



# jokebox.

Helper, v1.01

## 레퍼런스

1. JokeboxHelper 정의 타입
2. JokeboxHelper 정의 함수

## JOKEBOX 정의 타입

- 구조체
  - RECTANGLE *사각형 정보를 저장하는 구조체*
  - VECTOR2 *2차원 벡터 정보를 저장하는 구조체*

- 구조체(Struct)

- ✓ RECTANGLE

**요약:** 사각형 정보를 저장하는 구조체입니다.

**정의:**

```
typedef struct Rectangle
{
    float x;
    float y;
    float width;
    float height;
}RECTANGLE;
```

**설명:**

필드 이름	값
x	사각형 왼쪽-위 모서리의 x 좌표
y	사각형 왼쪽-위 모서리의 y 좌표
width	사각형의 너비
height	사각형의 높이

- ✓ VECTOR2

**요약:** 2차원 벡터 정보를 저장하는 구조체입니다.

**정의:**

```
typedef struct Vector2
{
    float x;
    float y;
}VECTOR2;
```

**설명:**

필드 이름	값
x	2차원 벡터의 x 성분
y	2차원 벡터의 y 성분

## JOKEBOX 정의 함수

JokeboxHelper는 정의 타입들을 조작하는 기본적인 함수들을 제공합니다. 이들 함수를 이용하여 제공되는 타입을 손쉽게 이용할 수 있습니다.

- 사각형 조작 함수 (rect)

- |                   |                          |
|-------------------|--------------------------|
| - rect_create     | RECTANGLE 구조체를 생성        |
| - rect_contains   | 사각형이 특정 위치를 포함하는지 여부를 반환 |
| - rect_intersects | 사각형들이 서로 교차하는지 여부를 반환    |

- ✓ **RECTANGLE** rect\_create (float x, float y, float width, float height);

**요약:** 입력된 값들을 갖는 RECTANGLE 구조체를 만듭니다.

**설명:**

인수 이름	전달할 값
x	사각형 왼쪽-위 모서리의 x 좌표
y	사각형 왼쪽-위 모서리의 y 좌표
width	사각형의 너비
height	사각형의 높이

**return:** 해당 값을 갖는 RECTANGLE 구조체

**사용예:** 다음 예시 코드는 rect를 x = 3, y = 5, 너비 = 20, 높이 = 10를 갖는 RECTANGLE로 초기화하는 코드입니다.

```
RECTANGLE rect;  
rect = rect_create(3, 5, 20, 10);
```

- ✓ **int** rect\_contains (RECTANGLE rect, float x, float y);

**요약:** 사각형이 점 (x, y)를 포함하는지 여부를 반환합니다.

**설명:**

인수 이름	전달할 값
rect	점을 포함하고 있는지 확인할 사각형
x	점의 x 좌표
y	점의 y 좌표

**return:** 점을 포함하고 있으면 1, 아니면 0

- ✓ **int** rect\_intersects (RECTANGLE rect1, RECTANGLE rect2);

**요약:** 두 사각형이 서로 교차되는지 여부를 반환합니다.

설명:

인수 이름	전달할 값
rect1	교차 여부를 확인할 사각형 1
rect2	교차 여부를 확인할 사각형 2

**return:** 두 사각형이 교차하고 있으면 1, 아니면 0

**참고:** 한 사각형이 다른 한 사각형을 포함하는 경우는 교차하지 않는 것으로 판정합니다.

- 2차원 벡터 조작 함수 (vec2)

• C/C++	
- vec2_create	VECTOR2 구조체를 생성
- vec2_signflip	벡터 부호를 반전시킨 것을 반환
- vec2_add	두 벡터를 더한 것을 반환
- vec2_subtract	두 벡터를 뺀 것을 반환
- vec2_multiply	벡터를 상수배한 것을 반환
- vec2_dot	두 벡터를 내적인 것을 반환
- vec2_normalize	벡터를 정규화(Normalize)한 것을 반환
- vec2_length	벡터의 길이를 반환
• C++	
- -(VECTOR2)	벡터 부호를 반전
- (VECTOR2) + (VECTOR2)	두 벡터를 더하기
- (VECTOR2) - (VECTOR2)	두 벡터를 빼기
- (VECTOR2) * (float)	벡터를 상수배하기
[ (VECTOR2) * (float) / (float) * (VECTOR2) ]	

✓ **VECTOR2** vec2\_create (float x, float y);

[C/C++]

**요약:** 입력된 값들을 갖는 VECTOR2 구조체를 생성합니다.

설명:

인수 이름	전달할 값
x	벡터의 x 성분
y	벡터의 y 성분

**return:** 해당 값을 갖는 VECTOR2 구조체

**사용예:** 다음 예시 코드는 vec를 x = 3, y = 5를 갖는 VECTOR2로 초기화하는 코드입니다.

```
VECTOR2 vec;  
vec = vec2_create(3, 5);
```

✓ <오버라이드된 연산자> -(VECTOR2)

[C++]

**요약:** 지정된 벡터의 부호를 반전시킨 것을 반환합니다.

**return:** 대상 벡터의 부호를 반전시킨 결과

**참고:** 이 함수는 연산 대상이 되는 벡터(들)의 값을 바꾸지 않습니다.

**사용예:** 다음 예시 코드는 vec=(1, 2)를 반전시킨 결과를 vec2에 저장하는 코드입니다.

```
VECTOR2 vec, vec2;  
vec = vec2_create(1, 2);  
vec2 = -vec;    // vec2 = (-1, -2)
```

✓ <오버라이드된 연산자> (VECTOR2) + (VECTOR2)

[C++]

**요약:** 두 벡터를 더한 결과를 반환합니다.

**return:** 두 벡터를 더한 결과

**참고:** 이 함수는 연산 대상이 되는 벡터(들)의 값을 바꾸지 않습니다.

**사용예:** 다음 예시 코드는 vec=(1, 2)와 vec2=(3, 4)를 더한 결과를 vec2에 저장하는 코드입니다.

```
VECTOR2 vec, vec2;  
vec = vec2_create(1, 2);  
vec2 = vec2_create(3, 4);  
vec2 = vec + vec2;    // vec2 = (4, 6)
```

✓ <오버라이드된 연산자> (VECTOR2) - (VECTOR2)

[C++]

**요약:** 두 벡터를 뺀 결과를 반환합니다.

**return:** 대상 벡터 1에서 대상 벡터 2를 뺀 VECTOR2

**참고:** 이 함수는 연산 대상이 되는 벡터(들)의 값을 바꾸지 않습니다.

✓ <오버라이드된 연산자> (VECTOR2) \* (float)

<오버라이드된 연산자> (float) \* (VECTOR2)

[C++]

**요약:** 지정된 벡터를 상수배 한 결과를 반환합니다.

**return:** 대상 벡터에 상수배 한 결과

**참고:** 이 함수는 연산 대상이 되는 벡터(들)의 값을 바꾸지 않습니다.

**사용예:** 다음 예시 코드는 vec=(1, 2)를 2배한 결과를 다시 vec에 저장하는 코드입니다.

```
VECTOR2 vec;  
vec = vec2_create(1, 2);  
vec = vec * 2;    // vec = (2, 4)
```

따라 이용할 수 있습니다. (원작자 **이광무, OrangeYellow**)