

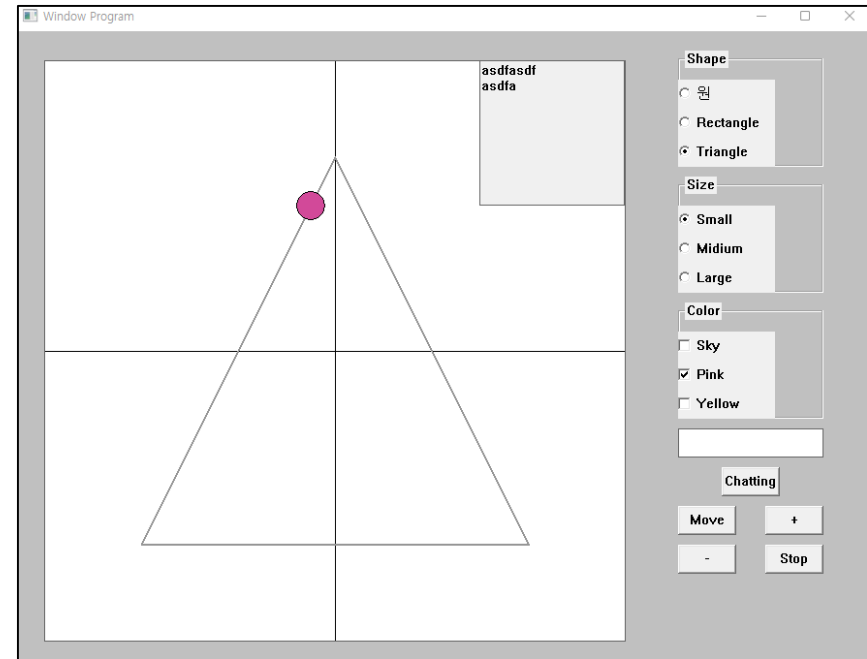
7장 차일드 윈도우

2022년도 1학기 윈도우 프로그래밍

실습 7-1

• 공전하는 원 만들기

- 좌측에 차일드 윈도우를 만들고 중앙을 원점으로 x축과 y축 좌표계를 그린다.
 - 좌표계에 선택된 형태의 곡선을 그린다.
 - 원이 곡선을 따라 이동한다.
- 우측에 컨트롤을 놓는다.
 - 라디오 버튼: 곡선 종류 (원 / 사각형 / 삼각형)
 - 라디오 버튼: 좌표계의 원의 크기 대 / 중 / 소
 - 색상: 움직이는 원의 색상 3종류
 - 에디트 박스: 문자열 입력
 - 버튼1: 에디트 박스의 문자열을 채팅 창에 출력
 - 버튼2: 원 이동
 - 버튼3: 이동 속도 증가
 - 버튼4: 이동 속도 감소
 - 버튼5: 이동 멈춤



• 차일드 윈도우와 컨트롤을 이용하여 연결된 그림 만들기

- 화면의 좌측에는 그림을 그릴 차일드 윈도우를 설정한다.
- 차일드 윈도우 좌/우에는 그림을 연결할 화살표 버튼을 놓는다.
 - 화살표 버튼을 누르면 좌우로 이동한다. (이미지를 좌우로 붙인다.)
 - 이미지 버튼을 만들 경우: 윈도우 스타일에 BS_BITMAP 을 추가하고, 대화상자의 컨트롤과 마찬가지로 SendMessage 함수로 비트맵 이미지를 버튼 위에 올려놓는다.

```
hButton[0] = CreateWindow ( L"button", L"1-이동", WS_CHILD | WS_VISIBLE | BS_PUSHBUTTON | BS_BITMAP, 10, 550, 50, 50, hwnd, (HMENU)IDC_BUTTON0, hInst, NULL);
```

```
hBit = LoadBitmap (hInst, MAKEINTRESOURCE(IDB_BITMAP1));
```

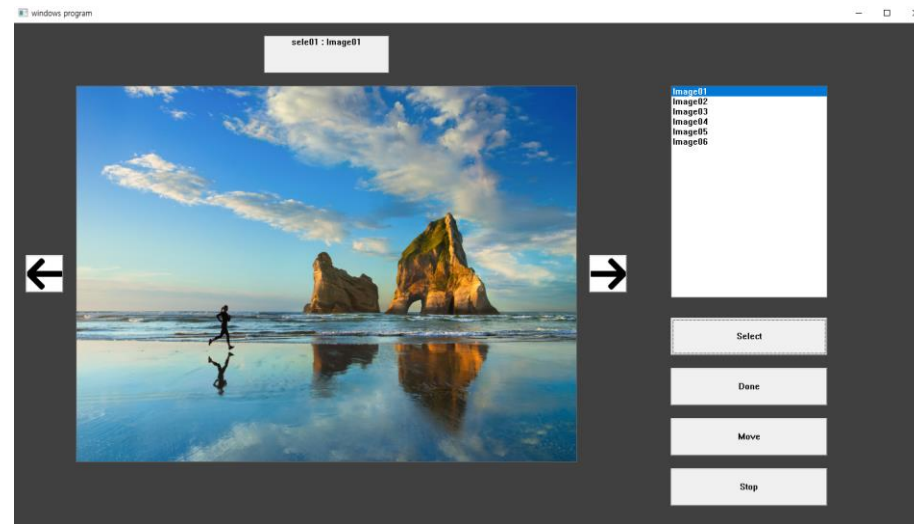
```
SendMessage (hButton[0], BM_SETIMAGE, 0 /*IMAGE_BITMAP*/, (LPARAM)hBit);
```

- 화면의 우측에는 콤보 박스 또는 리스트 박스를 이용하여 사용할 이미지 리스트를 선택할 수 있게 한다.
- 이미지를 선택 후 선택 버튼을 눌러 선택된 이미지를 좌측의 차일드 윈도우에 놓는다.
- 최대 10개의 이미지 (좌우로 움직일 수 있는 이미지)를 연결하여 놓을 수 있도록 한다.
- 버튼

- 선택: 리스트 박스에서 이미지 선택
- 이동: 이미지가 좌측으로 움직인다.
- 멈춤: 움직임이 멈춘다.
- 완성: 이미지 연결이 완성

- 에디트 박스

- 현재 선택된 이미지의 순서 출력



- **타뷰의 맵 만들기 (대화상자 또는 차일드 윈도우 사용하기)**
 - 맵 설정하기
 - **맵 크기**: 맵의 크기 설정하기 (가로와 세로 크기를 숫자로 입력)
 - **맵 격자 개수**: 맵의 가로와 세로의 격자 개수 (가로와 세로 격자의 개수 입력)
 - 배경 넣기
 - **배경 종류**: 이미지를 선택할 수 있는 리스트 박스 (배경 3개 사용)
 - 기능 컨트롤:
 - **그리드 그리기/해제하기**
 - 배경에 그리드 (격자)를 그리기 / 지우기
 - **장애물 넣기**
 - 장애물을 넣는 위치는 배경 이미지를 그리드에 맞춰 각 그리드 칸에 클릭하여 장애물을 배치 한다.
 - 장애물 종류 1) 통과할 수 없는 장애물 - 바위, 불
 - 장애물 종류 2) 통과할 수 있는 장애물 - 거미줄, 똥
 - **아이템 넣기**
 - 그리드에 맞춰 아이템을 배치한다.
 - 3종류의 아이템: 폭탄, 열쇠, 목숨
 - **적 배치하기**
 - 맵에 적을 배치한다. 보스도 배치하도록 한다.
 - 2종류의 몬스터: 좌우로 이동하는 몬스터, 상하로 이동하는 몬스터 (랜덤하게 이동하는 것도 무관함)
 - **테스트 버튼**
 - 제작한 맵을 테스트 하는 버튼: 팝업 윈도우를 띄우고 만든 맵을 테스트해본다.
 - 파일 입출력: 메뉴 또는 키보드 명령어 사용 → 다음 주, 파일 입출력을 배운 후 적용 (또는 C 표준 파일 입출력 사용)
 - 제작한 맵을 저장한다.
 - 저장된 맵을 읽는다.

• 버튼에 이미지 입히기

- 대화상자: 버튼의 속성에서 모양-이미지를 true로 바꿈

```
HBITMAP hBit;
```

```
HWND hButton;
```

```
hBit = LoadBitmap (hInst, MAKEINTRESOURCE(IDB_BITMAP1));
```

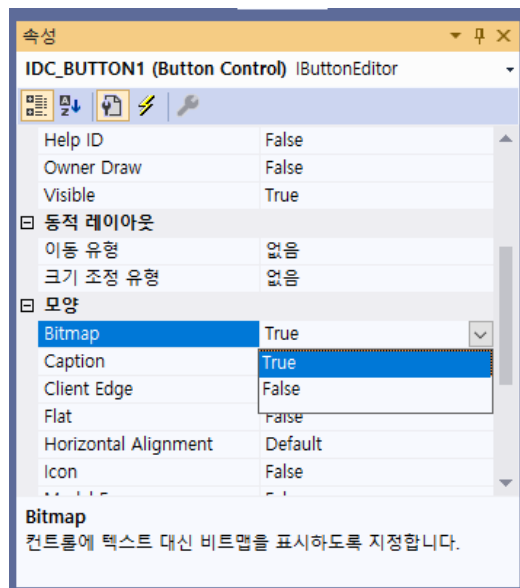
// 비트맵 로드

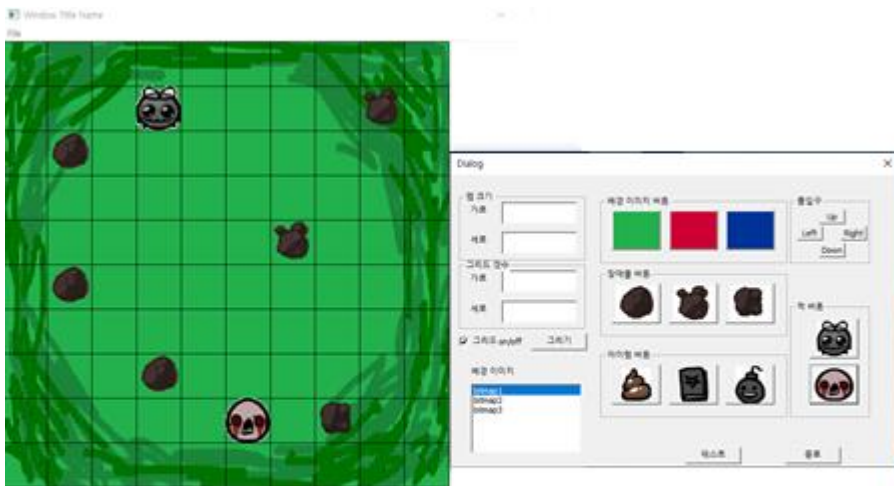
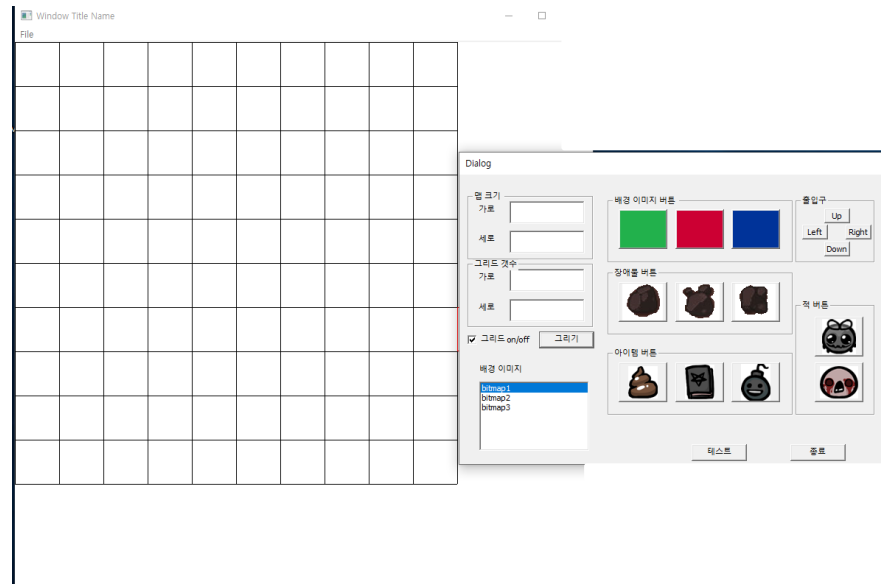
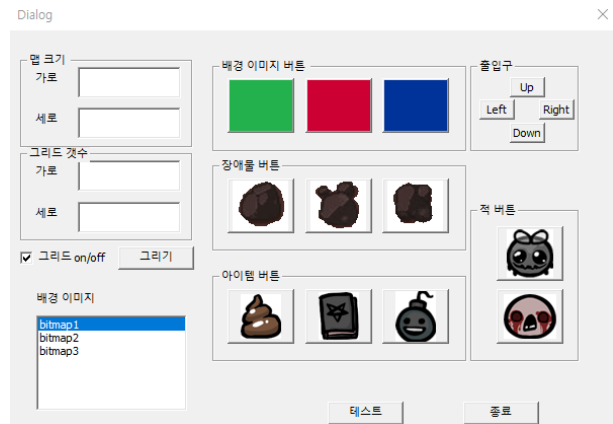
```
hButton = GetDlgItem (hwnd, IDC_BUTTON1);
```

// 버튼의 핸들 얻기

```
SendMessage (hButton, BM_SETIMAGE, 0, (LPARAM)hBit);
```

// 버튼에 메시지 보내기





- 대화상자 또는 차일드 윈도우 만들기
 - 맵틀 컨트롤 패널로 사용: 버튼, 리스트 박스, 그룹박스

- 보드 그리기
 - 이전 실습과 같이

- 몬스터 구조체 설정하기

```
typedef struct {
    POINT pos;           //--- 위치
    int type;             //--- 몬스터의 종류
    int direction;        //--- 몬스터의 이동 방향
    int status;           //--- 몬스터의 상태
    int speed;            //--- 몬스터의 이동 속도
} MONSTER;
```

- 마우스 이벤트 처리하기

- 마우스의 위치에 따라 보드의 위치 찾기
 - 보드의 칸이 비어 있다면, 선택된 객체를 넣기
 - 사용 예)

```
case WM_LBUTTONDOWN:
    mx = LOWORD (lParam);
    my = HIWORD (lParam);
    x = mx / stepX;           //--- stepX: 보드의 x축 한 칸 크기
    y = my / stepY;           //--- stepY: 보드의 y축 한 칸 크기
    if ( board[y][x] == 0 )   //--- 보드가 비어있다면
        board[y][x] = object; //--- 객체를 넣기
```

- 현재 어떤 객체 생성을 위한 마우스 선택인지 구분하여 처리하기
 - 장애물, 아이템, 적 인지에 따라 해당 객체 생성

- 객체 구조체 설정하기

- 장애물과 아이템 객체가 공용할 수 있도록 설정

```
typedef struct {
    POINT pos;           //--- 객체 위치
    int type;             //--- 객체 종류
    int status;           //--- 객체 상태
} OBJECT;
```

- 테스트하기

- 테스트 버튼 ➔ 팝업 차일드 윈도우 생성
 - `WS_POPUPWINDOW | WS_VISIBLE | WS_CAPTION` 스타일로 생성
 - `WM_CREATE` 메시지에서 타이머 설정하여 몬스터 이동하기

- 차일드 윈도우 생성 예)

```
hWndChild = CreateWindow (L"Child Window Class Name", L"Test Window Title Name", WS_POPUPWINDOW | WS_VISIBLE | WS_CAPTION, 0, 0, xWidth+20, yHeight+50, hWnd, NULL, hInst, NULL);
```

- 저장하기/읽기

- 맵 결과 저장하기
 - 보드 크기, 보드의 x축 y축 칸 수, 보드의 배경 이미지
 - 몬스터 데이터: 개수, 몬스터 구조체의 데이터
 - 객체 데이터: 개수, 객체 구조체의 데이터
- 맵 데이터 읽고 출력하기
 - 맵을 로드했을 때, 데이터 저장 형식에 따라 데이터들을 읽어 출력