

게임 기획서

DONGJAK90000



나은찬

소프트웨어학부 / 20150234

석영현

소프트웨어학부 / 20170764

최성률

소프트웨어학부 / 20150283

게임 기획서

DONGJAK90000

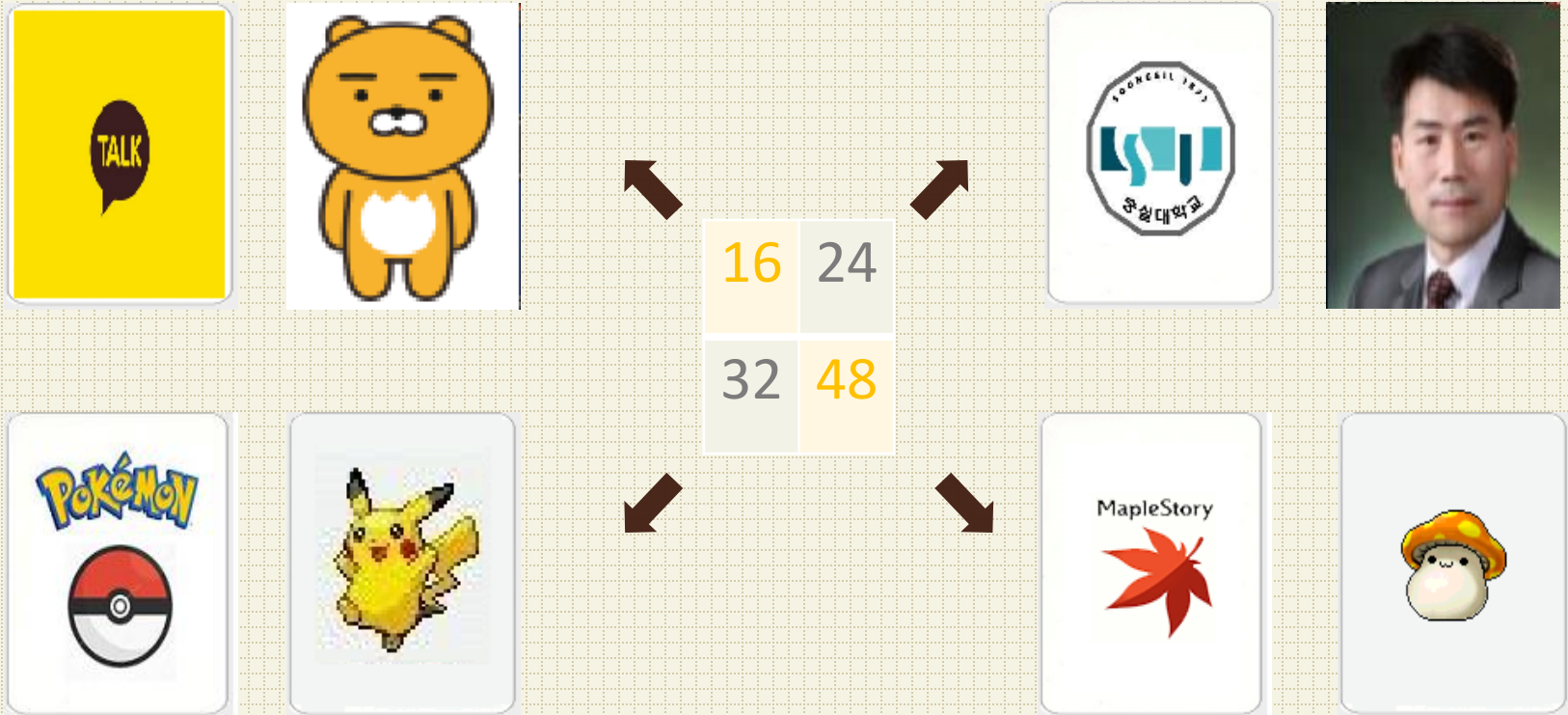


01. 게임 소개

02. 게임 진행 방법

03. 게임 화면 및 시스템

배경 및 컨셉



01. 게임 소개

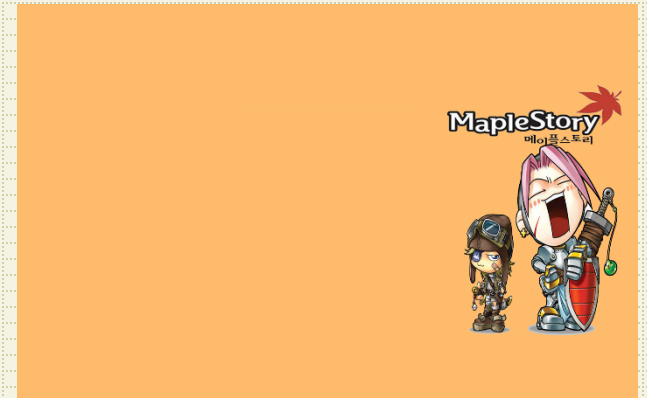
02. 게임 화면 및 시스템

03. 게임 진행 방법

배경 및 컨셉



16	24
32	48

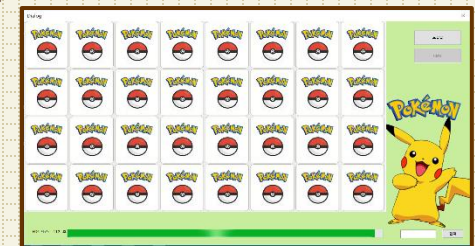
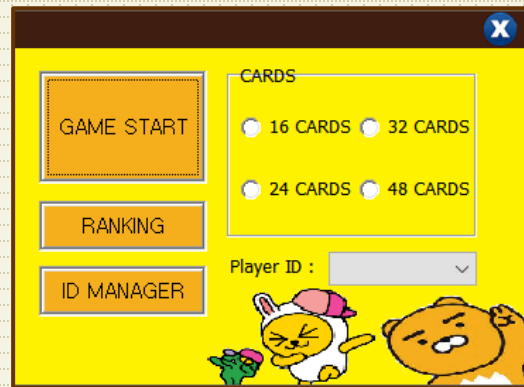


게임진행순서

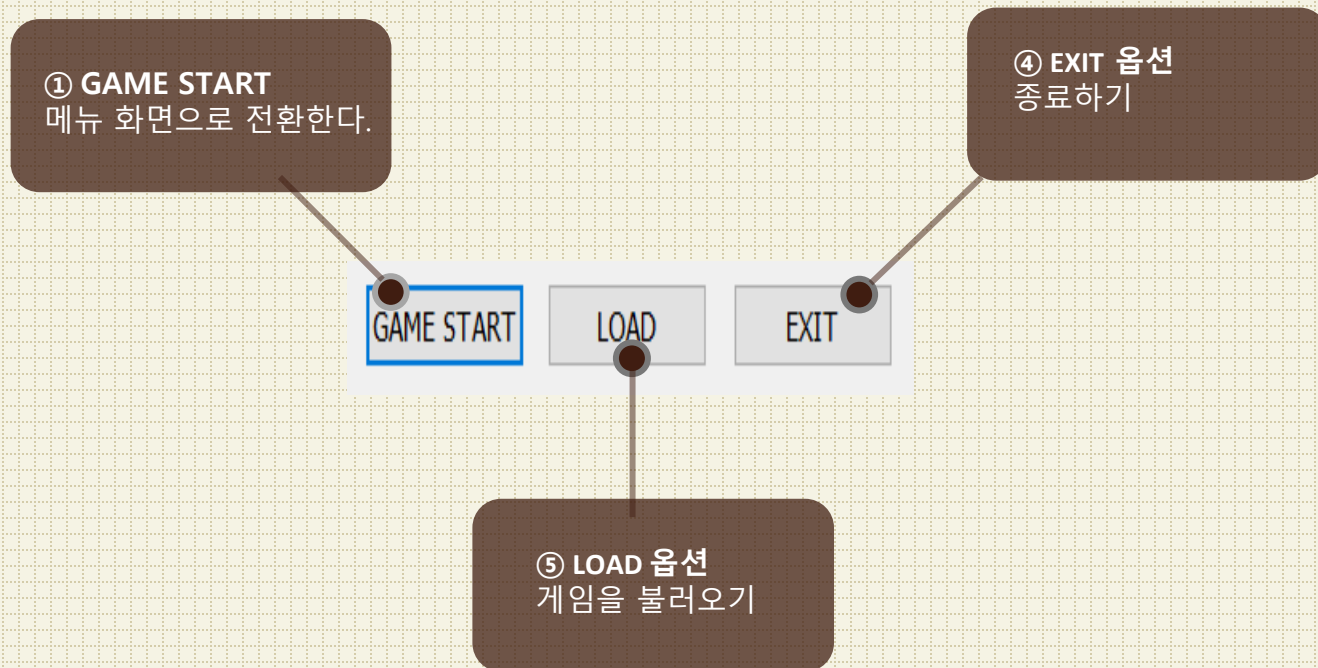
01. 게임 소개

02. 게임 진행 방법

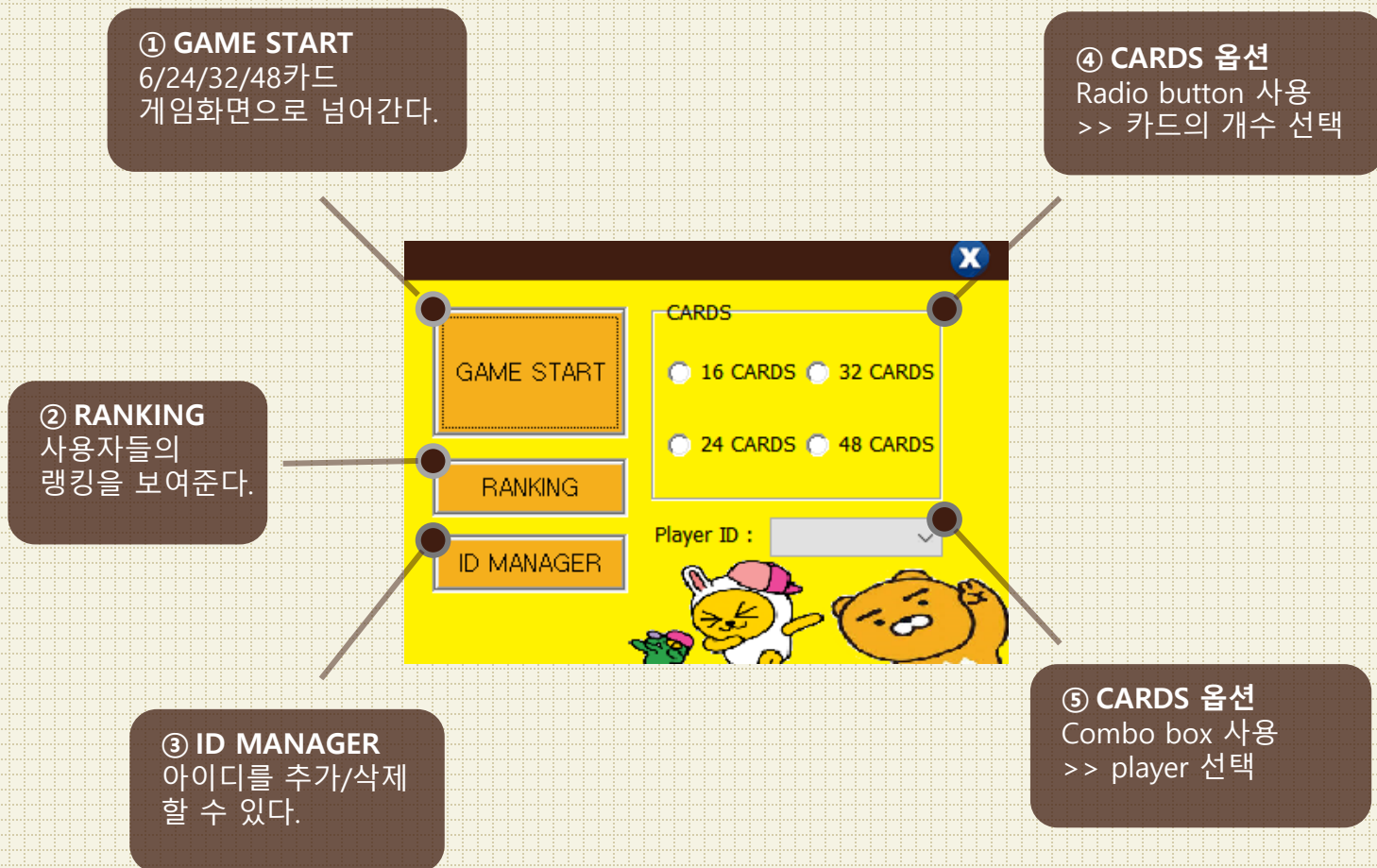
03. 게임 화면 및 시스템



게임 화면 UI



게임 화면 UI

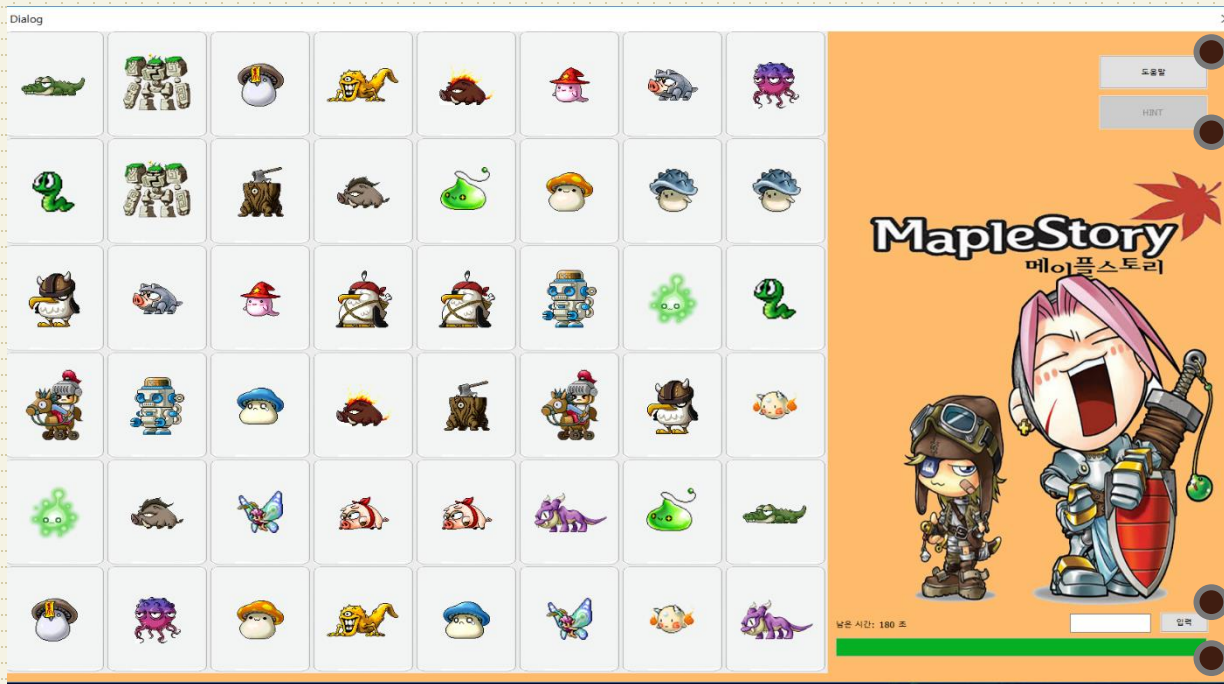


01. 게임 소개

02. 게임 진행 방법

03. 게임 화면 및 시스템

게임 화면 UI



도움말 표시

게임 설명 / 조작방법/ 힌트
사용 등에 대한 정보를 제공

힌트 사용

힌트 획득 후 버튼 클릭 시
카드를 다시 볼 수 있는 기회
제공

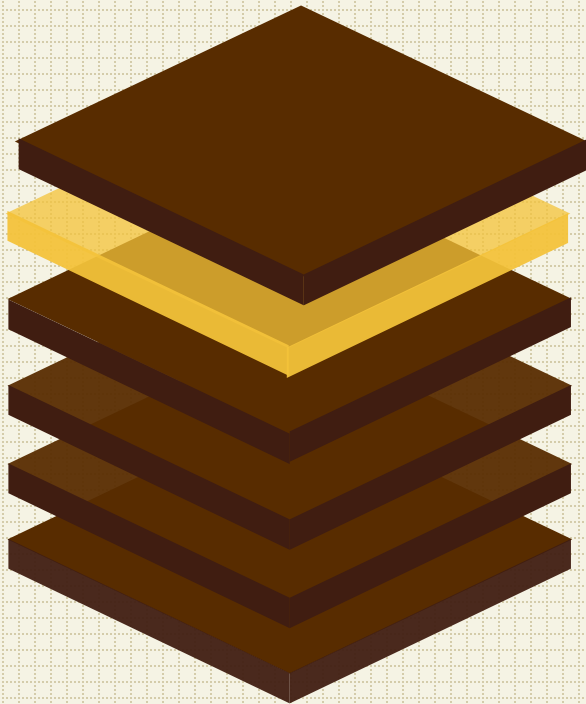
힌트 입력

이곳에 힌트의 답이
라고
생각되는 것을 입력

타이머

남은 시간을
시각적으로 표현

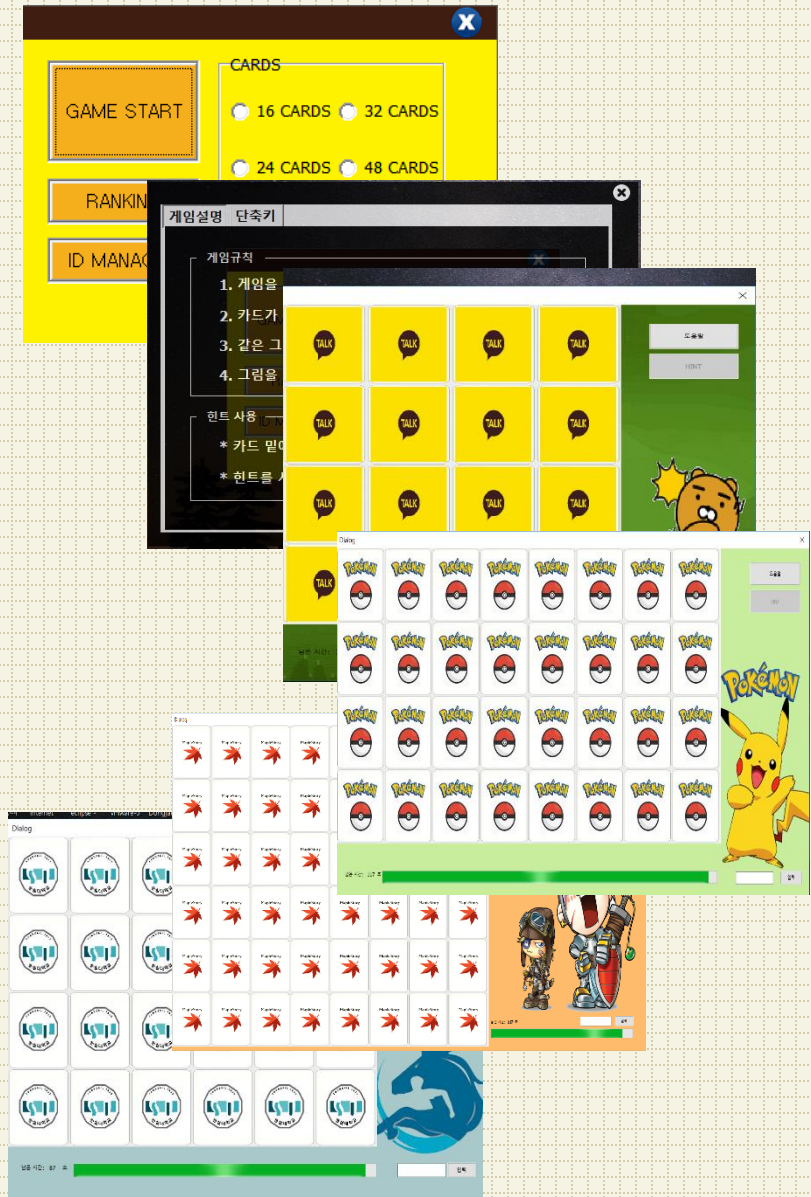
DLL 사용



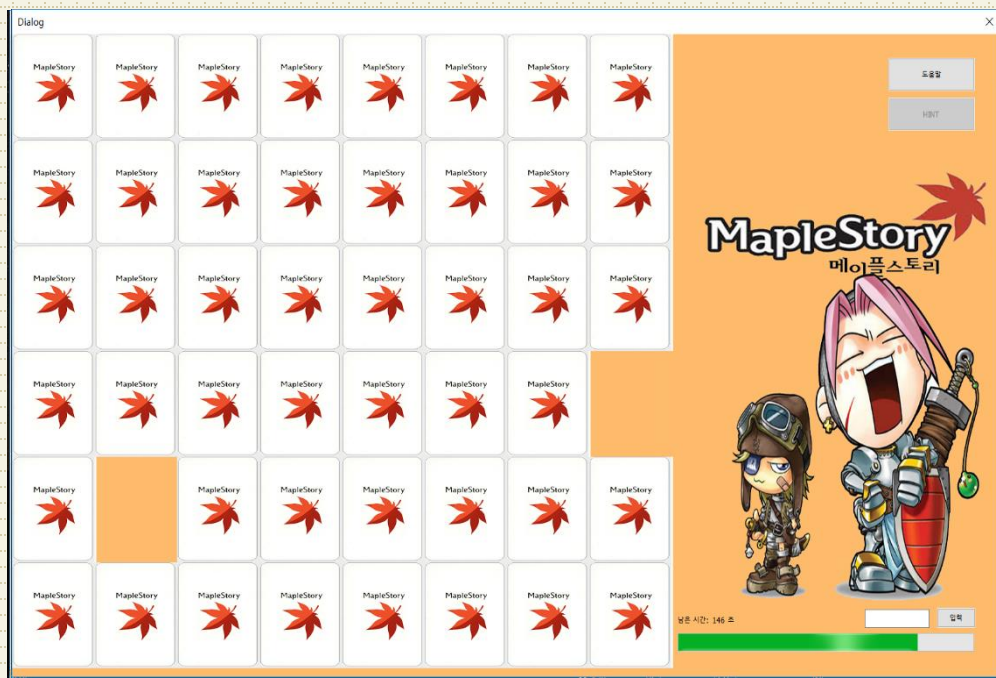
01. 게임 소개

02. 게임 진행 방법

03. 게임 화면 및 시스템



좌표 기반 게임 시스템



OnPaint()

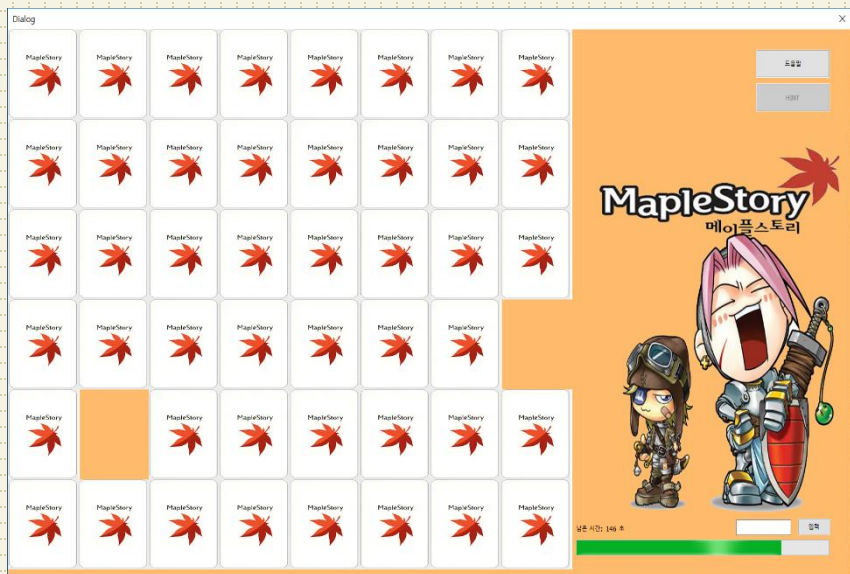
```
for (i = 0; i < REAL_CARD_COUNT * 2; i++) {
    if (m_CardTable[i] != -1) {
        index = m_CardTable[i] + 1;
        m_CardImage[index].Draw(dc, (i % 8)
            * 140, (i / 8) * 160);
    }
}
```

OnLButtonDown()

```
if (point.x >= 0 && point.y >= 0 && point.x <= 6 * 140
    && point.y <= 4 * 160){
    int x = point.x / 140;
    int y = point.y / 160;
    int selectIndex = y * 6 + x;
}
```

01. 게임 소개

게임 단축키



02. 게임 진행 방법

03. 게임 화면 및 시스템

```
BOOL Play24Cards::PreTranslateMessage(MSG* pMsg)
{
    ...
```

```
    if (pMsg->message == WM_KEYDOWN && pMsg->hwnd ==
        m_focusHint) {
```

```
        if (pMsg->wParam == VK_RETURN) {
            GetDlgItem(IDC_BUTTON_INPUT)->SetFocus();
            OnClickedButtonInput();
        }
        ...
```

```
    if (pMsg->message == WM_KEYDOWN && pMsg->wParam
        == VK_F1)
```

```
    {
        CHelpDialog CHelpDialog;
        CHelpDialog.DoModal();
        return TRUE;
    }
```

```
    switch (pMsg->wParam)
```

```
    {
        case 0x31:
```

```
        ...
```

```
        case 0x32:
```

```
        ...
```

파일 입출력

DataID - 메모장		
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)		
이영재	2	6880, 18000,
김현규	1	14490,
김현우B	1	6880,
최재현	1	6760,
슌거스	3	12960, 15660, 16560,
최유성	2	10530, 19260,

Load

```
void CDongjak90000Dlg::LoadTxt()
{
    FILE *set_file = NULL;
    set_file = fopen("../Data/DataID.txt", "r");
    ....
    for (int i = 0; !feof(set_file); i++) {
        fscanf(set_file, "%s\t\t\t%d\t\t\t", temp, &rankCnt);
        ....
        for (int j = 0; j < rankCnt; j++) {
            fscanf(set_file, "%d, ", &rankTemp[j]);
            ID[i].SetGameRank(rankTemp[j]);
        }
        fscanf(set_file, "\n", NULL);
        ....
    }
    fclose(set_file);
}
```

파일 입출력

DataID - 메모장		
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)		
이영재	2	6880, 18000,
김현규	1	14490,
김현우B	1	6880,
최재현	1	6760,
슌거스	3	12960, 15660, 16560,
최유성	2	10530, 19260,

Save

```
void CChoiceMode::SaveTxt()
```

```
{
    FILE *get_file = NULL;
    get_file = fopen("../Data\\DataID.txt", "w");
    ....
    for (int i = 0; i < IDCnt; i++) {
        fprintf(get_file, "%s\t\t\t%d\t\t\t",
            (CStringA)ID[i].GetUserID(), ID[i].m_RankCnt);
        for (int j = 0; j < ID[i].m_RankCnt; j++) {
            fprintf(get_file, "%d, ", ID[i].GetUserRank(j));
        }
        fprintf(get_file, "\n", NULL);
    }
    fclose(get_file);
}
```

랭킹 생성

RANKING			×
순위	ID	점수	
1	최유성	19260	
2	이영재	18000	
3	슌거스	16560	
4	슌거스	15660	
5	김현규	14490	
6	슌거스	12960	
7	최유성	10530	
8	김현우B	6880	
9	이영재	6880	
10	최재현	6760	

```
BOOL CRanking::OnInitDialog()
{
    ....
    < RK.totalRankCnt)
    {
        strCount.Format(_T("%d"), nCount + 1);
        m_listRanking.InsertItem(nCount, strCount);
        m_listRanking.SetItem(nCount, 1, LVIF_TEXT,
            RK.GetTotalName(nCount), 0, 0, 0, 0);
        m_listRanking.SetItem(nCount, 2, LVIF_TEXT,
            RK.GetTotalRank(nCount), 0, 0, 0, 0);
        nCount = m_listRanking.GetItemCount();
    }
    return TRUE;
}
```

랭킹 생성

```
void CRank::SetTotalRank(CUserID * ID, int IDCnt)
{
    CString *stemp;
    int *itemp;
    int totalCnt = 0;
    int index = 0;
    int tpi;
    CString tps;

    // 총 기록 갯수 구하기
    for (int i = 0; i < IDCnt; i++) {
        for (int j = 0; j < ID[i].m_RankCnt; j++) {
            totalCnt++;
        }
    }
    if (totalCnt < Rank_N) { this->totalRankCnt = totalCnt; }
    else { this->totalRankCnt = Rank_N; }
    stemp = new CString[totalCnt];
    itemp = new int[totalCnt];

    // 총 랭킹 다 넣기
    for (int i = 0; i < IDCnt; i++) {
        for (int j = 0; j < ID[i].m_RankCnt; j++) {
            stemp[index] = ID[i].GetUserID();
            itemp[index++] = ID[i].GetUserRank(j);
        }
    }
}
```

```
// 총 랭킹 정렬
for (int i = 0; i < totalCnt - 1; i++) {
    for (int j = i + 1; j < totalCnt; j++) {
        if (itemp[i] < itemp[j]) {
            tpi = itemp[i];
            itemp[i] = itemp[j];
            itemp[j] = tpi;
            tps = stemp[i];
            stemp[i] = stemp[j];
            stemp[j] = tps;
        }
    }
}

// 10개까지만 자르기
for (int i = 0; i < this->totalRankCnt; i++) {
    if (itemp[i] == 0) {
        m_TotalRank[i].Format(_T("(시간초과)"));
        m_TotalName[i] = stemp[i];
    }
    else {
        m_TotalRank[i].Format(_T("%d"), itemp[i]);
        m_TotalName[i] = stemp[i];
    }
}

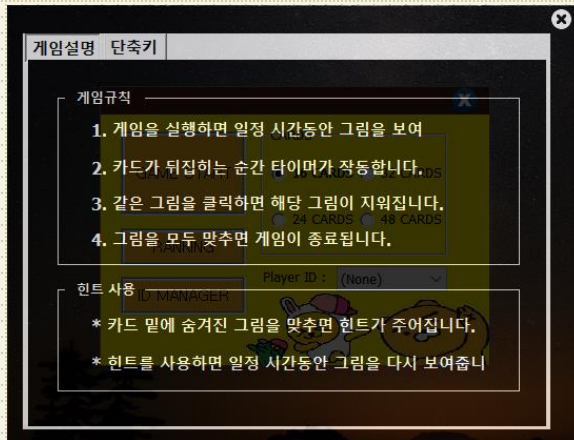
delete[] stemp;
delete[] itemp;
```

01. 게임 소개

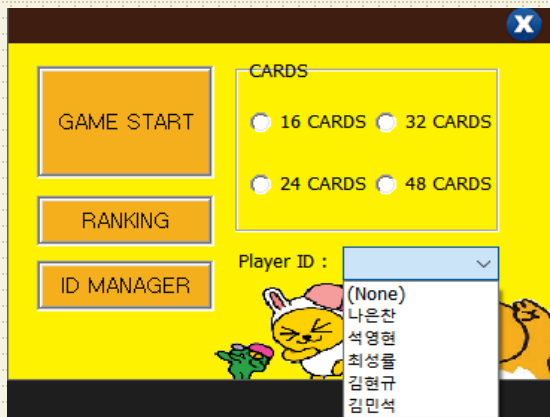
02. 게임 진행 방법

03. 게임 화면 및 시스템

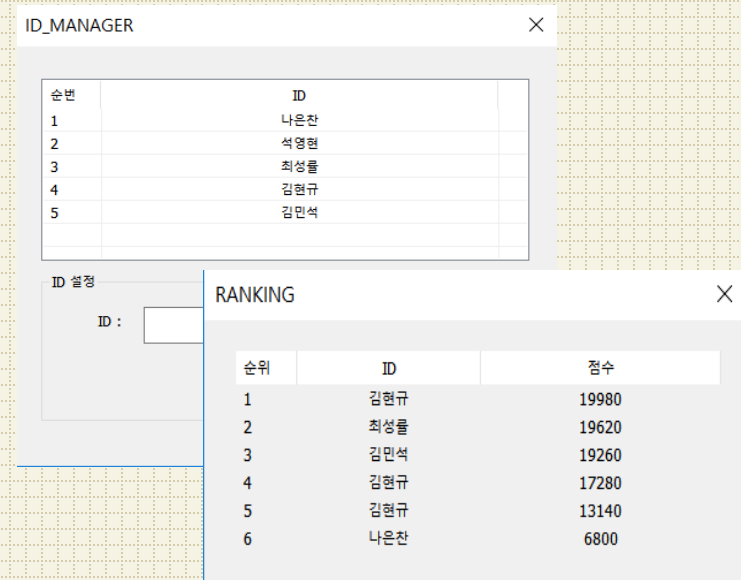
다양한 컨트롤 사용



탭 컨트롤을 활용한 도움말



콤보박스를 활용한 Player ID 리스트



리스트컨트롤을 활용한 ID_MANAGER / RANKING



프로그래스바를 활용한 타이머

DONJAK90000 ++

01. 게임 소개

02. 게임 진행 방법

03. 게임 화면 및 시스템

