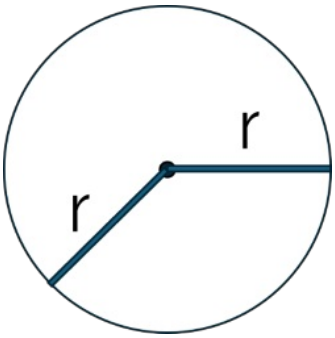
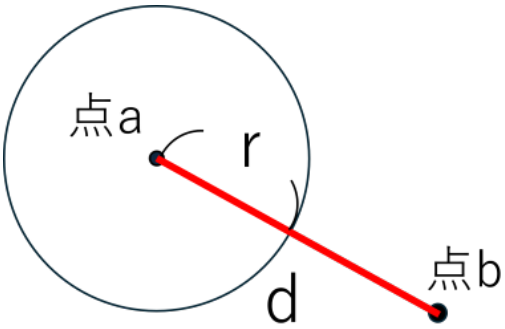
球と球の当たり判定(3D)（by チョコミント）

球と球の当たり判定

こんにちは、チョコミントです。 今回から当たり判定について軽く触れていこうかなと思います。 まずは比較的簡単な**球と球の当たり判定**についてです。 もう知っている方も多いと思いますが、よければ見ていってください。



球は**中点から円周の距離が全方向半径(r)の長さ**で構成されています。 では、球と点が衝突しているか確かめるにはどのように考えればいいのでしょうか。



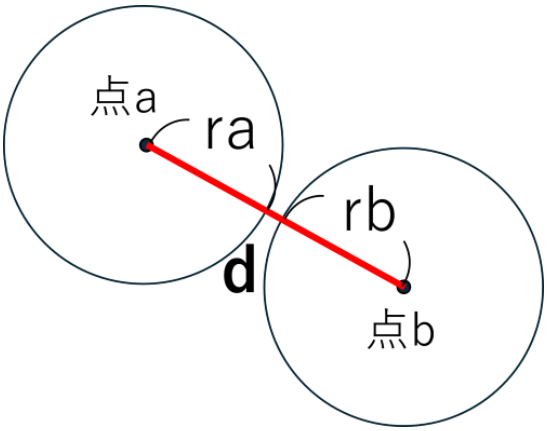
上の図では点a(球体に中点)から点bの距離をdとし、球体の半径をrと定義しています。 もうぴんっときた人もいるかもしません。 **dの距離がr(半径)以下だったら衝突**しているのです。 ってことは半径はもうわかっているので、**点aと点bの距離**が分かれば衝突しているかわかりそうですね！！

点と点の距離

点と点の距離は**三平方の定理**を利用して求めることができます。 3次元の場合はこのような感じです。

```
XMFLOAT3 c = { a.x - b.x,a.y - b.y,a.z - b.z };
return sqrtf((c.x * c.x) + (c.y * c.y) + (c.z * c.z));
```

簡単に説明するとX,Y,Z軸でそれぞれの始点(例:点a)から終点(例:点b)へのベクトルを求めて、 それらの合成ベクトルの長さが距離となるのです。 点と点の距離が分かってしまえばもう球と球の当たり判定は簡単です。



この図をみて理解した人も多いと思いますが、 **dの距離が(ra(半径) + rb(半径))以下だったら衝突**しているのです。

```
//点と点の距離を求める
XMFLOAT3 a, XMFLOAT3 b;
XMFLOAT3 c = { a.x - b.x,a.y - b.y,a.z - b.z };
float d = sqrtf((c.x * c.x) + (c.y * c.y) + (c.z * c.z));

float ra = 0.5f; //球体aの半径
float rb = 1.0f; //球体bの半径

//aとbの距離がそれぞれの半径を足した合計値以下なら当たっている
if (ra + rb >= d)
{
 //衝突している
}
```

サンプルプログラムです。

最後に

今回は球と球の当たり判定についての記事でした。 物足りない人も多かったのではないでしょうか。 次回は球と箱(AABB)の当たり判定について記事を書いてみます。

Revision #4
★Created 14 May 2025 03:45:48 by youe2
Updated 2 June 2026 17:30:23 by youe2