

NGN_COLLISIONS.H

```
uint32_t GetPixel(  
    NGN_CollisionMapData* cmap,  
    int32_t position_x,  
    int32_t position_y  
);
```

Devuelve el color del pixel del mapa dado en las coordenadas dadas. En caso que las coordenadas estén fuera del mapa, devolverá 0x00000000. El formato del color devuelto es 0xRRGGBBAA (RGBA)

```
uint32_t color = ngn->collisions->GetPixel(  
    collision_map,  
    wizard.sprite->position.x,  
    wizard.sprite->position.y  
);
```

```
Size2 GetMapSize(NGN_CollisionMapData* cmap);
```

Devuelve el tamaño del mapa dado en formato Size2 (width y height).

```
Size2 world_size = ngn->collisions->GetMapSize(collision_map);
```

```
bool HitBox(NGN_Sprite* spr1, NGN_Sprite* spr2);
```

Realiza una verificación de colisión por cajas entre 2 sprites. Devuelve TRUE en caso de colisión.

```
if (ngn->collisions->HitBox(player, coin)) {  
    hit = true;  
} else {  
    hit = false;  
}
```

```
bool PixelPerfect(NGN_Sprite* spr1, NGN_Sprite* spr2);
```

Realiza una verificación de colisión a nivel de pixel entre 2 sprites. Devuelve TRUE en caso de colisión.

```
if (ngn->collisions->PixelPerfect(player, coin)) {  
    hit = true;  
} else {  
    hit = false;  
}
```

Atención: La detección de colisiones a nivel de pixel consume una gran cantidad de recursos de la CPU. Usar solo en casos donde sea absolutamente imprescindible.

```
bool RaycastPoint(  
    NGN_Sprite* spr,  
    float position_x,  
    float position_y
```

```
);
```

```
bool RaycastPoint(NGN_Sprite* spr, Vector2 position);
```

Comprueba si hay un pixel visible del sprite proporcionado en las coordenadas dadas.

```
if (ngn->collisions->RaycastPoint(coin, 128, 96)) {  
    hit = true;  
} else {  
    hit = false;  
}
```