DAY");

outtextxy(400,190,"PLEASE DO YOUR FIRST DAY");

gohome(head,tail,hh);

}

outtextxy(400,150,"CHOOSE (MAX/MIN) --> ");

cin>>maxmin;

char \*bufferm = new char(maxmin.length()+1);

strcpy(bufferm, maxmin.c\_str());

outtextxy(650, 150, bufferm);

for(int i=0; i<maxmin.size(); i++)

{

char c=tolower(maxmin[i]);

maxmin[i]=c;

}

while(maxmin.compare("max")&&maxmin.compare("min"))

{

cleardevice();

outtextxy(400,150,"CHOOSE (MAX/MIN) --> ");

cin>>maxmin;

char \*bufferm = new char(maxmin.length()+1);

strcpy(bufferm, maxmin.c\_str());

outtextxy(650, 150, bufferm);

for(int i=0; i<maxmin.size(); i++)

{

char c=tolower(maxmin[i]);

maxmin[i]=c;

}

}

if(maxmin.compare("max")==0)

{

hh=head;

while(hh!=NULL)

{

if(maxi<hh->rate)

{

maxi=hh->rate;

}

hh=hh->movenext();

}

hh=head;

while(hh!=NULL)

{

if(maxi==hh->rate)

{

hh->name();

}

hh=hh->movenext();

}

gohome(head,tail,hh);

}

else if(maxmin.compare("min")==0)

{

hh=head;

while(hh!=NULL)

{

if(mini>hh->rate)

{

mini=hh->rate;

}

hh=hh->movenext();

}

hh=head;

while(hh!=NULL)

{

if(mini==hh->rate)

{

hh->name();

}

hh=hh->movenext();

}

gohome(head,tail,hh);

}

}

* **void load\_save(Emotion\* head,Emotion\* tail,Emotion\*hh)**

{

cleardevice();

ifstream file;

file.open("detail.txt",std::ios::in);

readimagefile("load save.jpg",85,50,300,390);

string mood\_s;

int rate\_s;

if(file)

{

if(z==0)

{

if(a==0)

{

while(file>>mood\_s>>rate\_s)

{

if(!head)

{

if(mood\_s.compare("happy")==0)

{

head=new happy(rate\_s);

hh=head;

tail=head;

}

else if(mood\_s.compare("sad")==0)

{

head=new sad(rate\_s);

hh=head;

tail=head;

}

else if(mood\_s.compare("angry")==0)

{

head=new angry(rate\_s);

hh=head;

tail=head;

}

else if(mood\_s.compare("stress")==0)

{

head=new stress(rate\_s);

hh=head;

tail=head;

}

}

else

{

if(mood\_s.compare("happy")==0)

{

tail->append(new happy(rate\_s));

tail=tail->movenext();

}

else if(mood\_s.compare("sad")==0)

{

tail->append(new sad(rate\_s));

tail=tail->movenext();

}

else if(mood\_s.compare("angry")==0)

{

tail->append(new angry(rate\_s));

tail=tail->movenext();

}

else if(mood\_s.compare("stress")==0)

{

tail->append(new stress(rate\_s));

tail=tail->movenext();

}

}

cleardevice();

readimagefile("load save.jpg",85,50,300,390);

outtextxy(500,200,"Load success");

}

a++;

}

else

outtextxy(500,200,"Your save already loaded.");

}

else

outtextxy(500,200,"You can't load save.");

}

else

outtextxy(500,200,"No Save File");

file.close();

gohome(head,tail,hh);

}

* **void delfile(Emotion\* head,Emotion\* tail,Emotion\*hh)**

{

cleardevice();

readimagefile("delete.jpg",85,50,300,390);

outtextxy(400, 130, "Your save file is deleted.");

remove("detail.txt");

a=0;

gohome(head,tail,hh);

}