An Approach to Modelling Collision Risk in Slither.io

Artin Ghadar

1 Introduction

1.1 Concept of the game

The online game Slither.io works as follows. You are in a lobby with other players (normally in the range of a few hundred) and your aim is to grow your snake's score, which in turn increases its length and width. This is done by collecting pellets that spawn in two ways: naturally on the map or through the death of another snake. It is also important to note that a snake in a multiplayer game is predominantly another person, but sometimes a bot. A snake dies by having their head collide into the body of another snake (the head of a snake can go through its own body safely) or by having their head hit the edge of the map, the latter being avoidable and uncommon alongside not dropping any pellets. A snake moves at a constant speed, but can sprint to go quicker, expending a constant minimal 15 pellets/second.

1.2 Methodology

In this essay, I aim to develop a quantitative model for the risk of collision experienced by a player. This risk depends on several moving variables and a key challenge of the task lies in translating these qualitative gameplay features into mathematical relationships.
To achieve this, I will define the following: formulae for length and width (separately) in terms of score; the density of lethal area; the 2 different velocities in the game and the impact of human reaction time. Through mathematical modelling using these fundamental variables, I will arrive to quantitative conclusions that will provide understanding into how different variables interact under the overall aim of modelling risk within the game (alongside determining the critical conditions at which death is inevitable).
While the model relies on a number of reasonably derived assumptions, it captures the essential dynamics of the system and demonstrates how we can analyse a real-time interactive environment using a variety of mathematical techniques.

2 Defining Variables

2.1 Length and Width

As the numerical score increases, the length and width of the in-game snake increases. After scouring the internet for the correlation between these two variables - I found no data and no formula. As a result I came up with a method to mathematically estimate a formula for the length of the snake. To do this, I first thought to measure the length of a snake in pixels compared to the score, plot this and then work from there. As length increases, the game dynamically zooms out. This brings about an issue in using pixels alone, because it is dependent on an unknown dynamic zooming factor. To fix this problem, I installed the NTL Mod for Slither.io which allowed me to turn off dynamic zooming. Using the software GIMP, I was able to measure the length of snakes up to a certain size, as shown in Figure 1.

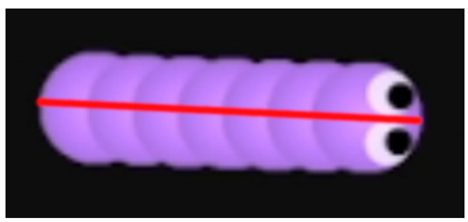

Figure 1: Snake of score 13 and length of 295 pixels

When the snake reaches the point at which the whole snake is not displayed on the screen because its length is too great, another problem arises. The map on the NTL Mod is segmented into many small sections, the number of which seen depending on the zoom factor. There is also the very useful option of manual zooming. With the sections as reference points, I was able to calculate the length of the snake in terms of how many sections long it is. We will from now on use a section, denoted by x, as the base unit of length within the game for analysis. This could also be done with dynamic zooming, but every screenshot would need the length of a section (in pixels) to be measured to find the ratio. Whereas with manual zooming, it can be set to a zoom and stay there - only needing the section length to be measured every time you change the zoom, being far more efficient. Let us call the length of a section in pixels x_p and the length of the snake in pixels L_p, therefore we get the length of the snake (L), in units of x, to be as follows:

$$L = \frac{L_p}{x_p}$$

Figure 2 shows an example of how this was executed:

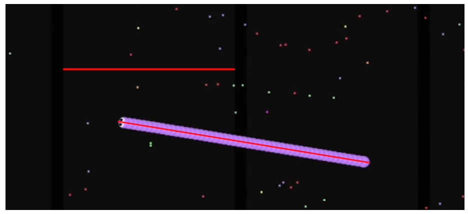

Figure 2: Snake of score 467, length of $1.481x$: $x_p = 675$, $L_p = 1000$

Using this formula, with more data that I acquired through the same methodology, I was a able to plot it on Desmos and therefore estimate a regression. Similarly, I repeated this technique for width(d), arriving to this formula:

$$d = \frac{d_p}{x_p}$$

In this case the zooming problem was irrelevant: the width would not go off screen because the game is based around the head of the snake. Figure 3 shows measuring width:

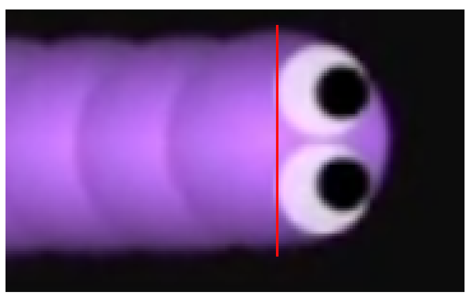

Figure 3: Snake of score 553 and width of $0.056x$: $x_p = 1610$, $d_p = 90$

This allowed me to derive an estimate formula for both width and length based on score. It is important to note how these formulae are estimates, they are not exact since they have been derived from limited data sets - however they are sufficient for the context of this model. They are as follows (rounded because as I said they will not be exact, therefore having many significant figures is irrelevant). Let S = score:

$$d = 0.26S^{0.14}$$

$$L = 0.020S^{0.71}$$

Below is Figure 4, with the x-axis representing S and the y-axis representing x:

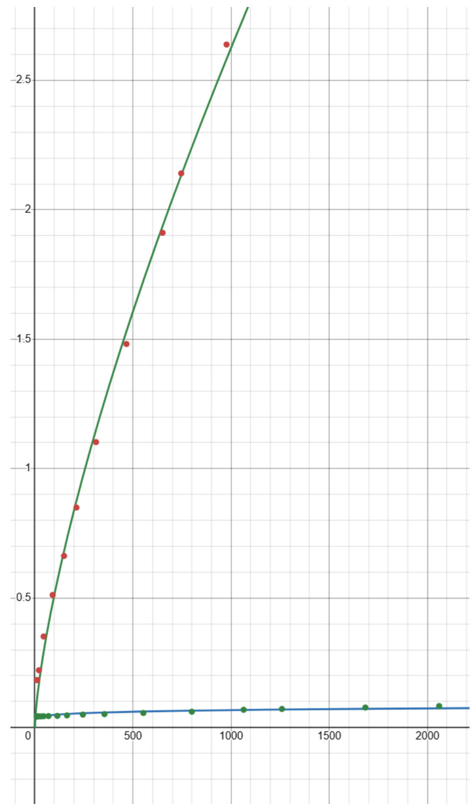

Figure 4: Regressions and Data Plots for Length (Green Curve, Red Plots) and Width (Blue curve, Green Plots)

2.2 Density

Using x as a base length, defined in Section 2.1, we can use x^2 as a base area. To define density (ρ) in this context we need to find a unit A such that $\rho = \frac{A}{x^2}$. The most obvious route is to use the number of snakes(N_s), but the issue with this is that every snake will be a different size with different percentages of their bodies within the area - leading to a useless measure of density. Instead, by using the total area in the region (excluding your own snake - it is not lethal) we can calculate a useful and accurate measure of the lethal density of a region in space. Even if this area is not accessible (i.e. in the width of a snake), it acts as lethal space taken up by a snake. If we let the fraction of a snake in the area to be f_s then density can be defined as follows:

$$\rho = \frac{\Sigma L d f_s}{x^2}$$

This essentially acts as a measure of how much of the area is made up of lethal 'snake'., thus we can derive density through score.

2.3 Velocity

Again using x as a base length, we can measure the time taken to travel a distance of x at the two different speeds. I used the image analysis software Kinovea to inspect the video in frames of 0.02 seconds. If the frame did not capture the exact point at which the snake arrived at the boundary, looking at the next frame allowed me to find the time rounded to the closest 0.01 seconds with sufficient accuracy for the purposes of the experiment. For the non-sprinting speed my measurements were between 3.07-3.09 seconds to travel x and had an average of 3.08 seconds (all measurements will be to the nearest 0.01 second because anything less than that is meaningless because of the resolution of the experiment) and the sprinting speed was 1.27 seconds consistently. We can look at the velocities in units of xs^{-1} as $v_0 = 0.32xs^{-1}$ and $v_1 = 0.79xs^{-1}$.

2.4 Human Reaction Time

Based on data from human benchmark, the median human reaction time is $273ms$ and the mean is $284ms$ (this includes the reaction time of the device used which would normally be about 10-$50ms$). However, this must be taken into account because the game will of course be played on a device. We should also consider that these are averages and not necessarily representative of the target demographic that plays video games, therefore a conservative estimate must be used. With the average gamer reaction time supposedly being 150-$250ms$, the figure of $175ms$ is more appropriate to use as a conservative estimate, let τ be used to denote this.

3 Mathematical Modelling

3.1 Modelling Turning Mechanics

The arc length of a circle between 2 radii is defined as the angle between the 2 radii multiplied by the length of the radius.

$$s = R\theta$$

Rearranging we see how:

$$\theta = \frac{s}{R}$$

Therefore:

$$\frac{d\theta}{ds} = \frac{1}{R}$$

If curvature is defined as the rate of change of direction per distance unit:

$$k = \frac{d\theta}{ds}$$

Then:

$$k = \frac{1}{R}$$

Using:

$$s = R\theta$$

Dividing by dt on both sides we get:

$$\frac{ds}{dt} = R\frac{d\theta}{dt}$$

Angular velocity is defined by change in angle per unit of time:

$$\omega = \frac{d\theta}{dt}$$

$$\frac{ds}{dt} = R\omega$$

And since s is a unit of distance (around the circumference of the circle):

$$v = \frac{ds}{dt}$$

$$v = R\omega$$

$$\omega = \frac{v}{R}$$

Using

$$k = \frac{1}{R}$$

We can also express ω as:

$$\omega = vk$$

To determine a reasonable estimate for a minimum turn radius, I turned a snake around in a circle as tightly as possible so that the radius of the circle is as close as possible to the minimum turn radius, see Figure 4.

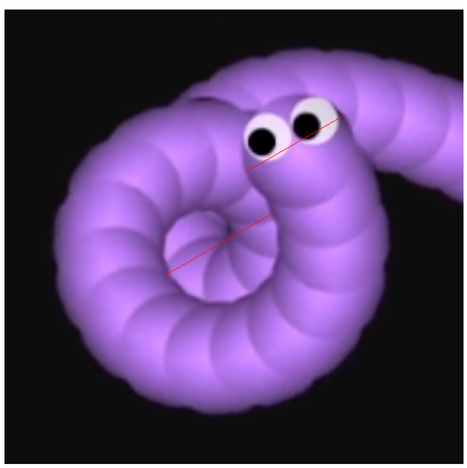

Figure 5: Snake of minimum turn radius relative to its width

Naturally, the turn radius will depend on the width of the snake. The image shows how the turn diameter is approximately equal to the width of the snake (there is not necessarily a clear point at which we define the width to start and end alongside the potential for this not to be the exact minimum turn, even if it is very close). Therefore:

$$R_{min} \sim \frac{d}{2}$$

To maximise curvature we must minimise the turn radius so:

$$k_{max} \sim \frac{2}{d}$$

Consequently, we can find the maximum angular velocity:

$$\omega_{max} \sim \frac{2v}{d}$$

3.2 Setting up survival conditions

We can also integrate human reaction time into the action of turning. Let us say that an event is recognised that requires turning, how can we model the lag? First, let us say that there is an snake with a distance D from the head of our snake. Consequently the time available to react is as such:

$$T = \frac{D}{v}$$

This means that

$$\tau \leq T$$

$$v\tau \leq D$$

must be true or else there will be a collision.
This allows us to find the minimum distance the head of our snake can be to the body of another snake without us being doomed (them moving is irrelevant because of the time it takes for the motion to move down the body). Using previous estimates of τ the results are as follows for v_0 and v_1:

$$v_0 : D \geq 0.056x$$

$$v_1 : D \geq 0.13825x$$

$$D_{min} = 0.056x$$

We can also set up a survival condition. If a change of direction($\Delta\theta$) must be made in the time available then:

$$\omega_{req} = \frac{\Delta\theta}{T}$$

If

$$\omega_{max} \gtrsim \omega_{req}$$

Then

$$\frac{2v}{d} \gtrsim \frac{\Delta\theta}{T}$$
$$\frac{2v}{d} \gtrsim \frac{\Delta\theta v}{D}$$
$$\frac{2}{d} \gtrsim \frac{\Delta\theta}{D}$$
$$D \gtrsim \frac{d\Delta\theta}{2}$$

Consequently the survival condition is:

$$D \gtrsim max(v\tau, \frac{d\Delta\theta}{2})$$

This also allows to model the change in direction needed to escape danger:

$$\Delta\theta \lesssim \frac{2D}{d}$$

But we have to consider the change in position of the other snake when there is this chance of collision. Since constant movement is mandatory, the snake that we would collide into may not be there at the time of collision depending on its length. The other snake may want to go at v_0 but in these short time scales, reaction time will limit any velocity change - let the other snake's velocity be v_b:

$$\Delta x = v_b T$$

If Δx is greater than the remaining snake (α), a collision will not happen. Thus the amended survival condition is:

$$D \gtrsim max(v\tau, \frac{d\Delta\theta}{2})$$

when

$$\alpha > \Delta x$$

3.3 Applying Density

Since every particle has the same mass (they don't literally have mass but for the sake of the modelling we need to express them as being the same, which they are). In terms of standard density we can model distribution as such: if each particle has mass m_m and the average distance between them is D_m then we can look at area as:

$$A_m \sim D_m^2$$

and thus density as:

$$\rho_m \sim \frac{m_m}{D_m^2}$$

so:

$$D_m \sim \sqrt{\frac{m_m}{\rho_m}}$$

Since the mass is arbitrary and we can just imagine it to be one:

$$D_m \sim \sqrt{\frac{1}{\rho_m}}$$

We have defined density in the context as the fraction of lethal area, so we can use it to model the average distance between your snake and the lethal area. Thus we get:

$$D \sim \sqrt{\frac{1}{\rho}}$$

Now into the survival condition:

$$1 \gtrsim \rho(max(v\tau, \frac{d\Delta\theta}{2})^2$$

when

$$\alpha > \Delta x$$

Expressing $\Delta\theta$ as:

$$\Delta\theta \sim \frac{2D}{d}$$

and therefore:

$$\Delta\theta \sim \frac{2}{d\sqrt{\rho}}$$

3.4 Conclusion

This allows us to represent a survival condition in the form of modelling risk purely on pre-definable characteristics of an area. Thus we can define our final function for risk as:

$$F(\rho, v, \tau) \sim \rho(max(v\tau, \frac{1}{\sqrt{\rho}}))^2$$

We can ignore

$$\alpha > \Delta x$$

as it is a parameter for determining whether or not a collision will happen, only really occurring with very small snakes and thus too uncommon and insignificant to be included in the final risk estimate.
We can now model the dominant factors at low and high densities. The critical density(ρ_c) is when:

$$v\tau = \frac{1}{\sqrt{\rho}}$$

Therefore:

$$\rho_c = \frac{1}{(v\tau)^2}$$

Low density is defined when:

$$\rho \ll \rho_c$$

Then:

$$\frac{1}{\sqrt{\rho}} \gg v\tau$$

so:

$$max(v\tau, \frac{1}{\sqrt{\rho}}) = \frac{1}{\sqrt{\rho}}$$

$$F \sim \rho(\frac{1}{\sqrt{\rho}})^2$$

$$F \sim 1$$

This shows with a density approaching 0, our risk approaches 1 - the lowest risk.
A high density region would be modelled as:

$$v\tau \gg \frac{1}{\sqrt{\rho}}$$

so:

$$max(v\tau, \frac{1}{\sqrt{\rho}}) = v\tau$$

$$F \sim \rho(v\tau)^2$$

This shows that with a maximum density approaching 1, alongside v_0 and τ at $175ms$:

$$F_{min} \sim 0.003136$$

Consequently:

$$0.003136 \lesssim F \lesssim 1$$

where a greater value of F correlates to a lower risk.
Using logical assumptions, we have been able to produce a coherent framework for modelling collision risk in the dynamic system of Slither.io which allows analysis of events with inputs on a macroscopic and microscopic level.