

# Snake Game

20141229  
IT학부 주혜원

# 목차

- 코드 설명
- 어려웠던 점
- 보완해야 할 점
- 수정한 부분

# snake\_gameView.h

```
#define FILED_WIDTH 30
#define FILED_HEIGHT 30

public:
    COLORREF fruit_color;
    int Speed;
    CString view_strName;

afx_msg void OnEntername();
afx_msg void OnGamerule();
afx_msg void OnTimer(UINT_PTR nIDEvent);
afx_msg void OnKeyDown(UINT nChar, UINT
    nRepCnt, UINT nFlags);
virtual void OnInitialUpdate();
void MakeSnake(void);
void SnakeFood(void);
int AddScore(int s);

afx_msg void OnApple();
afx_msg void OnPlum();
afx_msg void OnGrape();
afx_msg void OnUpdateApple(CCmdUI *pCmdUI);
afx_msg void OnUpdatePlum(CCmdUI *pCmdUI);
afx_msg void OnUpdateGrape(CCmdUI *pCmdUI);
afx_msg void Onlevel1();
afx_msg void Onlevel2();
afx_msg void Onlevel3();
afx_msg void Onlevel4();
afx_msg void Onlevel5();
afx_msg void OnUpdatelevel1(CCmdUI *pCmdUI);
afx_msg void OnUpdatelevel2(CCmdUI *pCmdUI);
afx_msg void OnUpdatelevel3(CCmdUI *pCmdUI);
afx_msg void OnUpdatelevel4(CCmdUI *pCmdUI);
afx_msg void OnUpdatelevel5(CCmdUI *pCmdUI);
```

# snake\_gameView.cpp

```
struct Snake
{
    int x, y;
    int len;
    int direct;
}Snake[50];

struct Food
{
    int x;
    int y;
    int isfood;
}Food;

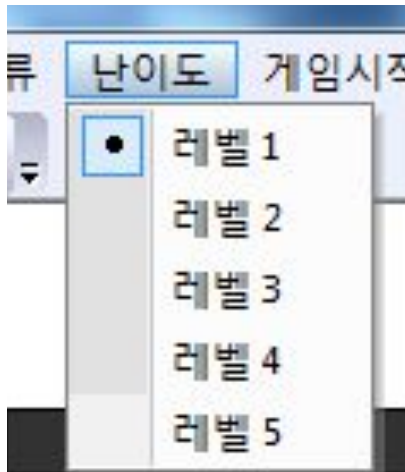
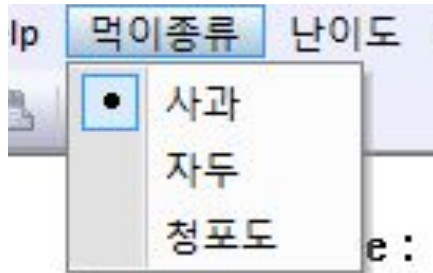
static int score = 10;
```

```
Csnake_gameView::Csnake_gameView()
{
    // TODO: add construction code here

    fruit_color = RGB(0, 0, 0);
    Speed = 200;
    Score = 10;

    Snake[0].x=10;
    Snake[0].y=10;
    Snake[1].x=11;
    Snake[1].y=10;
    Snake[2].x=12;
    Snake[2].y=10;
    Snake[0].direct=3;
    Snake[0].len=3;
    Food.isfood=1;
}
```

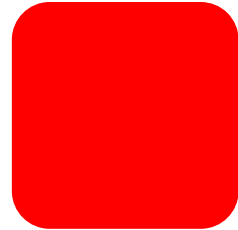
## Resource View - snake\_game



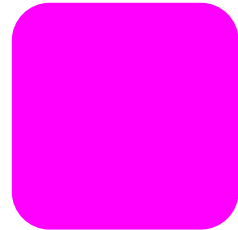
| Caption | ID        |
|---------|-----------|
| 사과      | ID_Apple  |
| 자두      | ID_Plum   |
| 청포도     | ID_Grape  |
| 레벨 1    | ID_level1 |
| 레벨 2    | ID_level2 |
| 레벨 3    | ID_level3 |
| 레벨 4    | ID_level4 |
| 레벨 5    | ID_level5 |

# MAINFRAME

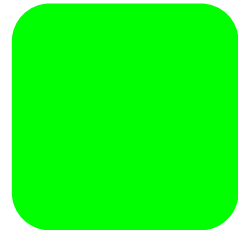
```
void Csnake_gameView::OnApple()  
{  
    // TODO: Add your command handler code here  
    fruit_color = RGB(255, 0, 0);  
}
```



```
void Csnake_gameView::OnPlum()  
{  
    // TODO: Add your command handler code here  
    fruit_color = RGB(255, 0, 255);  
}
```



```
void Csnake_gameView::OnGrape()  
{  
    // TODO: Add your command handler code here  
    fruit_color = RGB(0, 255, 0);  
}
```



```
void Csnake_gameView::OnUpdateApple(CCmdUI *pCmdUI)
{
    // TODO: Add your command update UI handler code here
    pCmdUI->SetRadio(fruit_color == RGB(255, 0, 0));
}
```

```
void Csnake_gameView::OnUpdatePlum(CCmdUI *pCmdUI)
{
    // TODO: Add your command update UI handler code here
    pCmdUI->SetRadio(fruit_color == RGB(255, 0, 255));
}
```

```
void Csnake_gameView::OnUpdateGrape(CCmdUI *pCmdUI)
{
    // TODO: Add your command update UI handler code here
    pCmdUI->SetRadio(fruit_color == RGB(0, 255, 0));
}
```

# MAINFRAME

```
void Csnake_gameView::Onlevel1()
{
    // TODO: Add your command handler code here
    Speed = 200;
}
```

```
void Csnake_gameView::Onlevel2()
{
    // TODO: Add your command handler code here
    Speed = 170;
}
```

```
void Csnake_gameView::Onlevel3()
{
    // TODO: Add your command handler code here
    Speed = 140;
}
```

```
void Csnake_gameView::Onlevel4()
{
    // TODO: Add your command handler code here
    Speed = 120;
}
```

```
void Csnake_gameView::Onlevel5()
{
    // TODO: Add your command handler code here
    Speed = 90;
}
```

SetTimer(0, Speed, NULL);



```
void Csnake_gameView::OnUpdatelevel1(CCmdUI *pCmdUI)
{
    // TODO: Add your command update UI handler code here
    pCmdUI->SetRadio(Speed == 200);
}

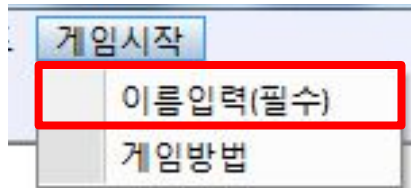
void Csnake_gameView::OnUpdatelevel2(CCmdUI *pCmdUI)
{
    // TODO: Add your command update UI handler code here
    pCmdUI->SetRadio(Speed == 170);
}

void Csnake_gameView::OnUpdatelevel3(CCmdUI *pCmdUI)
{
    // TODO: Add your command update UI handler code here
    pCmdUI->SetRadio(Speed == 140);
}

void Csnake_gameView::OnUpdatelevel4(CCmdUI *pCmdUI)
{
    // TODO: Add your command update UI handler code here
    pCmdUI->SetRadio(Speed == 120);
}

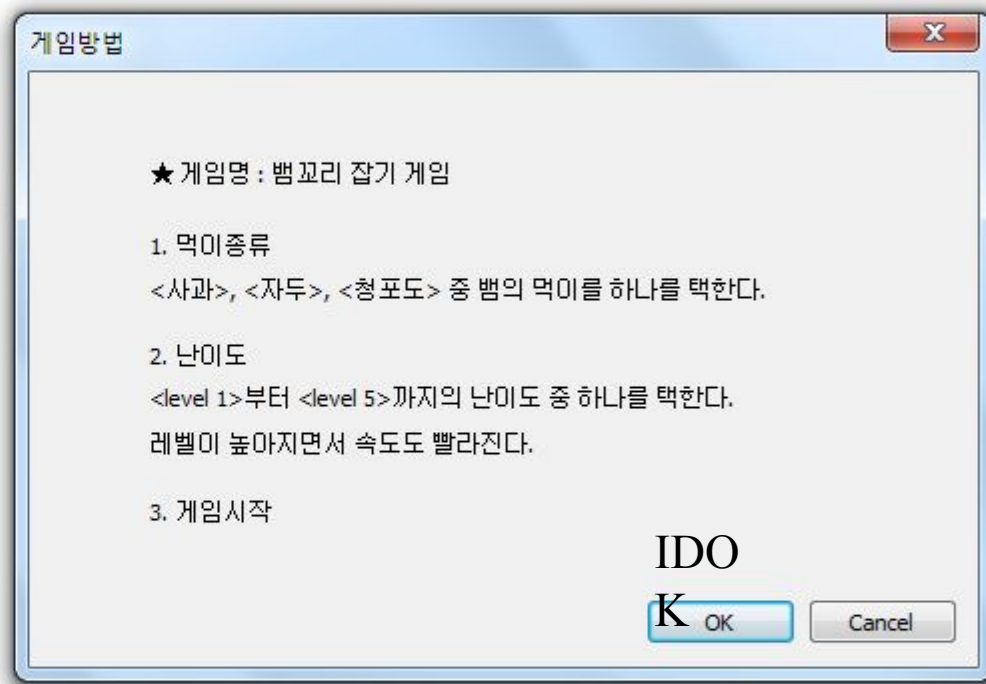
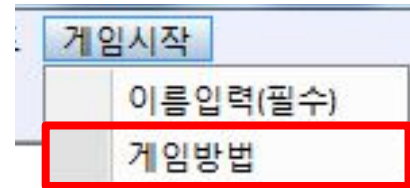
void Csnake_gameView::OnUpdatelevel5(CCmdUI *pCmdUI)
{
    // TODO: Add your command update UI handler code here
    pCmdUI->SetRadio(Speed == 90);
}
```

# CDlgName



IDD\_DIALOG1

# CDlgRule



IDD\_DIALOG2

# DlgName.h

```
#pragma once
// CDlgName dialog

class CDlgName : public CDialogEx
{
    DECLARE_DYNAMIC(CDlgName)

public:
    CDlgName(CWnd* pParent = NULL);    // standard constructor
    virtual ~CDlgName();

    // Dialog Data
    enum { IDD = IDD_DIALOG1 };

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV
    support

    DECLARE_MESSAGE_MAP()
public:
    CString strName;};
```

# DlgRule.h

```
#pragma once
// CDlgRule dialog

class CDlgRule : public CDialogEx
{
    DECLARE_DYNAMIC(CDlgRule)

public:
    CDlgRule(CWnd* pParent = NULL);    // standard constructor
    virtual ~CDlgRule();

// Dialog Data
    enum { IDD = IDD_DIALOG2 };

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support

    DECLARE_MESSAGE_MAP()
};
```

```

#include "DlgName.h"

void Csnake_gameView::OnEntername()
{
    // TODO: Add your command handler
    code here
    CDlgName dlg;
    dlg.strName = view_strName;

    if(dlg.DoModal()==IDOK){
        OnInitialUpdate();
        SetTimer(0, Speed, NULL);
    }
    else return;

    view_strName = dlg.strName;

    Invalidate();
}

void Csnake_gameView::OnInitialUpdate()
{
    CView::OnInitialUpdate();

    Snake[0].x=10;
    Snake[0].y=10;
    Snake[1].x=11;
    Snake[1].y=10;
    Snake[2].x=12;
    Snake[2].y=10;
    Snake[0].direct=3;
    Snake[0].len=3;
    Food.isfood=1;

    // TODO: Add your specialized code
    here and/or call the base class
}

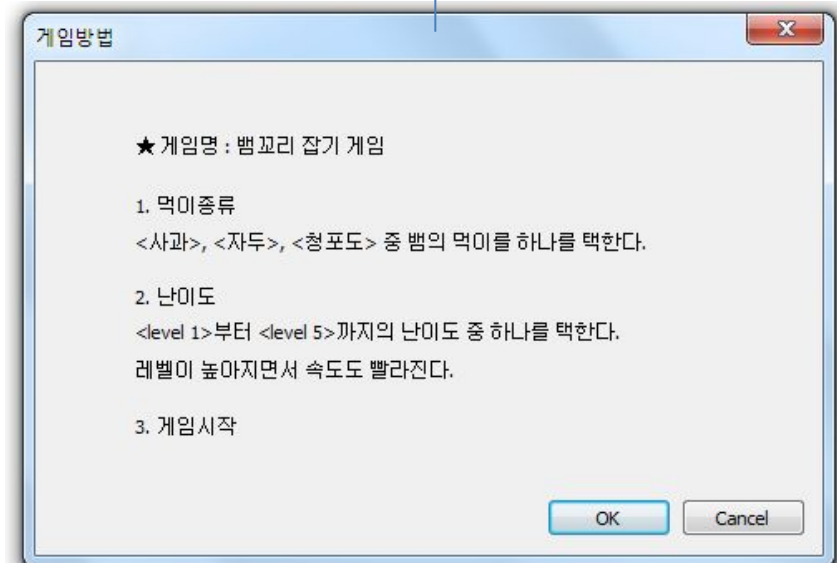
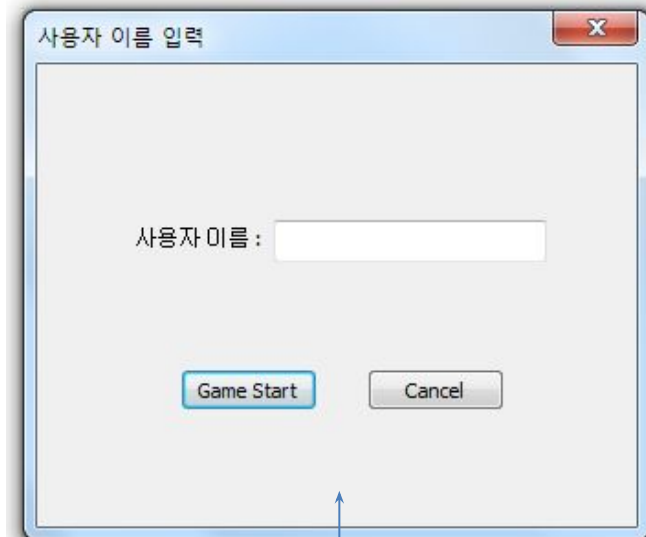
```

```
#include "DlgRule.h"

void Csnake_gameView::OnGamerule()
{
    // TODO: Add your command
    handler code here

    CDlgRule dlg;

    if(dlg.DoModal()==IDOK){
        OnEntername();
    }
}
```



```
void Csnake_gameView::OnDraw(CDC* pDC)
{
    Csnake_gameDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if (!pDoc)
        return;

    CString strMsg;
    strMsg.Format(_T("User Name : %s"), view_strName);
    pDC->TextOutW(20, 20, strMsg);

    pDC->TextOutW(220, 20, _T("Score : "));

    switch(Speed){
        case 200:pDC->TextOutW(560, 20, _T("level 1"));break;
        case 170:pDC->TextOutW(560, 20, _T("level 2"));break;
        case 140:pDC->TextOutW(560, 20, _T("level 3"));break;
        case 120:pDC->TextOutW(560, 20, _T("level 4"));break;
        case 90:pDC->TextOutW(560, 20, _T("level 5"));break;
    }
}
```

```
int i, j;
```

```
CPen myPen, *pOldPen; //Cbrush의 오브젝트 정의  
CBrush myBrush, *pOldBrush; //Cbrush의 오브젝트 정의
```

```
myPen.CreatePen(PS_SOLID, 1, RGB(0, 0, 0));  
myBrush.CreateSolidBrush(RGB(50, 50, 50));
```

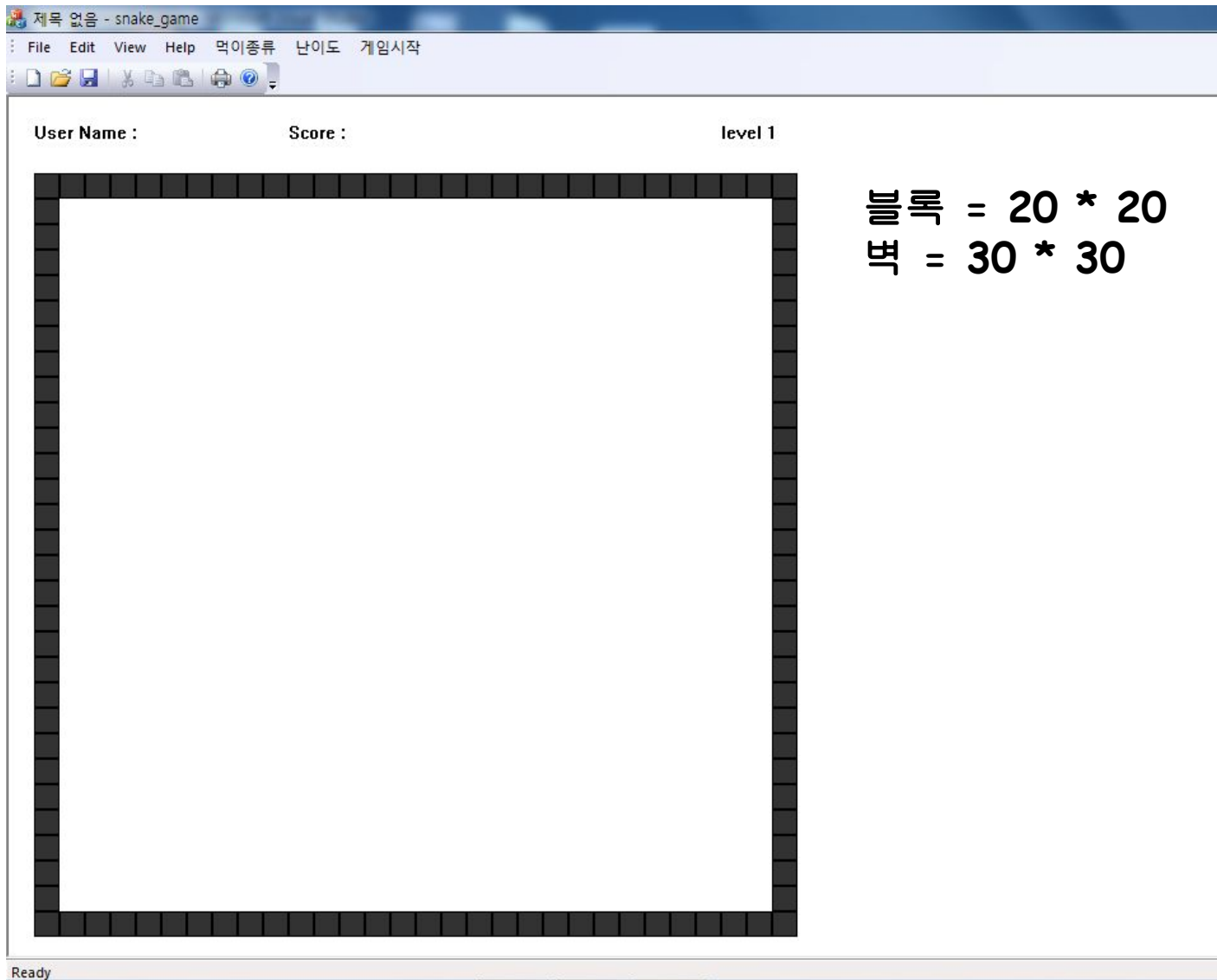
```
pOldPen = pDC->SelectObject(&myPen); //새로운 Pen으로 설정  
pOldBrush = pDC->SelectObject(&myBrush); //새로운 Brush로 설정
```

```
for(i = 0; i < FILED_HEIGHT; i++)  
{  
    for(j = 0; j < FILED_WIDTH; j++)  
    {  
        if(i == 0 || i == FILED_HEIGHT - 1 || j == 0 || j == FILED_WIDTH -  
1){  
            pDC->Rectangle(j * 20 + 20, i * 20 + 60, j * 20 + 40, i * 20 +  
80);  
        }  
    }  
}  
pDC->SelectObject(pOldPen);  
pDC->SelectObject(pOldBrush); //DC를 원래 상태로 복귀  
myPen.DeleteObject();  
myBrush.DeleteObject(); //객체 삭제
```

```
}
```



# 초기화면



```
void Csnake_gameView::MakeSnake(void)
{
    CDC *pDC=GetDC();

    pDC->SelectStockObject(WHITE_PEN); //DC에 할당
    //Snake[2].x * 20 = 240
    //Snake[2].y * 20 = 200
    //(Snake[2].x + 1) * 20 = 13*20 = 260
    //(Snake[2].y + 1) * 20 = 11*20 = 220
    //Rectangle(240, 200, 260, 220)
    pDC->Rectangle(Snake[Snake[0].len-1].x * 20, Snake[Snake[0].len-1].y * 20,
        (Snake[Snake[0].len-1].x + 1) * 20, (Snake[Snake[0].len-1].y + 1) * 20);

    //머리값 따라다니도록, i = 2 1
    //Snake[2].x = Snake[1].x
    //Snake[2].y = Snake[1].y
    //Snake[1].x = Snake[0].x
    //Snake[1].y = Snake[0].y
    for(int i=Snake[0].len-1; i>0; i--)
    {
        Snake[i].x=Snake[i-1].x;
        Snake[i].y=Snake[i-1].y;
    }
}
```

## snake\_gameView.cpp

### MakeSnake(void)

```
if(Snake[0].direct==1)Snake[0].y--; //↑  
if(Snake[0].direct==2)Snake[0].y++; //↓  
if(Snake[0].direct==3)Snake[0].x--; //←  
if(Snake[0].direct==4)Snake[0].x++; //→
```

```
CBrush DrawBrush=RGB(0,0,0);  
CBrush *pDrawBrush = pDC->SelectObject(&DrawBrush);
```

```
//Snake[0].x * 20 = 200  
//Snake[0].y * 20 = 200  
//(Snake[0].x + 1) * 20 = 11*20 = 220  
//(Snake[0].y + 1) * 20 = 11*20 = 220  
//Rectangle(200, 200, 220, 220)  
pDC->Rectangle(Snake[0].x * 20, Snake[0].y * 20, (Snake[0].x  
+ 1) * 20, (Snake[0].y + 1) * 20);
```

```
}
```

```
void Csnake_gameView::OnKeyDown(UINT nChar, UINT nRepCnt, UINT
nFlags)
{
    // TODO: Add your message handler code here and/or call default

    switch(nChar)
    {
        case VK_UP : if(Snake[0].direct != 2) Snake[0].direct = 1;
        break;
        case VK_DOWN : if(Snake[0].direct != 1) Snake[0].direct = 2;
        break;
        case VK_LEFT : if(Snake[0].direct != 4) Snake[0].direct = 3;
        break;
        case VK_RIGHT : if(Snake[0].direct != 3) Snake[0].direct =
        4; break;
    }

    CView::OnKeyDown(nChar, nRepCnt, nFlags);
}
```

```
void Csnake_gameView::SnakeFood(void)
{
    CDC *pDC=GetDC();

    //Snake[0].x * 20 = 200 같음 Food.x * 20
    //and
    //Snake[0].y * 20 = 200 같음 Food.y * 20
    if(Snake[0].x * 20 == Food.x * 20 && Snake[0].y * 20 ==
    Food.y * 20)
    {
        Snake[0].len++;
        Food.isfood = 1;
    }
}
```

```
if(Food.isfood==1)
{
    srand((unsigned)time(NULL));
    do
    {
        Food.x = rand() % (FILED_WIDTH - 4) + 2;
        Food.y = rand() % (FILED_HEIGHT - 4) + 4;
        //isfo = 2 1 0
        for(int isfo=Snake[0].len-1; isfo>=0; isfo--)
        {
            if(Food.x*20==Snake[isfo].x*20 &&
               Food.y*20==Snake[isfo].y*20)
            {
                Food.x=rand()% (FILED_WIDTH - 4) + 2;
                Food.y=rand()% (FILED_HEIGHT - 4) + 4;
            }
        }
    }
    while(Food.x*20<0 || Food.y*20<0 || Food.x*20>1000 ||
          Food.y*20>1000);
```

```
pDC->SelectStockObject(WHITE_PEN);
```

```
CBrush DrawBrush = fruit_color;
```

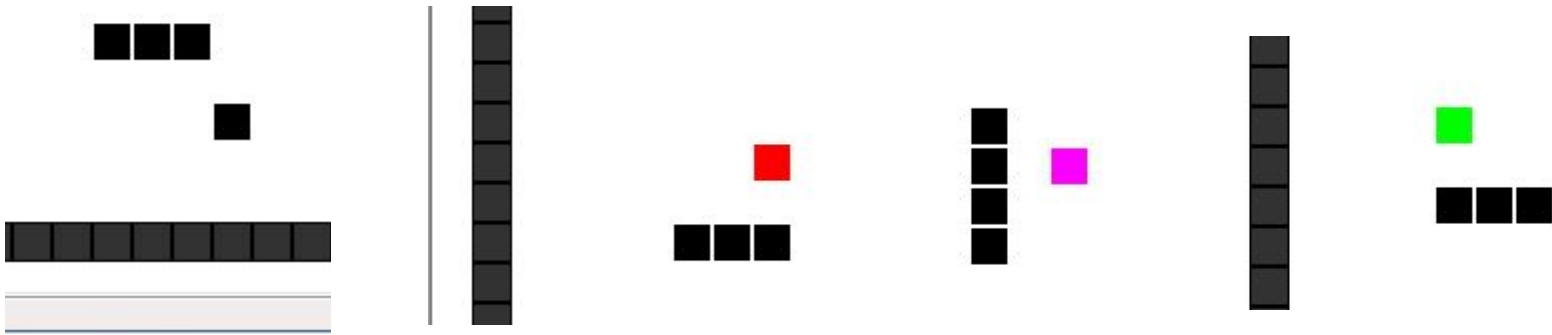
```
CBrush *pDrawBrush = pDC->SelectObject(&DrawBrush);
```

```
pDC->Rectangle(Food.x*20, Food.y*20, (Food.x+1)*20, (Food.y  
+1)*20);
```

```
Food.isfood=0;
```

```
}
```

```
}
```

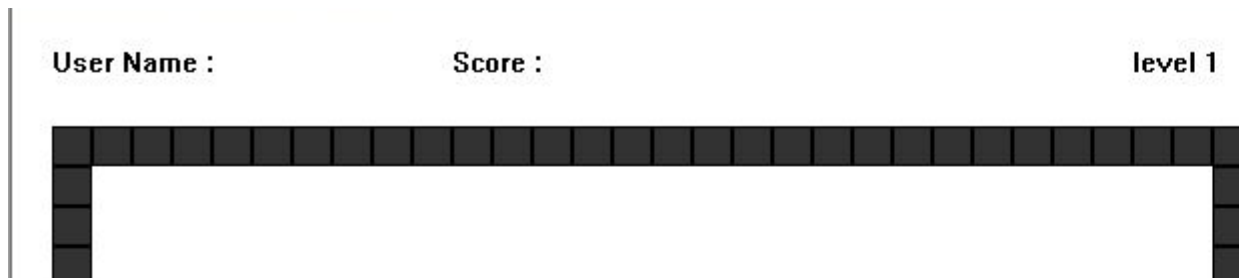


## snake\_gameView.cpp AddScore(int s)

```
int Csnake_gameView::AddScore(int s)
{
    CDC *pDC=GetDC();
    CString strMsg2;
    strMsg2.Format(_T("%d"), score);
    pDC->TextOutW(300, 20, strMsg2);

    score+=10;

    return score;
}
```





snake\_gameView.cpp  
OnTimer(..)

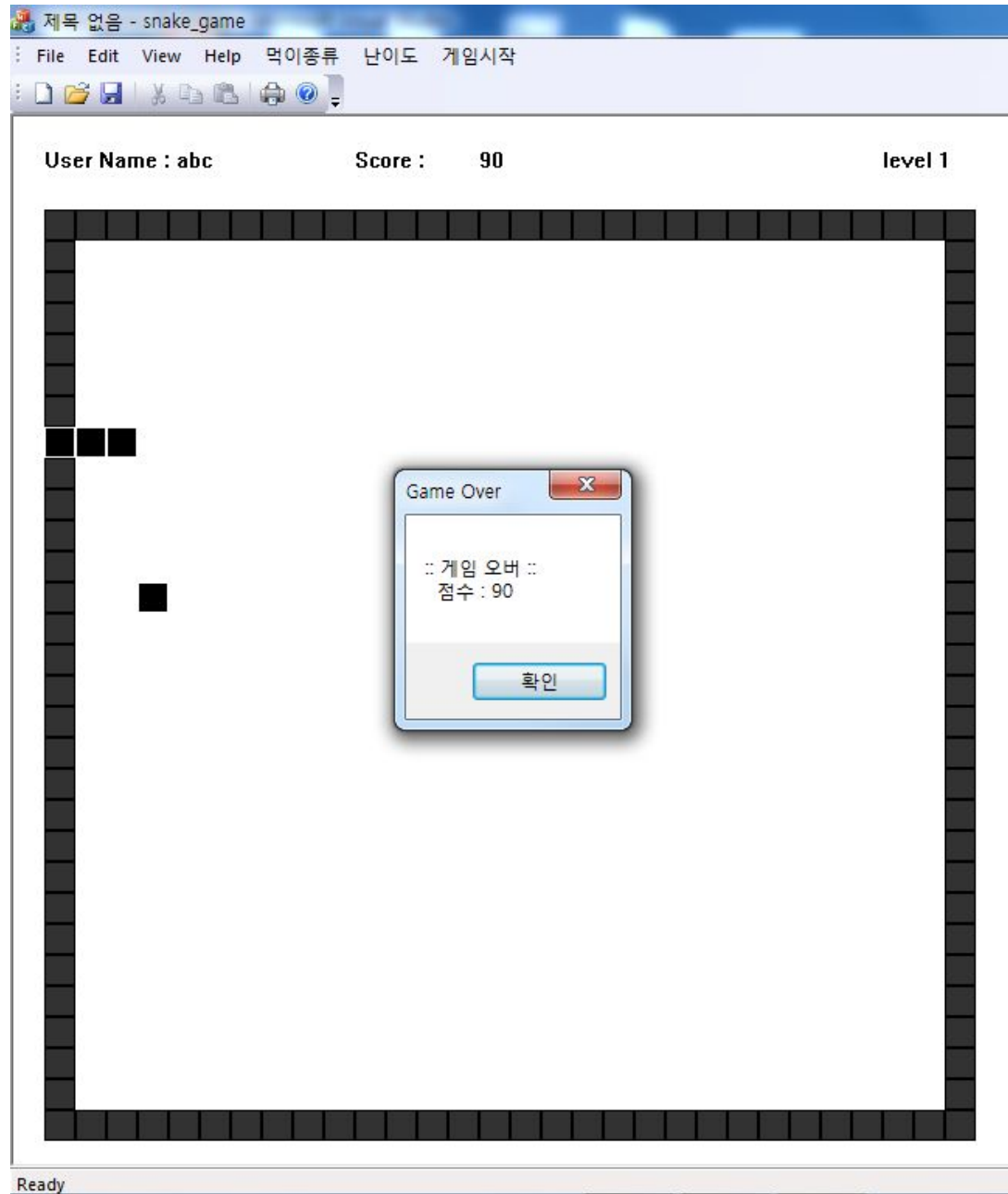
```
void Csnake_gameView::OnTimer(UINT_PTR nIDEvent)
{
    // TODO: Add your message handler code here and/or call default

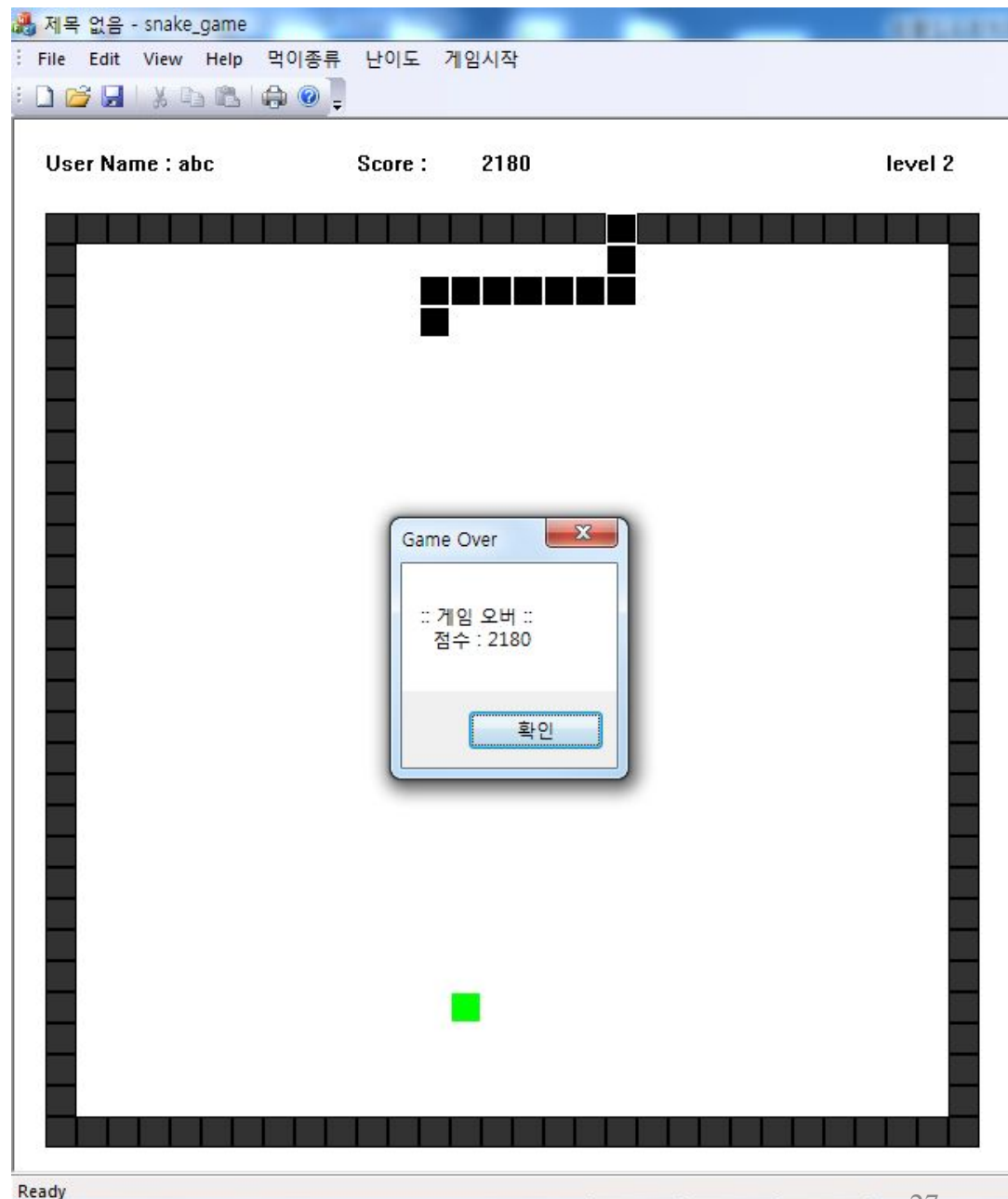
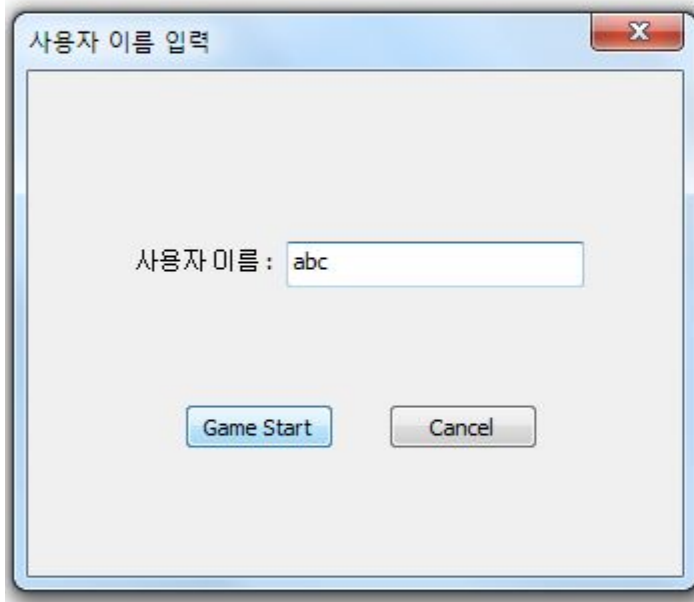
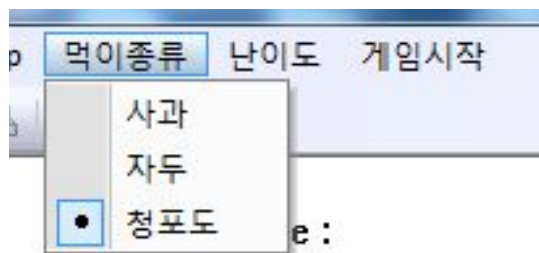
    MakeSnake();
    SnakeFood();
    AddScore(10);

    if((Snake[0].x * 20 < 40)||((Snake[0].y * 20 <
80)||((Snake[0].x+1) * 20 > 600)||((Snake[0].y+1) * 20 > 640))
    {
        KillTimer(0);

        CString strMsg;
        strMsg.Format(_T(":: 게임 오버 :: \n 점수 : %d"),
score-=10);
        MessageBox(strMsg, _T("Game Over"), MB_OK);
    }

CView::OnTimer(nIDEvent);
}
```





# 수정한 부분

- 클라이언트 화면 상단의 변화  
User Name:    Score:
- Score은 Snake의 움직이는 거리로 계산
- Snake의 머리와 먹이를 CPoint로  
활용하기보다구조체를 만들어 배열로 사용

# 보완해야 할 점

- ```
struct BODY{  
    int x;  
    int y;  
    BODY* next;  
};
```
- ```
BODY *BodyPos = (BODY*)SnakeHead->next;  
while(BodyPos != NULL)  
{  
    if(BodyPos->x == SnakeHead->x && BodyPos->y ==  
        SnakeHead->y)  
    {  
        return true;  
    }  
    BodyPos = BodyPos->next;  
}
```

😊 감사합니다 😊