Skill, Matchmaking, and Ranking

Dr. Josh Menke
Sr. Systems Designer
Activision Publishing

GAME DEVELOPERS CONFERENCE March 14–18, 2016 · Expo: March 16–18, 2016 #GDC16

Outline

I. Design Philosophy
II. Definitions
III. Skill
IV. Matchmaking
v. Ranking

Design Values

Easy to Learn, Hard to Master

- "Begin with the end in mind" (Stephen R. Covey, 7 Habits)
- "First try to add … years of replayability" (Rob Pardo, AGC 2006)
- Skill depth adds replayability, always something new to learn
- Good skill-based matchmaking allows skill depth

Less depth makes recognizing skill harder

- Less predictable
- Harder to separate out the good players, so they get to dominate

Don't ignore the core, even if it's small

- Mass listens to the core, judges the game by their opinion

Who's your Audience?

Picture just a fun exaggeration

A reminder to think about audience

Presentation contains tools

Adapt them based on your audience

Skill, Matchmaking, Ranking

Different But Related

Skill System

- Figuring out how good players are

Matchmaking System

- Putting players together into matches
- *Might* use skill system or ranking system
- Influences skill and ranking systems

Ranking System

- Telling players how "good" they are
- *Might* use skill system

Audience Dictates Use

Skill System

What is it?

- Any method to measure player ability
- Ideally predictive

Why?

- Matchmaking
- Ranking
- Understanding players
- Understanding game skill depth

Skill Systems

Simple stats

- K/D, Score
- Matchmaking interaction issues: Can't both rank and matchmake with them

Ratings

- Predictive
- Put made up numbers on people
- 1500, 2300
- -2.5, 3.2

E.L.O.

Electric Light Orchestra

Not a Skill System

Common Misnomer

Árpád Elo

- Physicist
- Chess master
- Pre-1960 Rating System
- Great for its time
- Core model is good
  - Though not his, Thurstone (1927)
- Inefficient by today's standards
- Doesn't address all today's needs

Good Skill Rating Systems

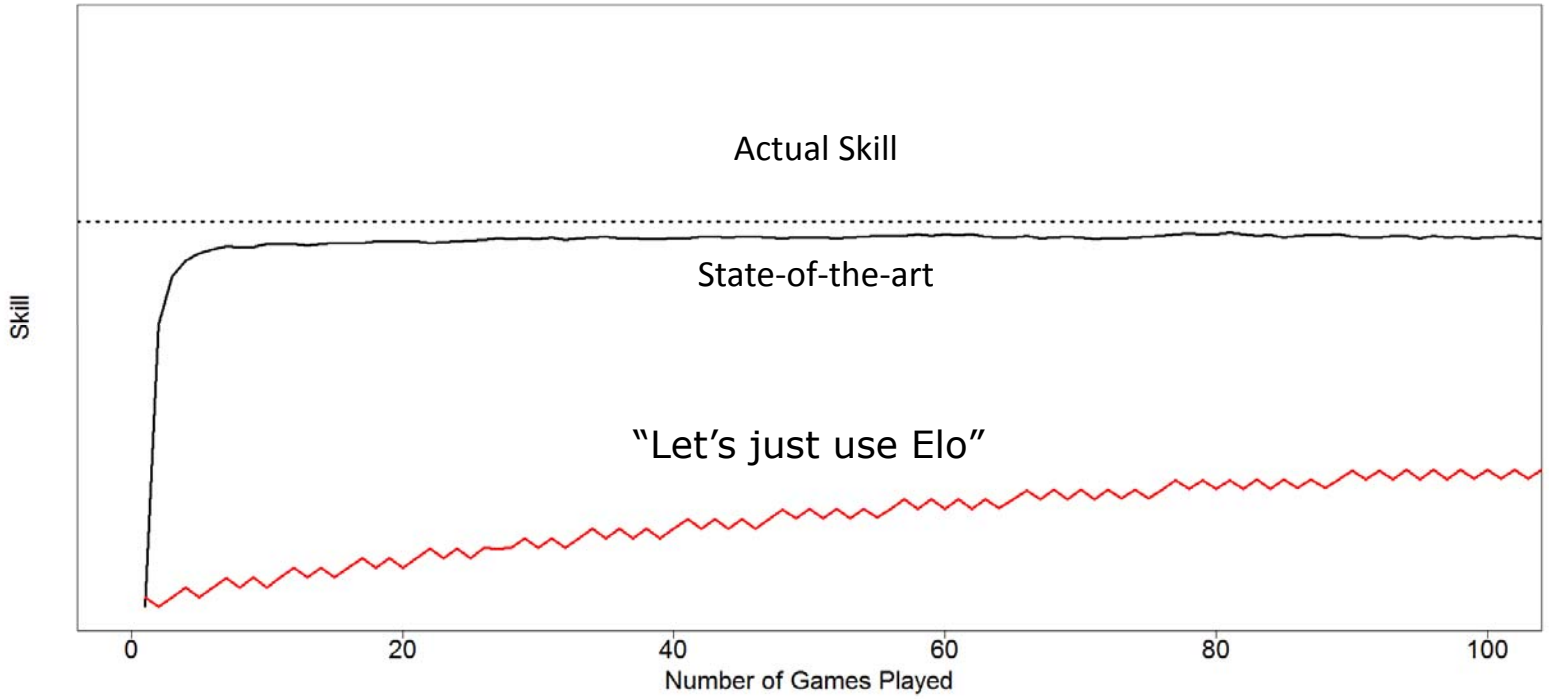
Find Player Skills Super SUPER fast

- Just ONE game is possible, can be even faster
- Players won't stay around if it takes too long

Predict match outcomes correctly

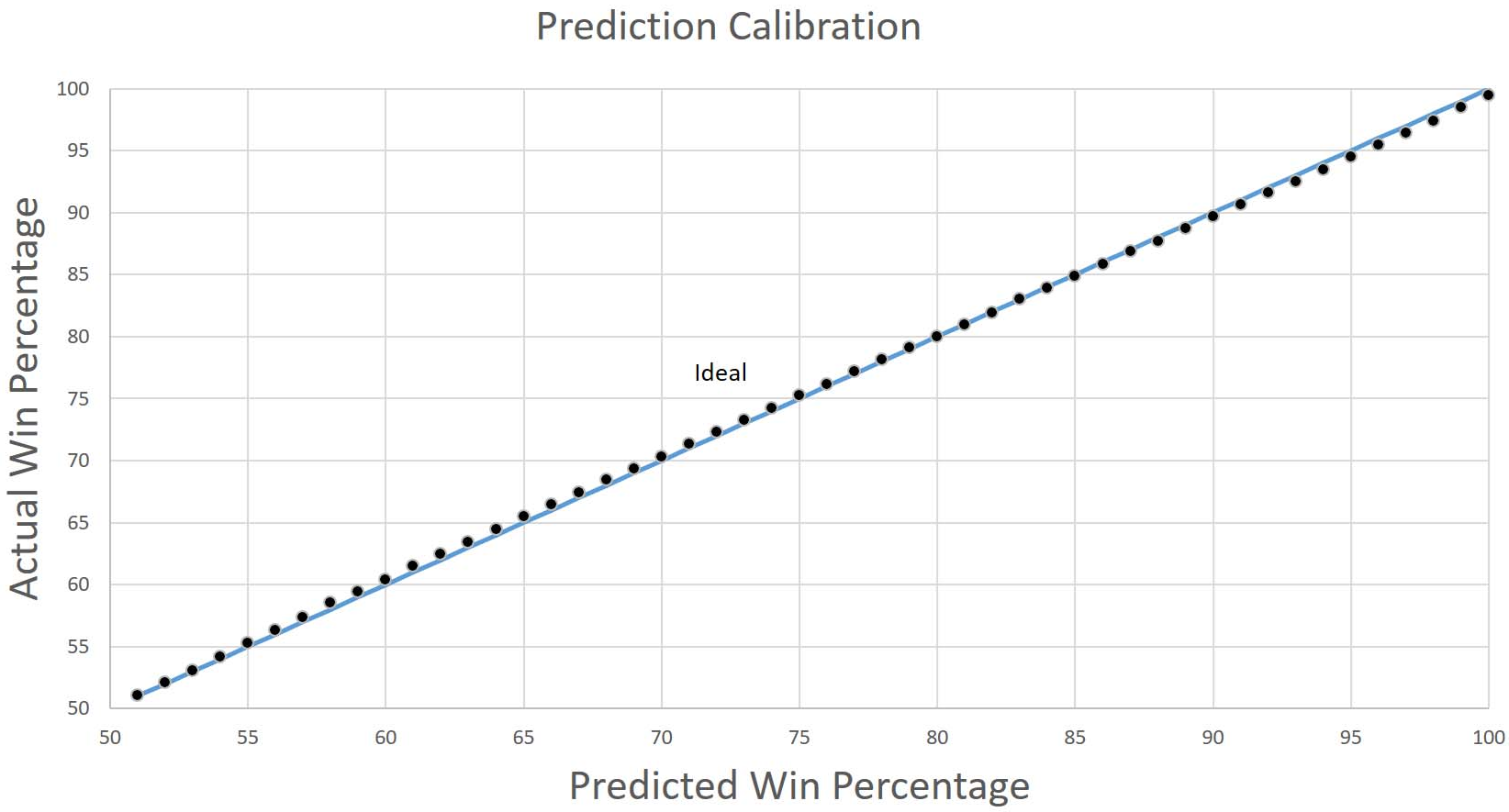
- Given the skill of two players, A and B, how often A beat B?
- Gives Probabilities: 75%? A should beat be 3 out of 4 matches
- Gives the ***RIGHT*** probability: 75% should NOT yield 9 out of 10 wins

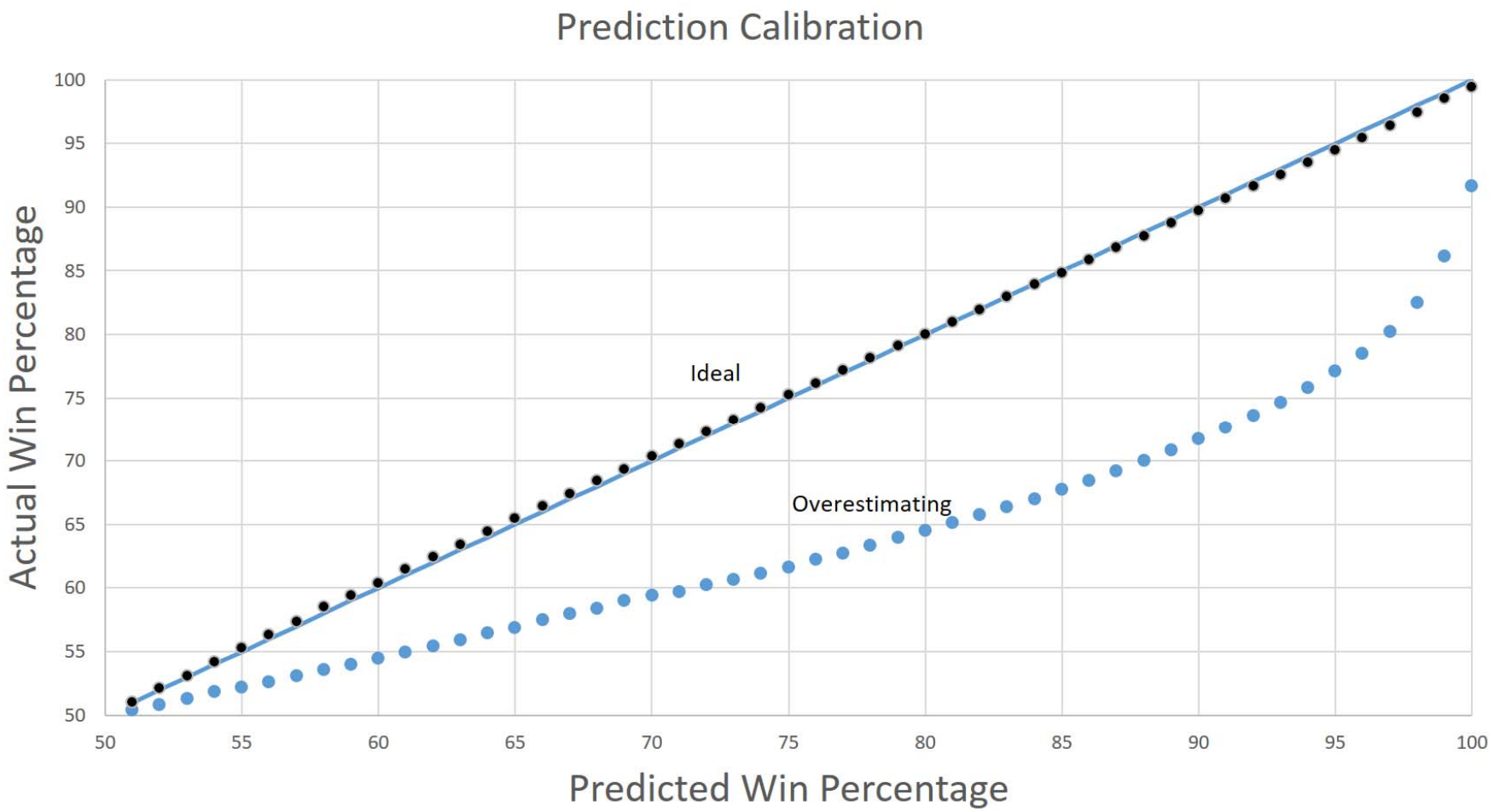
What I mean by Super SUPER Fast

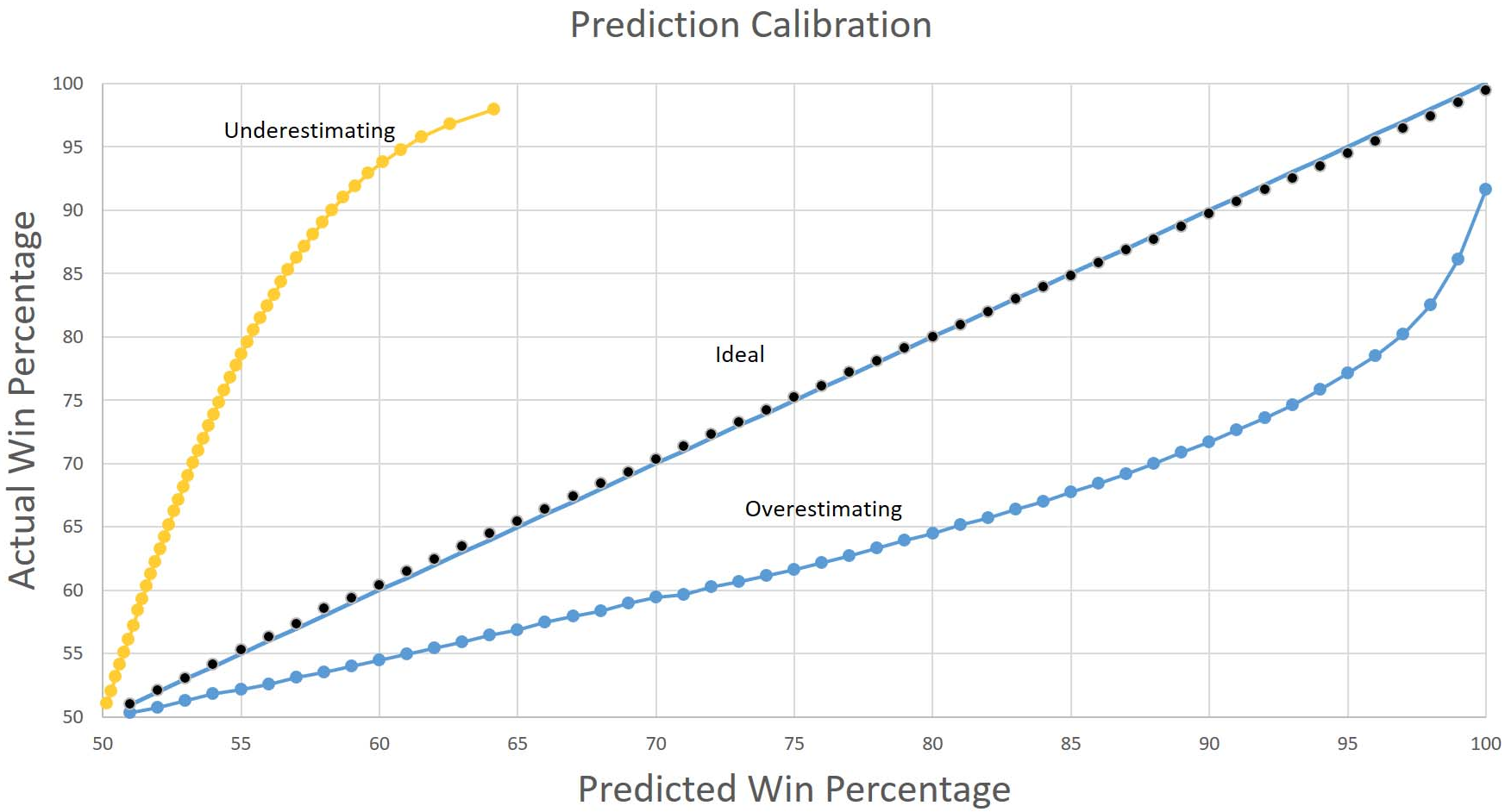


Test if it Predicts Match Outcomes Correctly

1. **Get a set of real matches that were NOT matchmade**
2. **Run your rating system after each match**
3. **Record predicted probability for each match (75%, 67%, 88%, etc.)**
4. **Count how many times:**
   - You see a prediction, in terms of the higher rated (e.g., seen 75% 100 times).
   - The higher-rated player wins (out of those 100, the higher-rated player won 74).
5. **Graph predictions vs. actuals win percentages at each prediction**


Prediction Calibration
Actual Win Percentage
Predicted Win Percentage
Ideal
100
95
90
85
80
75
70
65
60
55
50
50
55
60
65
70
75
80
85
90
95
100


Prediction Calibration
Actual Win Percentage
Predicted Win Percentage
Ideal
Overestimating
100
95
90
85
80
75
70
65
60
55
50
50
55
60
65
70
75
80
85
90
95
100


Prediction Calibration
Actual Win Percentage
Predicted Win Percentage
Underestimating
Ideal
Overestimating
100
95
90
85
80
75
70
65
60
55
50
50
55
60
65
70
75
80
85
90
95
100

Distribution over all Ratings

Create a histogram of all player ratings

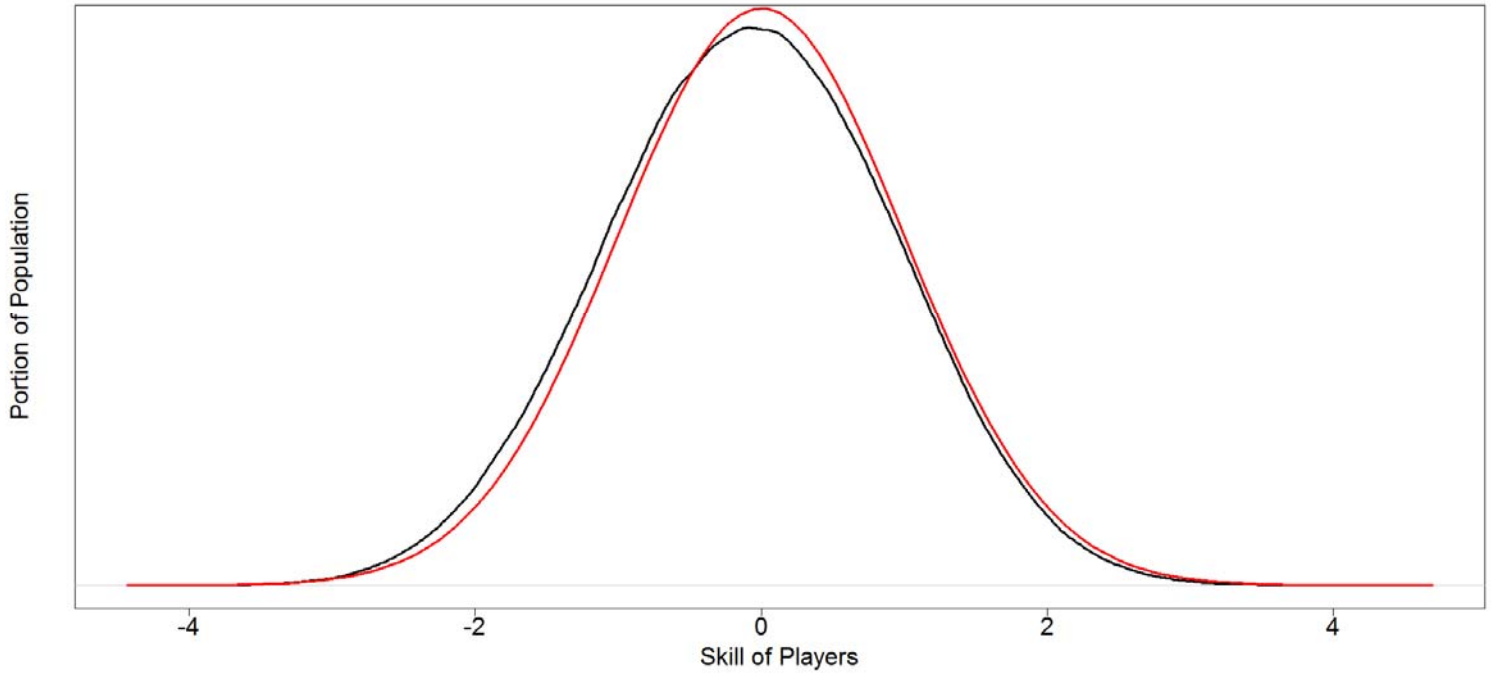
Nice if this distribution is normal (“bell curve”, also called Gaussian)

Tells you what skill looks like across all your players

If normal:

- Can use normal distribution stats to find a player’s relative rank without sorting
- Can create fake players by randomly sampling from a normal distribution
- Can create fake matches if your predictions are calibrated

Distribution over all Ratings



How do we find skill?

Like Elo's method but better

- Ratings go up on wins
- Ratings go down on losses
- Amount depends on opponent strength

Elo probably too slow

Place to start reading:

- Online Bayesian Ranking (JMLR, 2011)

Don't apply without understanding, lots of subtleties

1. Given a game result and the current $\mu_{ij}, \sigma_{ij}^2, \forall i, \forall j$. Given β^2 and $\kappa > 0$. Decide a way to set γ_q in (50)
2. For $i = 1, \ldots, k$, set
$$\mu_i = \sum_{j=1}^{n_i} \mu_{ij}, \quad \sigma_i^2 = \sum_{j=1}^{n_i} \sigma_{ij}^2.$$
3. For $i = 1, \ldots, k$,
   3.1. Team skill update: obtain Ω_i and Δ_i in (27) and (28) by the following steps.
      3.1.1. For $q = 1, \ldots, k, q \neq i$,
$$c_{iq} = (\sigma_i^2 + \sigma_q^2 + 2\beta^2)^{1/2}, \qquad \hat{p}_{iq} = \frac{e^{\mu_i/c_{iq}}}{e^{\mu_i/c_{iq}} + e^{\mu_q/c_{iq}}}, \tag{49}$$
$$\delta_q = \frac{\sigma_i^2}{c_{iq}}(s - \hat{p}_{iq}), \quad \eta_q = \gamma_q \left(\frac{\sigma_i}{c_{iq}}\right)^2 \hat{p}_{iq}\hat{p}_{qi}, \text{ where } s = \begin{cases} 1 & \text{if } r(q) > r(i), \\ 1/2 & \text{if } r(q) = r(i), \\ 0 & \text{if } r(q) < r(i). \end{cases} \tag{50}$$
      3.1.2. Calculate
$$\Omega_i = \sum_{q:q\neq i} \delta_q, \qquad \Delta_i = \sum_{q:q\neq i} \eta_q.$$
   3.2. Individual skill update
   For $j = 1, \ldots, n_i$,
$$\mu_{ij} \leftarrow \mu_{ij} + \frac{\sigma_{ij}^2}{\sigma_i^2}\Omega_i, \qquad \sigma_{ij}^2 \leftarrow \sigma_{ij}^2 \max\left(1 - \frac{\sigma_{ij}^2}{\sigma_i^2}\Delta_i, \kappa\right).$$

Matchmaking

A million players show up day 1 and want to play

Matchmaking is putting those games together

A lot of this is just good engineering

Design Still Important, Designers help decide:

- How to prioritize matchmaking: Skill? Veterancy? Social? Latency?
- What to communicate to players
- How long players should wait for a given drop in quality

Ideal Matchmaking

Designer Ideal:

- Put players into matches that are fun
- Have a planned experience with varying intensity (easy, even, hard)

Business Ideal

- Put a long term monetary value on each match for every player
- Match into matches that maximize that value over time
- Keep the most amount of players in our game having fun

Realistic Proxies

- Skill Gap: Hard to say what fun is, but we know what it isn't
- Churn: Build advanced predictive models of churn, use them

Need Fewer Buckets: Especially Low Pop

Main modes, Rotationals, One-offs

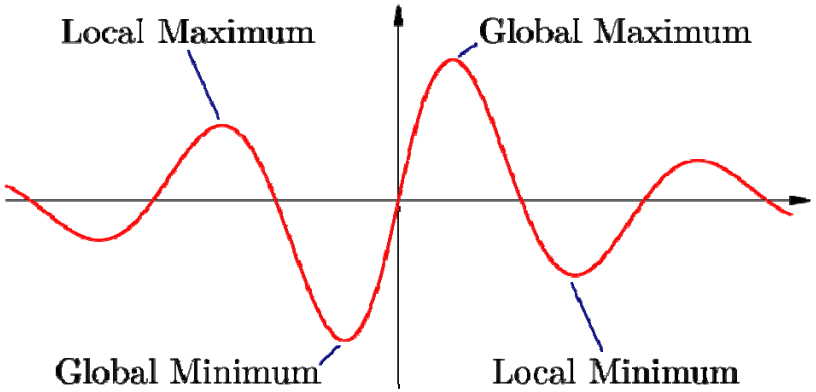
- Creates demand
- Focuses players seasonally
- Lets you try weird stuff

Thumbs up / down modes

- Sane defaults
- Matchmaker does the work
- Communicate it’s not guaranteed

Global Optimizer

1. Players (Parties) come in
2. Assigned Matches Go Out



Makes a bigger difference than you expect

- Easy to do a little envelope math and convince yourself the benefit is subtle
- In Practice, it's a much nicer impact
- Almost a necessity for good, low pop matchmaking

Hard to directly compare new to old: games don't use both at once

Simulation Testing: Since we trust model

Sample from the model

- Create fake players choosing "real" skills from a normal distribution.

Run fake players through the matchmaker

Decide who wins using the same model

- Sample from the normal each player has, the higher sample wins

Can test matchmaking, ranking, and alternative skill models

Measuring

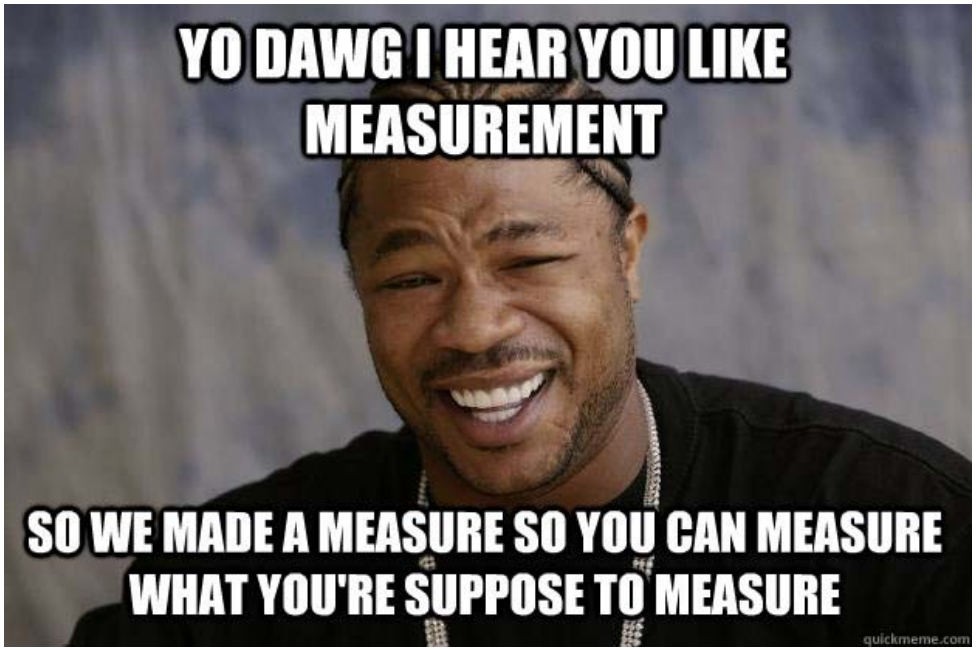
Predicting Right

- Otherwise, doesn't matter how tight we matchmake

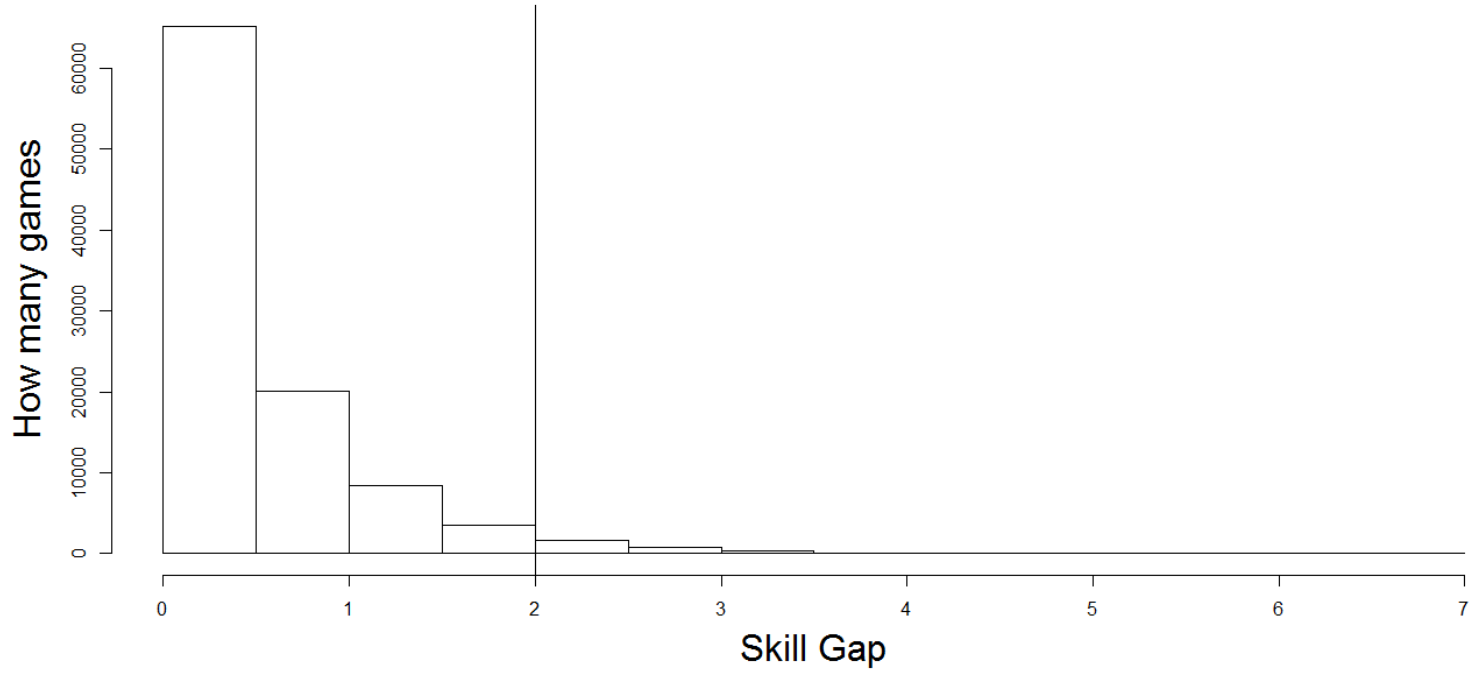
Matchmaking Tight

- Using our skill ratings right?

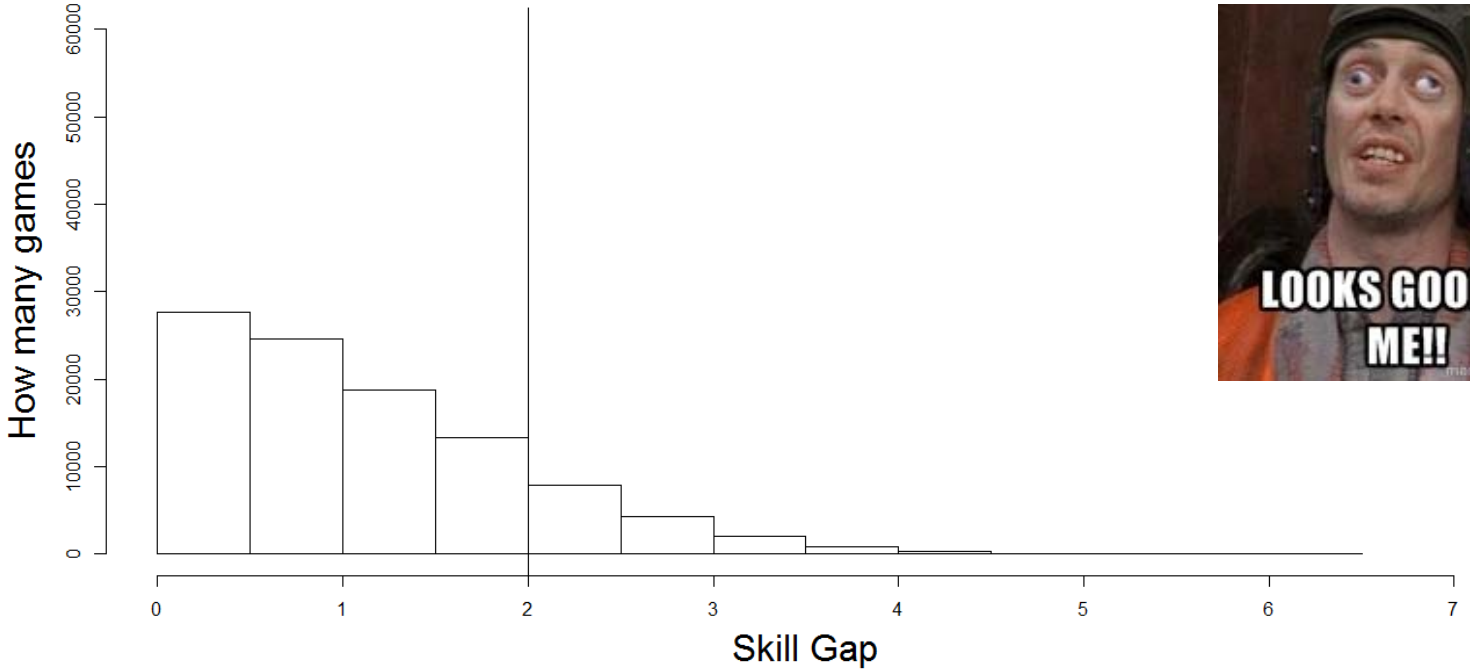
Finds bugs NOT design flaws



Matchmaking Tight



Matchmaking Tight



Tight Skill Matching Allows Depth

Common temptation in sandbox design:

1. Make a sweet new ability
2. Playtest it
3. Better players dominate the Playtest
4. Nerf or remove it since it dominates

With tight skill matchmaking

- Players that master dominating abilities get higher skill ratings
- They are matchmade away from those who haven't mastered them
- Weak players don't get dominated
- You don't have to remove cool stuff that adds skill depth, embrace it instead

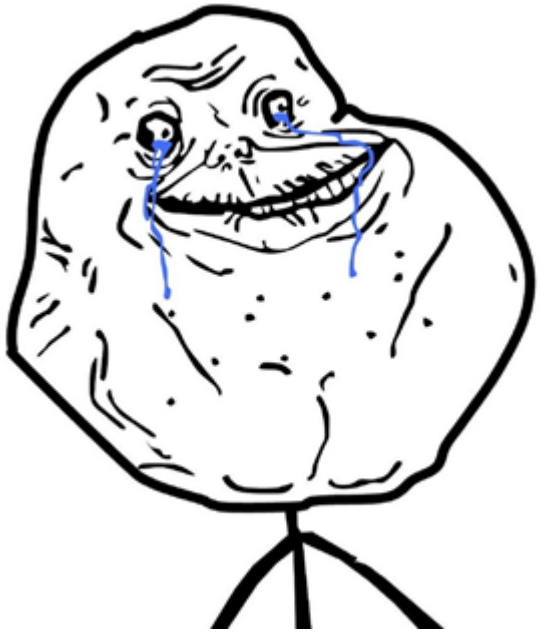
Can’t let you Play with Friends

Can’t let you Play with Friends

Can Play with Friends!

Can't Play Without Friends

1) Play with friends for a night
2) Skill inflates because of synergy
3) Play alone the next night
4) Get owned, don't know why
5) Conclude game isn't fun without friends
6) Only play when friends are on



Lose people this way

Play with Friends Solution

Track skill of friends together

Any party that queues

- create a new skill
- Estimate based on history
- Adapt quickly

Naturally fixes all these problems (doesn't fix perception)

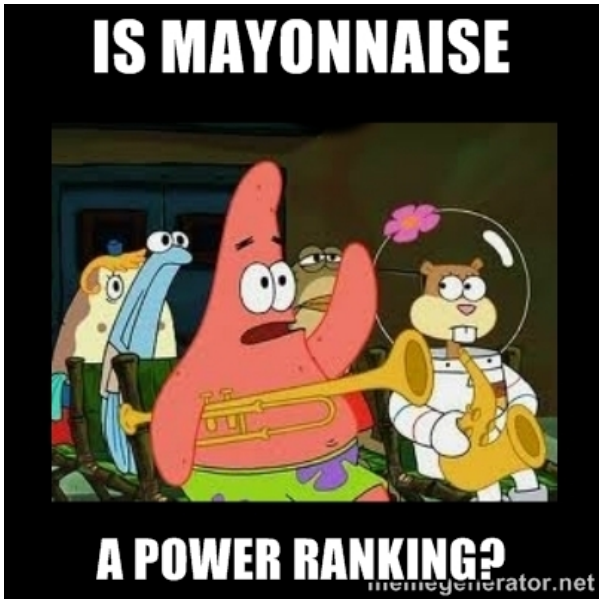
Ranking

We know how good players are, what should we tell them?

- Depends on the game.
- Progression, Hybrid, or Skill

Don't copy paste from your favorite game:

- Take inspiration
- Fully understand the mechanics
- Adapt to your own game's fiction and audience



Ranking Systems: Progression

Levels and unlocks based on purely time investment

For recreational play

Not a measure of skill, only veterancy

Pre-game / Loading Screen Implications

Ranking Systems: Hybrid

Start at the bottom like a progression system

Skill component

- usually tunable
- Ranks can be skill anchored or loose
- Losses can move you down

Good for:

- Games with no other progression system, so combine both
- Games where "competitiveness" is unknown
- Games that are all about progression and players expect it (RPGs)

Ranking Systems: Hybrids

Questionable for games that already have great progression systems

- Becomes just another system, forgettable

Ranks often meaningless except near the top

Pre-game Screens AND Matchmaking issues

Having progression is confusing when it stops

Hybrids: Skill, Rank, and Matchmaking

Visible rank separate from actual skill number

Causes visually strange match-ups:

- 1500 Rank (3.0 skill) vs. 2200 Rank (3.0 skill)
- 1500 wonders why matched vs. 2200?

Artifact of hybrid: Need to manage this

E.g. Show average team raw skill. Seems rough, but effective

Ranking Systems: Skill

Ranks tied to skill

Focus on current ability and placement rather than progression

Clearly competitive games or parts of your game (eSports focused games)

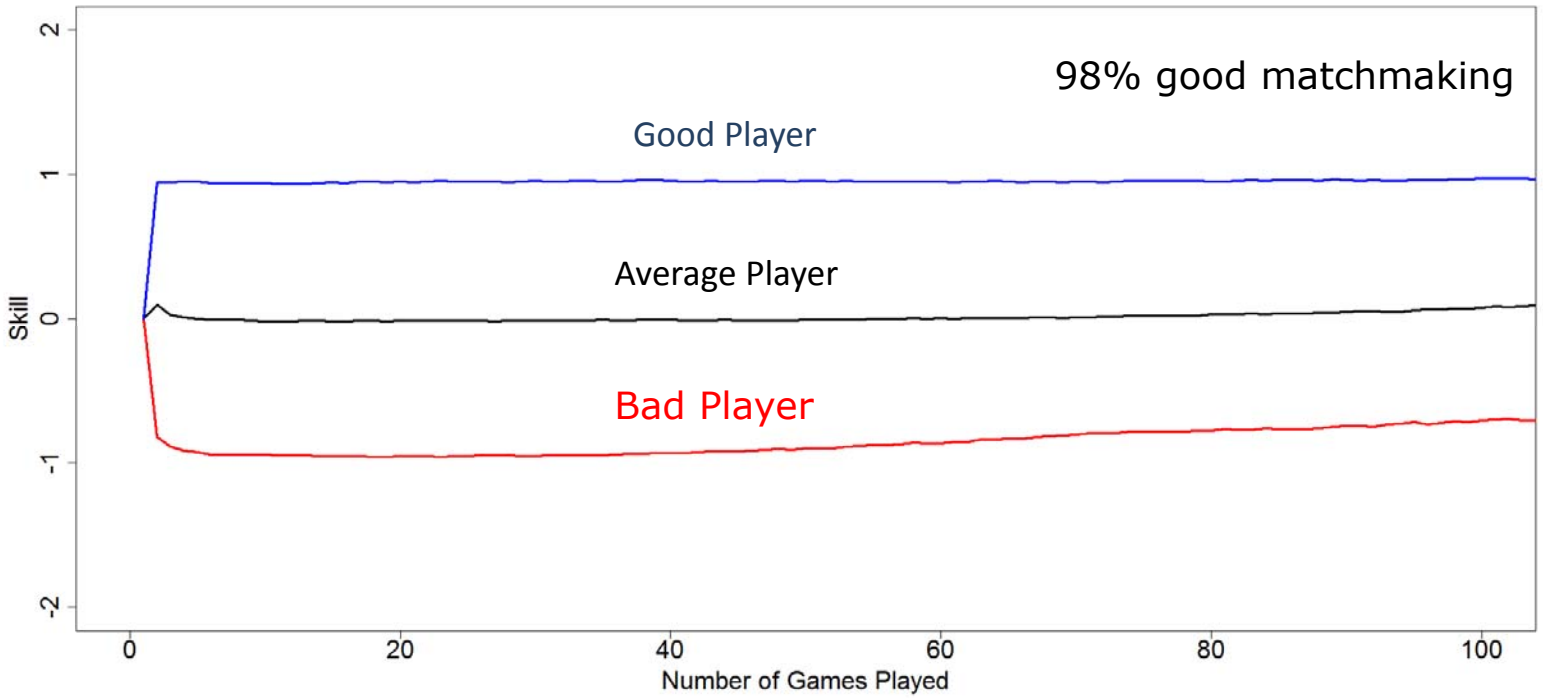
Games that already have rich progression and could use something unique

Transparency: Audience that cares about skill systems

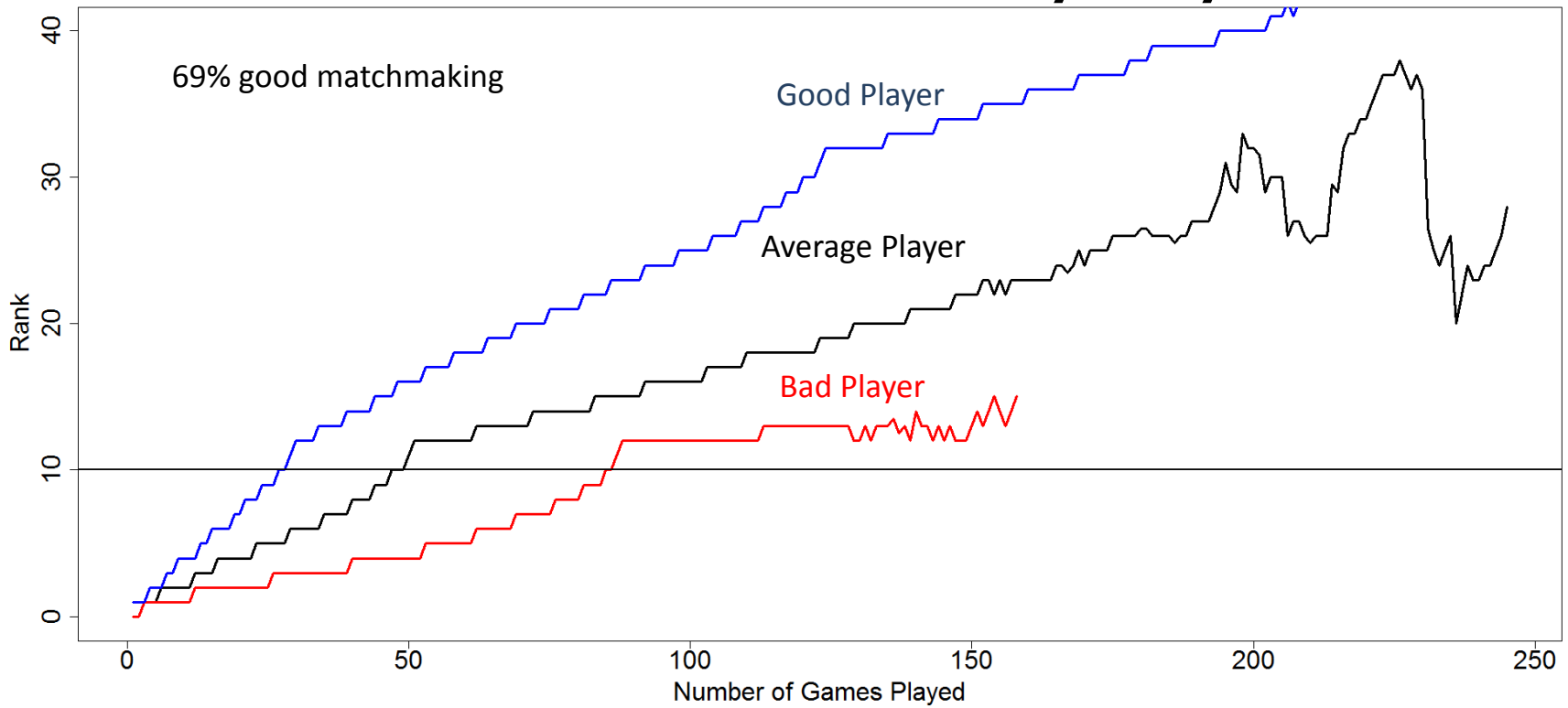
Comparing with Simulation

1. **Create a bunch of fake players (since we know how)**
2. **Matchmake them**
3. **Choose outcomes using calibrated ratings system**
4. **Update Visible Ranks**
5. **Gather results**
6. **Graph the rank experience over time for a given skill level**

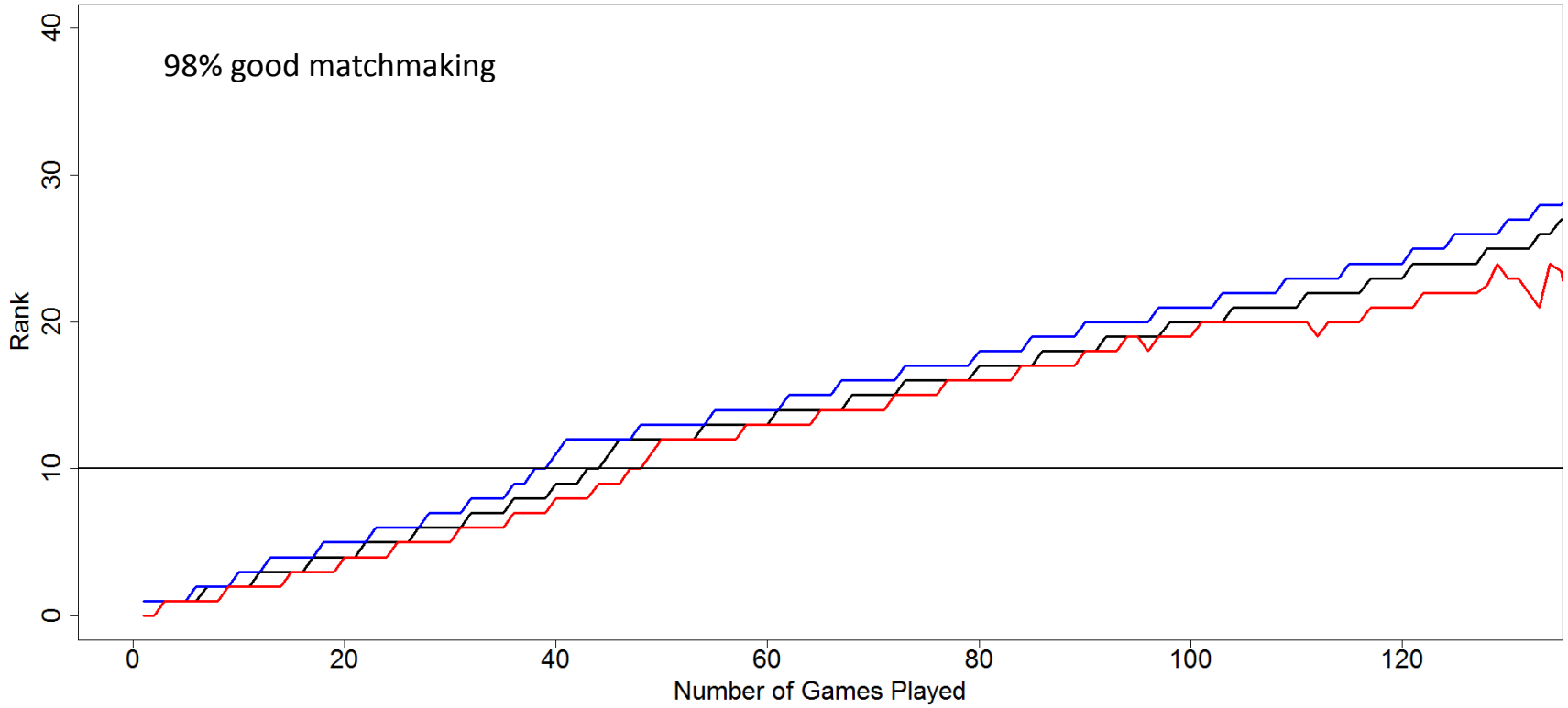
Simulation: Raw Skill for Ranking



Simulation: Inflationary Hybrid



Simulation: Hybrid with Skill Matchmaking



Let’s focus on Skill Based Ranking

We know what progression-based ranking is

Hybrid is a mix of both

So let’s talk about good attributes of a skill system

Hybridizing an exercise for designers who think their games need it

My Personal Ranking System Philosophy

If you are going Skill, go all the way

- Players who like skill will thank you
- Players that don't, don't care anyways: they'll ignore it

Link to Pros – Granted somewhat fictional

Current Skill Position, NOT progression, placement matches

- no confusion: only move up when you get better

Broad, Meaningful Ranks for Relatability

- Chunky skill target goals: 3:1 Odds between

More philosophy

High personal granularity to **see movement**

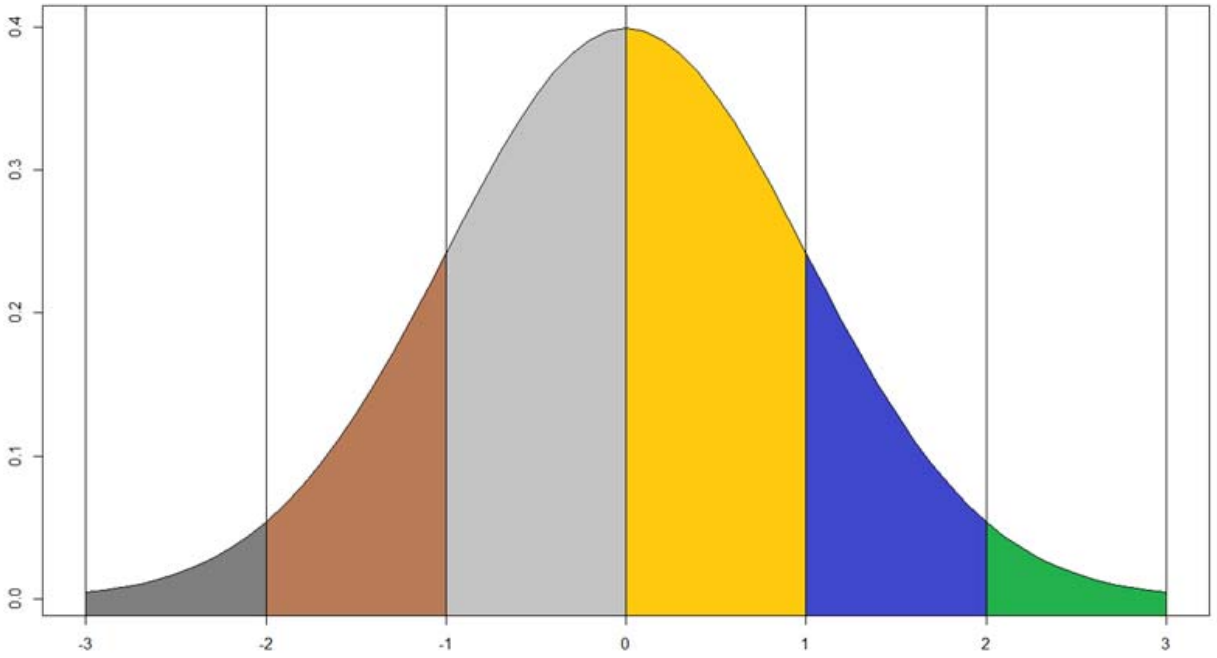
- Don’t feel level locked
- Tiers, Numbers
- For small sub goals
- Only meant for the individual, not for comparison

Endless ceiling for Pros

Transparency for the core

Skill Distribution

Portion of the population



Integrity means Moving up Feels Great!

Genuine feedback about improvement

Adds depth and replayability

1. Learn something new, or practice
2. Skill improves
3. Rank increases: Yay!

For warm fuzzies search:

promotion reaction

Thanks!

- Questions
- West Hall Overlook 2022

- twitter: @joshua_menke