

NGN_MATH.H

Clase Vector2

Esta clase incluye todos los métodos y propiedades necesarios para el uso básico de vectores en 2 dimensiones. Hay 3 variaciones de la clase, Vector2, Vector2I32 y Vector2I64, con sus respectivas sobrecargas. Todas las clases tienen soporte para las operaciones de suma y resta entre vectores del mismo tipo, así como soporte para las operaciones de multiplicación y división entre un vector y un escalar, además de las operaciones lógicas de igualdad y diferencia.

```
float Vector2.Magnitude();  
int32_t Vector2I32.Magnitude();  
int64_t Vector2I64.Magnitude();
```

Devuelve la magnitud del vector. En el caso de vectores de números enteros, devuelve el resultado más aproximado posible.

```
Vector2 my_vector = {10.0f, 25.0f};  
float d = my_vector.Magnitude();
```

```
void Vector2.Normalize();
```

Este método solo está disponible en los vectores de punto flotante. Normaliza y actualiza el contenido del vector (convierte el contenido de las coordenadas X e Y a valores entre -1.0f y 1.0f).

```
Vector2 my_vector = {10.0f, 25.0f};  
my_vector.Normalize();
```

```
Vector2.x;  
Vector2.y;
```

Propiedades principales del vector. Almacenan las coordenadas X e Y. Estas propiedades son del tipo del vector declarado (float, int32_t o int64_t).

```
Vector2 my_vector;  
my_vector.x = 13.5f;  
my_vector.y = 27.34f;
```

Clase NGN_Math

Esta clase incluye funciones matemáticas básicas para ayudar al desarrollo de cualquier videojuego, como el cálculo de ángulos o distancias.

```
float GetDistance(  
    float x1,  
    float y1,  
    float x2,  
    float y2  
);
```

```
float GetDistance(Vector2 a, Vector2 b);
```

```
uint32_t GetDistance(  
    int32_t x1,  
    int32_t y1,  
    int32_t x2,  
    int32_t y2  
);
```

```
uint32_t GetDistance(Vector2I32 a, Vector2I32 b);
```

Devuelve la distancia entre 2 puntos (4 sobrecargas). En el caso de usar números enteros, se devuelve el valor más aproximado redondeado.

```
float d = ngn->math->GetDistance(100.0f, 100.0f, 200.0f, 300.0f);
```

```
float GetAngle(  
    float x1,  
    float y1,  
    float x2,  
    float y2  
);
```

```
float GetAngle(Vector2 a, Vector2 b);
```

Devuelve el ángulo que forman 2 puntos (2 sobrecargas) respecto la horizontal. El primer parámetro determina el vértice del ángulo y el segundo parámetro la apertura de ese ángulo. El resultado es devuelto en **RADIANES**.

```
Vector2 origin = {10.0f, 10.0f};  
Vector2 destination = {100.0f, 15.0f};  
float a = ngn->math->GetAngle(origin, destination);
```

```
float GetManhattan(  
    float x1,  
    float y1,  
    float x2,  
    float y2  
);
```

```
float GetManhattan(Vector2 a, Vector2 b);
```

```
uint32_t GetManhattan(  
    int32_t x1,  
    int32_t y1,  
    int32_t x2,  
    int32_t y2  
);
```

```
uint32_t GetManhattan(Vector2I32 a, Vector2I32 b);
```

Devuelve la distancia Manhattan entre 2 puntos (4 sobrecargas). Esta distancia es la suma de las diferencias en valor absoluto de las diferentes coordenadas ($\text{abs}(x1 - x2) + \text{abs}(y1 - y2)$).

```
float d = ngn->math->GetGetManhattan(100.0f, 100.0f, 200.0f, 300.0f);
```