Introduction

osu! has been around for over 10 years now and it has evolved drastically over that time. As expected that means there now remain aspects of the game and its ecosystem that are severely outdated due to sheer age or neglect. Today, they either cause problems through exploitation or are simply not useful anymore.

There has been a lot of debate in the community, especially recently, about many of these issues:

- Difficulty calculation
- Performance points
- Score V1 / V2
- Score V2 gameplay changes
- Mapping meta
- Tournament rules and regulations
- Rule breaking and punishments
- *and more...*

Most, if not all, of these issues are the result of aspects of the game and ecosystem that have not kept up with the times:

- Difficulty calculation was a tremendous step forward when it was initially introduced, however development has been largely ignored for the past couple years due to several reasons.
- For the most part, performance points and the mapping meta are extensions of the difficulty calculation issues, although they have problems of their own as well.
- Score V1 is severely outdated (hasn't been touched in over 10 years), and V2 has been a heated topic since its introduction as an alternate scoring system, however it has steadily become the preferred scoring system for tournaments.

This essay will be split into 3 main sections in which I will discuss the difficulty calculation system, performance point system, and scoring system. Each section will identify the issues surrounding the topic, potential changes to help fix these issues, and the effects these changes could have on other aspects of the game.

Difficulty Calculation

In order to identify the issues with current difficulty calculation, we need to first define what the goal of a difficulty calculator is.

The current difficulty calculator implemented in osu! more or less defines difficulty as:

'the weighted sum of the physical strain on a player while performing the actions required to perfectly clear a beatmap'

In plain terms this means that lower time and higher spacing between consecutive objects makes for higher difficulty, and the more densely clustered the difficulty spikes are, the higher the difficulty will be. While this definition does accurately measure a large aspect of difficulty, it ignores various others which are not determinable with such simple values as distance and time alone.

A more appropriate definition of difficulty is perhaps:

'the weighted sum of the difficulty of the actions required to perfectly clear a beatmap'

While this may sound similar, it has one key difference. There is no strict binding to the raw physical strain as a scale; instead the playable difficulty of objects is emphasised.

The distinction between these definitions is often explained using the common example of aim angles. It is the general consensus that aim that is considered 'jumpy' is far easier to execute when there are low angles between the objects (as commonly seen in the current 'pp mapping' meta with triangle patterns), while high angles are significantly more difficulty. However when the aim is considered to be more 'flow-y' the opposite is true, high angles are far easier than low angles (if the objects are not overlapped enough to not allow aim shortcuts). If you only look at distance and time between objects, it is possible to create two maps that are of equal difficulty according to the current system, but worlds apart in playable difficulty.

Now that we have identified the problem and defined what the goal of a difficulty calculator should be, we must devise a system for quantifying the 'difficulty of the actions required to perfectly clear a beatmap' and how exactly to weight those actions when summing the result, that is, the difficulty of completing an individual hit object in the context of the beatmap and how much it contributes to the final star rating at the end.

Note that the rest of the difficulty calculation section discusses specifics about the implementation of various ideas and thus contains a fair bit of maths. If you don't wish to read about those details, feel free to skip to the next section (Performance points and Rankings).

Skills and totalling

We are going to use a structure that splits up the difficulty into multiple parts - or skills - that will each measure an aspect of difficulty independently. In the common play styles of osu! there are 3 skills which are physically separated. Aiming (the movement of the cursor to the hit objects), clicking (the clicking of the hit objects), and reading (the identification of hit object positions and timing).

First we need to explain how to get a final star value for the map from the difficulty of each individual object.

The current system processes hit objects in sections of 400ms. Each hit object adds an amount to the 'strain' value, and the peak strain value within that section is taken as the difficulty. After all sections have their difficulty calculated, they are summed up using a similar weighting formula to that of players' top performances. The difficulty values are sorted from highest to lowest and plugged into this formula:

$$\sum_{i=0} diff_i \cdot weight^i$$

where *weight* is 0.9

This means that if you had, for example, a beatmap with 3 sections of peak values 1, 6, and 3, the total difficulty for the beatmap would be:

$$\begin{aligned} &= (6 \times 0.9^0) + (3 \times 0.9^1) + (1 \times 0.9^2) \\ &= 6 + 2.7 + 0.81 \\ &= 9.51 \end{aligned}$$

This has a few issues though.

First, splitting the map into 400ms sections means that some difficulty spikes may be ignored if another occurs in the same 400ms section; while only a very minor issue, it is still an inaccuracy.

Second, since we want to be determining difficulty on a per object basis, this formula doesn't work for us at all since it is assumed that sections are of equal length which would bias maps with average section length shorter than 400ms. However, we can modify it to work for variable length sections (individual objects/actions) without too much trouble.

Additionally, modifying the difficulty calculator to work on a per object basis opens the door for more interesting performance point and score systems, but I will discuss those in later sections.

For each skill we will:

1. Convert each hit object into a 'difficulty point', which will contain all the information about this object that relates to the given skill.
   We do this is so that skills like aiming aren't limited to a single difficulty value per hit object if they themselves have multiple sections of difficulty. For example, slider ticks should be treated the same way hit circles are (except for the radius) for aiming. Additionally, this provides a good place to apply the effects of mods.
2. Calculate the value of each difficulty point using the skill specific methods.
   This is discussed in later sections.
3. Sort the difficulty points in descending order of value.
4. Plug them into the formula:

$$\sum_{i=0} diff_i \cdot weight^n$$

   where *weight* is 0.9, *n* starts at 0 and each point increases *n* by $\frac{difficulty\ point\ length}{400}$.

   This means that if you had a beatmap with 3 difficulty points of timings 100ms, 500ms, and 300ms and difficulty values of 4, 7, and 3 respectively, the total beatmap difficulty would be:

$$\begin{aligned} &= (7 \times 0.9^0) + (4 \times 0.9^{1.25}) + (3 \times 0.9^{1.5}) \\ &= 7 + 3.5 + 2.6 \\ &= 13.1 \end{aligned}$$

   First *n* increases by 1.25 because $\frac{500}{400} = 1.25$ and then by 0.25 because $\frac{100}{400} = 0.25$.

   This is essentially the same formula except modified to 'weight the weight' by section length to allow for variable length sections.

Physical skills

Both aiming and clicking are physical skills, so they can be treated similarly in some regards. Physical skills have a common aspect, in that they cause physical strain on the player as they exert the energy to perform an action. To avoid confusion with the current difficulty calculator, we will refer to the physical strain on a player as *exertion* instead. Strain in the current system tracks a single value since it is the only form of difficulty quantified. Instead, we will have exertion act as a bonus of sorts upon the raw difficulty of the object.

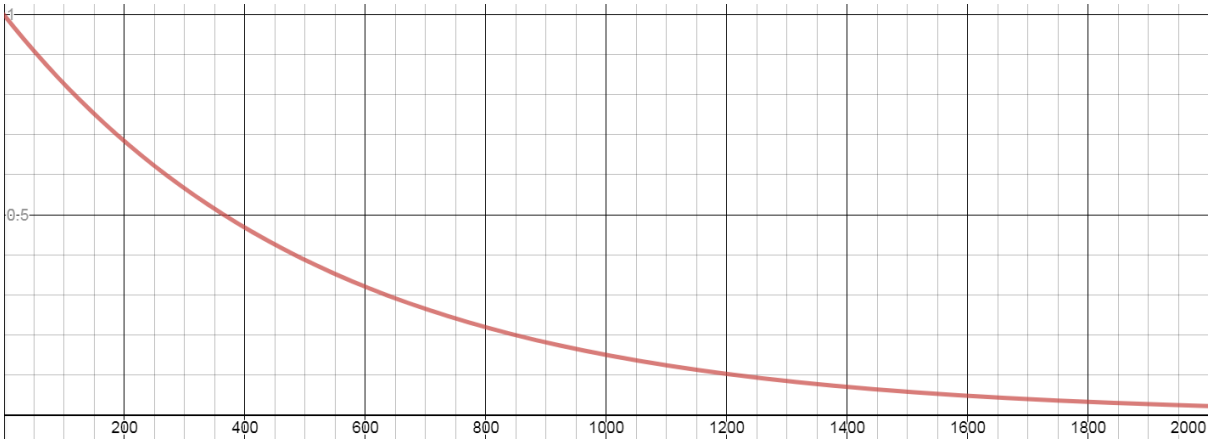
Difficulty for physical skills is made up of 3 parts: the raw difficulty of the action (which is used as the base value), an exertion bonus, and a semantic bonus. The raw difficulty is simply $\frac{energy\ exerted}{time}$, and the semantic bonus accounts for non-exertion based difficulty, such as angles in aim, and rhythm complexity in clicking. The exertion bonus comes in 2 parts: speed, and stamina. Speed is used to measure the difficulty increase of energy exerting actions in quick succession, whereas stamina is used to measure the physical

exhaustion after a long period of constant energy exertion (this is mostly relevant for streams). The energy exerted from an action is added to both of the exertion values and then both values are decayed over time, much like strain values in the current system. Both of the exertion values function in the same way, except for how fast they decay; speed will decay much faster than stamina.

Exertion decay uses the same formula as the current strain system:

$$decay\ portion(time) = base^{\frac{time}{1000}}$$

where *base* is the portion to decay to after 1 second and *time* is the decay time in milliseconds



Note this graph uses a base value of 0.15 as an example

This curve means that if the decay base is 0.15, the exertion value will decay to 15% of its original value over 1 second, with the decay rate slowing as it approaches 0.

Please note that the constants (coloured orange) used in the following sections' skill formula definitions are examples and in fact have only had minimal balancing. The purpose of these sections is to discuss potential methods of quantifying separate difficulty types. Balancing on this proposed system has yet to be conducted.

Aiming

Aiming uses a difficulty point type called 'aim points' which consist of a time, set of coordinates, and radius. Aim points are any point in a beatmap that is required for the cursor to move to as a specific time. This includes hit circles, slider heads and slider ticks.

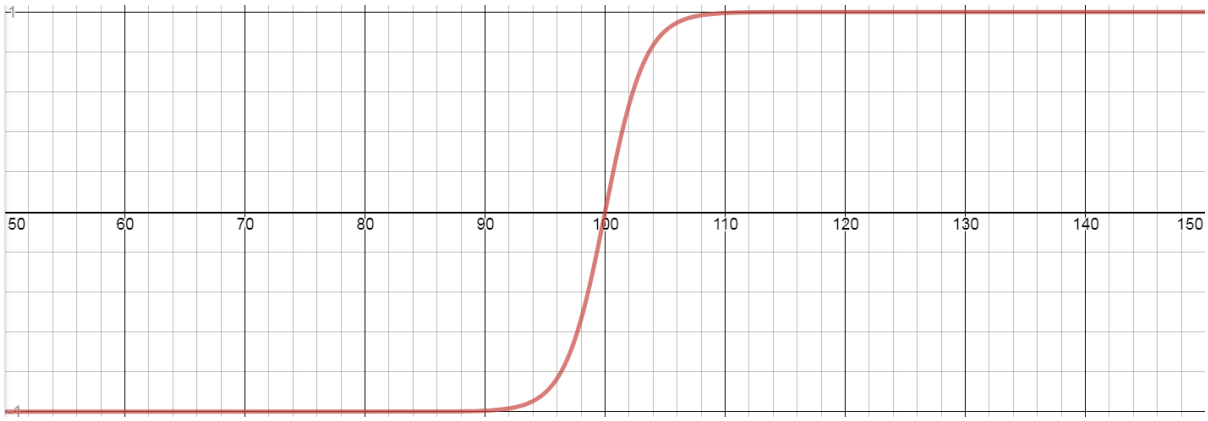
As aiming is a physical skill, it is made up of the 3 parts mentioned above, raw difficulty, exertion bonus, and semantic bonus.

The energy exerted for aiming is taken as the distance to the current aim point from the last one.

The semantic difficulty for aiming is made up of 2 parts - angle difficulty, and steadiness difficulty - which both depend on the 'snappiness' of the action. Snappiness is the degree to which players are likely to be able to 'snap' to the aim points rather than 'flowing' them. Snappiness is a function with range (-1, 1) where -1 is completely flow, 1 is completely snap, and is defined as:

$$snappiness(delay) = \tanh(harshness \cdot (delay - threshold))$$

where *delay* is the time of the action in milliseconds, *harshness* is 0.3, and *threshold* is 100



This function defines a curve such that snappiness is high when the delay between the objects is high, and low when the delay is low. The point of inflection is the *threshold* value, and the rate of change is the *harshness*. With the current values, the threshold sits at the equivalent of 300bpm 1/2 notes (single taps) or 150bpm 1/4 notes (streams).

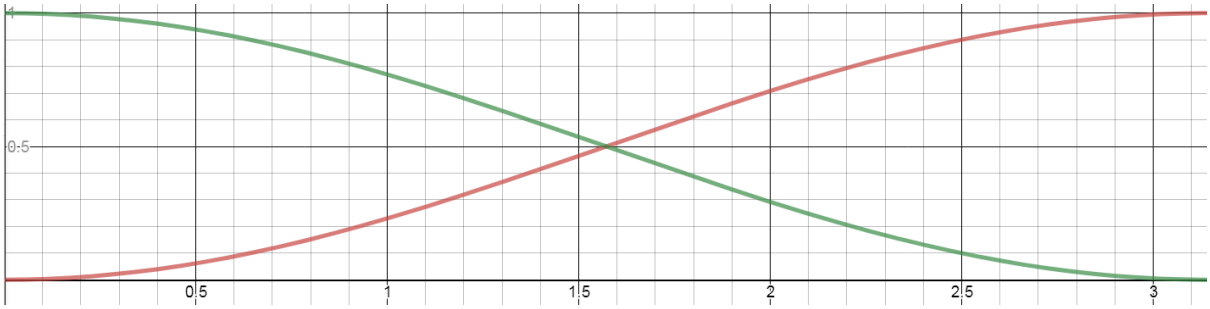
I believe the snappiness threshold value should also lower as exertion increases (especially speed) as it is one thing to snap a couple 300bpm single taps, but a whole different story to snap several consecutively.

Angle difficulty is the degree to which the angle formed by the previous aim action and the current one increases the difficulty.

It has a range of [0,1] and is defined as:

$$angle\ difficulty(a, snappiness) = \frac{snappiness \cdot \sin(a - \frac{\pi}{2}) + 1}{2}$$

where a is the angle in radians



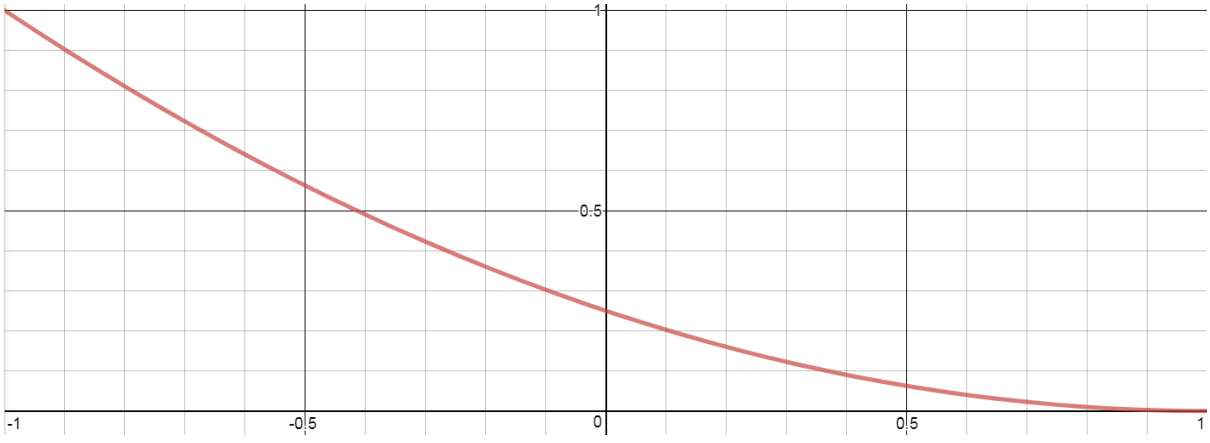
Red curve snappiness is 1
Green curve snappiness is -1

The reason the curve is non-linear is because there is a theory that the increase in angle makes snapping jumps more difficult because of the requirement to slow your momentum to a halt while approaching the aim point and then speed back up afterwards in the same direction meaning 2 shifts in force. For low angle jumps, the action of slowing down and speeding back up for the next aim point is one and the same, meaning there is only 1 shift in force as opposed to 2 for a high angle jump. As the angle increases, the amount of extra force shift is non-linear, as it follows a circular function, hence the non-linear angle difficulty.

Steadiness difficulty is the degree to which maintaining a constant aim speed over the course of subsequent actions affects difficulty, and thus is only relevant for low snappiness actions.

It has a range of (0,1) and is defined as:

$$steadiness\ difficulty(snappiness) = \frac{(snappiness - 1)^2}{4}$$



Steadiness could really be part of angle difficulty, since it is a subset of the domain and ties into the momentum shifting theory, however it has been split up for the sake of logical separation.

Clicking

Clicking uses a difficulty point type called 'click points' which only consist of a time. Click points are any point in a beatmap when the player is required to click. This includes hit circles and slider heads (spinners are not counted because they can be held from a previous note).

As clicking is also a physical skill, it is made up of the same 3 parts as aiming: raw difficulty, exertion bonus, and semantic bonus.

Unlike aiming, clicking doesn't require differing amounts of energy for different click points. All click points require a uniform amount of energy, which we can just say is a constant 1.

The semantic bonus for clicking is rhythmic complexity, usually seen in technical or old style beatmaps, as of writing this report, I have not determined an objective way to determine rhythmic difficulty, but I believe one exists and it will take some more research to find it.

Reading

Often you hear people talking about *objective* and *subjective* difficulty, however in reality, subjective difficulty doesn't really exist. Often when it is far too difficult to define a scale to objectively measure certain types of difficulty people just call it subjective instead.

Usually in osu! when you hear about subjective difficulty, it is relating to reading. Reading isn't subjective though: reading is literally how difficult it is to visually identify objects' aim position/order and click timing. Both of these things *can* be measured. However, it has another problem: exploitation.

The largest issue with reading is that it can be exploited. The act of memorising a map is seen by many as exploitation of the system since the difficulty is essentially removed. I am undecided about my opinion on this issue but I can see merit in both sides of the debate.

There is valid reasoning for reading to be regarded as a legitimate part of the system, but the fact that it can be exploited makes it extremely hard to allow it as part of a solid system.

There are also those who argue that memorising a map is in itself part of the process of mastering the map, so it should be considered. Regardless, I will discuss the methods I believe should be used for quantifying reading (and to some extent memorisation) difficulty, should it be desired.

First we will start with aim reading, which is identifying the position and order of the hit objects. Often you hear people say " low AR makes maps harder to read", however what they really mean is not low AR but instead high density, otherwise Easy and Normal difficulties would be objectively harder to read, which they clearly are not. Density is the amount of objects on the screen at any given time, the more objects to keep track of, the higher the density, the higher the reading difficulty.

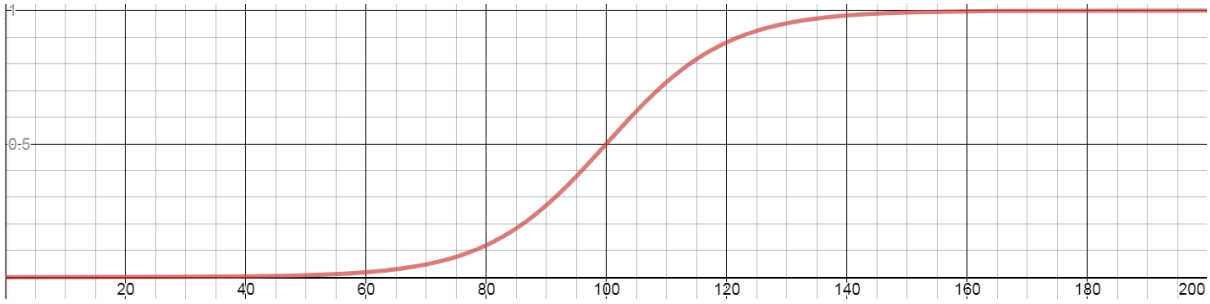
However, this has a problem. If we are to say that density is the primary factor in aim reading, then low AR stream beatmaps such as Mendes would be comparable to beatmaps like oddloop. The problem is that consecutive notes are easier to read if they are closely spaced. This is why triples are barely harder to read that single notes, and the same goes for streams.

In order to accurately measure density, instead of measuring the total amount of hit objects on the screen, we should measure the total focus required for each note. For this system I will refer to the focus required for a single hit object as its 'focal weight'. This way we can define the density of any given point in a beatmap as the sum of its visible hit objects' focal weights.

Focal weight of a hit object can be defines as:

$$focal\ weight(distance) = \frac{\tanh\big(harshness \cdot (distance - threshold)\big) + 1}{2}$$

where *distance* is the normalised distance from the last hit object, *harshness* is 0.05, and *threshold* is 100



Normalised distances are defined as the distance between 2 circles in osu!pixels if the target circle radius is 52 osu!pixels:

$$normalised\ distance(r_1, r_2, distance) = \frac{52 \cdot distance}{\frac{r_1 + r_2}{2}}$$

where *distance* is the distance between the circles in osu!pixels, r_1 is the radius of the first circle, and r_2 is the radius of the second circle

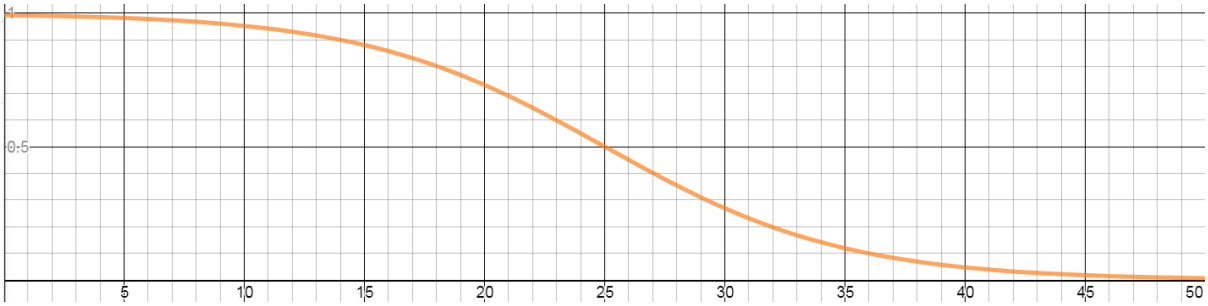
Using this definition for focal weight and density, we can get a good approximation for the aim reading difficulty of a map. However there is another factor that makes a large difference in the readability of a map: overlap.

Hit objects appearing underneath other hit objects are far more difficulty to see, however they do not have an increased focal weight. To account for this we can say that objects also have an overlap bonus if they are intersecting another hit object that is not part of a low focal weight chain of notes.

The overlap bonus can be defined as:

$$overlap\ bonus(distance) = \frac{-\tan(harshness \cdot (distance - threshold)) + 1}{2}$$

where *distance* is the normalised distance from the nearest object, *harshness* is 0.1, and *threshold* is 25



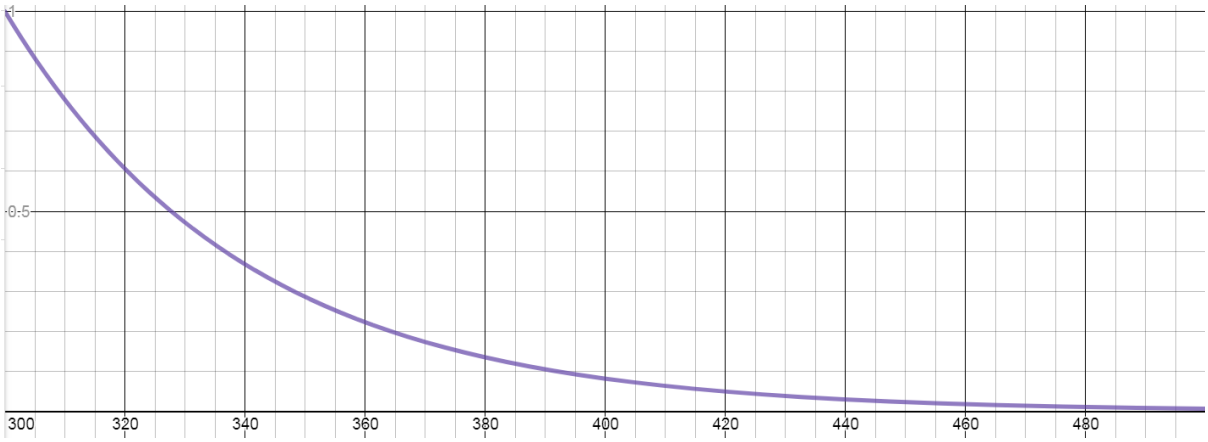
Click reading is a bit more difficult though. I initially had a theory that distance changes where no timing changes occurred would cause rhythmic misreads due to time-distance inequality, however the results were unsatisfying. As of writing this essay I have not come up with a full method of determining the reading difficulty of click timing, but perhaps it will tie into the physical rhythmic difficulty of clicking too.

On top of the aim and click reading, there is the sheer speed of the reading that comes into effect on extremely high AR beatmaps. As reading that quickly is a skill that needs to be trained to improve, a speed bonus is applied to the result of reading difficulty.

The speed bonus can be defined as:

$$speed\ bonus(time) = e^{\frac{-time+300}{rate}} + 1$$

where *time* is the approach time of the notes in milliseconds, and *rate* is 40



Note this graph is shifted -1 unit in the y-axis

This gives a 2x bonus at AR11 and a 1.082x bonus at AR10.3.

Visual mods are slightly more complicated however, especially Flashlight.

Hidden can be handled as a bonus based on AR. High AR giving almost no bonus while low AR gives a huge bonus. The bonus should cap out at a value that can be considered to be the bonus for memorising.

Flashlight is first and foremost a memorisation mod. It is practically impossible to sight read a map with Flashlight if its spacing is high enough. The bonus for Flashlight can be similar to the bonus for Hidden except instead of it scaling towards a memorisation bonus with AR, it would do so with visibility of the next hit object from the previous one.

Performance Points and Rankings

Performance points are the metric used for ranking players and are supposed to be seen as a representation of the skill of a player.

The current performance points system (ppv2) has been a part of osu! for approximately 5 years now and while it has been tweaked over that time, the core of the system has remained largely the same. Although it is criticised by many for being a flawed system (along with difficulty calculation), I believe the hype surrounding high pp plays has been a strong driving force in the game's growth.

Removing the current pp system would be a massive mistake as the community has grown too accustomed to it as a major part of the game. However I think it may be a step in the right direction to change the role it plays, and perhaps introduce a new system that can better fill its role as the primary metric for ranking players.

If most players find the need to grind specific beatmaps in order to rise in the rankings that are supposed to judge pure skill, then something is clearly wrong.

The player rankings are supposed to position players based on their skill. However does that mean that if a player is capable of clearing a map with 1 miss right in the middle that it should be considered vastly less skilful than without that single miss? Clearly not: they are displaying near identical levels of skill on every other object, yet that is nonetheless the system that we have known for all this time.

The reality is that the pp system is trying to accomplish 2 separate goals at the same time and does neither accurately. It attempts to judge players' skill, but also to reward plays that display a mastery of the map. This is why when people say "pp shouldn't be combo based", they are quickly rejected. Removing combo from pp *would* turn it into a true skill system, **but** it would also mean that many of the plays that are considered legendary due to the sheer mastery of the beatmap would be lost among the high skill but low combo plays.

I believe the solution to this issue is to split pp into 2 separate systems. This way each system can properly achieve its goal without needing to worry about accounting for the other. Current pp should weight combo and accuracy possibly even higher it current does to make plays with mistakes worth much less, and the new system (let's call it Skill Points, or sp for short) would eliminate most if not all combo weighting and emphasise the skill required to achieve the amount of misses, 50s and 100s in the play.

Performance points would display the level of mastery in a play, while skill points would display the raw skill required for the play and would be used as the player ranking metric. This would also be a step forward in solving the problem of players intentionally keeping

themselves at a low rank in order to play in low rank tournaments since it would be far more difficult to not gain skill points. But that's a topic for another time.

Additionally, if pp was a measure of mastery and not raw skill, it would be perfectly acceptable to consider reading difficulty as a legitimate part of the system, since memorisation is probably the highest level of mastery a player could display in a beatmap. This could potentially solve the problem of the reading ability affecting rankings debate.

The question then becomes what happens to pp? By no means does this imply that pp would or should be ignored, in fact the way high pp plays are hyped by the community is exactly the way it should be. I have a couple ideas about ways to implement this system such that it doesn't take away from the importance of pp, since after all the goal of osu is still to master beatmaps; gaining skill is just part of the journey.

Skill points should not replace or become the primary ordering for scores in player profiles; if anything, either add a secondary section much like there is for #1 scores, or add it as a secondary value on the scores the same way accuracy is. Also, creating a new leader board for the global top pp plays would be brilliant for keeping pp competitive and is long overdue.

There is also the topic of seasonal rankings that players have been wanting for years. I think that seasonal rankings would be an appropriate place to use pp as the metric for ranking players, as it wouldn't need to be concerned with ensuring the rankings were strictly tied to skill. Seasonal rankings and how they would be structured is a whole other topic to delve into however, so I will save that for another time.

I also believe that this would help with player satisfaction and retention in the long run as well, since one of the most common complaints and points of frustration for players is the existence of so called 'skill walls' that make players feel like they are not improving anymore. This causes them to turn to 'farm maps' to get that sense of improvement back, until they inevitable hit another skill wall. I - and I'm sure many of you reading this - have experience with this and know that even while you may not be climbing in the current system's rankings you are still gaining skill. Getting fewer misses on plays, becoming more consistent, getting higher accuracy, improving your weaker areas of play, etc... If the official player rankings were to reflect this progress properly it would go a long way toward helping shift players' mindsets in the direction of playing and improving, rather than desperately trying to increase a number.

The biggest problem with this system however is the fact that not all mistakes are equal and with the restrictions of the pp system, it can be difficult to judge how much it should effect the reward. When the player makes a mistake in a play, whether it's a miss, 100 hit, etc... without replay data it is not possible to know what caused the miss. This isn't the main issue though, since you can just assume they made the mistake on the most difficult part; it's

usually better to assume the worst in these scenarios. But the main issue is how to do the actual deduction in that case.

To illustrate this with a simplified example, imagine you have a beatmap with only 3 hit objects. The first two are a simple slow jump, with an aiming difficulty value of 2, and the third is an much harder jump, with an aiming difficulty value of 10 making for a total weighted aiming difficulty of 11 or so. It is obvious that if you see someone made a play with 1 miss that they missed the third note and should only be awarded skill points for only the first jump, however if the performance calculator only knows the total aiming difficulty of 11, it has no way of knowing if it was made from something like a 6 and a 6 or some other set of values. The best it could do is say you missed 1 out of 3 notes and reward you two thirds of the 11 points. This would obviously make maps with difficulty spikes extremely exploitable.

To accurately award skill points the calculator must be aware of the individual difficulties of each note, which is only possible using a per-object difficulty approach like the one I described in the difficulty calculation section, as well as a database system that keeps track of the individual difficulty values, not only total star values.

Scoring

Score V1 has been the scoring system used in osu! since it was released over 10 years ago. Prior to the introduction of ppv2, total ranked score was the metric players were ranked by. Score V1 had its time to be useful and that time has long since passed, and it is time to move onto a more modern score system.

When score V2 was first introduced and the announcement was made that it may become the default scoring system for newly ranked maps, there was outrage. This response was partially because players were displeased with how the new system functioned, however the vast majority of players (myself included) were against the system taking over because it was a massive change and the usefulness of score V2 didn't make up for it. Making change happen, especially in established ecosystems, is always going to be difficult but often it's necessary to move forward.

The purpose of a score system is to rank plays on a given beatmap. Most rhythm games' keep score using a simple summation system where the accuracy of hitting a note contributes a set amount of score. Score V1 is generally very similar to this summation approach, except it also scales with combo, which has its issues. Combo scaling means that you can score far higher on a map with far lower accuracy, and objective skill that lower scored plays, which in osu! is especially a problem because only the highest scoring play is considered for pp calculation. Score V1 is also problematic for tournaments due to the nature of the combo scaling meaning a single player getting a high combo could beat an entire team of players who had a couple unlucky combo breaks.

Score V2 is different in that it is capped at 1,000,000 points of which 30% is made up of your accuracy, and 70% is made up of your max combo achieved. While this system is far superior to score V1 for tournament play due to the score limit and accuracy portion, it still can be improved.

I think it would be a step into a new stage of scoring if score was difficulty-aware. Score V1 has a 'difficulty multiplier' that is based on the beatmap settings (CS, AR, OD, HP) but that's not what I'm referring to here. Rather, I mean that the amount each completed hit object contributes to the total score should be weighted by the difficulty of that object according to the difficulty calculator, which is made possible by the difficulty calculation system I discussed in the earlier section.

Each object completed has several types of difficulty (as described in the difficulty calculator section) and are relevant to certain hit judgements. In other words, if a player hits a 50, they should be rewarded the same amount of aiming points as hitting a 300, however it would reward far fewer clicking points. There comes the problem, however, that slider breaks judged with a 100 are, from a score perspective, indistinguishable from mistimed sliders. It

would be required for there to be modifications to the way slider breaks are judged for this system work properly, or for it to simply ignore aim difficulty of slider ticks, however this is clearly undesirable since slider aim is one of the most anticipated features of difficulty calculation.

Having score be difficulty-aware would do several things. It would give players a much better idea of how much skill they displayed in a given play (even more so that the skill point system I discussed earlier could due to its limitations). It would make tournaments far more interesting as players completing a difficulty spike would be rewarded with an amount of points that accurately reflect the difficulty of the feat. It would mean that mods would reward a scaling amount of score based on the actual difficulty increase, which would make freemod tournament beatmaps much more interesting.

The way the score would be totalled would depend on how combo is desired to be included, if at all.

A score V2 style system with a max score of 1,000,000 and portions dedicated to each difficulty skill is one option.

Another is to have a 'build up' system where there is a temporary score that is added to as a player clears notes; once the player has a combo break, the score is weighted against the player's breaking combo and adds to the score. This would mean that the order of the hit objects before the break would not matter, eliminating the issue of combo unfairly boosting easy sections.

Conclusion

With the release of osu!lazer and the new website, I believe it is time to start focusing on rebuilding the core systems of the game to improve the competitive rankings and tournaments scene as well. From the difficulty calculator, to player rankings, to the scoring system, there are improvements that can be made to help in solving many of the more deeply rooted problems osu! has.

Changing the difficulty calculator to a per object and skill based system opens the door to a more accurate performance point and scoring systems.

Changing the role of performance points and introducing the skill points system to the primary metric for skill would create more accurate rankings and help with player mindset and satisfaction.

Creating a new difficulty-aware score system would make scoring more accurately display players' ability and make tournaments more interesting.

Although this essay focused on the standard game mode, the systems I discussed are just as valid for other game modes too. All that would be needed is to have a difficulty system with skills that accurately quantify difficulty in the other game modes.

The ideas in this essay are mostly the result of my experimentation and research over the past couple months, but are also inspired by discussions from around the osu! community.

Samuel Cattini-Schultz (Syrin) - February 2018