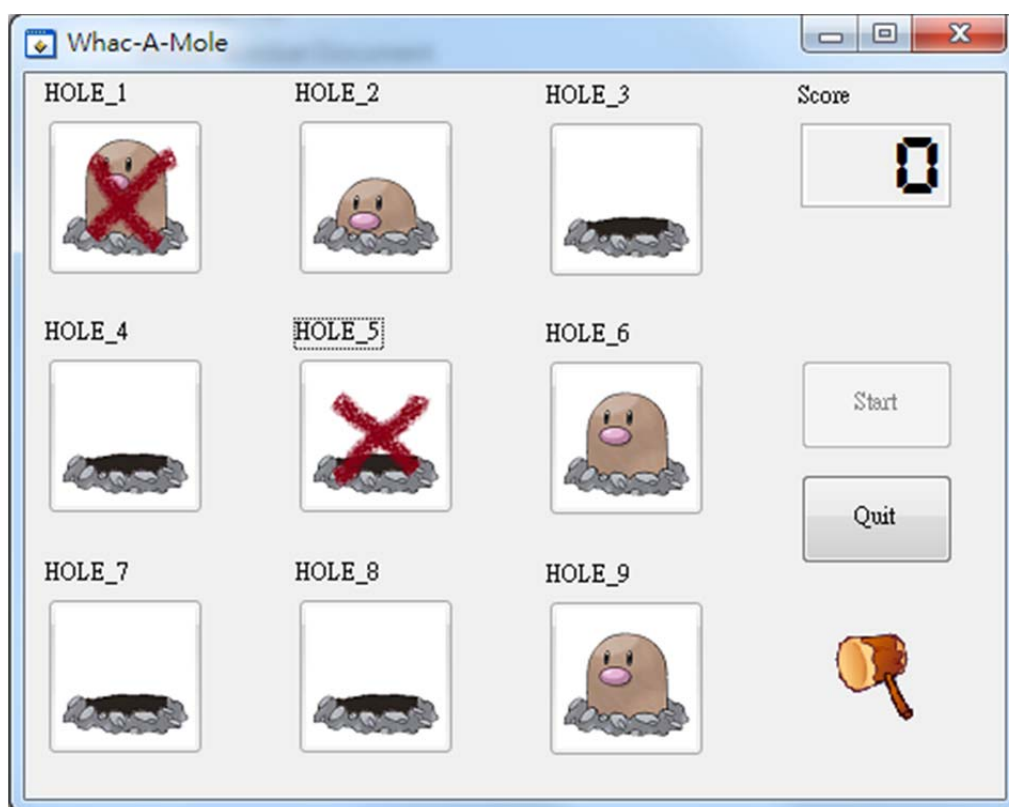


打地鼠

(Whac-A-Mole)



目錄

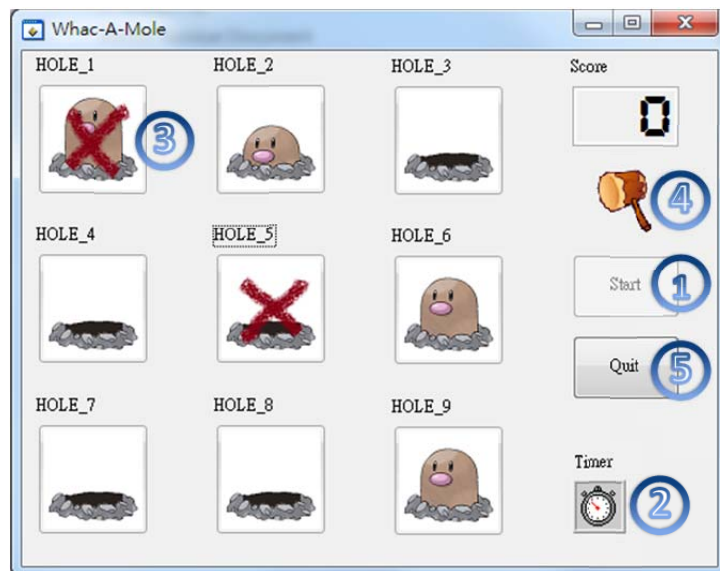
一、目的.....	3
二、內容.....	3
1. 功能解說:.....	3
2. 程式列表:.....	4
三、討論.....	12

一、目的

使用 NI LabWindows™ / CVI 軟體設計出打地鼠程式，擁有打地鼠特效、鼠槌游標、計算分數等功能，模擬真正的打地鼠機台，實際動手做出擁有互動人機介面以及功能的程式。

二、內容

1. 功能解說：



※面板物件：

Command Button	:	①	START、	⑤	QUIT
Timer	:	②	TIMER		
Picture Ring	:	③	HOLE_1 ~ HOLE_9		
滑鼠游標	:	④			
Numeric	:	SCORE			

※功能：地鼠隨機出現(動畫)；點擊正確加分、錯誤扣分；改變滑鼠游標(動畫)

2. 程式列表：

※總程式縮圖：

```
/* Include .h */
#include <ansi_c.h>
#include <cvirte.h>
#include <userint.h>
#include <windows.h> /* Load user.dll */
#include "Whac-A-Mole.h"

/* Start State */
#define TRUE 1
#define FALSE 0

/* Picture States */
#define EMPTY 0
#define POPOUT 1
#define HIT 2
#define ERROR 4

/* Declare Global Variable */
static int panelHandle;
static int start=FALSE;
static int score=0;
static int hole_1[2]={EMPTY, 0}; /* { Picture States, time } */
static int hole_2[2]={EMPTY, 0}; /* { Picture States, time } */
static int hole_3[2]={EMPTY, 0}; /* { Picture States, time } */
static int hole_4[2]={EMPTY, 0}; /* { Picture States, time } */
static int hole_5[2]={EMPTY, 0}; /* { Picture States, time } */
static int hole_6[2]={EMPTY, 0}; /* { Picture States, time } */
static int hole_7[2]={EMPTY, 0}; /* { Picture States, time } */
static int hole_8[2]={EMPTY, 0}; /* { Picture States, time } */
static int hole_9[2]={EMPTY, 0}; /* { Picture States, time } */
static int mouse_click_time[2]={FALSE, 0}; /* { Mouse Click States, time } */
static int ra_1=0; /* Appear Probability */
static int ra_2=0; /* Appear Probability */
static int ra_3=0; /* Appear Probability */
static int ra_4=0; /* Appear Probability */
static int ra_5=0; /* Appear Probability */
static int ra_6=0; /* Appear Probability */
static int ra_7=0; /* Appear Probability */
static int ra_8=0; /* Appear Probability */
static int ra_9=0; /* Appear Probability */

/* Main */
int main (int argc, char *argv[])

/* Start ? */
int WINAPI Start (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)

/* Timer */
int WINAPI ProcessLoop (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)

/* Hole */
int WINAPI Hole_1 (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
int WINAPI Hole_2 (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
int WINAPI Hole_3 (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
int WINAPI Hole_4 (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
int WINAPI Hole_5 (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
int WINAPI Hole_6 (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
int WINAPI Hole_7 (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
int WINAPI Hole_8 (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
int WINAPI Hole_9 (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)

/* Quit */
int WINAPI Quit (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
```

※各函式解說：

呼叫函式、宣告變數：

```
/* Include .h */
#include <ansi_c.h>
#include <cvirte.h>
#include <userint.h>
#include <windows.h> /* Load user.dll */ 1
#include "Whac-A-Mole.h"

/* Start State */
#define TRUE 1 2
#define FALSE 0

/* Picture States */
#define EMPTY 0
#define POPOUT 1 3
#define HIT 2
#define ERROR 4

/* Declare Global Variable */
static int panelHandle;
static int start=FALSE;
static int score=0;
static int hole_1[2]={EMPTY , 0}; /* { Picture States , time } */ 4
static int hole_2[2]={EMPTY , 0}; /* { Picture States , time } */
static int hole_3[2]={EMPTY , 0}; /* { Picture States , time } */
static int hole_4[2]={EMPTY , 0}; /* { Picture States , time } */
static int hole_5[2]={EMPTY , 0}; /* { Picture States , time } */
static int hole_6[2]={EMPTY , 0}; /* { Picture States , time } */
static int hole_7[2]={EMPTY , 0}; /* { Picture States , time } */
static int hole_8[2]={EMPTY , 0}; /* { Picture States , time } */
static int hole_9[2]={EMPTY , 0}; /* { Picture States , time } */
static int mouse_click_time[2]={FALSE , 0}; /* { Mouse Click States , time } */ 5
static int ra_1=0; /* Appear Probability */
static int ra_2=0; /* Appear Probability */
static int ra_3=0; /* Appear Probability */
static int ra_4=0; /* Appear Probability */
static int ra_5=0; /* Appear Probability */
static int ra_6=0; /* Appear Probability */
static int ra_7=0; /* Appear Probability */
static int ra_8=0; /* Appear Probability */
static int ra_9=0; /* Appear Probability */
```

1 #include <windows.h> : 呼叫 user.dll，使用更換滑鼠游標功能

2 /* Start State */ : TRUE 狀態為開始、FALSE 狀態為等待開始

3 /* Picture States */ : EMPTY 顯示為空洞圖片、POPOUT 顯示為 10 個地鼠鑽出圖片、HIT 顯示為鑽出時打擊圖片、ERROR 顯示為未鑽出時打擊圖片。

4 hole_1[2]={EMPTY , 0} : 索引值 0 為圖片狀態、索引值 1 為圖片持續的時間

5 mouse_click_time[2]={FALSE , 0} : 索引值 0 為滑鼠游標狀態、索引值 1 為圖片持續的時間

ra_1 : 宣告地鼠隨機出現的整數機率

Main 主函式、CVICALLBACK Start:


```
/* Main */
int main (int argc, char *argv[])
{
    if (InitCVIRTE (0, argv, 0) == 0)
        return -1; /* out of memory */
    if ((panelHandle = LoadPanel (0, "Whac-A-Mole.uir", PANEL)) < 0)
        return -1;
    DisplayPanel (panelHandle);
    RunUserInterface ();
    DiscardPanel (panelHandle);
    return 0;
}


/* Start ? */
int CVICALLBACK Start (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            start=TRUE;
            SetCtrlAttribute (panelHandle, PANEL_HOLE_1, ATTR_DIMMED, 0); /* Hole Undimmed */
            SetCtrlAttribute (panelHandle, PANEL_HOLE_2, ATTR_DIMMED, 0); /* Hole Undimmed */
            SetCtrlAttribute (panelHandle, PANEL_HOLE_3, ATTR_DIMMED, 0); /* Hole Undimmed */
            SetCtrlAttribute (panelHandle, PANEL_HOLE_4, ATTR_DIMMED, 0); /* Hole Undimmed */
            SetCtrlAttribute (panelHandle, PANEL_HOLE_5, ATTR_DIMMED, 0); /* Hole Undimmed */
            SetCtrlAttribute (panelHandle, PANEL_HOLE_6, ATTR_DIMMED, 0); /* Hole Undimmed */
            SetCtrlAttribute (panelHandle, PANEL_HOLE_7, ATTR_DIMMED, 0); /* Hole Undimmed */
            SetCtrlAttribute (panelHandle, PANEL_HOLE_8, ATTR_DIMMED, 0); /* Hole Undimmed */
            SetCtrlAttribute (panelHandle, PANEL_HOLE_9, ATTR_DIMMED, 0); /* Hole Undimmed */


            srand(time(NULL)); /* Set Seed Random */
            ra_1=(rand()%10)+2; /* Random from 2 to 11 */
            ra_2=(rand()%10)+2; /* Random from 2 to 11 */
            ra_3=(rand()%10)+2; /* Random from 2 to 11 */
            ra_4=(rand()%10)+2; /* Random from 2 to 11 */
            ra_5=(rand()%10)+2; /* Random from 2 to 11 */
            ra_6=(rand()%10)+2; /* Random from 2 to 11 */
            ra_7=(rand()%10)+2; /* Random from 2 to 11 */
            ra_8=(rand()%10)+2; /* Random from 2 to 11 */
            ra_9=(rand()%10)+2; /* Random from 2 to 11 */

            break;
    }
    return 0;
}
```

 start=TRUE : 遊戲開始

 SetCtrlAttribute (panelHandle, PANEL_HOLE_1, ATTR_DIMMED, 0) : 將洞變成可點擊的狀態

 srand(time(NULL)) : 設定程式每次開始都不同的種子數

 ra_1=(rand()%10)+2 : 產生範圍從 2 到 11 隨機的整數

CVICALLBACK ProcessLoop (一) :

```

2  /* Timer */
int CVICALLBACK ProcessLoop (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_TIMER_TICK: 1
        {
            if (mouse_click_time[0]==FALSE) /* UnHit */ 2
            {
                HCURSOR hCurs1 = LoadCursorFromFile("Mouse1.ani"); /* Load Mouse Cursor */
                SetCursor (hCurs1); /* Change Mouse Cursor */
                DestroyCursor(hCurs1); /* Release Mouse Cursor Memory */
            }

            if (mouse_click_time[0]==TRUE) /* Hit */ 3
            {
                if (mouse_click_time[1]<34) /* Mouse Cursor Keep Time */
                {
                    4 mouse_click_time[1]++;
                    HCURSOR hCurs2 = LoadCursorFromFile("Mouse2.ani"); /* Load Mouse Cursor */
                    SetCursor (hCurs2); /* Change Mouse Cursor */
                    DestroyCursor(hCurs2); /* Release Mouse Cursor Memory */
                }

                if (mouse_click_time[1]>=34) /* Mouse Cursor Keep Time */
                {
                    mouse_click_time[1]=0; /* Reset Keep Time to 0*/
                    mouse_click_time[0]=FALSE; /* Change To UnHit */
                }
            }
        }
    }
}

```

1 EVENT_TIMER_TICK :設定當 0.005 seconds 時執行一次

2 if (mouse_click_time[0]==FALSE) :當沒有點擊時顯示未點擊游標圖
 HCURSOR hCurs1=LoadCursorFromFile("Mouse1.ani") :讀取未點擊游標圖
 SetCursor (hCurs1) :設定未點擊游標圖給程式游標
 DestroyCursor(hCurs1) :釋放讀取游標圖的記憶體

3 if (mouse_click_time[0]==TRUE) :當點擊時顯示點擊游標圖
 if (mouse_click_time[1]<34) :顯示點擊游標圖的時間:小於
 0.005 * 34 = 0.17 seconds 時繼續
 顯示點擊的游標圖

4 mouse_click_time[1]++ :每 0.005 seconds 加一
 if (mouse_click_time[1]>=34) :超過 0.17 seconds 時顯示未點擊游標圖
 mouse_click_time[1]=0 :重置時間
 mouse_click_time[0]=FALSE :設定回未點擊游標

CVICALLBACK ProcessLoop (二) :

```

if (start==TRUE) /* Start */
{
    SetCtrlAttribute (panelHandle, PANEL_START, ATTR_DIMMED, 1); /* Start Dimmed */ 1
    SetCtrlAttribute (panelHandle, PANEL_STARTMESSAGE, ATTR_VISIBLE, 1); /* Startmessage Visible */

    static int shift=0;
    static int startmessage_visible=TRUE;

    if (startmessage_visible==TRUE) 2
    {
        shift=shift+2; /* StartMessage Distance */
        SetCtrlAttribute (panelHandle, PANEL_STARTMESSAGE, ATTR_LEFT, 400-shift); /* Set StartMessage Position */
    }

    if (shift>=800 || startmessage_visible==FALSE) /* StartMessage Position Over The Window */ 3
    {
        SetCtrlAttribute (panelHandle, PANEL_STARTMESSAGE, ATTR_VISIBLE, 0); /* Startmessage Hide */
        startmessage_visible=FALSE;
    }

    static double random_1=0; 4
    static int random_1_number=0; /* Declare Random Appear Probability */
    static double random_2=0;
    static int random_2_number=0; /* Declare Random Appear Probability */
    static double random_3=0;
    static int random_3_number=0; /* Declare Random Appear Probability */
    static double random_4=0;
    static int random_4_number=0; /* Declare Random Appear Probability */
    static double random_5=0;
    static int random_5_number=0; /* Declare Random Appear Probability */
    static double random_6=0;
    static int random_6_number=0; /* Declare Random Appear Probability */
    static double random_7=0;
    static int random_7_number=0; /* Declare Random Appear Probability */
    static double random_8=0;
    static int random_8_number=0; /* Declare Random Appear Probability */
    static double random_9=0;
    static int random_9_number=0; /* Declare Random Appear Probability */
}

```

1 SetCtrlAttribute (panelHandle, PANEL_START, ATTR_DIMMED, 1) : 讓 Start 按鈕變不能點擊的狀態

SetCtrlAttribute (panelHandle, PANEL_STARTMESSAGE, ATTR_VISIBLE, 1): 顯示出 Start 跑馬燈

startmessage_visible=TRUE : 設定 Start 跑馬燈變數為顯示狀態

2 if (startmessage_visible==TRUE) : 當跑馬燈變數為顯示時讓跑馬燈位移

shift=shift+2 : 設定跑馬燈一次要位移的距離

SetCtrlAttribute (panelHandle, PANEL_STARTMESSAGE, ATTR_LEFT, 400-shift): 跑馬燈位移

3 if (shift>=800 || startmessage_visible==FALSE) : 當跑馬燈變數為不顯示時讓跑馬燈停止

SetCtrlAttribute (panelHandle, PANEL_STARTMESSAGE, ATTR_VISIBLE, 0): 不顯示出 Start 跑馬燈

startmessage_visible=FALSE : 設定 Start 跑馬燈變數為不顯示狀態

4 static double random_1=0 : 即時產生 0 到最大為 ra (2~11 整數)隨機的浮點數

static int random_1_number=0 : 達成地鼠鑽出條件的總數

CVICALLBACK ProcessLoop (三) :

```

/* hole_1 */
if (hole_1[0]==EMPTY)
{
    random_1=ra_1*rand() / RAND_MAX; /* Generate Random Probability */ 1
    if (random_1>=(ra_1-1)) /* Appear Probability */
        random_1_number++;
    if (random_1_number>200) 2
    {
        hole_1[0]=POPOUT;
        random_1_number=0; /* Reset random_1_number */
    }
}

if (hole_1[0]==POPOUT)
{
    hole_1[1]=hole_1[1]+5;
    if ( hole_1[1]>0 && hole_1[1]<=100 ) 3
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 2);
    if ( hole_1[1]>100 && hole_1[1]<=200 )
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 3);
    if ( hole_1[1]>200 && hole_1[1]<=300 )
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 4);
    if ( hole_1[1]>300 && hole_1[1]<=400 )
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 5);
    if ( hole_1[1]>400 && hole_1[1]<=500 )
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 6);
    if ( hole_1[1]>500 && hole_1[1]<=600 )
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 7);
    if ( hole_1[1]>600 && hole_1[1]<=700 )
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 8);
    if ( hole_1[1]>700 && hole_1[1]<=800 )
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 9);
    if ( hole_1[1]>800 && hole_1[1]<=900 )
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 10);
    if ( hole_1[1]>900 && hole_1[1]<=1000 )
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 11);
    if ( hole_1[1]>1000 && hole_1[1]<=1100 )
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 11);
    if ( hole_1[1]>1100 && hole_1[1]<=1200 )
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 10);
    if ( hole_1[1]>1200 && hole_1[1]<=1300 )
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 9);
    if ( hole_1[1]>1300 && hole_1[1]<=1400 )
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 8);
    if ( hole_1[1]>1400 && hole_1[1]<=1500 )
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 7);
    if ( hole_1[1]>1500 && hole_1[1]<=1600 )
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 6);
    if ( hole_1[1]>1600 && hole_1[1]<=1700 )
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 5);
    if ( hole_1[1]>1700 && hole_1[1]<=1800 )
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 4);
    if ( hole_1[1]>1800 && hole_1[1]<=1900 )
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 3);
    if ( hole_1[1]>1900 && hole_1[1]<=2000 )
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 2);
    if ( hole_1[1]>2000 ) 4
    {
        hole_1[1]=0; /* Reset time */
        hole_1[0]=EMPTY;
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 1); /* Update Picture EMPTY State */
    }
}

```

1 random_1=ra_1*rand() / RAND_MAX :即時產生 0 到最大為 ra (2~11 整數)隨機的浮點數

if (random_1>=(ra_1-1)) :如果隨機浮點數大於 ra (2~11 整數)減一之條件

random_1_number++ :達成地鼠鑽出條件的總數加一

2 if (random_1_number>200) :達成地鼠鑽出條件的總數大於兩百

hole_1[0]=POPOUT :設定圖片狀態為地鼠鑽出

random_1_number=0 :重置鑽出條件的總數

hole_1[1]=hole_1[1]+5 :設定地鼠鑽出的時間

3 if (hole_1[1]> 0 && hole_1[1]<=100) :如果地鼠鑽出的時間在一範圍內

SetCtrlVal (panelHandle, PANEL_HOLE_1, 2) :顯示地鼠鑽出的圖片 (有十張連續圖)

4 if (hole_1[1]>2000) :超過地鼠鑽出的時間

hole_1[1]=0 :重置地鼠鑽出的時間

hole_1[0]=EMPTY :重置圖片狀態為空洞

機率問題將於後面頁數中討論

CVICALLBACK ProcessLoop (四) :

```

if (hole_1[0]==HIT) /* Good HIT */ 1
{
    SetCtrlVal (panelHandle, PANEL_HOLE_1, 12); /* Update Picture Good HIT State */

    if (hole_1[1]<1000) /* Keep for 1 seconds */ 2
        hole_1[1]=hole_1[1]+5;
    else /* Keep for 1 seconds after */
    {
        hole_1[0]=EMPTY;
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 1); /* Update Picture EMPTY State */
        hole_1[1]=0; /* Reset time */
    }
}

if (hole_1[0]==ERROR) /* Error HIT */ 3
{
    SetCtrlVal (panelHandle, PANEL_HOLE_1, 13); /* Update Picture Error HIT State */

    if (hole_1[1]<1000) /* Keep for 1 seconds */ 4
        hole_1[1]=hole_1[1]+5;
    else /* Keep for 1 seconds after */
    {
        hole_1[0]=EMPTY;
        SetCtrlVal (panelHandle, PANEL_HOLE_1, 1); /* Update Picture EMPTY State */
        hole_1[1]=0; /* Reset time */
    }
}

```

- | | | |
|---|--|----------------------|
| 1 | if (hole_1[0]==HIT) | : 當鑽出時打擊地鼠 |
| | SetCtrlVal (panelHandle, PANEL_HOLE_1, 12) | : 顯示正確打擊圖片 |
| 2 | if (hole_1[1] <1000) hole_1[1]=hole_1[1]+5 | : 持續顯示正確打擊圖片(保持一秒時間) |
| | else {hole_1[0] =EMPTY; hole_1[1]=0} | : 重置圖片為空洞、重置時間 |
| 3 | if (hole_1[0]== ERROR) | : 當未鑽出時打擊地鼠 |
| | SetCtrlVal (panelHandle, PANEL_HOLE_1, 13) | : 顯示錯誤打擊圖片 |
| 4 | if (hole_1[1]<1000) hole_1[1]=hole_1[1]+5 | : 持續顯示錯誤打擊圖片(保持一秒時間) |
| | else {hole_1[0]=EMPTY; hole_1[1]=0} | : 重置圖片為空洞、重置時間 |

CVICALLBACK Hole_1 :

```

3  /* Hole */
   int CVICALLBACK Hole_1 (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
   {
       switch (event)
       {
           4  case EVENT_LEFT_CLICK: 1
               mouse_click_time[0]=TRUE;

               if (hole_1[0]==POPOUT) /* Good HIT */ 2
               {
                   hole_1[0]=HIT;
                   hole_1[1]=0; /* Reset time */
                   score++;
                   SetCtrlVal (panelHandle, PANEL_SCORE, score); /* Update SCORE */
               }

               if (hole_1[0]==EMPTY) /* Error HIT */ 3
               {
                   hole_1[0]=ERROR;
                   hole_1[1]=0; /* Reset time */
                   score--;
                   SetCtrlVal (panelHandle, PANEL_SCORE, score); /* Update SCORE */
               }

               break;
           }
       return 0;
   }

```

1 EVENT_LEFT_CLICK

:觸發左鍵點擊事件

mouse_click_time[0]=TRUE

:設定滑鼠游標狀態為按下

2 if (hole_1[0]==POPOUT)

:如果左鍵點擊時為地鼠鑽出狀態

hole_1[0]=HIT

:設定圖片為正確打擊

hole_1[1]=0

:重置時間

score++

:分數加一

SetCtrlVal (panelHandle, PANEL_SCORE, score)

:更新面板分數顯示

3 if (hole_1[0]==EMPTY)

:如果左鍵點擊時為地鼠未鑽出狀態

hole_1[0]=ERROR

:設定圖片為錯誤打擊

hole_1[1]=0

:重置時間

score--

:分數減一

SetCtrlVal (panelHandle, PANEL_SCORE, score)

:更新面板分數顯示

CVICALLBACK Quit :

```

5  /* Quit */
   int CVICALLBACK Quit (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
   {
       switch (event)
       {
           case EVENT_COMMIT:
               QuitUserInterface (0);
               break;
       }
       return 0;
   }

```

三、討論

※隨機出現地鼠的機率是由我們自己定義的一套方式。

1. 首先設定 srand 讓程式每次開始都有不同的種子數。
2. 再產生範圍從 2 到 11 隨機的整數 $ra_1 = (\text{rand}() \% 10) + 2$ 。
3. 再即時產生 0 到最大為 ra (2~11 整數)隨機的浮點數， $\text{random_1} = ra_1 * \text{rand}() / \text{RAND_MAX}$ ，因為是放在 Timer 裡，所以會每 0.005 seconds 即時產生一個浮點數 random_1。
4. 將 random_1 與 ra_1 減一的數值比較，如果符合大於的條件，達成地鼠鑽出條件的總數就加一， $\text{random_1_number}++$ 。
5. 當 random_1_number 總數大於兩百，就會讓圖片變為地鼠鑽出的狀態

※舉例:先產生隨機整數 $ra_1 = 3$ ；再即時產生浮點數 $\text{random_1} = 2.51$ ； $ra_1 = 3$ 減一等於 2， $\text{random_1} = 2.51$ 大於 2；所以達成地鼠鑽出條件的總數就加一， $\text{random_1_number}++$ ；當達成條件總數大於兩百時地鼠鑽出。由此推斷每次即時產生的浮點數都達成條件，最短也需要 $0.005 * 200 = 1$ 秒。如果即時產生的浮點數沒有達成條件，最長時間就為不定值(甚至不出來)。

所以此遊戲每個洞地鼠出現機率可能為

$\frac{2}{3} \uparrow$	$\frac{3}{4} \uparrow$	$\frac{4}{5} \uparrow$
$\frac{5}{6} \uparrow$	$\frac{6}{7} \uparrow$	$\frac{7}{8} \uparrow$
$\frac{8}{9} \uparrow$	$\frac{9}{10} \uparrow$	$\frac{10}{11} \uparrow$

此隨機出現的機率由我們自己定義的，事實上還有更多方法，這裡就介紹此種。

※此報告只列出一個洞的相關程式，事實上還需要複製總共九個，以及使用九個不同的變數。

※將各個地鼠鑽出的圖存成 10 個不同的圖，利用 Picture Ring 的形式顯示索引值 1~10 的图片，即可達到動畫的效果。另外索引值 11~12 顯示打擊正確以及錯誤的图片。

※此報告另一重點是需要使用外部的.dll 檔，以便使用更換滑鼠游標的功能。一開始我們就在尋找適合的滑鼠游標，配合此遊戲所以需要槌子的圖樣，不過 CVI 內建的游標圖示都不是我們想要的，於是就尋找資料如何改善，發現需要呼叫

Windows 內建的 user32.dll 檔才能解決。不過發現解決方法後一開始沒這麼簡單就解決，因為能支援 CVI 的程式語法與 Windows 內建的不盡相同，我們試了好幾次語法才成功被 CVI 辨認成功。

※原本的 Timer 是設計成 0.001 second 執行一次，因為為了讓人機介面反應時間變快所以才如此快；另一個原因是更換滑鼠游標的程式必須寫在 Timer 函式裡，因為其他編譯語言只需要開始時讀取外部圖片一次進入程式，不過 CVI 比較與眾不同，如果只讀取一次外部圖片，一移動滑鼠又會變回原樣，推測應該為 CVI 的問題。如果視覺上滑鼠游標圖示要不被人看出來時時有在更新，所以就需要每 0.001 秒更新一次。不過最後發現學校的電腦要是 Timer 使用 0.001 秒會跑不動，但在其他雙核心以上的電腦就不會有此問題，所以最後折衷改為 0.005 秒，學校電腦既不會跑不動、游標的更新時間也不會很明顯。