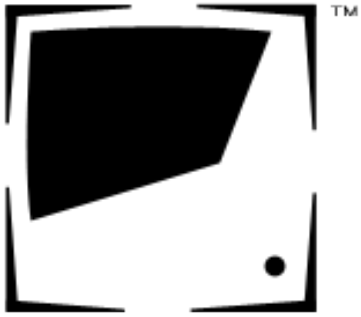


jokebox.

v4.05

라이브러리 레퍼런스

Powered by GPOS The World Creatives

JOKEBOX 정의 타입

- 열거형
  - COLORS *기본 색상을 정의한 열거형*
  - KEYS *키보드의 키코드(keycode)를 정의한 열거형*
  - MOUSEBUTTON *마우스 버튼을 정의한 열거형*
  - ANGLE *주요 각도를 정의한 열거형*
  - TRIANGLEDIR *삼각형의 방향을 정의한 열거형*

- 구조체
  - GAMETIME *게임 내 시간을 저장하는 구조체*

- 정의 타입
  - COLOR *색상을 나타내는 타입*
  - FONT *폰트의 ID를 저장하는 타입*
  - TEXTURE *텍스처의 ID를 저장하는 타입*
  - SOUND *사운드의 ID를 저장하는 타입*

JOKEBOX는 사용의 편의성을 높이거나, 높은 자유도로 인해 야기되는 복잡함을 방지하기 위하여 몇 가지 타입을 정의합니다. 이번 장에서는 JOKEBOX가 정의한 타입에 대하여 알아봅니다.

● 열거형(Enum)

✓ COLORS

요약: DirectX에 정의된 색을 열거한 열거형입니다.

상세: 열거형의 값은 unsigned int형으로 표현되어 있으며, 순서대로 Alpha, Red, Green, Blue 값을 1Byte 단위로 담고 있습니다.

정의:

```
typedef enum Colors
{
        ALICE_BLUE = 0xFFF0F8FF,
        ANTIQUE_WHITE = 0xFFFAEBD7,
        AQUA = 0xFF00FFFF,
        AQUAMARINE = 0xFF7FFFD4,
        AZURE = 0xFFF0FFFF,
        BEIGE = 0xFFF5F5DC,
        BISQUE = 0xFFFFE4C4,
        BLACK = 0xFF000000,
        BLANCHED_ALMOND = 0xFFFFEBCD,
        BLUE = 0xFF0000FF,
        BLUE_VIOLET = 0xFF8A2BE2,
        BROWN = 0xFFA52A2A,
```

```
        BURLY_WOOD = 0xFFDEB887,
        CADET_BLUE = 0xFF5F9EA0,
        CHARTREUSE = 0xFF7FFF00,
        CHOCOLATE = 0xFFD2691E,
        CORAL = 0xFFFF7F50,
        CORNFLOWER_BLUE = 0xFF6495ED,
        CORNSILK = 0xFFFFF8DC,
        CRIMSON = 0xFFDC143C,
        CYAN = 0xFF00FFFF,
        DARK_BLUE = 0xFF00008B,
        DARK_CYAN = 0xFF008B8B,
        DARK_GOLDENROD = 0xFFB8860B,
        DARK_GRAY = 0xFFA9A9A9,
        DARK_GREEN = 0xFF006400,
        DARK_KHAKI = 0xFFBDB76B,
        DARK_MAGENTA = 0xFF8B008B,
        DARK_OLIVE_GREEN = 0xFF556B2F,
        DARK_ORANGE = 0xFFFF8C00,
        DARK_ORCHID = 0xFF9932CC,
        DARK_RED = 0xFF8B0000,
        DARK_SALMON = 0xFFE9967A,
        DARK_SEA_GREEN = 0xFF8FBC8F,
        DARK_SLATE_BLUE = 0xFF483D8B,
        DARK_SLATE_GRAY = 0xFF2F4F4F,
        DARK_TURQUOISE = 0xFF00CED1,
        DARK_VIOLET = 0xFF9400D3,
        DEEP_PINK = 0xFFFF1493,
        DEEP_SKY_BLUE = 0xFF00BFFF,
        DIM_GRAY = 0xFF696969,
        DODGER_BLUE = 0xFF1E90FF,
        FIREBRICK = 0xFFB22222,
        FLORAL_WHITE = 0xFFFFFAF0,
        FOREST_GREEN = 0xFF228B22,
        FUCHSIA = 0xFFFF00FF,
        GAINSBORO = 0xFFDCDCDC,
        GHOST_WHITE = 0xFFF8F8FF,
        GOLD = 0xFFFFD700,
        GOLDENROD = 0xFFDAA520,
        GRAY = 0xFF808080,
        GREEN = 0xFF008000,
        GREEN_YELLOW = 0xFFADFF2F,
        HONEYDEW = 0xFFF0FFF0,
        HOT_PINK = 0xFFFF69B4,
        INDIAN_RED = 0xFFCD5C5C,
        INDIGO = 0xFF4B0082,
        IVORY = 0xFFFFFFF0,
        KHAKI = 0xFFF0E68C,
        LAVENDER = 0xFFE6E6FA,
        LAVENDER_BLUSH = 0xFFFFF0F5,
        LAWN_GREEN = 0xFF7CFC00,
        LEMON_CHIFFON = 0xFFFFFACD,
        LIGHT_BLUE = 0xFFADD8E6,
        LIGHT_CORAL = 0xFFF08080,
        LIGHT_CYAN = 0xFFE0FFFF,
        LIGHT_GOLDENROD_YELLOW = 0xFFFAFAD2,
```

```
        LIGHT_GREEN = 0xFF90EE90,
        LIGHT_GRAY = 0xFFD3D3D3,
        LIGHT_PINK = 0xFFFFB6C1,
        LIGHT_SALMON = 0xFFFFA07A,
        LIGHT_SEA_GREEN = 0xFF20B2AA,
        LIGHT_SKY_BLUE = 0xFF87CEFA,
        LIGHT_SLATE_GRAY = 0xFF778899,
        LIGHT_STEEL_BLUE = 0xFFB0C4DE,
        LIGHT_YELLOW = 0xFFFFFFE0,
        LIME = 0xFF00FF00,
        LIME_GREEN = 0xFF32CD32,
        LINEN = 0xFFFAF0E6,
        MAGENTA = 0xFFFF00FF,
        MAROON = 0xFF800000,
        MEDIUM_AQUAMARINE = 0xFF66CDAA,
        MEDIUM_BLUE = 0xFF0000CD,
        MEDIUM_ORCHID = 0xFFBA55D3,
        MEDIUM_PURPLE = 0xFF9370DB,
        MEDIUM_SEA_GREEN = 0xFF3CB371,
        MEDIUM_SLATE_BLUE = 0xFF7B68EE,
        MEDIUM_SPRING_GREEN = 0xFF00FA9A,
        MEDIUM_TURQUOISE = 0xFF48D1CC,
        MEDIUM_VIOLET_RED = 0xFFC71585,
        MIDNIGHT_BLUE = 0xFF191970,
        MINT_CREAM = 0xFFF5FFFA,
        MISTY_ROSE = 0xFFFFE4E1,
        MOCCASIN = 0xFFFFE4B5,
        NAVAJO_WHITE = 0xFFFFDEAD,
        NAVY = 0xFF000080,
        OLD_LACE = 0xFFFDF5E6,
        OLIVE = 0xFF808000,
        OLIVE_DRAB = 0xFF6B8E23,
        ORANGE = 0xFFFFA500,
        ORANGE_RED = 0xFFFF4500,
        ORCHID = 0xFFDA70D6,
        PALE_GOLDENROD = 0xFFEEE8AA,
        PALE_GREEN = 0xFF98FB98,
        PALE_TURQUOISE = 0xFFAFEEEE,
        PALE_VIOLET_RED = 0xFFDB7093,
        PAPAYA_WHIP = 0xFFFFEFD5,
        PEACH_PUFF = 0xFFFFDAB9,
        PERU = 0xFFCD853F,
        PINK = 0xFFFFC0CB,
        PLUM = 0xFFDDA0DD,
        POWDER_BLUE = 0xFFB0E0E6,
        PURPLE = 0xFF800080,
        RED = 0xFFFF0000,
        ROSY_BROWN = 0xFFBC8F8F,
        ROYAL_BLUE = 0xFF4169E1,
        SADDLE_BROWN = 0xFF8B4513,
        SALMON = 0xFFFA8072,
        SANDY_BROWN = 0xFFF4A460,
        SEA_GREEN = 0xFF2E8B57,
        SEA_SHELL = 0xFFFFF5EE,
        SIENNA = 0xFFA0522D,
```

```
        SILVER = 0xFFC0C0C0,
        SKY_BLUE = 0xFF87CEEB,
        SLATE_BLUE = 0xFF6A5ACD,
        SLATE_GRAY = 0xFF708090,
        SNOW = 0xFFFFFAFA,
        SPRING_GREEN = 0xFF00FF7F,
        STEEL_BLUE = 0xFF4682B4,
        TAN = 0xFFD2B48C,
        TEAL = 0xFF008080,
        THISTLE = 0xFFD8BFD8,
        TOMATO = 0xFFFF6347,
        TURQUOISE = 0xFF40E0D0,
        VIOLET = 0xFFEE82EE,
        WHEAT = 0xFFF5DEB3,
        WHITE = 0xFFFFFFFF,
        WHITE_SMOKE = 0xFFF5F5F5,
        YELLOW = 0xFFFFFF00,
        YELLOW_GREEN = 0xFF9ACD32,
        TRANS = 0x00000000
}COLORS;
```

✓ KEYS

요약: 게임에서 자주 사용되는 키의 명칭과 키코드 값을 대응시킨 열거형입니다.

정의:

```
typedef enum Keys
{
        KEY_ENTER       = 0x0D,  KEY_ESC          = 0x1B, KEY_SPACE       = 0x20,
        KEY_SHIFT       = 0x10, KEY_CTRL = 0x11, KEY_ALT          = 0x12,
        KEY_RSHIFT      = 0xA0, KEY_RCTRL       = 0xA2, KEY_RALT = 0xA4,
        KEY_LSHIFT      = 0xA1, KEY_LCTRL       = 0xA3, KEY_LALT = 0xA5,
        KEY_LEFT        = 0x25, KEY_UP          = 0x26, KEY_RIGHT       = 0x27,
        KEY_DOWN        = 0x28,
        KEY_0           = 0x30,
        KEY_1, KEY_2, KEY_3, KEY_4, KEY_5, KEY_6, KEY_7, KEY_8, KEY_9,
        KEY_A           = 0x41,
        KEY_B, KEY_C, KEY_D, KEY_E, KEY_F, KEY_G, KEY_H, KEY_I,
        KEY_J, KEY_K, KEY_L, KEY_M, KEY_N, KEY_O, KEY_P, KEY_Q,
        KEY_R, KEY_S, KEY_T, KEY_U, KEY_V, KEY_W, KEY_X, KEY_Y, KEY_Z,
        KEY_TAB         = 0x09,
        KEY_NUMPAD0     = 0x60,
        KEY_NUMPAD1, KEY_NUMPAD2, KEY_NUMPAD3, KEY_NUMPAD4, KEY_NUMPAD5,
        KEY_NUMPAD6, KEY_NUMPAD7, KEY_NUMPAD8, KEY_NUMPAD9,
        KEY_ADD         = 0x6B,  KEY_SUB         = 0x6D, KEY_MUL         = 0x6A,
        KEY_DIV         = 0x6F,
        KEY_F1          = 0x70,
        KEY_F2, KEY_F3, KEY_F4, KEY_F5, KEY_F6, KEY_F7,
        KEY_F8, KEY_F9, KEY_F10, KEY_F11, KEY_F12
}KEYS;
```

✓ MOUSEBUTTON

요약: 마우스 버튼을 지정하는 데 사용되는 열거형입니다.

상세:

값	설명
MOUSE_LEFT	마우스 왼쪽 버튼
MOUSE_RIGHT	마우스 오른쪽 버튼

✓ ANGLE

요약: 주요 각도를 정의한 열거형입니다.

상세:

값	설명
ANGLE_ZERO	0 도
ANGLE_CCW45	반시계 방향으로 45 도
ANGLE_CCW90	반시계 방향으로 90 도
ANGLE_CCW135	반시계 방향으로 135 도
ANGLE_REV	180 도
ANGLE_CW45	시계 방향으로 45 도
ANGLE_CW90	시계 방향으로 90 도
ANGLE_CW135	시계 방향으로 135 도

✓ TRIANGLEDIR

요약: 삼각형의 방향을 지정하는데 사용되는 열거형입니다.

상세:

값	설명
TDIR_UP	위쪽 방향 삼각형
TDIR_DOWN	아래쪽 방향 삼각형
TDIR_LEFT	왼쪽 방향 삼각형
TDIR_RIGHT	오른쪽 방향 삼각형

- 구조체(Struct)

✓ GAMETIME

요약: 게임과 관련된 시간을 저장하는 구조체입니다. GAMETIME 타입의 변수 gametime은 update 함수의 인수로 전달되며, 각 인수에 저장된 값은 다음과 같은 의미를 갖습니다.

정의:

```
typedef struct Gametime
{
        unsigned long elapsed;
        unsigned long total;
```

```
}GAMETIME;
```

설명:

필드 이름	값
elapsed	지난 update 호출부터 이번 update 호출까지 걸린 시간
total	게임이 시작된 이후에 지난 총 시간

- 정의 타입

✓ COLOR

요약: 색상을 나타내는데 사용되는 타입입니다.

상세: draw 함수에서 색상 정보를 받을 때 이용되는 타입이며, 내부적으로 unsigned int 형으로 정의되었습니다.

참고: RGB 함수나 RGBA 함수를 이용하면, Red, Blue, Green, Alpha 값을 각각 입력하여 COLOR 타입을 만들 수 있습니다.

RGB(r, g, b)	*[r, g, b는 0~255 사이의 적/녹/청색 값]*
RGBA(r, g, b, a)	*[r, g, b, a는 0~255 사이의 적/녹/청색 값]*

✓ FONT

요약: 폰트의 ID를 저장하는 타입입니다.

상세: content_load_font() 함수에서 할당되는 폰트의 ID를 저장하는 타입으로, 내부적으로 unsigned int 형으로 정의되었습니다.

✓ TEXTURE

요약: 텍스처의 ID를 저장하는 타입입니다.

상세: content_load_texture() 함수에서 할당되는 텍스처의 ID를 저장하는 타입으로, 내부적으로 unsigned int 형으로 정의되었습니다.

✓ SOUND

요약: 사운드의 ID를 저장하는 타입입니다.

상세: content_load_sound() 함수에서 할당되는 사운드의 ID를 저장하는 타입으로, 내부적으로 unsigned int 형으로 정의되었습니다.

JOKEBOX 정의 함수

JOKEBOX에서는 사용자가 그래픽을 출력하고 키보드 입력을 확인하는 등의 작업을 할 수 있도록 몇 가지 함수를 제공합니다. 이 장에서는 용도별로 나누어, JOKEBOX에서 정의된 함수에 대해 알아봅니다.

- 그래픽 출력 관련 함수 (draw)

  - 공통
    - draw_begin — *그리기(draw) 시작을 선언*
    - draw_clear — *화면을 특정 색상으로 지우기*
    - draw_end — *그리기(draw) 종료를 선언*

  - Level A
    - draw_char — *문자를 출력*
      [draw_char / draw_charc]
    - draw_string — *문자열을 출력*
      [draw_string / draw_stringc]
    - draw_area — *지정한 사각형 영역에 문자를 출력*
      [draw_area / draw_areac]

  - Level B
    - draw_rect — *사각형을 그리기*
      [draw_rect / draw_rectr]
    - draw_triangle — *삼각형을 그리기*
      [draw_triangle / draw_triangler]
    - draw_polygon — *다각형을 그리기*
      [draw_polygon / draw_polygonr]
    - draw_ellipse — *타원을 그리기*
      [draw_ellipse / draw_ellipser]
    - draw_line — *선을 그리기*
    - draw_text — *텍스트를 그리기*
      [draw_text / draw_textr]
    - draw_texture — *텍스처를 그리기*
      [draw_texture / draw_texturer / draw_texturep / draw_texturepr]

그래픽을 출력할 때 사용하는 함수들입니다. 이 분류의 함수들은 공통적으로 함수 이름이 **'draw'**로 시작됩니다.

✓ **void** draw_begin()

요약: 그리기 시작을 선언합니다. **화면에 그리기 전에 반드시 이 함수가 먼저 호출되어야 합니다.**

✓ **void** draw_clear(COLOR color)

요약: 인수로 전달한 색상(color)로 화면을 지웁니다. 주로 draw_begin을 호출한 직후에 호출되어, 지난번 게임 루프(update↔draw)에서 그렸던 화면을 지우는 역할을 수행합니다.

설명:

인수 이름	전달할 값
color	화면을 지울 색깔

✓ **void** draw_end()

요약: 그리기 끝을 선언합니다. **그리기를 모두 끝낸 후에 반드시 이 함수를 호출해야 합니다.** 통상적으로 draw 함수의 가장 마지막에서 호출합니다.

✓ **void** draw_char(int word, int x, int y) [LEVEL: **A**]

요약: 좌표 (x, y) 지점에 글자 word를 출력합니다. 이 함수를 통해 출력할 경우, 자동으로 글자색은 흰색, 배경색은 검은색으로 설정됩니다.

설명:

인수 이름	전달할 값
word	출력할 글자
x	출력할 x 좌표
y	출력할 y 좌표

✓ **void** draw_charc(int word, int x, int y, COLOR color, COLOR backcolor) [LEVEL: **A**]

요약: 좌표 (x, y) 지점에 글자 word를 출력합니다. 글자색은 color로, 배경색은 backcolor로 설정됩니다.

설명:

인수 이름	전달할 값
word	출력할 글자
x	출력할 x 좌표
y	출력할 y 좌표
color	글자색

backcolor	배경색

참고: 함수 이름 마지막의 c는 'Color'의 약식 표기입니다.

✓ **void** draw_string(int x, int y, char* string, ...) [LEVEL: **A**]

요약: 좌표 (x, y) 지점에 포맷 문자열 string 출력합니다. 이 함수를 통해 출력할 경우, 자동으로 글자색은 흰색, 배경색은 검은색으로 설정됩니다.

설명:

인수 이름	전달할 값
string	출력할 포맷 문자열
x	출력할 x 좌표
y	출력할 y 좌표
...	포맷 문자열에 사용될 추가 인수

✓ **void** draw_stringc(int x, int y, COLOR color, COLOR backcolor, char* string, ...) [LEVEL: **A**]

요약: 좌표 (x, y) 지점에 문자열 string을 출력합니다. 글자색은 color로, 배경색은 backcolor로 설정됩니다.

설명:

인수 이름	전달할 값
string	출력할 문자열
x	출력할 x 좌표
y	출력할 y 좌표
color	글자색
backcolor	배경색
...	포맷 문자열에 사용될 추가 인수

참고: 함수 이름 마지막의 c는 'Color'의 약식 표기입니다.

✓ **void** draw_area(char word, int x1, int y1, int x2, int y2) [LEVEL: **A**]

요약: (x1, y1) 칸부터 (x2, y2) 칸 사이에 있는 사각형 영역을 글자 word로 채웁니다. 이 함수를 통해 출력할 경우, 자동으로 글자색은 흰색, 배경색은 검은색으로 설정됩니다.

설명:

인수 이름	전달할 값
word	출력할 글자
x	출력할 사각형의 왼쪽 위의 x 좌표
y	출력할 사각형의 왼쪽 위의 y 좌표
width	사각형의 너비

height	사각형의 높이

✓ **void** draw_areac(char word, int x1, int y1, int x2, int y2, COLOR color, COLOR backcolor) [LEVEL: **A**]

요약: (x1, y1) 칸부터 (x2, y2) 칸 사이에 있는 사각형 영역을 글자 word로 채웁니다. 글자색은 color로, 배경색은 backcolor로 설정됩니다.

설명:

인수 이름	전달할 값
word	출력할 글자
x	출력할 x 좌표
y	출력할 y 좌표
width	사각형의 너비
height	사각형의 높이
color	글자색
backcolor	배경색

참고: 함수 이름 마지막의 c는 'Color'의 약식 표기입니다.

✓ **void** draw_rect (float x, float y, float width, float height, COLOR color, float stroke);
void draw_rectr (float x, float y, float width, float height, COLOR color, float stroke, float cx, float cy, float rotation); [LEVEL: **B**]

요약: 사각형을 그립니다.

상세: (x, y) 위치를 왼쪽 위 꼭지점으로 하여 사각형을 그립니다.

좌표 숫자와 가로/세로 폭에 소수점이 허용됩니다. 즉 (x, y) = (4.5, 3.2)와 같은 소수점 칸을 입력하는 것도 가능합니다.

회전 중심 좌표 (cx, cy) 역시 기준점은 화면의 왼쪽 위이며, 회전 각도 rotation은 반시계방향으로 증가하며, 도(°, degree) 단위를 사용합니다.

설명:

인수 이름	전달할 값
x	출력할 x 좌표 (단위: 칸)
y	출력할 y 좌표 (단위: 칸)
width	사각형의 너비 (단위: 칸)
height	사각형의 높이 (단위: 칸)
color	채우기 색
stroke	테두리 굵기 (단위: px)
cx	회전 중심 x 좌표 (기준: 화면의 왼쪽 위)
cy	회전 중심 y 좌표 (기준: 화면의 왼쪽 위)

rotation	회전 각도 (단위: 도(deg))

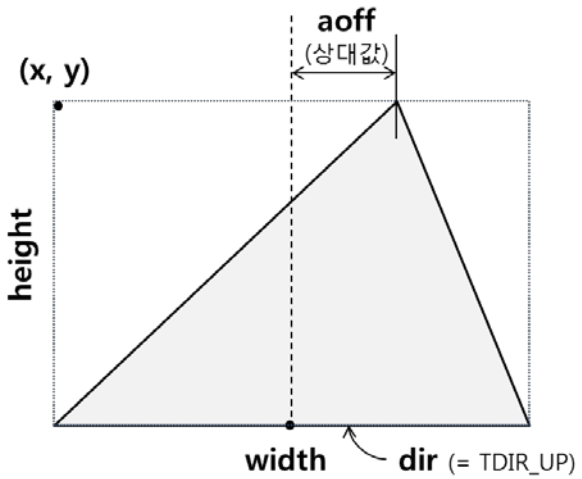
참고: 함수 이름 마지막의 r는 'Rotation'의 약식 표기입니다.

✓ **void** draw_triangle (float x, float y, float width, float height, float aoff, TRIANGLEDIR dir, COLOR color, float stroke);

void draw_triangler (float x, float y, float width, float height, float aoff, TRIANGLEDIR dir, COLOR color, float stroke, float cx, float cy, float rotation); [LEVEL: **B**]

요약: 삼각형을 그립니다.

상세:



사각형 (x, y, width, height)을 기준으로 삼각형을 그립니다.

dir은 삼각형의 방향을 나타내는 변수입니다. 삼각형의 방향은 꼭지점이 기준 사각형의 어느 변 위에 있는 지로 결정되는데, 가령 그림과 같이 꼭지점이 기준 사각형의 윗(UP)변에 있으면 '위쪽' 방향이 됩니다. (반대로 이와 마주보는 변인 아랫변은 삼각형의 밑변이 됩니다.)

aoff는 밑변의 중심에 대한 꼭지점(Apex)의 상대적인 오프셋으로, dir에 따라 아래와 같은 값을 갖습니다.

dir	값
TDIR_UP, TRIR_DOWN	width에 대한 상대값. (오른쪽으로 증가)
TDIR_LEFT, TDIR_RIGHT	height에 대한 상대값. (아래쪽으로 증가)

가령 dir이 TDIR_UP인 상황에서 aoff가 0이면 이등변 삼각형, aoff가 0.5이면 오른쪽 각이 직각인 직각 삼각형이 됩니다.

좌표 숫자와 가로/세로 폭에 소수점이 허용됩니다. 즉 (x, y) = (4.5, 3.2)와 같은 소수점 칸을 입력하는 것도 가능합니다.

회전 중심 좌표 (cx, cy) 역시 기준점은 화면의 왼쪽 위이며, 회전 각도 rotation은 반시계방향으로 증가하며, 도(°, degree) 단위를 사용합니다.

설명:

인수 이름	전달할 값
x	출력할 x 좌표 (단위: 칸)
y	출력할 y 좌표 (단위: 칸)
width	사각형의 너비 (단위: 칸)
height	사각형의 높이 (단위: 칸)
aoff	밑변 중심에 대한 꼭지점의 상대적 위치 (단위: 밑변 길이에 대한 상대값)
dir	삼각형의 방향
color	채우기 색
stroke	테두리 굵기 (단위: px)
cx	회전 중심 x 좌표 (기준: 화면의 왼쪽 위)
cy	회전 중심 y 좌표 (기준: 화면의 왼쪽 위)
rotation	회전 각도 (단위: 도(deg))

참고: 함수 이름 마지막의 r는 'Rotation'의 약식 표기입니다.

✓ **void** draw_polygon (float x, float y, float vertices[], int vCount, COLOR color, float stroke);
void draw_polygonr (float x, float y, float vertices[], int vCount, COLOR color, float stroke
float cx, float cy, float rotation); [LEVEL: **B**]

요약: 지정된 점을 꼭지점으로 갖는 다각형을 그립니다.

상세: vertices에 입력된 좌표에서 (x, y)만큼 이동된 점으로 이루어진 다각형을 그립니다.
vertices에는 다음과 같은 순서로 점이 입력되어야 합니다.

{ (1번 점의 x), (1번 점의 y), (2번 점의 x), (2번 점의 y), ..., (n번 점의 x), (n번 점의 y) }

좌표 숫자와 가로/세로 폭에 소수점이 허용됩니다. 즉 (x, y) = (4.5, 3.2)와 같은 소수점 칸을 입력하는 것도 가능합니다.

회전 중심 좌표 (cx, cy) 역시 기준점은 화면의 왼쪽 위이며, 회전 각도 rotation은 반시계방향으로 증가하며, 도(°, degree) 단위를 사용합니다.

설명:

인수 이름	전달할 값
x	출력할 x 좌표 (단위: 칸)
y	출력할 y 좌표 (단위: 칸)
vertices	꼭지점 목록 (단위: 칸)
vCount	꼭지점의 개수
color	채우기 색
stroke	테두리 굵기 (단위: px)
cx	회전 중심 x 좌표 (기준: 화면의 왼쪽 위)

cy	회전 중심 y 좌표 (기준: 화면의 왼쪽 위)
rotation	회전 각도 (단위: 도(deg))

✓ **void** draw_ellipse (float x, float y, float radX, float radY, COLOR color, float stroke);

void draw_ellipser (float x, float y, float radX, float radY, COLOR color, float stroke, float cx, float cy, float rotation); [LEVEL: **B**]

요약: 타원을 그립니다.

상세: 좌표 숫자와 가로/세로 폭에 소수점이 허용됩니다. 즉 (x, y) = (4.5, 3.2)와 같은 소수점 칸을 입력하는 것도 가능합니다.

회전 중심 좌표 (cx, cy) 역시 기준점은 화면의 왼쪽 위이며, 회전 각도 rotation은 반시계방향으로 증가하며, 도(°, degree) 단위를 사용합니다.

설명:

인수 이름	전달할 값
x	출력할 x 좌표 (단위: 칸)
y	출력할 y 좌표 (단위: 칸)
radX	x축 방향의 반지름 (단위: 칸)
radY	y축 방향의 반지름 (단위: 칸)
color	채우기 색
stroke	테두리 굵기 (단위: px)
cx	회전 중심 x 좌표 (기준: 화면의 왼쪽 위)
cy	회전 중심 y 좌표 (기준: 화면의 왼쪽 위)
rotation	회전 각도 (단위: 도(deg))

참고: 함수 이름 마지막의 r는 'Rotation'의 약식 표기입니다.

✓ **void** draw_line (float x1, float y1, float x2, float y2, float width, COLOR color, float stroke); [LEVEL: **B**]

요약: 선을 그립니다.

상세: 좌표 숫자와 가로/세로 폭에 소수점이 허용됩니다. 즉 (x, y) = (4.5, 3.2)와 같은 소수점 칸을 입력하는 것도 가능합니다.

설명:

인수 이름	전달할 값
x1	시작점의 x 좌표 (단위: 칸)
y1	시작점의 y 좌표 (단위: 칸)
x2	끝점의 x 좌표 (단위: 칸)
y2	끝점의 y 좌표 (단위: 칸)
width	선의 굵기 (단위: px)

color	채우기 색
stroke	테두리 굵기 (단위: px)

✓ **void** draw_text (char* text, FONT font, float x, float y, COLOR color, float stroke);

void draw_textr (char* text, FONT font, float x, float y, COLOR color, float stroke, float cx, float cy, float rotation); [LEVEL: **B**]

요약: 텍스트을 출력합니다.

상세: 이용하고자 하는 폰트(font)는 사전에 만들어져 있어야 합니다. 자세한 내용은 content_load_font() 함수를 참조하십시오.

font로 **NULL**을 전달하는 경우 **LEVEL A**에서 사용되는 기본 폰트가 사용되며, 크기는 단위 셀의 크기를 갖습니다.

텍스트 출력 위치 (x, y)는 텍스트 첫 글자의 왼쪽 위 모서리의 위치입니다.

좌표 숫자와 가로/세로 폭에 소수점이 허용됩니다. 즉 (x, y) = (4.5, 3.2)와 같은 소수점 칸을 입력하는 것도 가능합니다.

회전 중심 좌표 (cx, cy) 역시 기준점은 화면의 왼쪽 위이며, 회전 각도 rotation은 반시계방향으로 증가하며, 도(°, degree) 단위를 사용합니다.

설명:

인수 이름	전달할 값
text	출력할 텍스트
font	텍스트 출력에 이용될 폰트
x	출력할 x 좌표 (단위: 칸)
y	출력할 y 좌표 (단위: 칸)
color	채우기 색
stroke	테두리 굵기 (단위: px)
cx	회전 중심 x 좌표 (기준: 화면의 왼쪽 위)
cy	회전 중심 y 좌표 (기준: 화면의 왼쪽 위)
rotation	회전 각도 (단위: 도(deg))

참고: 함수 이름 마지막의 r는 'Rotation'의 약식 표기입니다.

✓ **void** draw_texture (TEXTURE texture, float x, float y, float width, float height, float opacity);

void draw_texturer (TEXTURE texture, float x, float y, float width, float height, float opacity, float cx, float cy, float rotation);

void draw_texturep (TEXTURE texture, float x, float y, float width, float height, float opacity, float sourceX, float sourceY, float sourceWidth, float sourceHeight);

void draw_texturepr(TEXTURE texture, float x, float y, float width, float height, float opacity,

float sourceX, float sourceY, float sourceWidth, float sourceHeight,

float cx, float cy, float rotation); [LEVEL: **B**]

요약: 텍스처을 출력합니다.

상세: 이용하고자 하는 텍스처(texture)는 사전에 만들어져 있어야 합니다. 자세한 내용은 content_load_texture() 함수를 참조하십시오.

텍스처 출력 위치 (x, y)는 텍스처의 왼쪽 위 모서리의 위치입니다.

부분만 출력하는 경우(texturep, texturepr), source로 시작하는 변수들의 값은 모두 가로(혹은 세로)를 1로 놓은 상태에서의 0~1 사이의 **상대값**입니다.

좌표 숫자와 가로/세로 폭에 소수점이 허용됩니다. 즉 (x, y) = (4.5, 3.2)와 같은 소수점 칸을 입력하는 것도 가능합니다.

회전 중심 좌표 (cx, cy) 역시 기준점은 화면의 왼쪽 위이며, 회전 각도 rotation은 반시계방향으로 증가하며, 도(°, degree) 단위를 사용합니다.

설명:

인수 이름	전달할 값
texture	출력할 텍스처
x	출력할 x 좌표 (단위: 칸)
y	출력할 y 좌표 (단위: 칸)
width	텍스처의 너비 (단위: 칸)
height	텍스처의 높이 (단위: 칸)
opacity	투명도 (최소 0 ~ 최대 1)
sourceX	(텍스처 부분 출력시) 부분 텍스처의 x 좌표 (단위: 상대값, 0~1)
sourceY	(텍스처 부분 출력시) 부분 텍스처의 y 좌표 (단위: 상대값, 0~1)
sourceWidth	(텍스처 부분 출력시) 부분 텍스처의 가로 길이 (단위: 상대값, 0~1)
sourceHeight	(텍스처 부분 출력시) 부분 텍스처의 세로 길이 (단위: 상대값, 0~1)
cx	회전 중심 x 좌표 (기준: 화면의 왼쪽 위)
cy	회전 중심 y 좌표 (기준: 화면의 왼쪽 위)
rotation	회전 각도 (단위: 도(deg))

참고: 함수 이름 마지막의 r는 'Rotation'의 약식 표기입니다.

- 사운드 출력 함수 (sound)

- Level B
  - sound_fx_play *[사용되지 않음] 사운드를 재생*
  - sound_play *사운드를 재생*
  - sound_pause *사운드를 일시정지*
  - sound_stop *사운드를 정지*
  - sound_is_playing *사운드가 재생중인지 여부를 반환*

사운드 출력과 관련된 함수입니다. 이 분류의 함수들은 공통적으로 함수 이름이 **'sound'**로 시작됩니다.

✓ **void** sound_fx_play (SOUND sound); **[사용되지 않음]** [LEVEL: **B**]

요약: 사운드를 재생합니다.

상세: **이 함수는 사용되지 않습니다.** sound_play를 사용하세요.

설명:

인수 이름	전달할 값
sound	재생할 사운드

✓ **void** sound_play (SOUND sound, int rewind, int loopEnabled); [LEVEL: **B**]

요약: 사운드를 재생합니다.

상세: 처음으로 사운드를 재생할 때에는 자동으로 사운드의 첫 부분부터 재생됩니다. 만약 rewind를 false로 하고 이 함수를 호출할 경우, 마지막으로 재생되었던 위치에서부터 다시 재생되기 시작합니다.

만약 이전에 sound_stop으로 사운드를 정지했다면, rewind와 관계 없이 처음부터 재생됩니다.

설명:

인수 이름	전달할 값
sound	재생할 사운드
rewind	처음부터 다시 재생할지 여부 (0: false, 1: true)
loopEnabled	자동 반복 재생 여부 (0: false, 1: true)

✓ **void** sound_pause (SOUND sound); [LEVEL: **B**]

요약: 사운드을 정지합니다.

상세: 사운드가 재생 중이었다면, 사운드 재생을 일시정지시킵니다.

설명:

인수 이름	전달할 값
sound	일시정지할 사운드

✓ **void** sound_stop (SOUND sound); [LEVEL: **B**]

요약: 사운드을 정지합니다.

상세: 사운드가 재생 중이었다면, 사운드 재생을 멈추고, 다음 재생할 위치를 가장 처음으로 돌려놓습니다.

설명:

인수 이름	전달할 값
sound	정지할 사운드

✓ **int** sound_is_playing (SOUND sound); [LEVEL: **B**]

요약: 사운드가 재생 중인지 여부를 반환합니다.

설명:

인수 이름	전달할 값
sound	재생 여부를 조사할 사운드

return: 재생 중이면 1, 아니면 0.

● 컨텐트 관리 함수 (content)

- Level B
  - content_load_font *폰트를 불러오기*
  - content_load_texture *텍스처를 불러오기*
  - content_load_sound *사운드를 불러오기*

폰트, 텍스처, 사운드와 같은 게임 내 컨텐트를 관리하는 함수입니다. 이 분류의 함수들은 공통적으로 함수 이름이 **'content'**로 시작됩니다.

✓ **FONT** content_load_font (char* name, float weight, float pxSize); [LEVEL: **B**]

요약: 폰트를 생성합니다.

상세: 폰트 파일을 불러와 폰트를 생성합니다. 인수 name에는 **폰트 이름**이 들어와야 합니다. **폰트 파일의 이름이 아님**에 주의하십시오. 개발 중에 **(메인 프로젝트 경로)₩Content₩Sound** 폴더에 파일을 넣어두면 자동으로 아웃풋 폴더(Debug 혹은 Release)에 복사됩니다.

이 함수는 폰트를 먼저 **Content₩Font** 폴더에서 검색하며, 해당 폰트가 없으면 System 폰트에서 검색하여 로드합니다. 만약 폰트를 찾지 못한 경우, 윈도우 시스템에 설정된

기본 폰트로 표시됩니다.

폰트 굵기(weight)는 0과 1 사이의 값이어야 합니다. 일반적인 폰트의 굵기는 0.3f의 값을 갖습니다.

반환하는 것은 폰트 ID로서, 차후 이 폰트를 이용할 때 여기서 반환된 ID를 사용해야 합니다. (draw_text() 참조)

설명:

인수 이름	전달할 값
name	폰트 이름
weight	폰트 굵기 (0 ~ 1)
pxSize	폰트 크기 (단위: 픽셀)

return: 생성된 폰트의 ID

✓ **TEXTURE** content_load_texture (char* name); [LEVEL: **B**]

요약: 텍스처를 생성합니다.

상세: 외부 그림 파일을 불러와 텍스처를 생성합니다. 텍스처 이름(name)으로는 **확장명이 포함된 파일 이름**을 적어야 하며, 자동으로 **Content₩Texture** 폴더에서 파일을 찾습니다. 개발 중에 **(메인 프로젝트 경로)₩Content₩Texture** 폴더에 파일을 넣어두면 자동으로 아웃풋 폴더(Debug 혹은 Release)에 복사됩니다.

반환하는 것은 생성된 텍스처의 ID입니다. draw_texture() 등의 함수를 사용할 때 이 ID를 이용하여 텍스처를 출력할 수 있습니다.

(투명색이 포함된)PNG, JPG, GIF등, 윈도우가 지원하는 거의 모든 이미지 포맷을 지원합니다.

설명:

인수 이름	전달할 값
name	텍스처 이름 (확장명이 포함된 파일 이름)

return: 생성된 텍스처의 ID

✓ **SOUND** content_load_sound (char* name); [LEVEL: **B**]

요약: 사운드를 생성합니다.

상세: 외부 사운드 파일을 불러와 사운드를 생성합니다. 사운드 이름(name)으로는 **확장명이 포함된 파일 이름**을 적어야 하며, 자동으로 **Content₩Sound** 폴더에서 파일을 찾습니다. 개발 중에 **(메인 프로젝트 경로)₩Content₩Sound** 폴더에 파일을 넣어두면 자동으로 아웃풋 폴더(Debug 혹은 Release)에 복사됩니다.

반환하는 것은 생성된 사운드의 ID입니다. sound_fx_play() 등의 함수를 사용할 때 이 ID를 이용하여 사운드를 재생할 수 있습니다.

WAV, OGG 포맷을 지원합니다.

설명:

인수 이름	전달할 값
name	사운드 이름 (확장명이 포함된 파일 이름)

return: 생성된 사운드의 ID

- 키보드 입력 관련 함수 (keybd)

- 공통
  - keybd_is_key_down *키가 눌려 있는지 여부를 반환*
  - keybd_is_key_pressed *키가 눌린 상태로 바뀌었는지 여부를 반환*
  - keybd_is_key_up *키가 떼져 있는지 여부를 반환*
  - keybd_is_key_released *키가 떼진 상태로 바뀌었는지 여부를 반환*

키보드의 입력을 확인하는데 사용하는 함수입니다. 이 분류의 함수들은 공통적으로 함수 이름이 '**keybd**'로 시작됩니다.

✓ **int** keybd_is_key_down(KEYS keycode)

요약: keycode에 해당하는 키가 눌린 상태인지 확인합니다.

설명:

인수 이름	전달할 값
keycode	확인할 키

return: 눌린 상태이면 1, 아니면 0

✓ **int** keybd_is_key_pressed(KEYS keycode)

요약: keycode에 해당하는 키가 방금 눌렸는지 확인합니다.

설명:

인수 이름	전달할 값
keycode	확인할 키

return: 방금 눌렸으면 1, 아니면 0

✓ **int** keybd_is_key_up(KEYS keycode)

요약: keycode에 해당하는 키가 눌리지 않은 상태인지 확인합니다.

설명:

인수 이름	전달할 값
keycode	확인할 키

return: 눌리지 않은 상태이면 1, 아니면 0

✓ **int** keybd_is_key_released(KEYS keycode)

요약: keycode에 해당하는 키가 방금 떼졌는지 확인합니다.

설명:

인수 이름	전달할 값
keycode	확인할 키

return: 방금 떼졌으면 1, 아니면 0

- 마우스 입력 관련 함수 (mouse)

- 공통
  - mouse_is_btn_down *버튼이 눌려 있는지 여부를 반환*
  - mouse_is_btn_pressed *버튼이 눌린 상태로 바뀌었는지 여부를 반환*
  - mouse_is_btn_up *버튼이 떼져 있는지 여부를 반환*
  - mouse_is_btn_released *버튼이 떼진 상태로 바뀌었는지 여부를 반환*
  - mouse_get_x *마우스 포인터의 x 좌표를 반환 (단위: 칸)*
  - mouse_get_y *마우스 포인터의 y 좌표를 반환 (단위: 칸)*

마우스의 입력을 확인하는데 사용하는 함수입니다. 이 분류의 함수들은 공통적으로 함수 이름이 '**mouse**'로 시작됩니다.

✓ **int** mouse_is_btn_down(MOUSEBUTTON button)

요약: button에 해당하는 버튼이 눌린 상태인지 확인합니다.

설명:

인수 이름	전달할 값
button	확인할 버튼

return: 눌린 상태이면 1, 아니면 0

✓ **int** mouse_is_btn_pressed(MOUSEBUTTON button)

요약: button에 해당하는 버튼이 방금 눌렸는지 확인합니다.

설명:

인수 이름	전달할 값
button	확인할 버튼

return: 방금 눌렸으면 1, 아니면 0

✓ **int** mouse_is_btn_up(MOUSEBUTTON button)

요약: button에 해당하는 버튼이 눌리지 않은 상태인지 확인합니다.

설명:

인수 이름	전달할 값
button	확인할 버튼

return: 눌리지 않은 상태이면 1, 아니면 0

✓ **int** mouse_is_btn_released(MOUSEBUTTON button)

요약: button에 해당하는 버튼이 방금 떼졌는지 확인합니다.

설명:

인수 이름	전달할 값
button	확인할 버튼

return: 방금 떼졌으면 1, 아니면 0

✓ **int** mouse_get_x()

요약: 마우스의 x 좌표를 칸 단위로 가져옵니다.

✓ **int** mouse_get_y()

요약: 마우스의 y 좌표를 칸 단위로 가져옵니다.

- 게임 시스템 관련 함수 (game)

- 공통
  - game_exit — *게임 프로그램을 종료*
  - game_is_playing — *게임 프로그램이 실행 중인지 여부를 반환*
  - game_set_window_title — *게임 프로그램의 제목을 설정*
  - game_set_cell_size — *한 칸의 크기를 설정*
  - game_set_cell_count — *화면의 가로/세로 칸 개수를 설정*
  - game_get_cell_x_count — *화면의 가로 칸의 개수를 반환*
  - game_get_cell_y_count — *화면의 세로 칸의 개수를 반환*
  - game_get_cell_width — *칸의 가로 길이를 반환*
  - game_get_cell_height — *칸의 세로 길이를 반환*

게임 프로그램에 관련된 함수입니다. 이 분류의 함수들은 공통적으로 함수 이름이 **'game'**로 시작됩니다.

✓ **void** game_exit()

요약: 게임 프로그램을 즉시 종료합니다.

✓ **int** game_is_playing()

요약: 게임 프로그램이 현재 실행 중인지 반환합니다.

설명:

return: 게임 프로그램이 실행 중이면 1, 아니면 0

✓ **void** game_set_window_title(char* title)

요약: 게임 프로그램의 제목을 title로 설정합니다. 초기화(initialize) 함수 내에서 사용할 것을 권장합니다.

설명:

인수 이름	전달할 값
title	게임 프로그램의 제목

✓ **void** game_set_cell_size(int width, int height)

요약: 한 칸의 크기를 설정합니다. 초기화(initialize) 함수 내에서 사용할 것을 권장합니다.

설명:

인수 이름	전달할 값
width	한 칸의 너비 (픽셀 단위)
height	한 칸의 높이 (픽셀 단위)

✓ **void** game_set_cell_count(int x, int y)

요약: 화면의 x/y 방향 칸 개수를 설정합니다. 초기화(initialize) 함수 내에서 사용할 것을 권장합니다.

설명:

인수 이름	전달할 값
x	x 방향의 화면 칸 개수
y	y 방향의 화면 칸 개수

✓ **int** game_get_cell_x_count()

요약: 화면의 가로 칸의 개수를 반환합니다.

설명:

return: 화면의 가로 칸의 개수

✓ **int** game_get_cell_y_count()

요약: 화면의 세로 칸의 개수를 반환합니다.

설명:

return: 화면의 세로 칸의 개수

✓ **int** game_get_cell_width()

요약: 한 칸의 폭을 반환합니다.

설명:

return: 한 칸의 폭 (단위: 픽셀)

✓ **int** game_get_cell_height()

요약: 한 칸의 높이를 반환합니다.

설명:

return: 한 칸의 높이 (단위: 픽셀)

● 엔진 구동 관련 함수 (engine)

- 공통
  - engine_set_init_func — *초기화(initialize) 콜백 함수를 설정*
  - engine_set_draw_func — *그리기(draw) 콜백 함수를 설정*
  - engine_set_update_func — *업데이트(update) 콜백 함수를 설정*
  - engine_set_hwnd — *화면을 출력할 윈도우의 핸들을 설정*
  - engine_run — *엔진 메인 루프를 시작*

JOKEBOX 엔진을 구동시킬 때 사용되는 함수들입니다. Framework를 이용해 개발 중이라면 Framework 내부적으로 이 함수들을 호출하여 엔진을 구동시키므로, 별도로 호출할 필요가 없습니다.

✓ **void** engine_set_init_func(void (*init_func)())

요약: 엔진에서 자동으로 호출할 초기화(initialize) 함수를 등록합니다.

설명:

인수 이름	전달할 값
init_func	등록할 초기화(initialize) 함수

✓ **void** engine_set_draw_func(void (*draw_func)())

요약: 엔진에서 자동으로 호출할 그리기(draw) 함수를 등록합니다.

설명:

인수 이름	전달할 값
draw_func	등록할 그리기(draw) 함수

✓ **void** engine_set_update_func(void (*update_func)())

요약: 엔진에서 자동으로 호출할 업데이트(update) 함수를 등록합니다.

설명:

인수 이름	전달할 값
update_func	등록할 업데이트(update) 함수

✓ **void** engine_set_hwnd(HWND hwnd)

요약: 게임 화면을 출력할 창의 핸들을 지정합니다.

설명:

인수 이름	전달할 값
hwnd	창의 핸들

✓ **void** engine_run()

요약: 최종적으로 엔진 구동에 필요한 초기화를 수행한 후 메인 루프를 시작하여 엔진을 구동합니다.

- 보조 함수

게임에 직접적인 영향은 끼치지 않으나, 원활한 코딩에 도움을 줄 수 있는 함수들입니다.

✓ **COLOR** RGB (r, g, b)

COLOR RGBA (r, g, b, a)

요약: 입력된 Red, Green, Blue(, Alpha) 색상 값을 갖는 COLOR를 반환하는 함수입니다.

상세: 매크로(#define) 함수로서, r, g, b(, a) 값은 각각 1Byte를 갖는 0~255 사이의 숫자입니다.

설명:

인수 이름	전달할 값
r	적색 성분 (0~255)
g	녹색 성분 (0~255)
b	청색 성분 (0~255)
a	투명색 성분 (0~255)

return: (r, g, b, a) 색상 정보를 갖는 COLOR