Basic AI:

Create an Enum:

enum AIState { Idle, Moving, Combat}

AIState myState = AIState.Idle;

Inside update:

Void Update()

{

Switch (myState)

{

Case AIState.Idle:

DoIdle();

Break;

Case AIState.Moving:

DoMoving();

Break;

Case AIState.Combat:

DoCombat();

Break;

}

// Then inside of Idle function

* Create if statements to check for when you would want to transition to another state.
  + myState = AIState.Moving;

// For Idle function, have the AI raycast towards the player, and check if the AI can see the player

* Once the AI sees the player, the AI changes state to chasing or combat, ect…

Find a player with AI

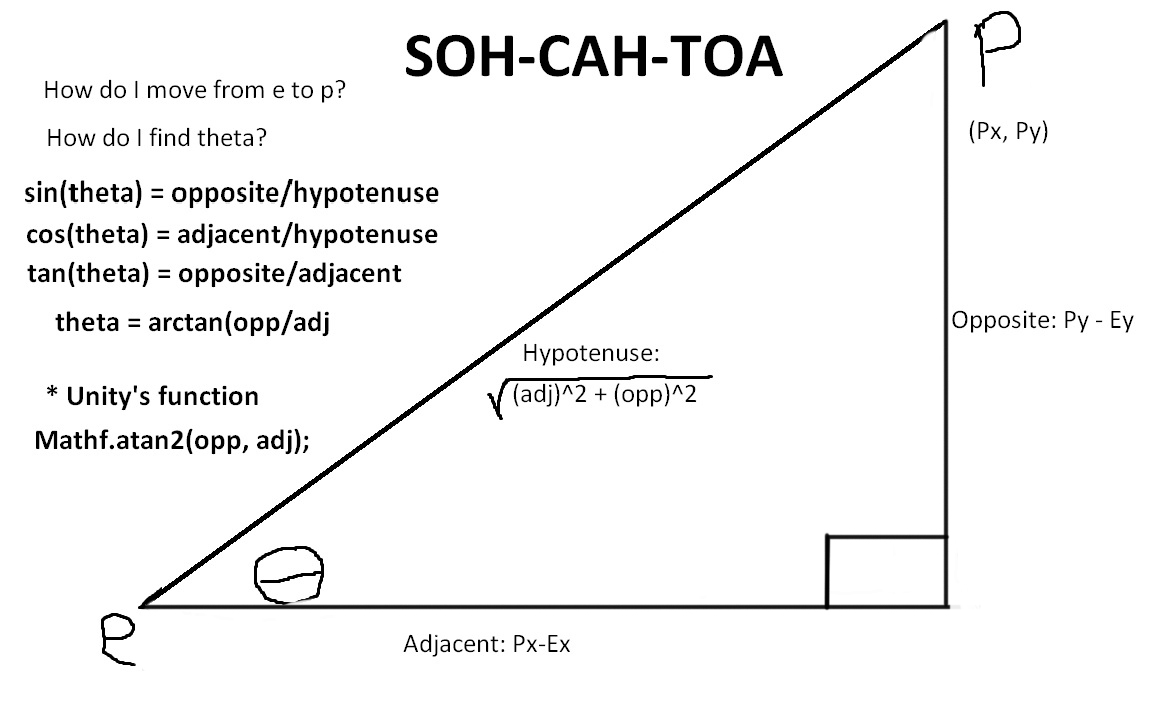
Vector3.Lookat() // doesn’t always work.

**Trigonometry:**

**SOH-CAH-TOA**

How do you move from E to P?

You need the angle to find direction (theta), and the distance



Find the angle using SOH\_CAH\_TOA. You know (Px, Py), and (Ex, Ey).

* tan(theta) = adjacent/hypotenuse
* theta = arctan (opposite/adjacent)
* Unity’s equivalent of the arctan is:
  + Mathf.Atan2(opp, adj);

Find the distance if given the speed, by creating a new Vector:

* Vector3 dist = new Vector3 (Px-Ex, Py-Ey, Pz-Ez);
* dist.Normalize() // shrinks it to 1
* Vector3 distance = dist \* speed;