



jokebox.

v4.04

사용설명서

1. Jokebox에 대해
2. 프로젝트 설치
3. 프레임워크 구조
4. Level B에 대하여
5. 게임 배포

Jokebox에 대해

Jokebox는 유사 CUI 및 2D 그래픽 게임 제작을 위한 **DirectX 랩핑 게임제작 프레임워크**입니다. Jokebox는 코더의 측면에서 '**최소한의 사전지식을 통한 최대한의 구현**'을 지향합니다. 본 프레임워크는 ▲원형 솔루션을 수정하여 사용하는 방식으로 별도의 설치가 필요 없으며, ▲그리기, 문자 열 쓰기 등의 거의 모든 절차를 함수 호출 한 번으로 통합하여 학습하기 쉽고, 사용하기 쉽습니다. 또한 ▲2D 게임 구현의 핵심만을 지원하여 배워야 할 양이 매우 작으며, ▲빌드 결과 폴더를 압축하여 배포하면 Windows Vista SP2 이상의 모든 컴퓨터에서 동작이 보장됩니다.

● Jokebox의 용도

Jokebox는 극-간소화를 추구하는 게임 프레임워크로서, 다음과 같은 게임 개발자에게 유용하게 사용될 수 있습니다.

- ✓ C 혹은 C++ 언어를 개발 체험과 함께 학습하고자 하는 경우
- ✓ 게임의 디테일보다 아이디어 구현에 초점을 둔 개발자
- ✓ 본 게임 제작에 앞서 간단한 프로토타입 제작이 필요한 개발자
- ✓ 취미용으로 간단한 게임을 제작하고 싶은 개발자

● Jokebox의 저작권

Jokebox는 [크리에이티브 커먼즈 저작자표시-동일조건변경허락 3.0 Unported 라이선스](#)에 따라 이용하실 수 있습니다. 라이선스를 준수하는 조건 하에서 자유롭게 Jokebox의 기능을 추가, 보완하거나, 이를 배포할 수 있습니다.

프로젝트 설치

Jokebox는 별도의 설치가 필요하지 않은 프레임워크입니다. Jokebox는 원본 소스형태로 배포되며, 새로 만들 게임은 Jokebox 원본 프로젝트의 **game.c**에 구현하면 됩니다.

TIP 확장명을 **.cpp**로 변경하면 C++ 환경에서 게임을 제작할 수 있습니다.

Jokebox 를 이용해 개발하기 위해서는 **DirectX 10.1** 이상 버전과 **Visual C++ 2010** 이상이 설치되어 있어야 합니다.

프레임워크 구조

게임 구현은 **game.c** 소스파일을 기반으로 이루어지며, 필요하다면 소스파일을 추가하여 구현할 수 있습니다. 이번 장은 **game.c** 소스파일과 관련된 프레임워크의 작동 방법에 대하여 알아봅니다.

```
#define JOKEBOX_LEVEL_A
#define JOKEBOX_LEVEL_B
#include "jokebox.h"
#include "jokeboxhelper.h"
#include "game.h"

void initialize()
{
    // TODO: 게임이 최초로 시작되었을 때 초기화할 것을 작성.
    //   ex) 캐릭터의 처음 위치 세팅, 변수값 초기화..
}

void update(GAMETIME gametime)
{
    // TODO: 게임 화면이 출력되기 전에 필요한 연산들을 작성.
    //   ex) 시간을 측정해서 캐릭터의 위치 변경..
}

void draw()
{
    draw_begin();
    draw_clear(BLACK);

    // TODO: 이 사이에 draw_draw() 함수를 이용해 출력할 것을 작성.

    draw_end();
}
```

Table 1. game.c 원형

- Jokebox의 함수 호출

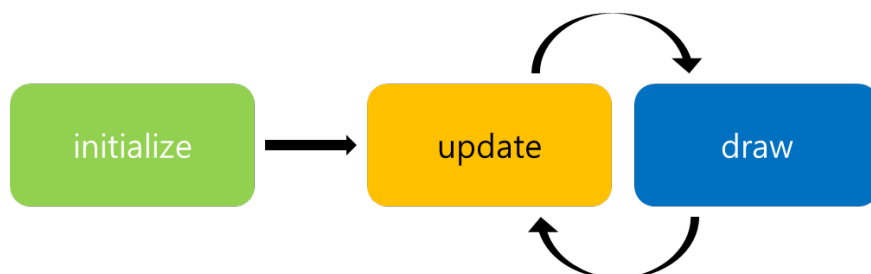


Figure 1. Jokebox의 함수 호출 순서

game.c에는 Jokebox 프레임워크에 의해 자동으로 호출되는 함수¹ 세 개가 기본으로 정의되어 있습니다. 위 그림은 Jokebox가 함수를 호출하는 순서를 간단히 표현한 것입니다.

➤ **initialize()**

initialize 함수는 게임 프로그램 실행 이후 최초에, 단 한 번만 실행됩니다. 이 함수 내부에는 게임에 필요한 각종 변수를 초기화하는 코드를 작성합니다.

예를 들어, 캐릭터가 등장하는 게임의 경우, 처음 게임이 시작했을 때 캐릭터의 위치 등을 이 함수에서 작성합니다.

➤ **update(GAMETIME gametime)**

initialize 함수가 종료된 후부터 게임 프로그램이 종료되기 전까지, Jokebox는 **update↔draw** 함수 순서로 무한히 호출하는데, 호출은 컴퓨터가 호출이 가능할 때 실시간으로 일어납니다.

update 함수에는 게임 동작을 위한 연산 코드나, 변수 값을 새로 갱신하는 코드를 작성합니다. 가령 방향키가 눌렸는지 체크한 후에, 방향키가 눌렸을 경우 캐릭터의 위치를 조정하도록 하는 코드를 작성해야 하는 경우, 이 함수 내에 작성합니다.

인수로 전달되는 **gametime**은 update와 관련된 시간 정보를 **ms(밀리세컨드)** 단위로 담고 있습니다. GAMETIME 구조체에 대한 정보는 레퍼런스 문서에서 확인하실 수 있습니다.

➤ **draw()**

update 함수가 종료된 후에 자동으로 **draw** 함수가 호출됩니다.

draw 함수에는 그래픽을 그리는 코드를 작성합니다. 그리는 종류에 따라 다양한 그리기 함수를 이용할 수 있으며, 이들 함수의 목록과 사용 방법은 레퍼런스 문서에서 확인하실 수 있습니다.

● Jokebox의 함수 체계

Jokebox은 난이도와 복잡도에 따라 함수를 두 가지 레벨로 나누어 제공합니다. **Level A** 함수들은 유사 CUI 환경의 게임을 제작하는 데 용이한 함수들입니다. 위치와 색상과 같은 기본적인 정보만으로 직관적인 그래픽 출력을 할 수 있으며, 엄격한 칸(cell) 단위 그래픽 처리가 이루어져 그래픽적인 복잡성이 배제되어 있습니다. 한편 **Level B** 함수들은 2D 게임을 제작할 수 있는 정도의

¹ '콜백Callback 함수'라고 칭합니다.

함수들로서, Level A 함수들에게서 지켜졌던 엄격한 칸 단위 그래픽 처리가 완화되고, 외부 리소스 개념이나 간단한 변형(Transform)과 같은 개념이 추가되었습니다. Level B에서는 Level A에 비하여 훨씬 더 자유로운 표현이 지원됩니다.

두 가지 레벨은 엔진의 헤더 파일인 "jokebox.h"가 include 되는 줄 앞에서, 레벨에 해당하는 상수(define constant)를 정의해줌으로써 사용할 수 있습니다. 참고로 두 레벨을 동시에 사용하는 것도 허용됩니다.

```
#define JOKEBOX_LEVEL_A
//#define JOKEBOX_LEVEL_B
#include "jokebox.h"
#include "jokeboxhelper.h"
#include "game.h"
```

Table 2. Level A 함수를 활성화하고, Level B 함수를 비활성화한 상태.

Level B를 사용하기 위해선 "#define JOKEBOX_LEVEL_B"의 주석을 해제합니다.

- 그래픽 출력

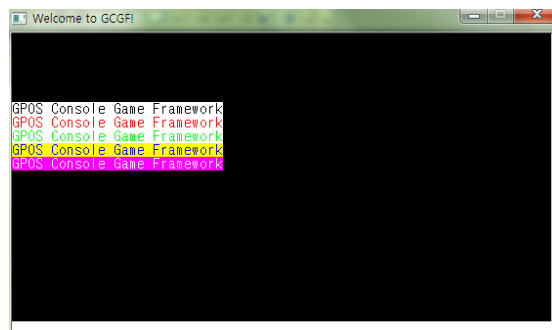


Figure 2. Jokebox 데모 화면

Jokebox는 **칸(cell) 단위**를 대부분의 경우 위치나 길이의 기본 단위로 삼습니다. 칸을 기본 단위로 삼는 것은 유사 CUI 환경에서 그래픽을 출력할 때 직관성을 확보하기 위한 것으로, 유사 CUI 환경뿐 아니라 2D 환경에서도 위치나 길이의 지나친 디테일을 미연에 방지하는 역할을 합니다. 칸의 크기는 **기본값 가로 14 px, 세로 14 px**로 설정되어 있지만, game_set_cell_size() 함수를 이용하여 칸의 크기를 조절할 수 있습니다.

Jokebox의 2D 좌표계는 **왼쪽 위**를 항상 원점으로 하고, x축의 증가 방향을 **왼쪽→오른쪽**으로, y축의 증가 방향을 **위쪽→아래쪽**으로 정의하고 있습니다.

LEVEL B에 대하여

Level B는 Level A에 비해 보다 자유로운 2D 표현이 지원되고, 사운드, 이미지, 폰트와 같은 외부 리소스 사용 방법이 제공됩니다. 이 장에서는 Level A와 구별되는 Level B의 특징을 살펴보고, Level B 함수 사용 방법을 간단히 알아봅니다.

- Level B에서만 지원되는 사항

대표적으로 Level B에서 새롭게 지원되는 기능들은 아래와 같습니다.

- 콘텐츠(content) 생성/사용: 폰트, 사운드, 텍스처
- 엄격한 칸 규제 해제
- 2D 도형 그리기

Level B에서 새롭게 지원되는 함수들은 **엄격한 칸 규제가 해제**되어 있습니다. Level A에서 모든 것이 칸에 묶여있었다면, Level B에서는 좌표 값으로 소수가 지원되어 길이, 위치 표현에 실질적인 제약이 없습니다. 또한 사각형, 타원과 같은 **2D 기본 도형 그리기** 함수가 추가되어 손쉽게 2D를 표현할 수 있으며, 외부 리소스(**컨텐츠**) 사용도 지원되어 디스크에 저장된 폰트나 사운드, 텍스처 파일을 불러와 게임상에서 사용할 수 있습니다.

- 함수 명명규칙

Level B에는 함수에 특정 옵션을 추가로 지정할 수 있도록 한 함수들이 함께 제공됩니다. 이러한 함수들은 원래 함수의 이름 마지막에 특정 문자가 덧붙여진 형태로 원래 함수와 구분되는데, 아래는 덧붙여진 글자들의 의미입니다.

| 표기 | 의미 | 관련 옵션(인수가 추가됨) |
|----|----------|-------------------|
| c | Color | 색상에 관련된 옵션 |
| p | Partial | 텍스처 부분 출력과 관련된 옵션 |
| r | Rotation | 회전과 관련된 옵션 |

- 콘텐츠(content) 이용하기

컨텐츠는 게임상에서 사용되기에 앞서 먼저 생성되어야 합니다. **컨텐츠 생성**은 생성하고자 하는 컨텐츠 분류에 맞는 content_load_... 함수를 이용해 생성할 수 있는데, 실제 생성된 컨텐츠는 엔진

내부에서 관리되고, 사용자는 이 콘텐츠들의 ID만 이 함수들의 반환값으로 받을 수 있습니다. 생성한 콘텐츠를 이용하기 위해서 사용자는 이 때 반환받은 ID를 보관하고 있어야 합니다.

콘텐츠 사용에서 한 가지 유의해야 할 점은, 콘텐츠 종류마다 **불러오는 경로가 고정되어** 있다는 점입니다. Visual Studio에서 곧바로 게임 프로그램을 실행시키는 경우, 콘텐츠를 불러오는 고정된 경로는 아래와 같습니다.

| 콘텐츠 | 경로 |
|-----|------------------------------|
| 폰트 | (메인 프로젝트 경로)\Content\Font |
| 사운드 | (메인 프로젝트 경로)\Content\Sound |
| 텍스처 | (메인 프로젝트 경로)\Content\Texture |

* (메인 프로젝트 경로)는 게임을 작성중인 프로젝트 파일이 있는 경로

단 빌드 후 생성되는 게임 프로그램의 경우, Content 폴더를 **해당 프로그램이 존재하는 폴더 내**에서 검색하게 됩니다. Visual Studio에서 빌드가 일어나면 메인 프로젝트 경로에 있던 Content 폴더가 그대로 빌드 결과 폴더로 복사되므로 일반적으로는 신경 쓸 필요가 없지만, 코드 수정 없이 **콘텐츠만 바뀌는 경우에는**, 수동으로 빌드 결과 폴더의 Content에 존재하는 콘텐츠와 메인 프로젝트 경로의 Content에 존재하는 콘텐츠 **두 쪽 모두를 갱신**해주어야 합니다. 참고로, 배포되는 프로그램에 사용되는 콘텐츠는 빌드 결과 폴더의 Content에 존재하는 콘텐츠들입니다.

같은 콘텐츠(ex. 같은 사운드 파일, 같은 이미지 파일)는 게임 실행 중 **단 한 번만** 생성되도록 권장됩니다. 따라서 콘텐츠 생성은 실질적으로 initialize(초기화) 함수에서 모두 이루어지는 것이 좋으며, 이 때 나중에 이용될 수 있는 콘텐츠까지 미리 생성해두는 것이 좋습니다.

```
// #define JOKEBOX_LEVEL_A
#define JOKEBOX_LEVEL_B
#include "jokebox.h"
#include "game.h"

FONT test_font;
TEXTURE test_texture;
SOUND test_sound;

void initialize()
{
    test_font = content_load_font("궁서체 ", 0.3f, 20);
    test_sound = content_load_sound("test.wav");
    test_texture = content_load_texture("test.png");
}
```

Table 3. initialize 함수에서 콘텐츠를 생성하는 코드.

콘텐츠 사용은 함수들을 통해 제공되며, 이 함수들의 사용 방법은 Jokebox의 다른 함수들의 사용 방법과 크게 다르지 않습니다. 가령 draw_texture 함수를 이용해 텍스처를 그리고, sound_play

함수를 이용해 소리를 출력하는 방식입니다. 어떠한 함수가 콘텐츠 사용에 이용되는지 알아보려면 [레퍼런스 문서](#)에서 Level B에서 제공되는 함수들의 목록을 참조해 주십시오.

함수 이름이 'draw_'로 시작되는 폰트나 텍스처 그리기 함수의 경우, 여타 'draw_' 함수들처럼 draw_begin()와 draw_end() 사이에서 사용되어야 합니다. 소리 출력과 관련된 'sound_' 함수들은 실질적으로 어느 곳에서나 사용될 수 있지만, 성능 및 코드 일관성을 위해 되도록 update 함수 내에서만 사용하는 것을 권장합니다.

```
void initialize()
{
    [...]
    sound_fx_play( test_sound);
}

void draw(GAMETIME gametime)
{
    draw_begin();
    draw_clear(BLACK);

    draw_texture(test_texture, 3, 5, 2, 1.3f, 1);
    draw_text("Hello World", test_font, 10, 1, BLACK, 1);

    draw_end();
}
```

Table 4. 콘텐츠를 사용하는 예.

게임 배포

Jokebox로 제작된 게임은 배포를 위해 별도의 과정을 필요로 하지 않으며, **빌드 결과가 만들어지는 폴더(ex: Debug 혹은 Release)를 그대로 압축**하여 묶는 것만으로 모든 배포 준비가 완료됩니다. 순수 Jokebox 프레임워크만을 이용하는 게임의 경우, Windows Vista SP2 이상의 모든 환경에서 구동이 보장됩니다.

이 저작물은 [크리에이티브 커먼즈 저작자표시-비영리-동일조건변경허락 3.0 Unported 라이선스](#)에 따라 이용할 수 있습니다. (원작자 **이광무, OrangeYellow**)