**AABB**

* Axis aligned bounding box

**Axis-Aligned Bounding Box (AABB) Explained**

An AABB (Axis-Aligned Bounding Box) is a fundamental concept in computer graphics and game development used to approximate the spatial extent of objects. It's called "axis-aligned" because the box's edges align with the coordinate axes (X, Y, and Z in 3D space), making it straightforward to work with. Below, we'll explain AABB with illustrations.

**1. 3D AABB Bounding Box**

In 3D space, an AABB is defined by two points: the minimum point (Min) and the maximum point (Max). These points represent the corners of the box, and the edges are always parallel to the coordinate axes (X, Y, and Z). The AABB encompasses the entire object, regardless of its orientation.

**2. 2D AABB Bounding Box**

In 2D space, the concept is the same. An AABB is defined by two points: the minimum point (Min) and the maximum point (Max), with edges aligned to the X and Y axes.

**Use Cases for AABB:**

* **Collision Detection**: AABBs are used for efficient collision detection between objects.
* **Bounding Volume Hierarchy**: Complex scenes use hierarchical bounding volumes like AABBs to optimize intersection tests.
* **Frustum Culling**: In computer graphics, AABBs are used to quickly remove objects outside the view frustum.

**3. AABB Representation in Code**

In code, an AABB can be represented as a structure or class containing two points: Min and Max. For example, in C++:

cppCopy code

struct AABB { glm::vec3 Min; glm::vec3 Max; };

**4. AABB vs. OBB (Oriented Bounding Box)**

While AABBs are aligned with coordinate axes, OBBs can be oriented arbitrarily in 3D space. OBBs provide a tighter fit around objects but are more complex to work with.

In summary, an AABB bounding box is a simple geometric shape used in computer graphics and game development for collision detection, optimization, and other purposes. Its key feature is its alignment with coordinate axes, making it easy to use in many scenarios.

* https://developer.mozilla.org/en-US/docs/Games/Techniques/3D\_collision\_detection