Title: Engagement Score as a Quality Assurance Metric

Case Study: What is Engagement Score and how can it be relevant for Quality Assurance within a product-based establishment?

Problem Statement:

A product-based tech company has recently launched an interactive quiz platform where users can engage in quiz games either against AI-powered bots or other real users. The platform's success largely depends on how well it can sustain user interest and provide a satisfying, competitive experience. Early user feedback and behavior analysis indicates that while multiplayer games are popular, a significant portion of users frequently interact with bots, especially during off-peak hours or when matching with other users takes longer.

However, the company faces a key challenge: ensuring that bot behavior aligns with user expectations to maintain or improve user engagement. If the bots are too easy, users may get bored; if too difficult or robotic, they may become frustrated. Striking the right balance in bot behavior is essential to improve user retention and platform satisfaction.

To address this issue, the company aims to:

- Use *Engagement Score metrics* to quantify how users interact with the bot games.
- Identify patterns indicating *positive or negative engagement* based on different bot behavior models.
- Modify and implement various bot behavior strategies (e.g., adaptive difficulty, human-like delay in answers, varied question types, or emotional reactions).
- Evaluate the impact of these changes by comparing *engagement scores* before and after implementation.

This case study will explore how *engagement score analytics* can be leveraged to fine-tune bot behavior, enabling the platform to deliver a more enjoyable and personalized quiz experience. The goal is to create a data-driven feedback loop between user interaction, bot behavior, and platform quality assurance, thereby enhancing the overall user experience and long-term success of the quiz platform.

Author: Shazia Parween
Shaziaparween900@gmail.com

Objectives:

1. Quantify user interaction with the quiz platform using a defined Engagement Score.
2. Model bot behavior that adapts based on user engagement, aiming to create a more interactive experience.
3. Evaluate the impact of different bot strategies: random, performance-based, and engagement-based, on overall user engagement.
4. Use Reinforcement Learning (RL) to train bots to optimize for higher engagement rather than simply maximizing win rate.
5. Determine whether engagement-focused bot behavior leads to better user retention and satisfaction.

Author: Shazia Parween
Shaziaparween900@gmail.com

An overview of the quiz platform:

The quiz starts with the players (here, user and bot) competing against one another to answer questions correctly while making use of some in-game special mechanics to sabotage the opponent. The game starts with both players on maximum health (10,000) and as the game progresses, they may lose or gain health points while maintaining a max value of 10,000. The possible actions available for players are categorized in two ways:

- Card Actions: There are 6 available cards, and user or bot may pick either one at any point in game. For the bot this choice has to be made strategically, and not at random. Once a card is played it is removed from the deck and cannot be played again. Each card represents a key function and the players can read its description at any point in game. The cards show their key abilities in main screen, so as to assist players in remembering correctly.
- Answer Actions: The answer actions are represented in the main code as an array called action. Here, action[0] represents "unallowed," which is the only available option when the active player is the opponent. This is because only the active player can answer the question, while the other player must wait for their turn. During their turn, the player can choose to answer the question or leave it unanswered. In the code, this is represented as follows: action[1] is assigned when the question is answered correctly, action[2] is assigned for an incorrect attempt, and action[3] is assigned when the timer runs out and the question is skipped. (Note: There is no manual skip button available to the players; a question can only be skipped if the timer runs out. In Bot strategy development, time delay handles how long bot takes to responds, and if it should skip the question.)

The game ends when either of the two conditions are met:

- The timer runs out. A typical Quiz game is timed at 10 minutes, with the timer starting at 10:00:00 and ending at 00:00:00. If the timer runs out without either player losing all their health, then the winner is determined by comparing the amount of health left for the two players, the player with the more health left wins the game. (Note: in case of same health, game is declared a tie and both players share reward points)
- One of the players reach the zero-health bar. The game ends and the one with zero health *loses*, leading the opponent to victory.

Author: Shazia Parween
Shaziaparween900@gmail.com

Methodology:

1. Defining Metrics

- User Engagement: User engagement is defined as a quantified value of a detection of any change in user health, and whether the user has used any playable cards in this turn. The change is detected and encoded as True(1) or False(0) value and this value is used to determine the strength of user engagement, ranging from 0 to 2. Below is the formula for user engagement that is used in this study:

  User engagement = Change in User Health (0 or 1) + Cards Used (0 or 1)

- Bot Engagement: Bot engagement is set to be dependent of user engagement. If the user has lost health, or played any cards (leading to increased engagement value) bot is encouraged with increased engagement value of its own. So we quantify bot;s engagement with change in its own health, it playing a card plus its correspondent user engagement value. The formula that comes about is:

  Bot engagement = Change in Bot Health (0 or 1) + Cards Used (0 or 1) + User Engagement

- Match Quality: A quality score to keep track of the game quality, it identifies if the game is dragging for too long and encourages the bot to end game early, or if there is no difference in the both player's health, indicating the bot is not playing competitively enough, so to encourage bot to play more competitively. Match Quality is directly referenced in the final engagement score calculation and affects bot behaviour significantly.

  Match Quality = 0.1 × (|Difference in Health| / Rounds Played)

- Engagement Score = Final formulae of the engagement score, is dependent on the three variables, user engagement, bot engagement and match quality. The formulae is represented as:

  Engagement score = (User Engagement + Bot Engagement + Match Quality) / 3

  The bot when learning through reinforcement learning is encouraged to increase this engagement score and the game progress is tracked against bot behaviour. The study further discusses the outcomes of this approach in later sections.

2. Game Structure

Both the Bot and the User start with 10,000 health points and 6 cards. On every turn, players can choose whether or not to use a card, since card usage is allowed on any turn. However, only the active player (whose turn it is) is allowed to answer the question. Health changes are determined by the correctness of the answer and the

Author: Shazia Parween
Shaziaparween900@gmail.com

consequence of any cards used, such as dealing extra damage, reducing incoming damage, or altering health directly. Cards are for one-time use and carry specific strategic functions, which are visible on the main screen to help players make informed choices. Answering actions depend on the player's turn: correct, incorrect, or timed-out responses lead to different consequences, with no manual skip option, only allowing for an unanswered question when the timer expires. The game continues until either the timer runs out (10 minutes) or one player's health reaches zero, in which case the winner is decided based on remaining health.

3, Bot Behavior Strategies Tested

To explore the impact of different decision-making strategies on bot behavior, three distinct policies were implemented and tested within the gameplay environment. Each policy guided how the bot selected actions such as card usage and answering choices, and provided a unique basis for comparing interaction patterns and system performance.

The first policy, referred to as the *Random Choice Policy*, involved the bot selecting both its cards and answers entirely at random. This policy required minimal logic and served as a control mechanism to benchmark more advanced strategies. It allowed for observation of unstructured behavior and provided a baseline for assessing how strategic decision-making could influence outcomes.

The second policy, known as the *Maximum Health Policy*, was designed with a performance-oriented objective. Under this policy, the bot selected actions that were expected to maximize damage to the user while minimizing its own health loss. The bot evaluated potential outcomes of its actions and prioritized those that would preserve its health advantage or quickly reduce the opponent's. This required the bot to be sensitive to changes in health values and to favor high-impact card effects or answer choices that contributed directly to game dominance.

The third strategy introduced was the *Maximum Engagement Policy*, which aimed to optimize the calculated Engagement Score. Rather than focusing solely on health-related outcomes, this policy guided the bot to choose actions that would contribute to a higher level of interaction between the user and the system. The bot considered factors such as whether the user had used a card or experienced a health change, and integrated these into its own engagement calculation. This setup encouraged the bot to respond in ways that maintained or amplified user activity, making engagement, not victory, its primary operational goal.

Each of these strategies was implemented within the same game environment, with consistent health values, card sets, and turn logic, ensuring that the variations in bot behavior could be attributed solely to the policy in effect.

Author: Shazia Parween
Shaziaparween900@gmail.com

Reinforcement Learning Implementation:

To enable intelligent decision-making by the quiz bot and enhance its responsiveness, a Deep Q-Learning (DQN) approach was implemented. The goal was to train the bot to select actions, a combinations of card usage and answer responses, that either improve user engagement or maximize its own chances of winning. This was achieved by designing two separate reward strategies: one based on the engagement score, which prioritizes meaningful interaction, and another based on health performance, which rewards damage dealt and penalizes health loss.

The model architecture was built using Keras and consisted of a simple three-layer neural network. The input layer encoded the current state of the game, represented by discretized values of the bot's health, the user's health, and the number of remaining cards. The output layer predicted Q-values for each of the 24 possible actions, derived from combining 6 available cards with 4 possible answer outcomes. Two hidden layers with ReLU activation functions allowed the network to learn non-linear relationships between states and actions.

Training was conducted using an ε-greedy policy, which balances exploration of new strategies with exploitation of known good actions. Experience replay was employed to stabilize learning by storing past interactions and randomly sampling batches for training, reducing correlation between consecutive samples. A target network was also maintained and periodically synced with the main network to further stabilize the Q-value updates.

The simulation environment mirrored the actual game mechanics, where the bot and user alternated turns, and cards could be used on any turn while answer actions were restricted to the active player. Health changes were driven by the correctness of the answer and the effects of any cards played. During training, the bot interacted with this environment repeatedly, receiving feedback based on the selected reward policy and updating its Q-values accordingly. Over time, this allowed the bot to learn strategies that aligned either with maximizing user engagement or optimizing win rates, depending on the chosen reward model.

Author: Shazia Parween
Shaziaparween900@gmail.com

Experimental Results:

(Note: For this analysis, timer is replaced with number of rounds played, which is fixed at 50, as we wanted to experiment with bot's time delay options without it affecting the experimental results.)

Engagement-Oriented Bot

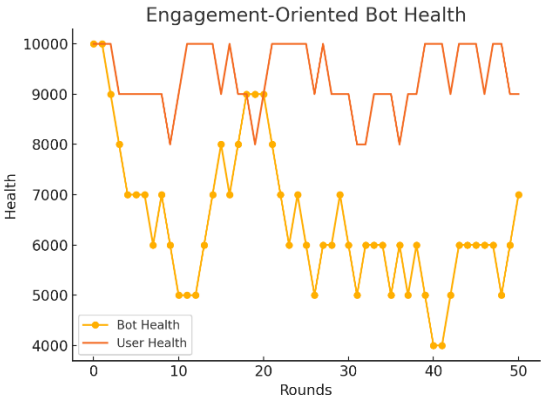

Figure 1: Bot vs User health

Shows balanced gameplay, where the bot adapts to user actions.

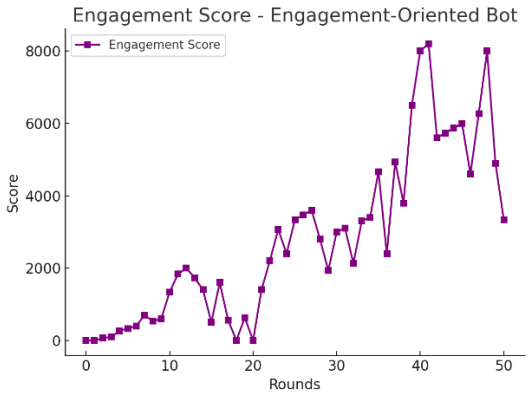

Figure 2: Engagement Score over Time

Steady increase in engagement confirms that the bot is enhancing interactivity.

Performance-Oriented Bot

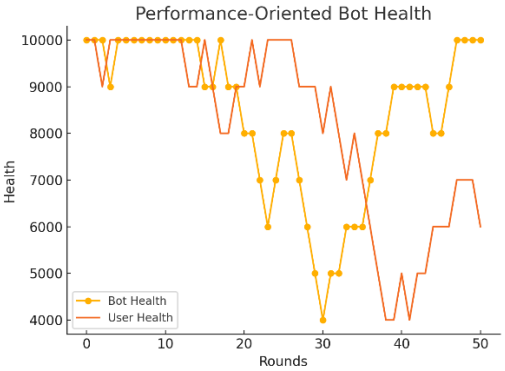

Figure 3: Bot vs User Health

Bot focuses on winning; results in greater health.

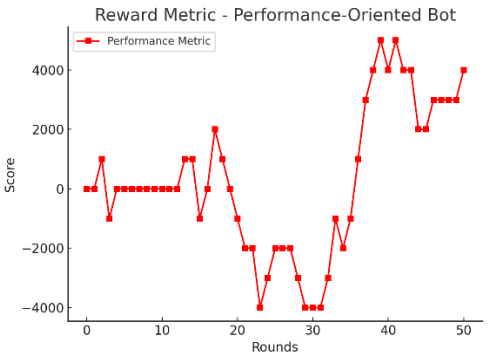

Figure 4: Reward Metric

Sharp spikes indicate high win-focused behavior but may reduce user enjoyment.

Author: Shazia Parween
Shaziaparween900@gmail.com

Results Observed:

a. When the bot operated using random strategies, engagement scores were generally low and varied widely. The gameplay lacked structure, often feeling unpredictable and unresponsive, which reduced the overall sense of involvement for the user.

b. With the maximum health reward strategy, the bot became highly focused on winning. While it was effective at reducing the user's health quickly, the engagement score remained suboptimal. The gameplay felt overly aggressive and robotic, leading to user frustration and a lack of meaningful interaction.

c. In contrast, when the bot was trained using the engagement score as its reward function, it consistently achieved higher engagement scores. The bot adapted its actions in response to user behavior, making the interaction feel more dynamic and personalized. Games became more balanced in terms of health outcomes and lasted longer, contributing to better match quality and a more enjoyable user experience.

Author: Shazia Parween
Shaziaparween900@gmail.com

Conclusions of the case study:

This case study demonstrates that engagement-oriented bot behavior improves user interaction quality more effectively than performance-focused or random strategies. By integrating Engagement Score as a reward signal in reinforcement learning, the bot can evolve into a more dynamic, adaptive opponent that responds to user behavior, enhancing overall satisfaction and platform retention.

The visualizations and simulation results highlight that while performance-focused bots can win more often, engagement-focused bots lead to a better user experience. By embedding engagement score as a core reward signal in the reinforcement learning loop, we have significantly improved the interactive quality of quiz games.

Engagement Score as an emerging QA metric:

Using the Engagement Score as a core quality assurance (QA) metric allows the company to validate and refine the bot's decision-making logic based on actual user interaction data. This ensures that improvements are grounded in real behavioral patterns rather than theoretical assumptions. It also helps direct QA efforts toward the areas where user engagement is most active, making testing more targeted and efficient. Ultimately, this approach enhances the platform's overall value by catering to both casual users seeking a smooth experience and competitive users looking for a more interactive challenge.

Author: Shazia Parween
Shaziaparween900@gmail.com