AI Implementation

AI Aiming

For the aiming, the AI works out the angle between the pivot (the game object 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 \* 0.01f) where x <= 15/25/45 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 X 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 stone and the target as it travels along the z axis (y is not important as it never changes its y position). I next 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 on target, then it will not curl.

If there are several stones blocking the target, the chance of the AI sliding straight is reduced, and instead the chance of sliding between the angles of 6o & 20o is increased, to simulate some assessment of the game state.

Sound Effects

**Win round sound effect** –  
<https://pixabay.com/sound-effects/aww-8277/>

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

**Stone sliding sound –**

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