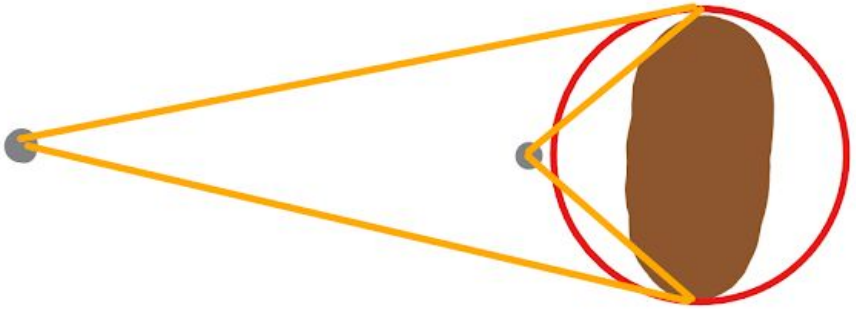
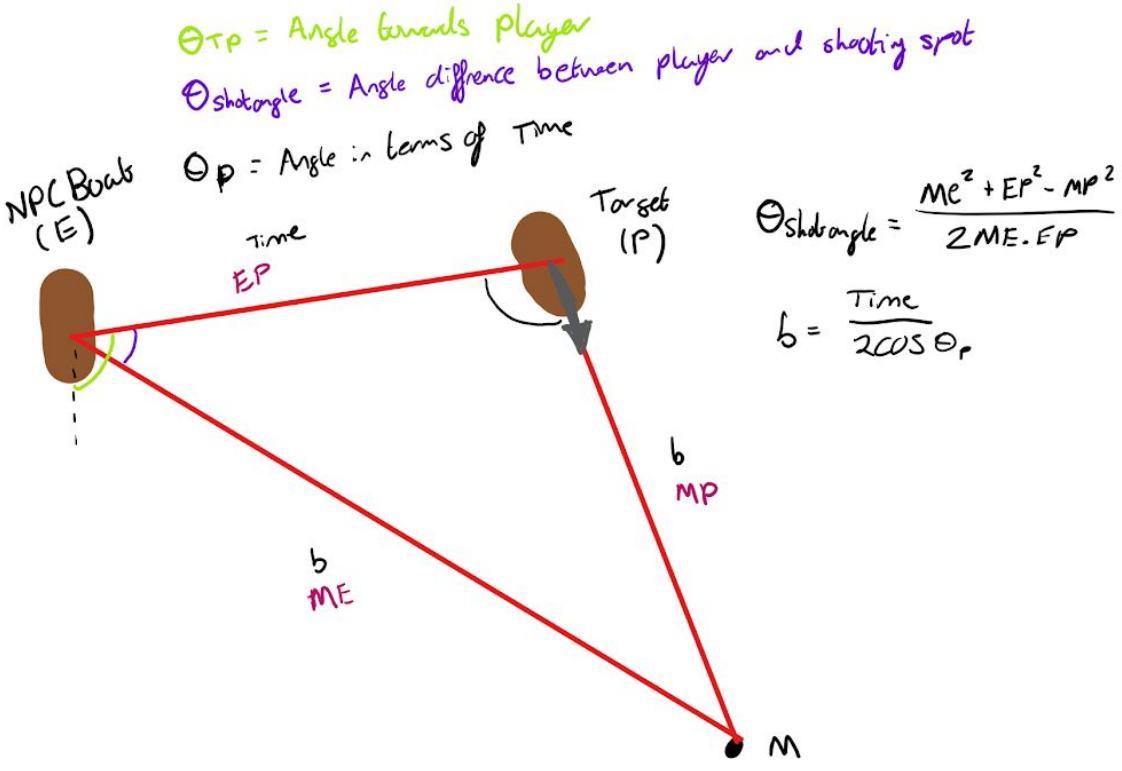
1.



Due to the distance of the boat from the projectiles perspective the angle that means that the projectile will hit changes drastically. This is why the function being described is used. First of all it finds out the radius to which it would hit. It then compares this with the actual radius of the boat and if smaller then is classed as a projectile to dodge.

2.

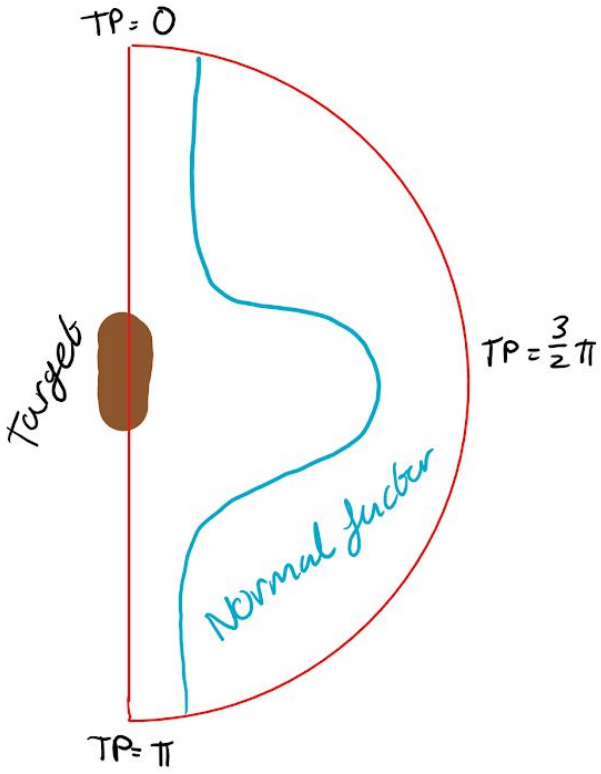


First of all this function is used to be able to find the collision point of two objects, in this case it is for the shot angle for the NPC (E) to fire and hit the Target (P). The point M is where the shot will collide with the Target. Due to it colliding with the Target we know that both ME and MP take the same time labelled as b and with simple math we know EP being just the distance between them divided by the projectile's speed that is equipped to that boat.

We first work out thetaP in terms of time which then gives us b which is the time taken for the shot. This function is timeForPerfectShotToCollide().

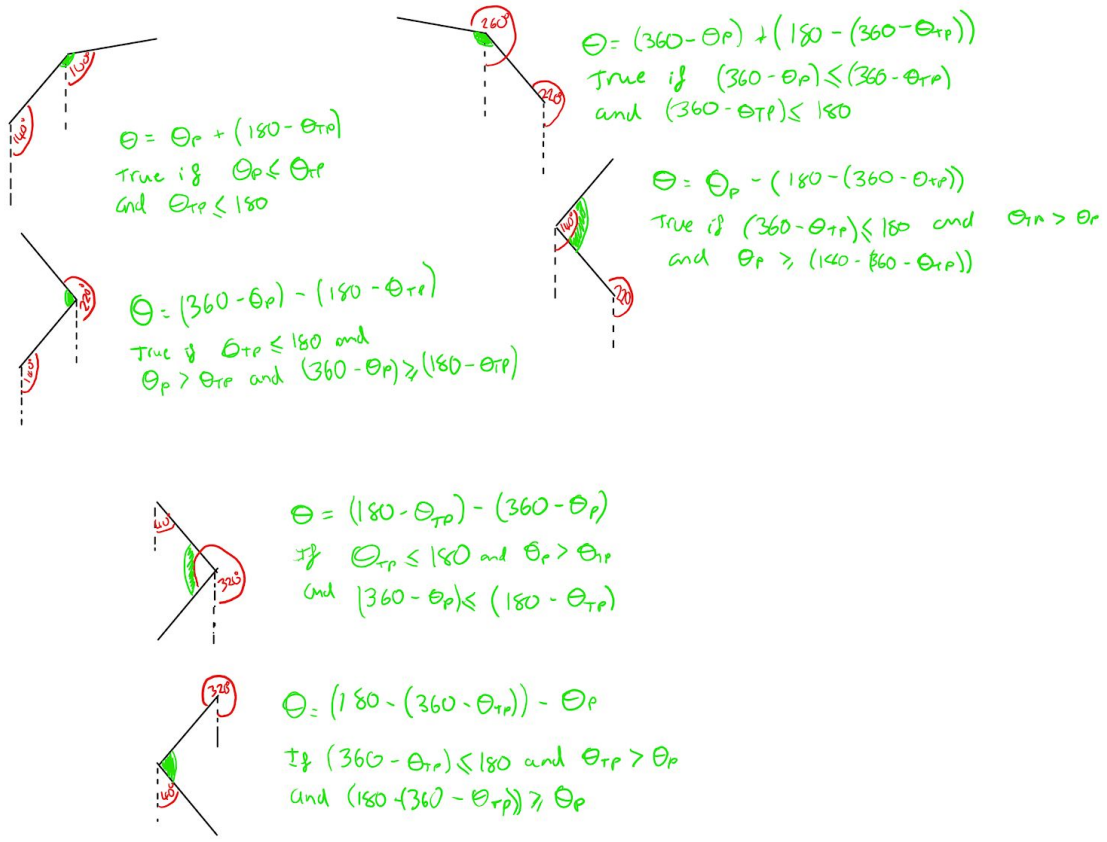
Knowing the stuff we do at the moment could give us an angle however due to the different speeds of the objects rather than being an isosceles in time in actual distances they could be varied. This means we convert them back into distances ME, EP and MP then work out shotAngle from there. Then we add shotAngle onto thetaTP which is the angle from the normal to the Target which then gives us the angle from the normal to M. This is where the NPC should fire for the perfect shot.

3.



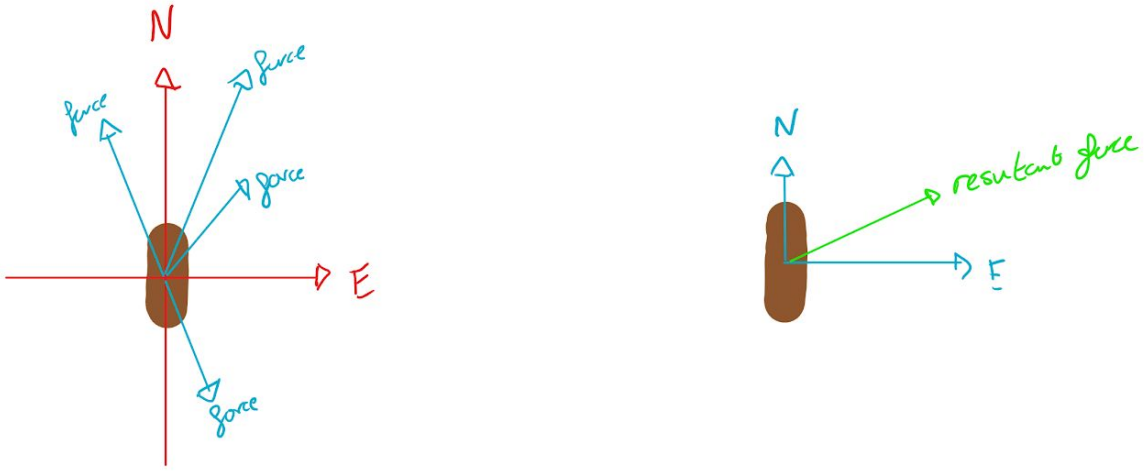
For the movement of the AI to make it behave the way intended and to get the AI to move at the right speed compared to the target this function was created. This is the same as the normal distribution in NPC Boat Behaviour 1. However takes the the normal distribution of either 3/2 pi or ½ pi depending on the side and puts the value in of TP which is angle of source to player. This is later on used like before to be able to rather than work out angle to work out speed of which the AI wants to go at.

4.



It takes 2 angles one being thetaP and the other thetaTP, these are labelled like that cause usually it is used to calculate the green angle where the green is where player is. Made so hideous with if statements due to edge cases and the way the angle system works.

5.



The way this function works is it is separated into two parts. First of all it goes through every force which when talking about here is both the force and the angle of which it acts like a vector. It then calculates how much of its force applies to E and to N which are the things which will be passed to the next function. Once all of the forces have been acted on N and E it moves to the next section.

We now class N and E being the only forces on the object. We then calculate the resultant force the same sort of way we did the N and E but this time its how much it affects the resultant.

The forces talked about here aren't usually actual forces applied to the boat but are usually the AI's way of knowing which way it wants to go towards e.g. how much the AI wants to go that direction.