あいつは自分から見て右にいるの？左にいるの？

☒目的

自分の向いている方向から見て、「ターゲットが右にいるのか、左にいるのか」を判断し、その方向に回転させたい。

☒前提知識

1. `atan2(y, x)` の意味

ベクトル `(x, y)` の**角度（ラジアン）を、原点から見たときのx軸との角度**として返します。範囲は `[-π, π]`。

2. 基準ベクトル

自分が向いている方向をベクトル `forward` とし、右方向を基準にする場合は、通常 `right = {1, 0}` などとします（右向き単位ベクトル）。

☒方法1：`atan2`による角度の差で判断

```
// 自分の向きとターゲット方向
Vec2 forward = 自分の向きベクトル;     // 例: {1, 0} → 右向き
Vec2 toTarget = targetPos - selfPos;   // 自分からターゲットへのベクトル

// それぞれの角度を求める
float angleForward = atan2(forward.y, forward.x);
float angleToTarget = atan2(toTarget.y, toTarget.x);

// 差を求める（-π〜π の範囲に自動でなる）
float delta = angleToTarget - angleForward;

// 回転方向を判断
if (delta > 0) {
    // 左にターゲットがある（左回りに回転すべき）
} else if (delta < 0) {
    // 右にターゲットがある（右回りに回転すべき）
}
```

☒この方法のメリット：

角度を厳密に扱える。滑らかに回転アニメーションしたいときに使いやすい。

☒方法2：右向きベクトルとの **外積 or 内積** を使って右左判定

2Dにおける**外積（cross product）**で符号を利用する方法：

```
float cross = forward.x * toTarget.y - forward.y * toTarget.x;
```

- `cross > 0` → 左側
- `cross < 0` → 右側
- `cross == 0` → 同一直線上

または、**右向きベクトルと角度比較**

基準として右向きのベクトル `right = {1, 0}` を使って、ターゲットとの角度を見る：

```
Vec2 right = {1, 0};
float angleToTarget = atan2(toTarget.y, toTarget.x);
float angleRight = atan2(right.y, right.x); // = 0

// 結果的に angleToTarget だけ見れば良い
if (angleToTarget > 0) {
    // ターゲットは上（左回り）
} else {
    // ターゲットは下（右回り）
}
```

☒補足：`atan2` と外積の使い分け

方法	特徴	回転方向判定	回転量取得
`atan2` 差分	正確な角度取得。平滑回転に◎	◎	◎（角度そのもの）
外積（2D）	軽量。方向判定だけなら高速	◎（符号）	×（角度は出せない）

Revision #1
★Created 25 June 2025 06:43:45 by youe2
Updated 17 June 2026 04:10:34 by youe2