Dice probabilities in relation to geometry

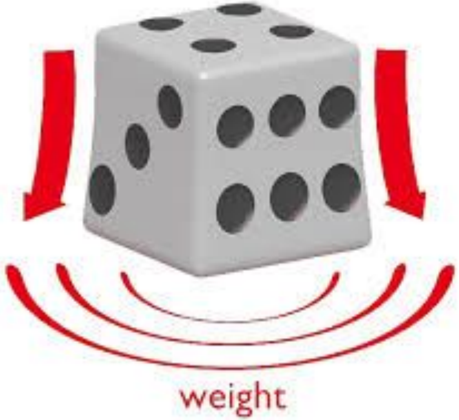
By Danny Schultz

Loaded dice:

- Dice with weights in them
- Provides uneven odds
- Often used for cheating
    - Provides users with an unfair advantage

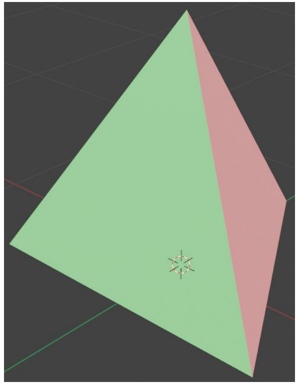
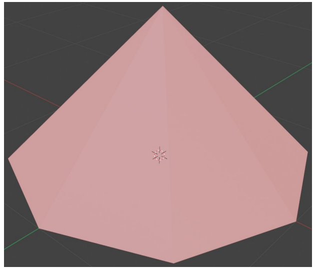
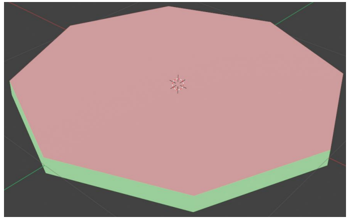
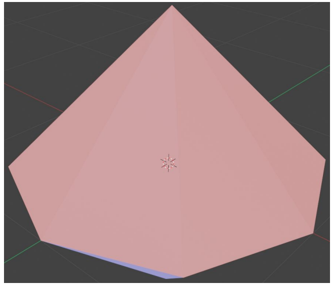
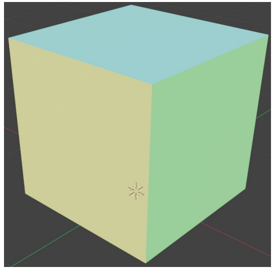
Geometry relationships

- Changing geometry of dice changes weight distribution
- Dice with uneven sides have uneven odds
- Is there a relationship between unevenness and probability?



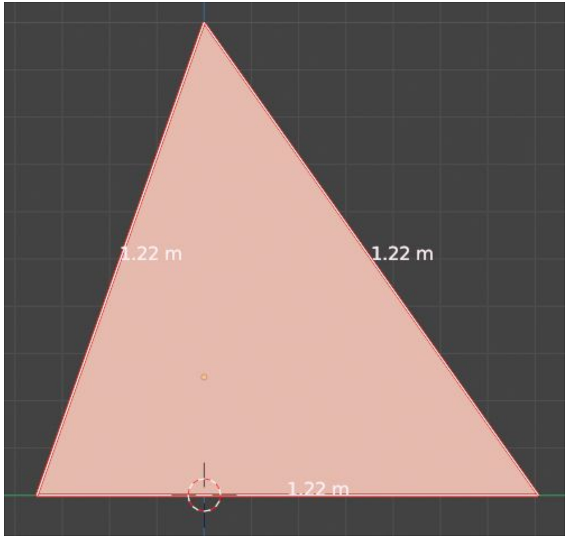
The experiment

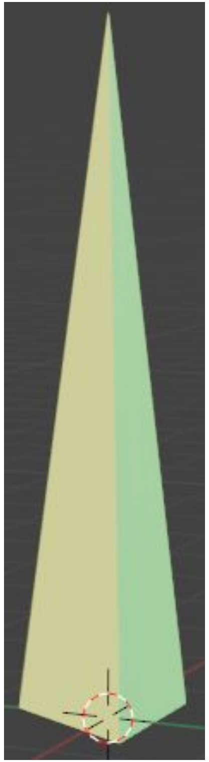
- 5 types of dice
    - Tetrahedron
    - Cone
    - Frustum
    - Bicone
    - Rectangular Prism
- 25 values of unevenness
    - 0.1-2.0
        - Increments of 0.1
    - 2.5-4.0
        - Increments of 0.5
    - 5.0
- 10,000 Simulated attempts per dice
- 1,250,000 total dice rolls

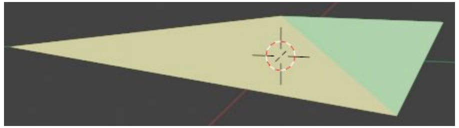






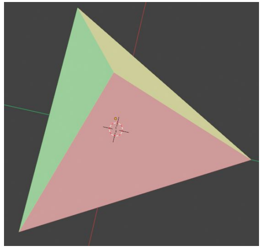
Tetrahedron dice

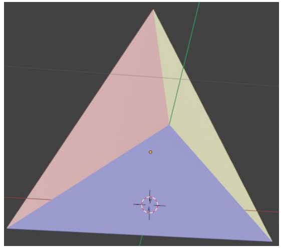
- Base height of 1:
  - Height is 1
  - All side lengths are the same
    - 1.22 each
- 4 Sides
  - Blue base (constant)
  - Red, green, and yellow sides (dynamic)



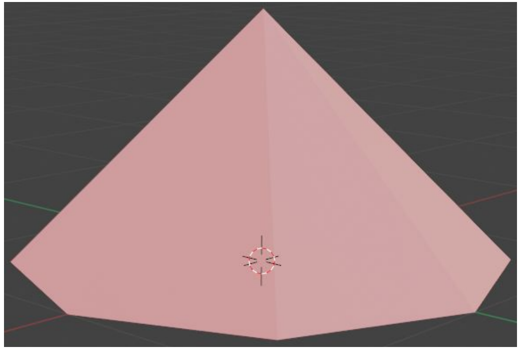




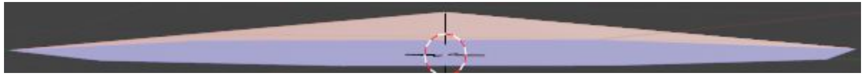


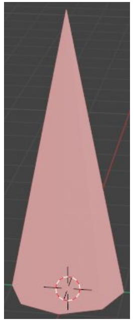


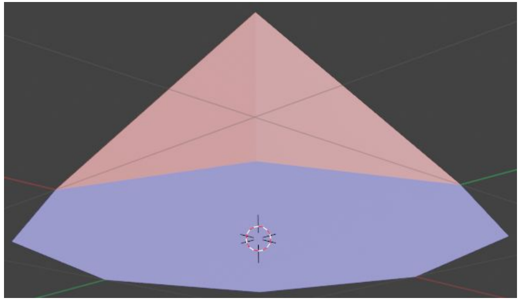
Cone dice



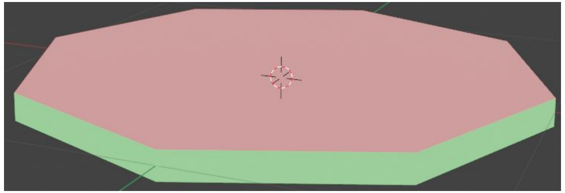
- Base height of 0
  - Would be a perfectly flat plane
  - Would not work with my simulation
  - Not used
    - If it was used it should yield results seen in a standard coin
  - Lowest height used is 0.1
- 2 sides
  - Blue base (constant)
  - Red top (dynamic)
- Other stats
  - Radius of the base is 1
  - Base of the cone is an octagon in place of a circle
    - Due to performance



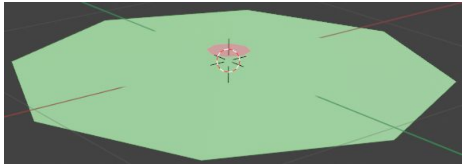


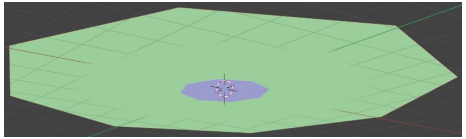


Frustum dice



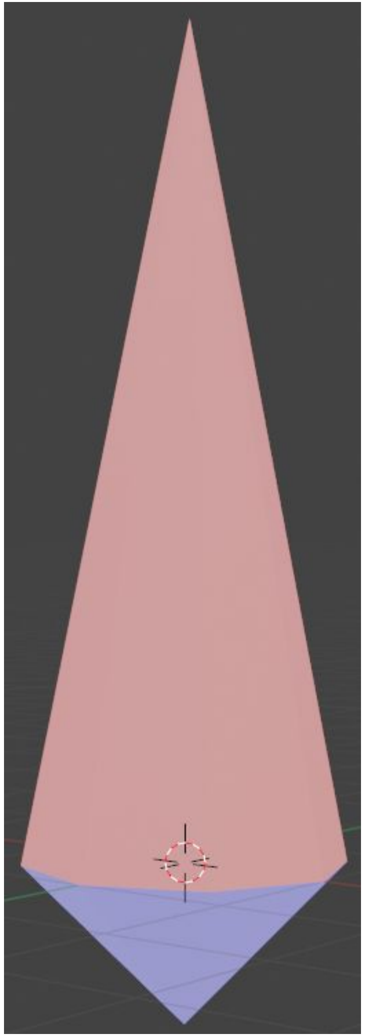
- What a frustum is
  - The bottom part of a cone
  - A frustum can be made by taking a cone and cutting the top of it off
- Base radius is 1
  - The blue face’s radius remains constant
  - The red face’s radius is what is being adjusted
  - When the red face’s radius is 1 the dice is technically a cylinder / and models an even coin
- 3 sides
  - Blue base (constant)
  - Red top (dynamic)
  - Green sides (dynamic)
- Other stats
  - Blue and red faces are octagons
  - Height is 0.1
  - The face the frustum “lands on” is the face that is down

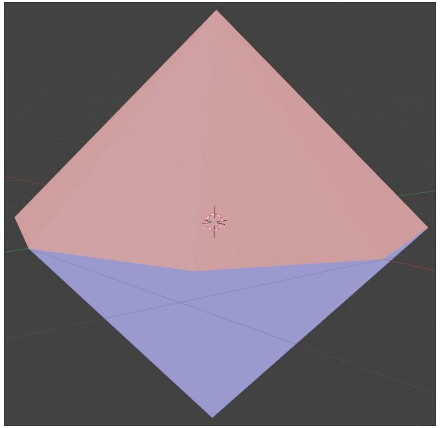


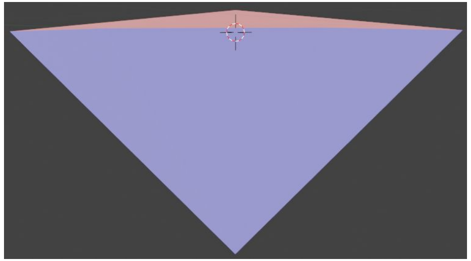


Bicone dice

- Base height of 1
  - Height of both red and blue cones are 1
- 2 sides
  - Blue cone (constant)
  - Red cone (dynamic)
- Other stats
  - Bicone's center ring is an octagon
  - Blue cone's height is always 1

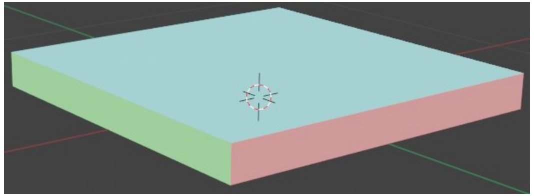


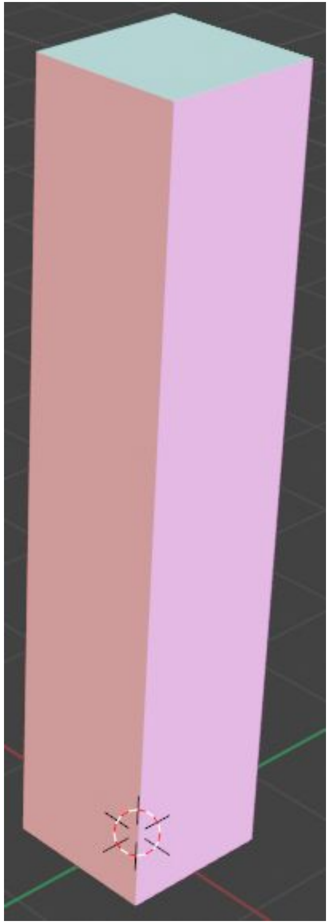


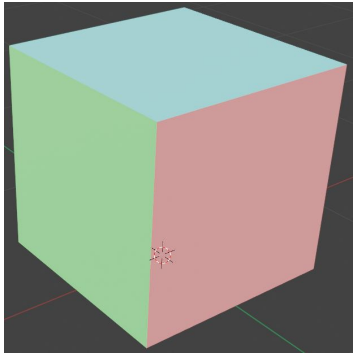


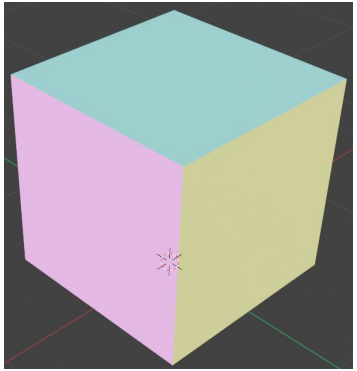
Cube dice

- Base height 1
  - Cube with all side lengths 1
- 6 sides
  - Blue base (constant)
  - Cyan top (constant)
  - Red, Green, Yellow, and Magenta sides (dynamic)



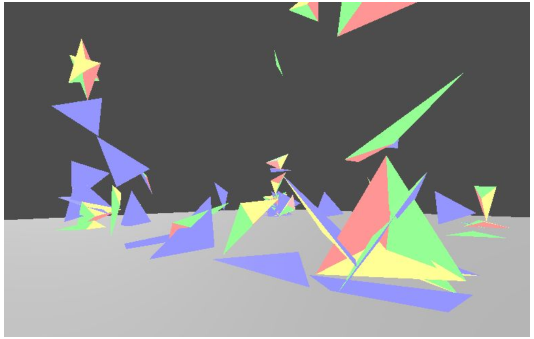


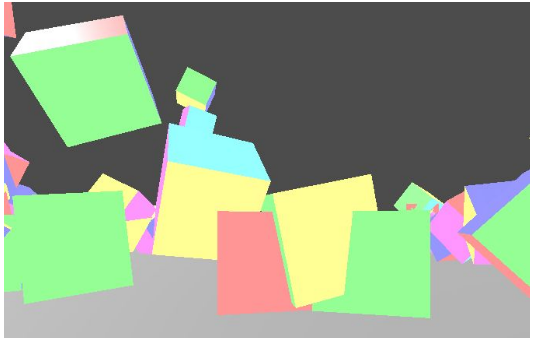




The simulation

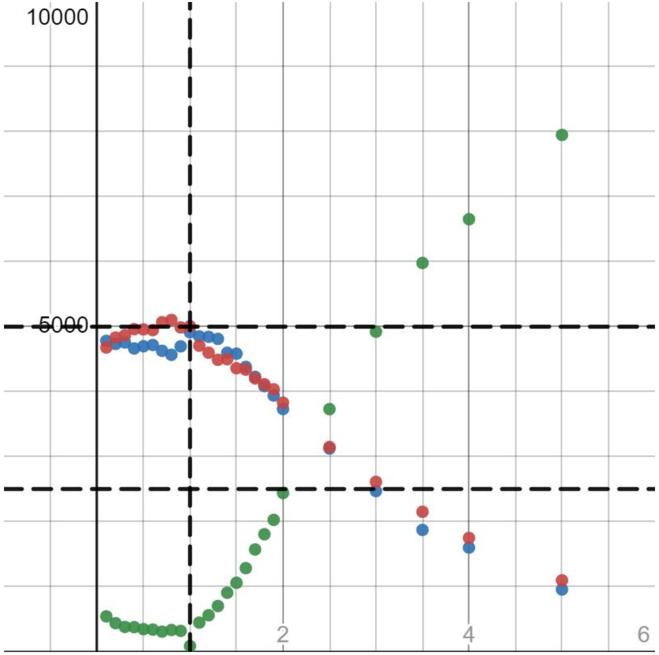
- Made in godot
  - Game engine
- Constantly spawn in up to 100 dice.
  - Give each dice a random velocity and angular velocity.
- Keep the dice within bounds.
- When a dice loses all speed record its result.
  - Calculate which face of the mesh is lowest.
- Took a few days to run all 1,250,000 dice
- Link to the project
  - https://github.com/DannySchultzDev/Dice-Simulation





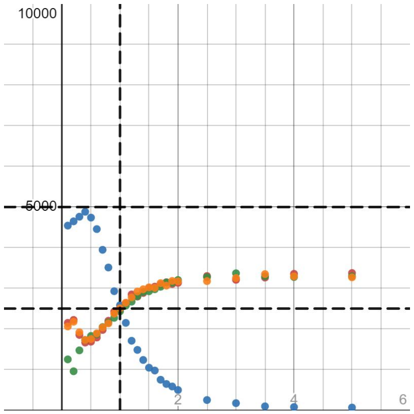
Spinning

- To prevent dice from getting stuck in the ground, if a die was ever alive for too long its side was recorded as if it had landed.
- Due to the large angular speed given to the frustum dice, they had a tendency to spin for so long that they would be deleted while spinning.
- On my initial simulation run, the frustum dice frequently landed on their green side as an outcome, especially as the difference between radiuses increased.
- Due to this issue I reran the simulation with a higher cutoff time, this mostly fixed the issue and gave me twice as much data

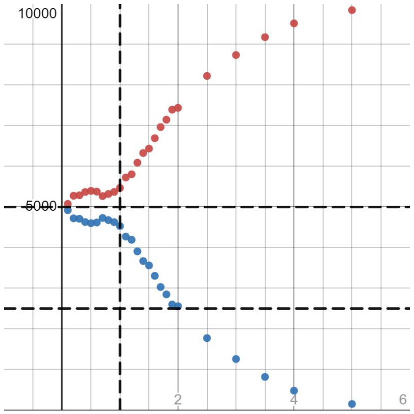


Results

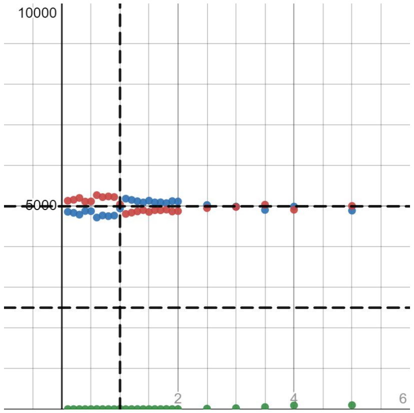
Tetrahedron



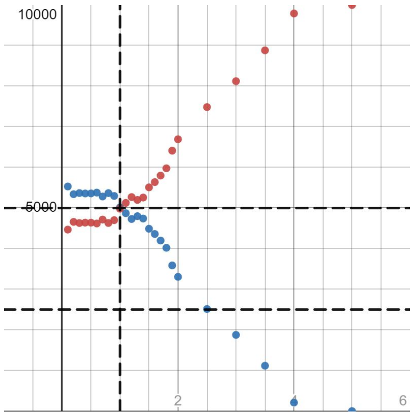
Cone



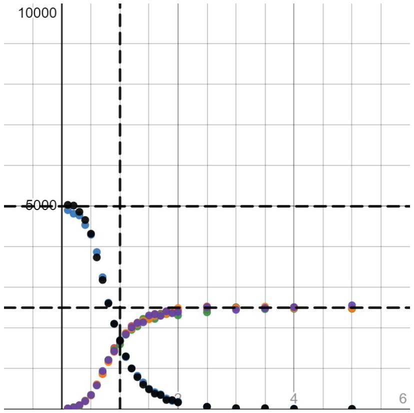
Frustum



Bicone

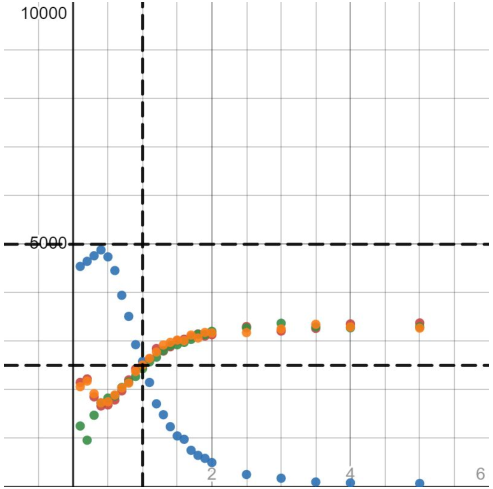


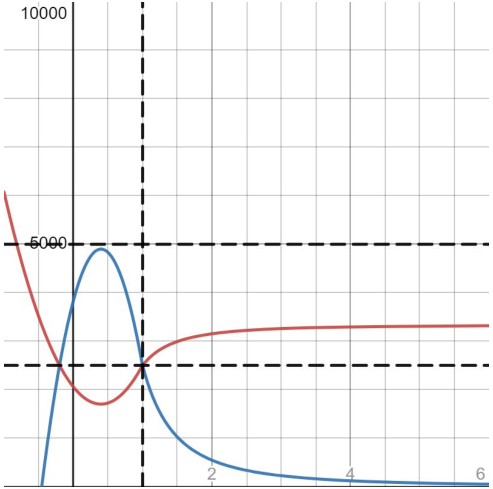
Cube



Tetrahedron (results)

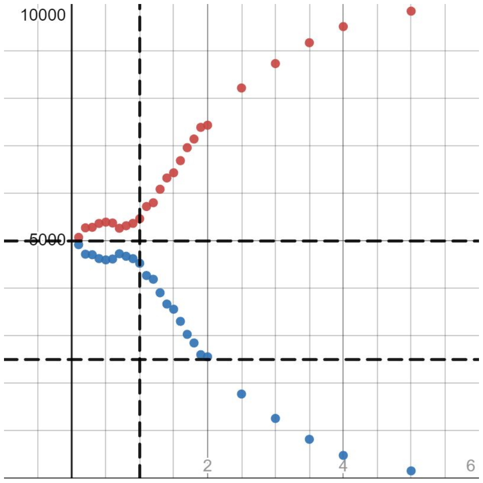
- One oddity is that the green side consistently came up less often than other sides when the height was <.4
  - The red, yellow, and green sides are identical so this is a strange result
- Another oddity that landing on the constant face did not approach .5 as the dice's height approached 0
  - It was expected to model the results of a coin as the dice became flat
- These oddities could be due to running too many objects in the simulation at once, or possibly other errors. More testing is required.

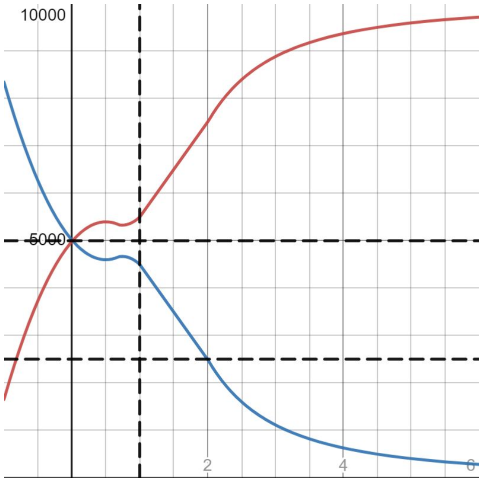




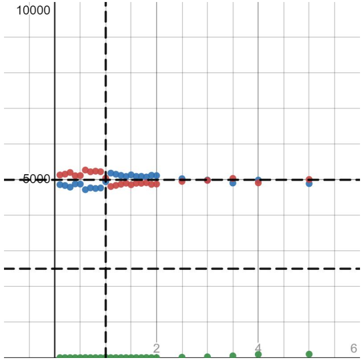
Cone dice

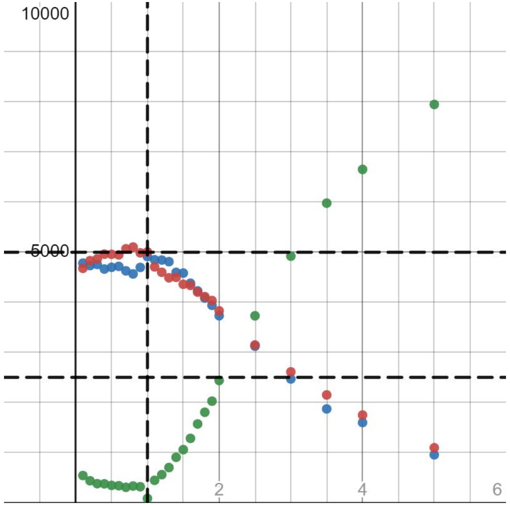
- An interesting observation is that the odds become more fair when the cone approaches a height of .7

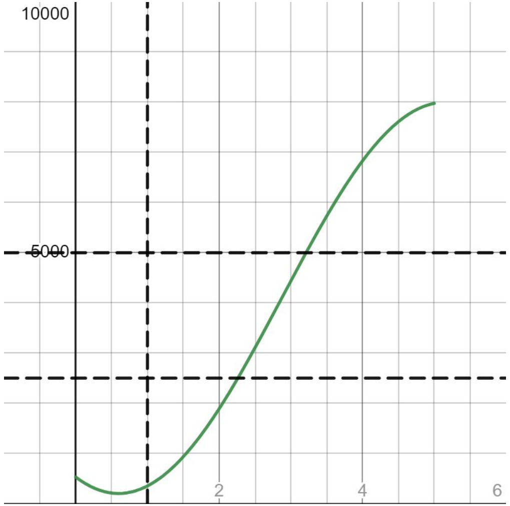




Frustum dice

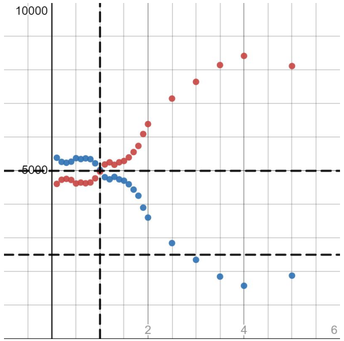


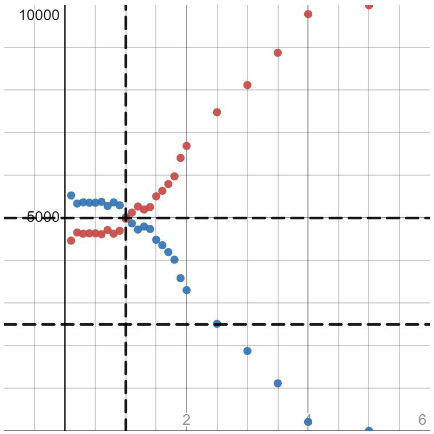


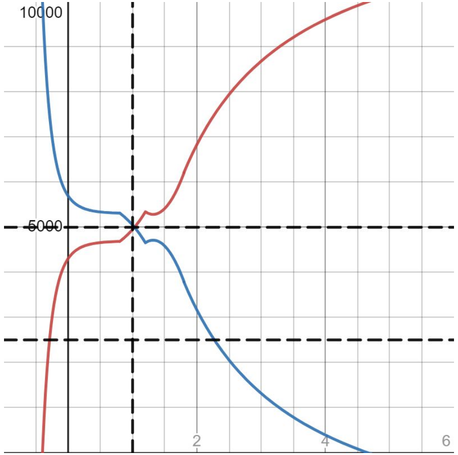


- Surprisingly changing the radius of one side of the frustum did not seem to change the probability of the dice landing on the top or bottom face.
- Changing the radius of the frustum did change the time it took for the dice to come to a halt.
  - This can be seen from the program hitting the failsafe and recording the bottom face as green (a side face).
  - When the dice was at its base position (the dynamic face had a radius of 1) it was never still spinning.

Bicone dice



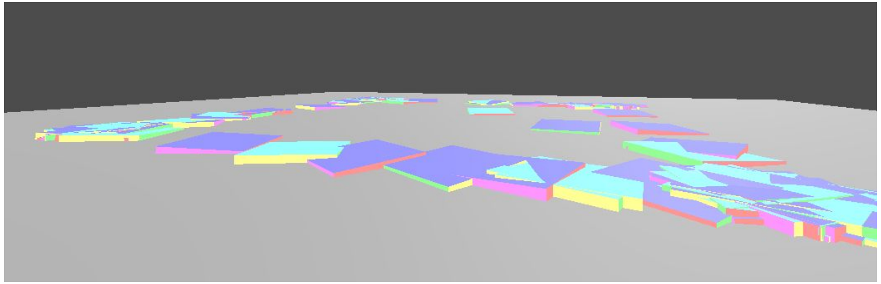


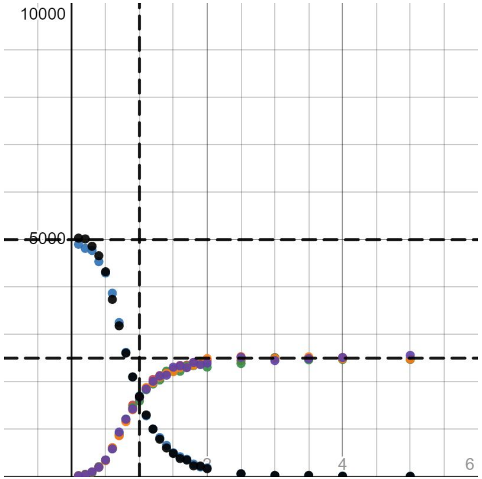


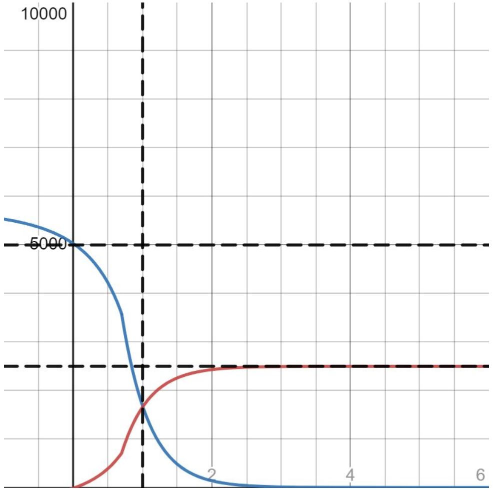
- Odds remain relatively constant from heights of .2 to .8.
- Odds become more fair as height approaches 1.3.
- Similar to the frustum dice the bicone dice can spin so long that they exceed the initial failsafe time as they get longer.

Cube dice

- Most simple distribution
- Desmos does not have cyan as a selectable color
  - I represented it with black in the presented graph
- Shorter cubes got stuck in the ground frequently







Thank you for listening!