Ball Skill - Anti-Exploitation & Fair Competition System

Competition Fairness Algorithm

1. Target Selection Penalty System

```javascript
// Rating gain/loss adjustment based on opponent selection patterns
function calculateExploitationPenalty(player, opponent, matchHistory) {
  let penaltyMultiplier = 1.0;

  // Age Gap Penalty (if age data available)
  const ageDiff = Math.abs(player.age - opponent.age);
  if (ageDiff > 10 && player.age > opponent.age) {
    penaltyMultiplier *= (1.0 - Math.min(0.3, ageDiff / 50)); // Up to 30% penalty
  }

  // Experience Gap Penalty
  const experienceGap = player.gamesPlayed - opponent.gamesPlayed;
  if (experienceGap > 50) {
    penaltyMultiplier *= (1.0 - Math.min(0.25, experienceGap / 500)); // Up to 25% penalty
  }

  // Losing Streak Targeting Penalty
  if (opponent.currentStreak < -3 && player.recentOpponentStreaks.filter(s => s < -2).length > 3) {
    penaltyMultiplier *= 0.7; // 30% penalty for consistently targeting struggling players
  }

  // Basketball IQ Gap Penalty
  const iqGap = player.basketballIQ - opponent.basketballIQ;
  if (iqGap > 200) {
    penaltyMultiplier *= (1.0 - Math.min(0.2, iqGap / 1000)); // Up to 20% penalty
  }

  return penaltyMultiplier;
}
```

2. Basketball IQ Integration

IQ Assessment Categories:

```javascript
```

```
const IQ_CATEGORIES = {
  RULES_KNOWLEDGE: {
    weight: 0.2,
    questions: 50,
    topics: ['fouls', 'violations', 'court_awareness', 'shot_clock']
  },
  STRATEGY_UNDERSTANDING: {
    weight: 0.3,
    questions: 40,
    topics: ['pick_and_roll', 'defensive_schemes', 'spacing', 'transition']
  },
  SITUATIONAL_AWARENESS: {
    weight: 0.3,
    questions: 35,
    topics: ['clock_management', 'foul_strategy', 'momentum_shifts']
  },
  SHOOTING_MECHANICS: {
    weight: 0.2,
    questions: 25,
    topics: ['form', 'arc', 'follow_through', 'shot_selection']
  }
};

// IQ-Adjusted Rating Calculation
effectiveRating = baseRating * (1.0 + (basketballIQ - 1000) / 2000)
// IQ of 1200 = 10% rating boost
// IQ of 800 = 10% rating penalty
```

3. Matchmaking Fairness Score

javascript

```
function calculateMatchFairness(player1, player2) {
  const factors = {
    ratingGap: Math.abs(player1.effectiveRating - player2.effectiveRating),
    experienceGap: Math.abs(player1.gamesPlayed - player2.gamesPlayed),
    ageGap: Math.abs(player1.age - player2.age),
    iqGap: Math.abs(player1.basketballIQ - player2.basketballIQ),
    streakDifference: Math.abs(player1.currentStreak - player2.currentStreak)
  };

  // Fairness score (higher = more fair matchup)
  const fairnessScore = 100 - (
    (factors.ratingGap / 10) +         // 100 rating points = -10 fairness
    (factors.experienceGap / 20) +      // 20 game difference = -1 fairness
    (factors.ageGap * 2) +             // 1 year difference = -2 fairness
    (factors.iqGap / 50) +             // 50 IQ points = -1 fairness
    (factors.streakDifference * 3)      // 1 streak difference = -3 fairness
  );

  return Math.max(0, fairnessScore);
}
```

4. Dynamic Entry Fee Adjustment

javascript

```
// Entry fees adjusted based on competition fairness
function calculateDynamicEntryFee(player, opponent, baseEvent) {
  let adjustedFee = baseEvent.entryFee;

  const fairnessScore = calculateMatchFairness(player, opponent);

  if (fairnessScore < 70) { // Unfair matchup
    // Higher-rated player pays premium for easy competition
    if (player.effectiveRating > opponent.effectiveRating) {
      adjustedFee *= (1.0 + (70 - fairnessScore) / 100); // Up to 70% premium
    }
    // Lower-rated player gets discount for facing stronger competition
    else {
      adjustedFee *= Math.max(0.5, fairnessScore / 70); // Up to 50% discount
    }
  }

  return adjustedFee;
}
```

Anti-Exploitation Database Schema

Player Behavior Tracking

sql

```sql
CREATE TABLE player_behavior_metrics (
  id SERIAL PRIMARY KEY,
  user_id INTEGER REFERENCES users(id),

  -- Target Selection Patterns
  avg_opponent_rating INTEGER,
  avg_opponent_experience DECIMAL(5,1),
  avg_opponent_age DECIMAL(3,1),
  avg_opponent_iq INTEGER,

  -- Exploitation Indicators
  weak_target_percentage DECIMAL(4,2), -- % of games vs significantly weaker opponents
  newbie_target_count INTEGER,        -- Games vs players with <10 total games
  losing_streak_target_count INTEGER,  -- Games vs players on 3+ game losing streaks

  -- Fairness Metrics
  avg_match_fairness_score DECIMAL(4,1),
  exploitation_penalty_total DECIMAL(4,2), -- Cumulative penalty applied

  updated_at TIMESTAMP DEFAULT NOW()
);

CREATE TABLE basketball_iq_scores (
  id SERIAL PRIMARY KEY,
  user_id INTEGER REFERENCES users(id),

  -- Category Scores (0-1000 scale)
  rules_knowledge INTEGER DEFAULT 500,
  strategy_understanding INTEGER DEFAULT 500,
  situational_awareness INTEGER DEFAULT 500,
  shooting_mechanics INTEGER DEFAULT 500,

  -- Overall Metrics
  overall_iq INTEGER DEFAULT 500,
  assessments_taken INTEGER DEFAULT 0,
  last_assessment TIMESTAMP,

  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);

CREATE TABLE match_fairness_log (
  id SERIAL PRIMARY KEY,
  event_id INTEGER REFERENCES events(id),
  player1_id INTEGER REFERENCES users(id),
  player2_id INTEGER REFERENCES users(id),

  fairness_score DECIMAL(4,1),
  rating_gap INTEGER,
  experience_gap INTEGER,
  age_gap INTEGER,
  iq_gap INTEGER,

  player1_penalty_applied DECIMAL(3,2),
  player2_penalty_applied DECIMAL(3,2),

  created_at TIMESTAMP DEFAULT NOW()
);
```

Basketball IQ Assessment System

Progressive Difficulty Questions

javascript

```javascript
const SAMPLE_IQ_QUESTIONS = {
  BEGINNER: [
    {
      question: "How many points is a free throw worth?",
      options: ["1", "2", "3", "Depends on the foul"],
      correct: 0,
      category: "RULES_KNOWLEDGE"
    },
    {
      question: "What's the best shooting form for consistency?",
      options: ["Low arc, quick release", "High arc, follow through", "Side spin for better bounce", "Jump as high
      correct: 1,
      category: "SHOOTING_MECHANICS"
    }
  ],
  INTERMEDIATE: [
    {
      question: "When should you use a pick and roll?",
      options: ["Only in transition", "When the shot clock is low", "To create mismatches and open shots", "Neve
      correct: 2,
      category: "STRATEGY_UNDERSTANDING"
    }
  ],
  ADVANCED: [
    {
      question: "You're down 3 with 24 seconds left. Best strategy?",
      options: ["Take quick 3-pointer", "Drive for 2, then foul", "Set up best 3-point shooter", "Call timeout to adv
      correct: 2,
      category: "SITUATIONAL_AWARENESS"
    }
  ]
};
```

IQ-Based Matchmaking Tiers

javascript

```
const IQ_MATCHMAKING_TIERS = {
  ROOKIE: { iqRange: [0, 600], description: "Learning the fundamentals" },
  STUDENT: { iqRange: [600, 800], description: "Understanding basic strategy" },
  PLAYER: { iqRange: [800, 1000], description: "Solid basketball knowledge" },
  COACH: { iqRange: [1000, 1200], description: "Advanced understanding" },
  ANALYST: { iqRange: [1200, 1500], description: "Expert-level basketball IQ" }
};
```

Implementation Safeguards

1. Graduated Penalty System

- **First offense**: Warning + 10% rating penalty
- **Pattern behavior**: 25% rating penalty + entry fee premium
- **Persistent exploitation**: Temporary matchmaking restrictions

2. Appeal and Review Process

```javascript
// Auto-review system for contested penalties
function reviewExploitationClaim(playerId, contestedMatch) {
  const playerHistory = getPlayerBehaviorMetrics(playerId);
  const matchContext = getMatchContext(contestedMatch);

  // If player generally has fair matchup history, reduce penalty
  if (playerHistory.avg_match_fairness_score > 80) {
    return "PENALTY_REDUCED";
  }

  // If this was an isolated incident in otherwise fair play
  if (playerHistory.weak_target_percentage < 0.2) {
    return "WARNING_ONLY";
  }

  return "PENALTY_UPHELD";
}
```

3. Educational Integration

- IQ quiz results unlock advanced features
- Higher IQ scores get access to strategy content

- Knowledge assessments become part of player progression
- Create educational content around fairness and sportsmanship

Business Benefits

1. Player Retention

- New players aren't immediately discouraged by being farmed
- Experienced players face appropriate challenges
- Creates sustainable skill development environment

2. Revenue Protection

- Prevents prize money concentration among exploiters
- Encourages fair play through economic incentives
- Builds trust in platform integrity

3. Community Building

- Rewards knowledge and improvement, not just exploitation
- Creates educational value beyond just shooting practice
- Builds reputation as serious basketball development platform

4. Data Insights

- Basketball IQ data helps with content creation
- Player behavior patterns inform product development
- Fairness metrics become selling point for platform quality