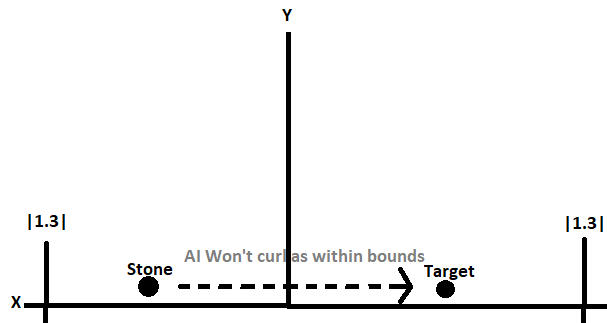
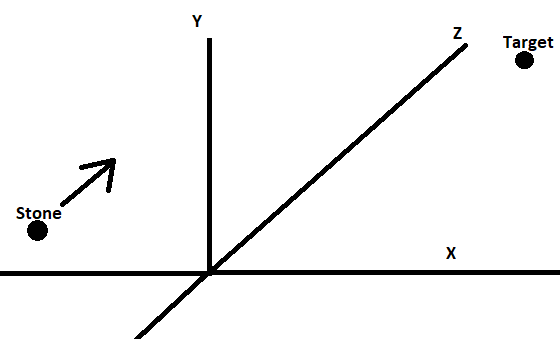
AI Implementation

AI Aiming

For the aiming, the AI works out the angle between the pivot (the gameobject that rotates the direction to be fired) and the target, as this value increases and decreases, an exponential function I trial and errored *(e^(((-x+0.001) /60 \* 1 - x \* 0.04f) where x <= 30 and x = angle)* increases the chance of the AI deciding to slide when it is closer to facing the direction of the target, and decreases as it moves further away, this allows for a reasonable amount of variation whilst maintaining fairness.

AI Directing

Starts by taking the XY coords of both the target and the stone, and works out the distance between them, this essentially allows me to work out the distance between the stones as they travel along the z axis, by ignoring it, to visualise it it’s like this:



When in reality, in the 3d plane it looks like this:  
I then work out which side of the target the stone is on, by comparing their x coordinates, this determines which direction the AI will curl in, as if it is on the left of the stone, it will curl right and towards it, if the stone is already within the bounds seen in the first image, then it will not curl, as it should already be on track to be close to / hit the target.

Sound Effects

**Miss sound effect -**  
<https://pixabay.com/sound-effects/aww-8277/>

**Hit sound effect** –   
<https://www.youtube.com/watch?v=2Cj9pXhIXus> – trimmed from this video

**Stone sliding sound –**

<https://www.youtube.com/watch?v=7cAnlZ1U1Mg>