



2D Game Programming 07

- 2D Game Engine

afewhee@gmail.com





- 2D Game Engine
 - ◆ 2D Game 제작을 위한 Library 집합
 - ◆ Game 제작의 편리성, 성능을 일정하게 유지

- 2D Game Engine 제작 방법
 - ◆ 추상 클래스와 전역 함수를 적절히 적용
 - ◆ 객체의 관리는 STL를 활용
 - ◆ 게임과 병행해서 라이브러리 제작 → Pair Coding



- 기본 함수 → 확장 함수
- 기본 객체 → 확장 객체

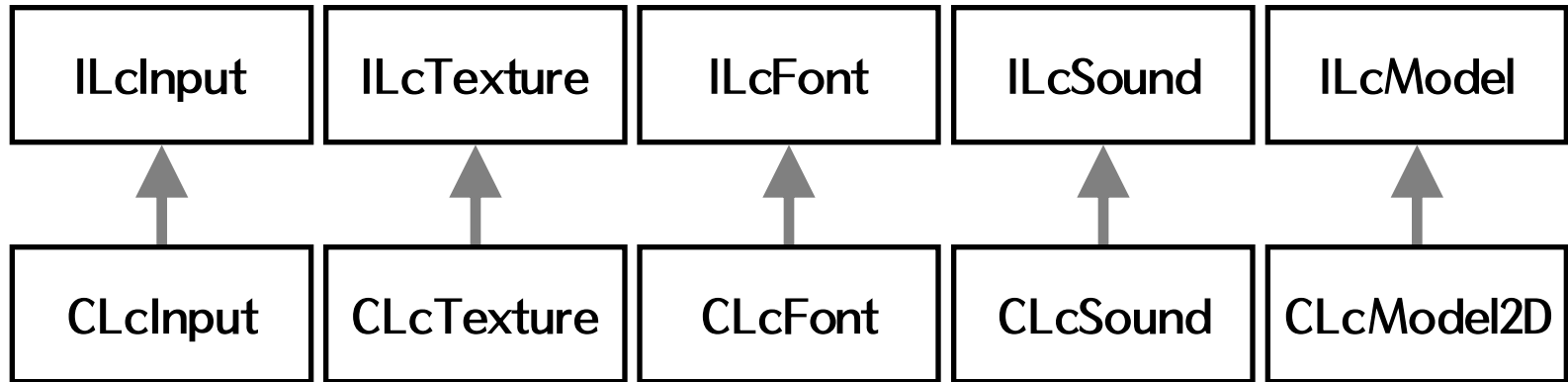
- 윈도우 생성/소멸 → 내부에서 DirectX를 초기화/해제
 - ◆ Sprite의 생성/소멸은 내부에서 처리

- 기본 객체를 생성하는 함수
 - ◆ 기본 객체: DirectX에서 제공되는 객체, 개발자가 정한 기본객체
 - ◆ 개발자가 정한 객체는 인터페이스만 있는 추상 클래스를 반드시 정함

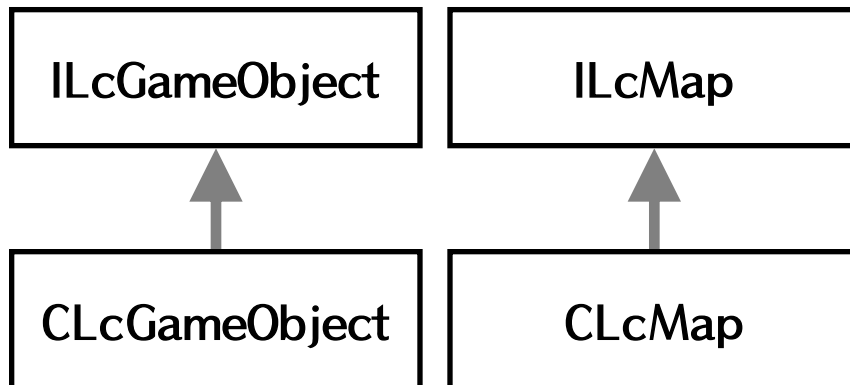
- 함수 포인터를 사용하여 게임 라이브러리와 연동
 - ◆ 주요 함수 포인터: 게임 객체 생성-초기화/해제, 데이터 갱신, 렌더링



● 기본 객체



● 확장 객체



3. 라이브러리 구성 - 인터페이스

- 윈도우
 - ◆ SpLib_CreateWin(): // 윈도우와 DirectX생성
 - ◆ SpLib_DestroyWin(): // 윈도우와 DirectX소멸
 - ◆ SpLib_Run(): // 게임 루프
- 함수 포인터 설정
 - ◆ SpLib_SetFrameMove(): // 게임 데이터 update 함수
 - ◆ SpLib_SetRender(): // 게임 데이터 렌더링 함수
 - ◆ SpLib_SetKeyboard(): // 게임엔진에 키보드 이벤트 함수 연결
 - ◆ SpLib_SetMouse(): // 게임엔진에 마우스 이벤트 함수 연결해
- Input(키보드, 마우스)
 - ◆ SpLib_GetKeyboard(): // 키보드(0~255)
 - ◆ SpLib_GetMouseX(): // 마우스 위치 X
 - ◆ SpLib_GetMouseY(): // 마우스 위치 Y
 - ◆ SpLib_GetMouseZ(): // 마우스 위치 Z
 - ◆ SpLib_GetMouseEvent(): // 마우스 이벤트 (0,1,2: L, R, M Button)
- 텍스처
 - ◆ SpLib_TextureLoad(): // 텍스처 객체 로딩
 - ◆ SpLib_TextureRelease(): // 텍스처 해제
 - ◆ SpLib_TextureWidth(): // 이미지 파일 너비
 - ◆ SpLib_TextureHeight(): // 이미지 파일 높이
 - ◆ SpLib_Draw2D(): // 텍스처 그리기
- 폰트
 - ◆ SpLib_FontCreate():
 - ◆ SpLib_FontDrawText():
- 사운드
 - ◆ SpLib_SoundLoad(): // 사운드 객체 로딩
 - ◆ SpLib_SoundRelease(): // 사운드 해제
 - ◆ SpLib_SoundPlay(): // 사운드 재생
 - ◆ SpLib_SoundStop(): // 사운드 멈춤
 - ◆ SpLib_SoundReset(): // 사운드 리셋
 - ◆ SpLib_SoundsPlaying(): // 사운드 실행중?





- 기본 객체 생성/소멸 함수들
 - ◆ Type 엔진이름_객체+{Create|Destroy} (...)

Ex)

```
int SpLib_TextureLoad(char* sFileName, DWORD dc)
int SpLib_TextureRelease(int _nKey)
```

```
BYTE SpLib_GetKeyboard();
int SpLib_GetMouseX();
int SpLib_GetMouseY();
int SpLib_GetMouseZ();
int SpLib_GetMouseEvent(int nMouse);
```

```
int SpLib_FontCreate(char* sName, LONG iH, BYTE iltalic)
Int SpLib_FontDrawText(int nIdx, ..., , const char *format, ...)
```

```
int SpLib_SoundLoad(char* sFileName)
int SpLib_SoundRelease(int _nKey)
void SpLib_SoundPlay(int _nKey)
```

- Direct3D의 객체를 기본 객체로 정한 게임 라이브러리 사용 예

```
void main()
{
    SpLib_SetClearColor(0xFF336699);
    SpLib_CreateWin(100, 100, 800, 600, "SpLib App", false);

    nTx_01 = SpLib_TextureLoad("Texture/lena.png");
    imgW_01 = SpLib_TextureWidth(nTx_01);
    imgH_01 = SpLib_TextureHeight(nTx_01);

    nTx_02 = SpLib_TextureLoad("Texture/mario2.png");
    imgW_02 = SpLib_TextureWidth(nTx_02);
    imgH_02 = SpLib_TextureHeight(nTx_02);

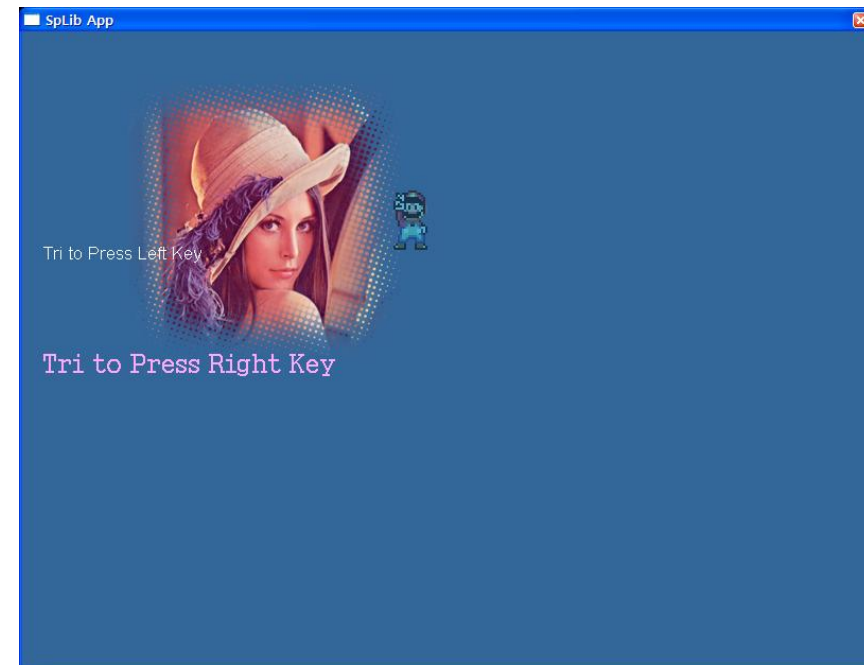
    nSound1 = SpLib_SoundLoad("sound/move3.wav");
    nSound2 = SpLib_SoundLoad("sound/trample.wav");

    nFont1 = SpLib_FontCreate("Arial", 20, 0);
    nFont2 = SpLib_FontCreate("궁서", 25, 0);

    SpLib_SetFrameMove(FrameMove);

    SpLib_SetRender(Render);
    SpLib_Run();

    SpLib_TextureRelease(nTx_01);
    SpLib_DestroyWin();
}
```





- DirectX의 객체를 Wrapping해서 기본 객체들을 구성할 경우 적절한 클래스 다이어그램과 연관관계를 표현하시오.
 - ◆ (3일차 클래스 디자인에서 실습한 내용을 적극 활용하시오.)

