期中報告

組員:陳敬橋、林明宏、許遠健

MODE INTRODUTION

1	2	3
NORMAL	SPECIAL-1	SPECIAL-2
正常模式	消除玩法 搶三分	大吃小

開始畫面

```c
char a[3][3];//OOXX棋盤二維陣列
char b[]={'O','X'};//判斷輸贏時使用
char b2[]={'o','x'};
int mode,k=0,O=0,X=0,X_quantity,x_quantity,O_quantity,o_quantity;//O X模式1計分 k平手
int x_memory[4][2]={{-1,-1},{-1,-1},{-1,-1},{-1,-1}};//特殊模式時用來記憶先後順序，設-1為空值
int o_memory[4][2]={{-1,-1},{-1,-1},{-1,-1},{-1,-1}};//特殊模式時用來記憶先後順序，設-1為空值
```

```c
int main(void)
{
    while(1)
    {
        clean();
        printf("%45s","<Weclome to OOXX>\n");
        puts("================================================================================");
        printf("%52s","<Please Choose Mods You Want>\n");
        puts("|------------------------------------------------------------------------------|");
        puts("|                                                                              |");
        puts("|                                  0:Normal                                    |");
        puts("|                                  1:Special-1                                 |");
        puts("|                                  2:Special-2                                 |");
        puts("|                                 -1:Exit                                      |");
        puts("|                                                                              |");
        puts("|------------------------------------------------------------------------------|");
        scanf("%d",&mode);
        if(mode<3&&mode>-2){
            printf("Loding Game");
            printf("....");
            Sleep(1000);
            printf(".....");
            Sleep(1000);
            printf(".......");
            Sleep(1000);
            printf("......\n\n");
            printf("棋盤演示\n\n");
            printf("00|01|02\n");
            printf("--------\n");
            printf("10|11|12\n");
            printf("--------\n");
            printf("20|21|22\n\n");
            puts("================================================================================
        }
```

```
                                <Weclome to OOXX>
================================================================================
                            <Please Choose Mods You Want>
|------------------------------------------------------------------------------|
|                                                                              |
|                                  0:Normal                                    |
|                                  1:Special-1                                 |
|                                  2:Special-2                                 |
|                                 -1:Exit                                      |
|                                                                              |
|------------------------------------------------------------------------------|
0
Loding Game.....

棋盤演示

00|01|02
--------
10|11|12
--------
20|21|22
```

Mode1

```
X|O|
-----
 | |

Enter the row and column for X:0 0
X| |
-----
X|O|
-----
 | |

Enter the row and column for O:0 2
X| |O
-----
X|O|
-----
 | |

Enter the row and column for X:2 0
X| |O
-----
X|O|
-----
X| |

X win
```

```
if(mode==0)
{
    show();
    while(1)
    {
        int end=0;//設end參數
        stepx();//輸入X
        k++;
        show();//顯示棋盤
        end=win();//判斷連線並接收end值
        if (end==1)//當end=1結束
        {
            break;
        }
        else if(k==9)
        {
            printf("This game ended in a tie.\n");
            break;
        }
        stepO();//輸入O
        k++;
        show();//顯示棋盤
        end=win();//判斷連線並接收end值
        if (end==1)//當end=1結束
        {
            break;
        }
        else if(k==9)
        {
            printf("This game ended in a tie.\n");
            break;
        }
    }
}
```

Mode2

New Tab

什麼是搶三分模式

一、當叉或圈放超過四個，消除第一個放的叉或圈

二、當叉或圈其中一個連成一條，則該條會消除，並記錄叉或圈贏一次

三、當叉或圈其中一個先達到三分就勝利

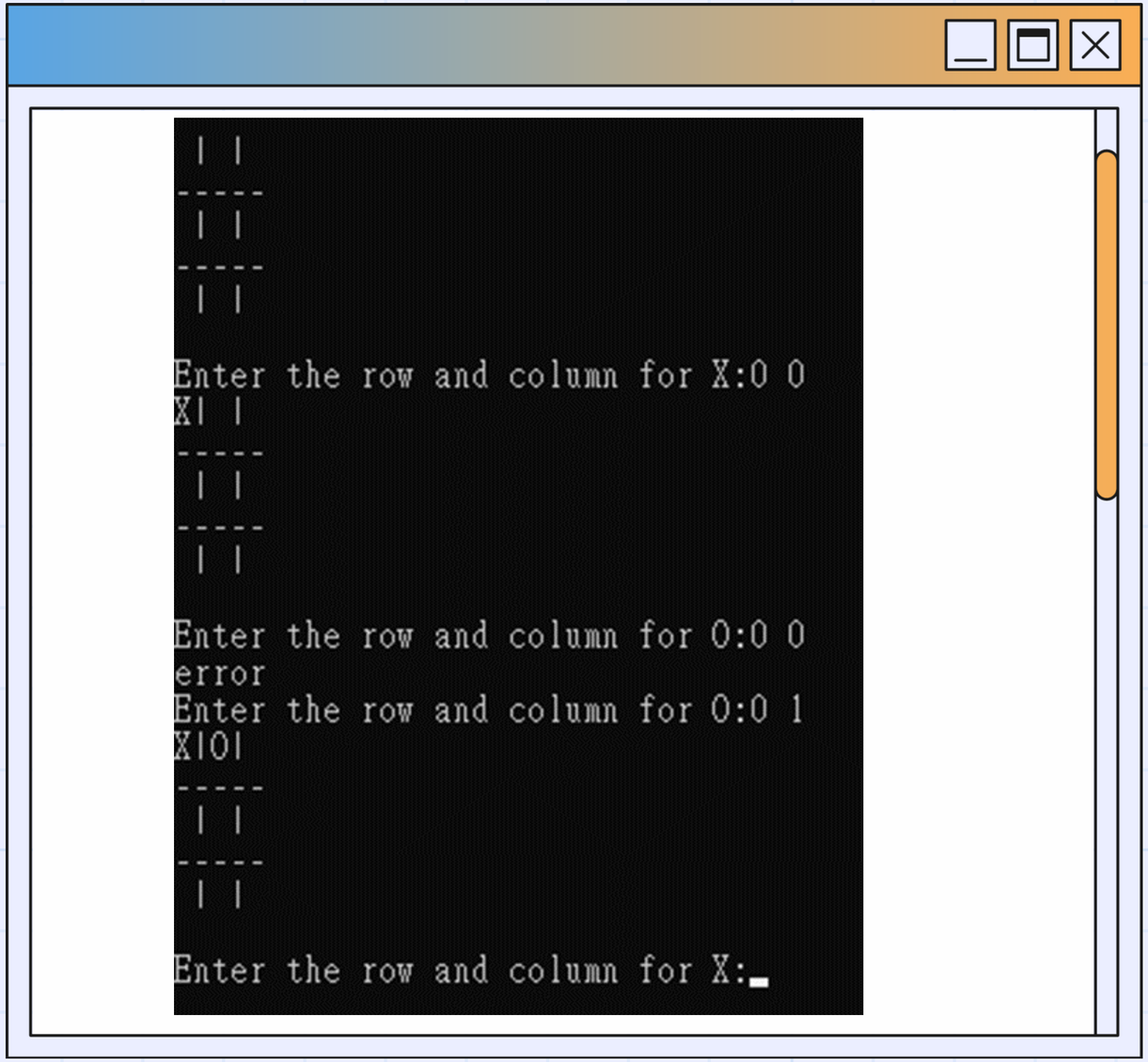
輸入X

```c
void stepx(void)//輸入X
{
    int x,y,i,j;//設XY為座標變數
    while(1)//直到正確前不斷重複
    {
        printf("Enter the row and column for X:");
        getchar();
        scanf("%c %c",&x,&y);
        if (a[x-48][y-48]!='\0')//判斷是否下在空格上
        {
            printf("error\n");
        }
        else if(x>50||x<48||y>50||y<48)//判斷是否超出範圍
        {
            printf("error\n");
        }
        else//將座標設為X，並跳出
        {
            x=x-48;
            y=y-48;
            a[x][y]='X';
            j=0;
            while(1)//特殊模式使用
            {
                if(x_memory[j][0]==-1&&x_memory[j][1]==-1)//如果第j格為空值，放入第j
                {
                    x_memory[j][0]=x;
                    x_memory[j][1]=y;
                    break;
                }
                j++;
            }
            break;
        }
    }
}
```

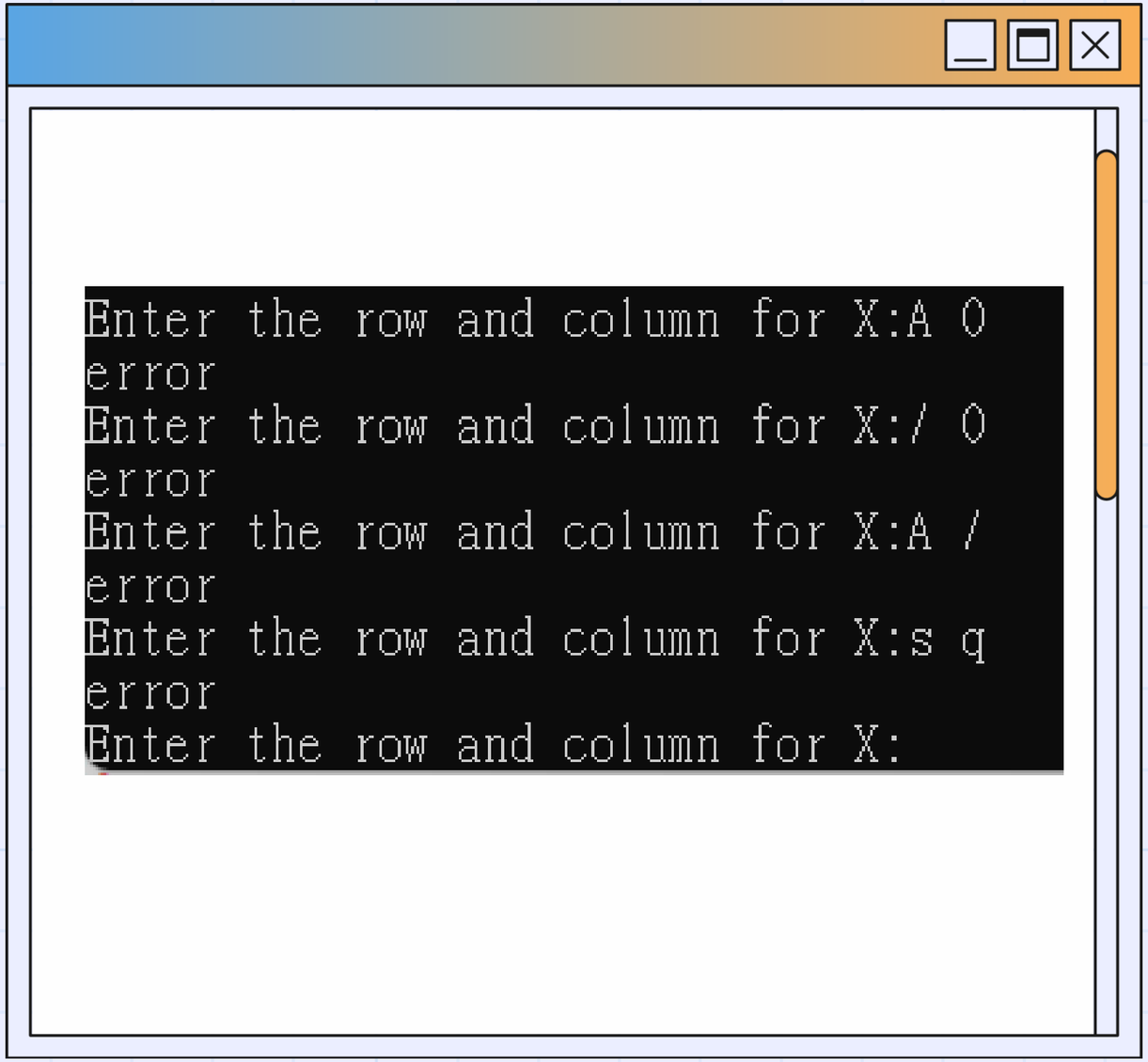
輸入O

```c
void stepO(void)//輸入O
{
    int x,y,i,j;
    while(1)//直到正確前不斷重複
    {
        printf("Enter the row and column for O:");
        getchar();
        scanf("%c %c",&x,&y);
        if (a[x-48][y-48]!='\0')//判斷是否下在空格上
        {
            printf("error\n");
        }
        else if(x>50||x<48||y>50||y<48)//判斷是否超出範圍
        {
            printf("error\n");
        }
        else//將座標設為O，並跳出
        {
            x=x-48;
            y=y-48;
            a[x][y]='O';
            j=0;
            while(1)//特殊模式使用
            {
                if(o_memory[j][0]==-1&&o_memory[j][1]==-1)//如果第j格為空值，放入第
                {
                    o_memory[j][0]=x;
                    o_memory[j][1]=y;
                    break;
                }
                j++;
            }
            break;
        }
    }
}
```

O和X輸入頁面



過濾頁面



判斷橫線連線

```
int win(void)//判斷連線
{
    int i,OX=0,re,j;//i直橫計數共三次//OX判斷OX共兩次//回傳值
    for(OX=0;OX<2;OX++)
    {
        for(i=0;i<3;i++)
        {
            if(a[i][0]==a[i][1]&&a[i][0]==a[i][2]&&a[i][0]==b[OX])//判斷橫線
            {
                printf("%c win\n\n",b[OX]);
                re=1;
                a[i][0]='\0';//特殊模式時，將連線格清空
                a[i][1]='\0';//特殊模式時，將連線格清空
                a[i][2]='\0';//特殊模式時，將連線格清空
                if(b[OX]=='O')//特殊模式時，將連線格記憶清空
                {
                    for(j=0;j<4;j++)
                    {
                        if(o_memory[j][0]==i)//如果記憶與連線在同一橫線上，刪除此格記憶
                        {
                            o_memory[j][0]=-1;
                            o_memory[j][1]=-1;
                        }
                    }
                    O++;
                }
                else if(b[OX]=='X')//特殊模式時，將連線格記憶清空
                {
                    for(j=0;j<4;j++)
                    {
                        if(x_memory[j][0]==i)//如果記憶與連線在同一橫線上，刪除此格記憶
                        {
                            x_memory[j][0]=-1;
                            x_memory[j][1]=-1;
                        }
                    }
                    X++;
                }
            }
        }
    }
```

判斷直線連線

```
        for(i=0;i<3;i++)
        {
            if(a[0][i]==a[1][i]&&a[0][i]==a[2][i]&&a[0][i]==b[OX])//判斷直線
            {
                printf("%c win\n\n",b[OX]);
                re=1;
                a[0][i]='\0';//特殊模式時，將連線格清空
                a[1][i]='\0';//特殊模式時，將連線格清空
                a[2][i]='\0';//特殊模式時，將連線格清空
                if(b[OX]=='O')//特殊模式時，將連線格記憶清空
                {
                    for(j=0;j<4;j++)
                    {
                        if(o_memory[j][1]==i)//如果記憶與連線在同一直線上，刪除
                        {
                            o_memory[j][0]=-1;
                            o_memory[j][1]=-1;
                        }
                    }
                    O++;
                }
                else if(b[OX]=='X')//特殊模式時，將連線格記憶清空
                {
                    for(j=0;j<4;j++)
                    {
                        if(x_memory[j][1]==i)//如果記憶與連線在同一直線上，刪除
                        {
                            x_memory[j][0]=-1;
                            x_memory[j][1]=-1;
                        }
                    }
                    X++;
                }
            }
        }
    }
```

判斷斜線連線

```
if(a[0][0]==a[1][1]&&a[0][0]==a[2][2]&&a[1][1]==b[OX] )//判斷斜線
{
    printf("%c win\n\n",b[OX]);
    re=1;
    a[0][0]='\0';//特殊模式時，將連線格清空
    a[1][1]='\0';//特殊模式時，將連線格清空
    a[2][2]='\0';//特殊模式時，將連線格清空
    if(b[OX]=='O')//特殊模式時，將連線格記憶清空
    {
        for(j=0;j<4;j++)
        {
            if(o_memory[j][0]==o_memory[j][1])//如果記憶與連線在同一
            {
                o_memory[j][0]=-1;
                o_memory[j][1]=-1;
            }
        }
        O++;
    }
    else if(b[OX]=='X')//特殊模式時，將連線格記憶清空
    {
        for(j=0;j<4;j++)
        {
            if(x_memory[j][0]==x_memory[j][1])//如果記憶與連線在同一
            {
                x_memory[j][0]=-1;
                x_memory[j][1]=-1;
            }
        }
        X++;
    }
}
```

```
if(a[0][2]==a[1][1]&&a[0][2]==a[2][0]&&a[1][1]==b[OX])
{
    printf("%c win\n\n",b[OX]);
    re=1;
    a[0][2]='\0';//特殊模式時，將連線格清空
    a[1][1]='\0';//特殊模式時，將連線格清空
    a[2][0]='\0';//特殊模式時，將連線格清空
    if(b[OX]=='O')//特殊模式時，將連線格記憶清空
    {
        for(j=0;j<4;j++)
        {
            if(o_memory[j][0]==0&&o_memory[j][1]==2)//如果記憶與連線在同一斜線上，刪除此格記憶
            {
                o_memory[j][0]=-1;
                o_memory[j][1]=-1;
            }
            if(o_memory[j][0]==1&&o_memory[j][1]==1)//如果記憶與連線在同一斜線上，刪除此格記憶
            {
                o_memory[j][0]=-1;
                o_memory[j][1]=-1;
            }
            if(o_memory[j][0]==2&&o_memory[j][1]==0)//如果記憶與連線在同一斜線上，刪除此格記憶
            {
                o_memory[j][0]=-1;
                o_memory[j][1]=-1;
            }
        }
        O++;
    }
else if(b[OX]=='X')//特殊模式時，將連線格記憶清空
{
    for(j=0;j<4;j++)
    {
        if(x_memory[j][0]==0&&x_memory[j][1]==2)//如果記憶與連線在同一斜線上，刪除此格記憶
        {
            x_memory[j][0]=-1;
            x_memory[j][1]=-1;
        }
        if(x_memory[j][0]==1&&x_memory[j][1]==1)//如果記憶與連線在同一斜線上，刪除此格記憶
        {
            x_memory[j][0]=-1;
            x_memory[j][1]=-1;
        }
        if(x_memory[j][0]==2&&x_memory[j][1]==0)//如果記憶與連線在同一斜線上，刪除此格記憶
        {
            x_memory[j][0]=-1;
            x_memory[j][1]=-1;
        }
    }
    X++;
}
```

Mode2消除連線

當有連線時:

```
if(re==1)//當有連線時
{
    if(mode==1)//特殊模式時
    {
        show();//顯示連線之後的結果
        push_x();//將o往前推
        push_o();//將x往前推
        printf("O:X\n%d:%d\n\n",O,X);//顯示目前比分
    }
    return(1);
}
```

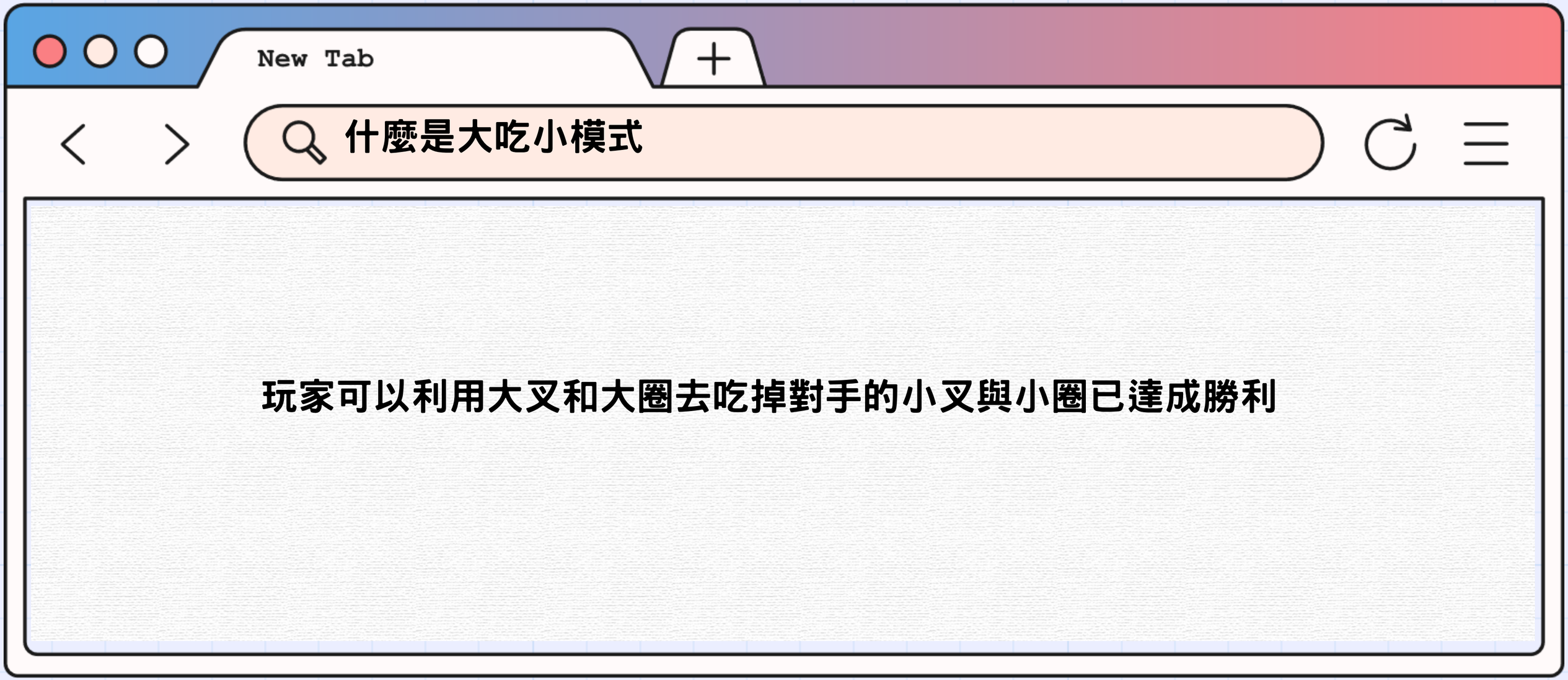
當沒有連線時:

```
else//當沒有連線時
{
    if(mode==1)//特殊模式時
    {
        if(x_memory[3][0]!=-1&&x_memory[3][1]!=-1)//當x下到第4步
        {
            a[x_memory[0][0]][x_memory[0][1]]='\0';//第1步位置清空
            x_memory[0][0]=-1;//第1步位置記憶清空
            x_memory[0][1]=-1;//第1步位置記憶清空
            show();//顯示刪除之後的結果
            push_x();//將x往前推
        }
        if(o_memory[3][0]!=-1&&o_memory[3][1]!=-1)//當下o到第4步
        {
            a[o_memory[0][0]][o_memory[0][1]]='\0';//第1步位置清空
            o_memory[0][0]=-1;//第1步位置記憶清空
            o_memory[0][1]=-1;//第1步位置記憶清空
            show();//顯示刪除之後的結果
            push_o();//將x往前推
        }
    }
    return(0);
}
```

Mode2-Push

```
void push_x(void)//將後面位置往前遞補
{
    int i,j;
    for(j=0;j<3;j++)
    {
        for(i=0;i<3;i++)
        {
            if(x_memory[i][0]==-1&&x_memory[i+1][0]!=-1)//如果前一格是空值，遞補
            {
                x_memory[i][0]=x_memory[i+1][0];
                x_memory[i][1]=x_memory[i+1][1];
                x_memory[i+1][0]=-1;
                x_memory[i+1][1]=-1;
            }
        }
    }
}
```

Mode3



Mode3

主程式

```
Chose X or x(X:2 x:4):error
Chose X or x(X:2 x:4):x
Enter the row and column for X:1 1
 | |
-----
 |x|
-----
 | |

Chose O or o(O:2 o4):error
Chose O or o(O:2 o4):o
Enter the row and column for O:1 2
 | |
-----
 |x|o
-----
 | |

Chose X or x(X:2 x:3):error
Chose X or x(X:2 x:3):X
Enter the row and column for X:0 0
X| |
-----
 |x|o
-----
 | |

Chose O or o(O:2 o3):error
Chose O or o(O:2 o3):O
Enter the row and column for O:1 1
X| |
-----
 |O|o
-----
 | |

Chose X or x(X:1 x:3):error
Chose X or x(X:1 x:3):
```

```
        else if(mode==2)
        {
            O_quantity=2;
            X_quantity=2;
            o_quantity=4;
            x_quantity=4;
            show();
            int end=0;
            k=0;
            while(1)
            {
                stepx2();//輸入X
                k++;
                show();//顯示棋盤
                end=win2();
                if (end==1)//當end=1結束
                {
                    break;
                }
                else if(k==9&&X_quantity==0)
                {
                    printf("This game ended in a tie.\n");
                    break;
                }
                step02();//輸入O
                k++;
                show();//顯示棋盤
                end=win2();
                if (end==1)//當end=1結束
                {
                    break;
                }
                else if(k==9&&O_quantity==0)
                {
                    printf("This game ended in a tie.\n");
                    break;
                }
            }
        }
```

Mode3

叉叉的副程式

```
Chose X or x(X:2 x:4):error
Chose X or x(X:2 x:4):X
Enter the row and column for X:1 1
 | |
-----
 |X|
-----
 | |

Chose O or o(O:2 o4):error
Chose O or o(O:2 o4):O
Enter the row and column for O:0 0
O| |
-----
 |X|
-----
 | |

Chose X or x(X:1 x:4):error
Chose X or x(X:1 x:4):X
Enter the row and column for X:0 2
O| |X
-----
 |X|
-----
 | |

Chose O or o(O:1 o4):error
Chose O or o(O:1 o4):o
Enter the row and column for O:2 0
O| |X
-----
 |X|
-----
o| |

Chose X or x(X:0 x:4):error
Chose X or x(X:0 x:4):X
X is not enough
```

```c
void stepx2(void)//輸入X
{
    int x,y,i,j;//設XY為座標變數
    char Xx;
    while(1)
    {
        printf("Chose X or x(X:%d x:%d):",X_quantity,x_quantity);
        scanf("%c",&Xx);
        if(Xx=='X')
        {
            if(X_quantity==0)
            {
                printf("X is not enough\n");
            }
            else
            {
                X_quantity--;
                break;
            }
        }
        else if(Xx=='x')
        {
            if(x_quantity==0)
            {
                printf("x is not enough\n");
            }
            else
            {
                x_quantity--;
                break;
            }
        }
        else
        {
            printf("error\n");
        }
    }
```

Mode3

判斷是否輸入正確

副程式

```
while(1)//直到正確前不斷重複
{
    printf("Enter the row and column for X:");
    scanf("%d %d",&x,&y);
    if (a[x][y]!='\0'&&Xx=='x')//判斷是否下在空格上
    {
        printf("error1\n");
    }
    else if(a[x][y]=='O'&&Xx=='X')
    {
        printf("error2\n");
    }
    else if(a[x][y]=='x'&&Xx=='X')
    {
        printf("error3\n");
    }
    else if(x>2||x<0||y>2||y<0)//判斷是否超出範圍
    {
        printf("error4\n");
    }
    else//將座標設為X，並跳出
    {
        if(Xx=='x')
        {
            a[x][y]='x';
        }
        else if(Xx=='X')
        {
            if(a[x][y]=='o')
            {
                k--;
            }
            a[x][y]='X';
        }
        break;
    }
}
```

Mode3

圈圈的副程式

```
Chose O or o(O:2 o4):error
Chose O or o(O:2 o4):O
Enter the row and column for O:1 0
 | |
-----
O|X|
-----
 | |

Chose X or x(X:1 x:4):error
Chose X or x(X:1 x:4):X
Enter the row and column for X:0 1
 |X|
-----
O|X|
-----
 | |

Chose O or o(O:1 o4):error
Chose O or o(O:1 o4):O
Enter the row and column for O:2 1
 |X|
-----
O|X|
-----
 |O|

Chose X or x(X:0 x:4):error
Chose X or x(X:0 x:4):x
Enter the row and column for X:1 2
 |X|
-----
O|X|x
-----
 |O|

Chose O or o(O:0 o4):error
Chose O or o(O:0 o4):O
O is not enough
Chose O or o(O:0 o4):error
Chose O or o(O:0 o4):
```

```c
void step02(void)//輸入O
{
    int x,y,i,j;//設XY為座標變數
    char Oo;
    while(1)
    {
        printf("Chose O or o(O:%d o%d):",O_quantity,o_quantity);
        scanf("%c",&Oo);
        if(Oo=='O')
        {
            if(O_quantity==0)
            {
                printf("O is not enough\n");
            }
            else
            {
                O_quantity--;
                break;
            }
        }
        else if(Oo=='o')
        {
            if(o_quantity==0)
            {
                printf("o is not enough\n");
            }
            else
            {
                o_quantity--;
                break;
            }
        }
        else
        {
            printf("error\n");
        }
    }
```

Mode3

判斷連線的副程式

```
int win2(void)//判斷連線
{
    int i,OX=0,re,j;//i直橫計數共三次//OX判斷OX共兩次//回傳值
    for(OX=0;OX<2;OX++)
    {
        for(i=0;i<3;i++)
        {
            if((a[i][0]==b[OX]||a[i][0]==b2[OX]) && (a[i][1]==b[OX]||a[i][1]==b2[OX]) && (a[i][2]==b[OX]||a[i][2]==b2[OX]))//判斷橫線
            {
                printf("%c win\n\n",b[OX]);
                re=1;
            }
        }
        for(i=0;i<3;i++)
        {
            if((a[0][i]==b[OX]||a[0][i]==b2[OX]) && (a[1][i]==b[OX]||a[1][i]==b2[OX]) && (a[2][i]==b[OX]||a[2][i]==b2[OX]))//判斷直線
            {
                printf("%c win\n\n",b[OX]);
                re=1;
            }
        }
        if((a[0][0]==b[OX]||a[0][0]==b2[OX]) && (a[1][1]==b[OX]||a[1][1]==b2[OX]) && (a[2][2]==b[OX]||a[2][2]==b2[OX]) )//判斷斜線
        {
            printf("%c win\n\n",b[OX]);
            re=1;
        }
        if((a[0][2]==b[OX]||a[0][2]==b2[OX]) && (a[2][0]==b[OX]||a[2][0]==b2[OX]) && (a[1][1]==b[OX]||a[1][1]==b2[OX]))
        {
            printf("%c win\n\n",b[OX]);
            re=1;
        }
    }


    if(re==1)//當有連線時
    {
        return(1);
    }
    else//當沒有連線時
    {
        return(0);
    }
```

初始化設定

```
void clean(void)
{
    int i,j;
    for (i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            a[i][j]='\0';
        }
    }
    for (i=0;i<4;i++)
    {
        for(j=0;j<2;j++)
        {
            x_memory[i][j]=-1;
            o_memory[i][j]=-1;
        }
    }
    k=0;
    O=0,X=0;//得分歸0
}
```

分工表與製作流程:

1

程式開發與設計

陳敬橋

2

講解程式邏輯和玩法

陳敬橋

3

設計開頭頁面與字符過濾系統

林明宏

4

梳理程式邏輯並製作PPT DEMO

許遠健

5

設計並優化PPT頁面

林明宏

6

最終報告

許遠健