**Genshin Impact Damage Optimization — Using math in the “real” world**

**Teacher:** Daniel O’Malley

**Time:** 1 hr

**# of Students:** whatever the classroom can fit

**Overview:** “When will we use this in the real world?” We’ve all heard this phrase too many times, it’s inevitable! No one cares about how many watermelons Sally has left, or the beard area under a sinusoidally shaped mustache. Today, we’ll use calculus to do something that really matters . . . winning videogames. If you’re interested in videogames, math, or both be sure to come by and see how we actually use math in the “real” world.

* Knowledge of Genshin Impact is not required to enjoy this class

**Goals:** The goal of the class is to walkthrough a demonstration of using calculus under the topic of videogames. Hopefully, students will understand that the skills they learn in class are not useless and will be encouraged to apply them to their own hobbies.

**Materials:**

* Projector
* Whiteboard / Chalkboard
* maybe some printouts in the future; idk I’ll figure it out.

**Class Outline:** This class will be 100% a lecture and listen format. Students will be encouraged to follow along the math by taking notes, but it is not required.

Curriculum Notes:

* Before we begin gauge the detail of audience Genshin knowledge
  + Ask them
    - Play genshin
    - AR 50+
    - Know what KeqingMains community is
    - Know what Elemental Gauge Theory is
* Chapter 0: Motivation
  + What is Genshin?
    - This is more in case, there are people who don’t play
    - “Genshin impact is an open world action rpg where you explore the 7 nations of tey-BLAH BLAH BLAH”
    - Who cares about all that, skip the dialogue
  + What we care about is that when you hit things, you see your damage . . . in NUMBERS
  + So my ape brain’s gotta chase BIGGER number, more DAMAGE
  + Where there’s numbers, there’s math, so today I’ll show you how we use math in the “real” world
* Chapter 1: Damage calculation
  + Numbers are great and all, but how do we get it?
  + Well there’s a lot going into it
    - Overwhelming amount of stat sources
    - Base attack
    - Character level
    - Ascension stat
    - Character constellation
    - Talent multiplier
    - Talent abilities
    - Talent passives
    - Weapon base attack
    - Weapon secondary stat
    - Weapon passive
    - Artifact main stat
    - Artifact sub stats
    - Artifact set effects
    - Elemental resonance
    - Teammate buffs
    - Elemental reactions
    - Enemy level
    - Enemy resistance
    - Enemy defense
    - Abyss effects
    - Commission effects
  + You don’t need to know these. There just isn’t enough time to explain everything
  + What you need to know is that there are different categories of damage statistics, which the player has some limited control in obtaining and exchanging.
  + A screenshot of a computer program

    Description automatically generated
  + This is the damage formula!
  + A bit scary
  + Mathematics Secret Technique #12
    - Make assumptions to make it easier
  + Show modified version with written explanations for those how know the game and are curious
  + Simplify
    - Reduce additive terms
      * )
    - Math Secret Technique #7: Algebra trickery
  + )
  + Now we got an interesting formula
  + Q: Lets say you want the largest DMG possible. If you can choose, would you like a D of 1.0 or a D of 1.2 and why
    - Bigger numbers = more damage! EZ
  + Q: If you can choose to add 0.5 to D or to A, which would give you a larger DMG.
    - Harder to answer . . .
* Chapter 2: Fences and Boars
  + Introduce a common calculus problem
    - Xiangling wants to build a rectangular enclosure for fresh meat. The meat tastes best when they have more area to move around, but Xiangling wants to use as little fencing as possible. What is the optimal length to width ratio of the enclosure.
    - Max area, min perimeter
    - Max A when , or on boundary
    - Boundary: and 🡪
      * 🡪
    - In other words, the square shape the calculus problem shows that the square shape optimizes area with minimal perimeter
  + Now lets look at a the crit multiplier
    - The equation represents it as
    - How does Crit work?
      * When you hit, you get a % chance of a critical hit determined by Crit Rate
      * The Crit hit is increased by a percentage determined by Crit Damage
      * To streamline it, we use average damage
    - Now let’s optimize the crit multiplier
      * Often in the game we have to choose between crit rate and crit damage
      * Is it better to hit BIG but less often, or to hit smaller but more often??
      * Well doesn’t the crit multiplier look familiar?
      * YES!
      * Optimizing the crit multiplier is like optimizing the area of a rectangle
        + Let crit rate be length
        + Let crit damage be width
        + The area of that rectangle is the output crit multiplier
      * Max CM, min IL
      * Max A when , or on boundary
      * Boundary: and 🡪
        + 🡪
    - Awesome! So ideally we want equal crit rate and crit damage
      * But . . . this calculus uses a certain assumption. Someone who plays Genshin may know what that is
      * The model assumes that crit rate and crit damage take the same amount of investment
        + i.e. 1% crit rate can be exchanged with 1% crit damage
      * However, in game, we often find that crit damage is 2x easier than crit rate to find
        + i.e. 31% crit rate can be exchanged with 62% crit damage
    - Adjust the parameters
      * Max CM, min IL
      * Max A when , or on boundary
      * Boundary: and 🡪
        + 🡪
      * Now you want same number of units of each crit. Which means you want a 1:2 ratio of crit rate to damage
* Chapter 3: Investment and Benefit
  + Units of investment is an interesting idea, lets apply them to other fields
* Chapter 4: Optimizing ATK and CRIT
  + Max damage when max
  + Quadratic Equation
  + Critical point
  + Optimized DAMAGE!!!!!!
  + Desmos to show the rest

|  |  |  |
| --- | --- | --- |
| * + .   + Max area, min perimeter   + .   + .   + Max A when , or on boundary   + Boundary: and 🡪   + 🡪 | * + .   + .   + Max CM, min IL   + Max A when , or on boundary   + Boundary: and 🡪   + 🡪 | * + .   + Max CM, min IL   + Max A when , or on boundary   + Boundary: and 🡪   + 🡪 |