Proceedings of CONF-SEML 2026 Symposium: Multimodal Data Acquisition: Applications in Physiological and Behavioral Research
DOI: 10.54254/2755-2721/2026.AD33922

RivalCraft: LLM-Assisted Design of Personalized Racing Rivals Through Human-Readable Personality Dimensions

Yuxuan Zhou

Suzhou North America High School, Suzhou, China
zhuiye005@gmail.com

Abstract. In traditional racing games, rival characters are authored by designers during development, and players race against whoever shipped with the game. Large language models make per-player rival generation feasible, but turning that possibility into a usable tool depends on a question of shape: what kind of input should humans and the model share, so the user can specify a rival in terms they can relate to and the model can act on? RivalCraft is our attempt at an answer. A rival is specified through six named personality sliders (aggression, patience, risk-taking, sportsmanship, psychological play, consistency) layered on an archetype; these are the shared vocabulary between the user and the LLM. The model returns a structured profile that the user can edit in place. In a formative study with eight participants using SUS, a six-item perception questionnaire, and short open-ended questions, the mean SUS was 78.1 (SD = 11.9); six of eight scored above the 68-point above-average threshold. Participants rated the tool highest on alignment with the sliders (M = 4.4/5) and satisfaction (M = 4.3), and lowest on believability (M = 3.25). In open-ended answers, participants consistently preferred the rival they had themselves shaped through the sliders and described AI-only drafts as "formulaic" or "lacking humanity." We argue that the sliders work less as parameters and more as a shared language that carries user taste into the draft.

Keywords: Racing Games, Large Language Models, RivalCraf, Personalized Racing Rival

1. Introduction

In traditional racing games, rivals are designer-authored. During development, a small team writes backstories, tunes driving styles, and playtests each character; the roster that ships is the roster players get. Players pick a car. They rarely pick a rival. This fixed-roster approach limits how personal a race can feel, because whoever the player ends up dueling against was neither chosen nor shaped by them. Player engagement with in-game characters depends on recognition, alignment, and allegiance [1], and a pre-authored rival leaves little room for any of the three to form.

Large language models change what is feasible. Recent work has used LLMs for level generation [2, 3], quest generation [4], and long-running character behavior [5]. Applied to rivals, the same capability means that, for the first time, players and designers can generate rival characters on demand, in seconds, instead of waiting for a studio to author them. The drafting bottleneck is gone.

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.AD33922

The real bottleneck is about input shape, not output speed. An LLM has to be told who to write, and a free-form prompt is a poor vehicle for that instruction. It hides what the user is actually asking for, and it gives the user nothing to react to before the draft arrives. In our study, participants who read pure LLM output called it "formulaic and conventional" (P1) and "still lacking a bit of humanity" (P6). The model is capable enough; what is missing is a vocabulary that the user and the model can share to specify a person.

RivalCraft is an attempt to build that shared vocabulary. A rival is specified through six named personality dimensions (aggression, patience, risk-taking, sportsmanship, psychological play, consistency), each rendered as a slider with a plain-language description. The dimensions were chosen to be words a person would naturally use to describe another person, so the user can relate to them and form a view of the rival before the model writes anything. They are also a compact structured signal the LLM can condition on, so the model can use them. In the framing of controllable generation [6], the sliders are control codes; in the framing of human-AI co-creation [7], they are the handle through which the human contributes to the draft.

We ran a formative study with eight participants using SUS, a six-item perception questionnaire, and short open-ended questions. The paper contributes:

RivalCraft, a rival-design tool that treats LLM-backed character generation as a shared-language problem: the user and the model communicate through six human-readable personality sliders.

Formative evidence from eight participants that, under this design, the most valued rivals are the ones users feel they themselves shaped through the sliders, and alignment with the sliders is the highest-rated perception item (M = 4.4/5).

Design implications for LLM-backed character tools: the controls that do the work are the ones a person would use to describe a person, and the tool should be built around a draft-and-edit loop rather than one-shot quality.

2. Related work

2.1. Racing opponents and player immersion

Racing opponents need to feel both fair and challenging, but what turns them from filler into characters is feeling like individuals. Lankoski [1] frames player engagement with in-game characters around recognition, alignment, and allegiance; for a rival, each one has to read as someone, not as a difficulty setting. On the behavioral side, rule-based opponents still lean on pathfinding. Alali [8] describes a modified A* pipeline that balances track adherence and speed and notes that a small amount of unpredictability improves engagement. Learning-based work goes further on raw speed: Samak et al. [9] combine imitation learning and reinforcement learning and produce agents that learn from humans and then outrun them. In both lines, driving behavior alone does not make a rival memorable; style and character do. RivalCraft sits one step earlier, on specifying what that character is.

2.2. PCG and LLM-based content generation

Procedural content generation has a long history in games. Summerville et al. [10] survey the shift from search- and rule-based methods toward machine learning trained on existing content. LLMs are the latest branch: Todd et al. [2] fine-tune language models to generate Sokoban levels; Sudhakaran et al. [3] introduce MarioGPT, which produces playable Mario levels from text prompts; Värtinen et al. [4] use GPT-family models for role-playing game quests. Character profiles fit this pipeline well

DOI: 10.54254/2755-2721/2026.AD33922

because the output is small and structured, but their value depends almost entirely on personality, which the pipeline itself does nothing to surface. Our sliders are an attempt to surface it.

2.3. Controllable generation and co-creative tools

Controllable text generation is a long-running problem in NLP. Keskar et al. [6] train CTRL with explicit control codes (domain, style, task) alongside the prompt and show that structured conditioning gives the user real control over the output. HCI work on human-AI writing tools points the same way. Wordcraft [11] lets writers invoke the LLM for targeted rewrites; TaleBrush [12] lets writers sketch a protagonist's fortune as an input curve. Yannakakis et al. [7] describe this broader pattern as mixed-initiative co-creativity: the human and the system take turns shaping the same artifact. RivalCraft applies the same idea to rival profiles, with the sliders as its control codes and an editable card as its co-creation surface.

2.4. LLMs for characters and agents

Park et al. [5] show LLM-driven agents producing believable social behavior in long-running simulations. Shanahan et al. [13] argue that careful prompting produces characters with reasonably stable personalities. Compared with these, RivalCraft produces a small structured profile rather than a dialogue tree or a live agent, and its input is a fixed set of named personality dimensions rather than a free prompt. Whether that kind of input is useful is the question our study tries to answer.

3. Rivalcraft

3.1. Design goals

We set three goals for the tool. First, draft speed: a user should be able to produce a rival profile in a minute or two. Second, differentiation: any two rivals generated from different settings should feel different. Third, legibility: personality should be visible to the user in a form they can read and relate to, not hidden behind a prompt or a latent vector.

3.2. Interface

The main screen is one short flow shown in Figure 1. The user picks an archetype (for example, "The Ruthless Speedster" or "The Calculating Strategist"), adjusts the six sliders shown in Figure 2, writes a short note if they want, picks a racing goal, and clicks generate. The tool sends this specification to the LLM and renders the result as a card the user can edit inline, shown in Figure 3.

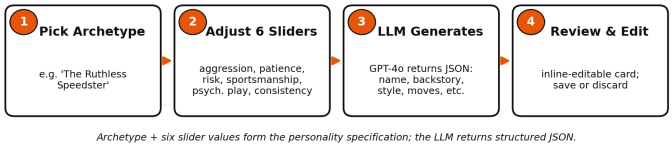

Figure 1. The four-step generation flow

The archetype and six slider values together form the personality specification the LLM conditions on; the model's structured JSON output is rendered as an editable card in step 4.

DOI: 10.54254/2755-2721/2026.AD33922

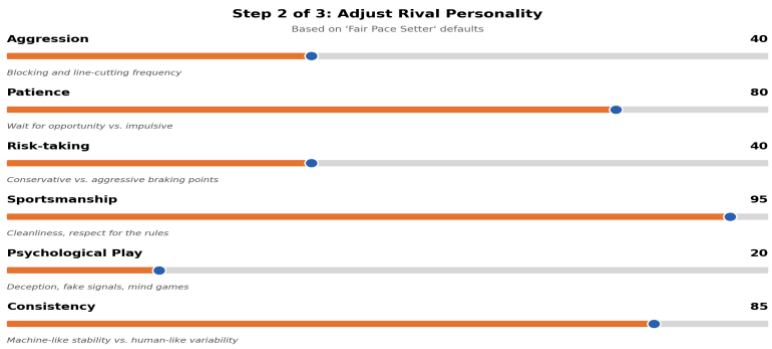

Figure 2. The six personality sliders (step 2 of the flow)

Each slider has a plain-language label and a short description shown next to the control.

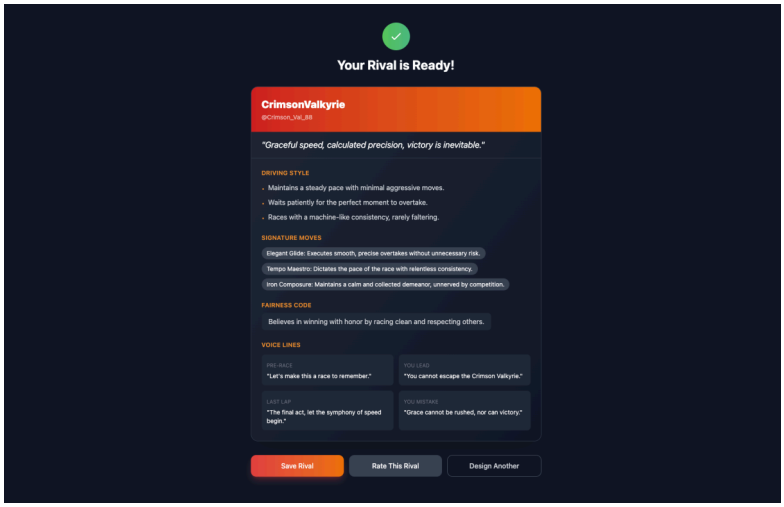

Figure 3. Generated rival card (step 4)

Every field — name, driving style, signature moves, fairness code, voice lines — is editable in place. The user can save the rival, rate it, or design another.

3.3. Personality dimensions

Six dimensions, each on a 0-100 scale with a default of 50: aggression, patience, risk-taking, sportsmanship, psychological play, consistency. They cover what most changes how a rival reads in a race. How hard they push. How long they wait. How much they gamble on a line. How they treat the player. How often they try to rattle the player. Whether their behavior stays the same across laps. Each slider ships with a short plain-language description shown next to the control, as illustrated in Figure 2, so that the user can relate the value to the kind of person it describes.

3.4. Generation flow

The flow has four steps, summarized in Figure 1.

1. Pick an archetype. Its default slider values fill the sliders.
2. Adjust the sliders. Keep the defaults or move each one individually. Together with the archetype, they form the personality specification.

DOI: 10.54254/2755-2721/2026.AD33922

3. Generate. The specification is inserted into a prompt template and sent to the LLM (GPT-4o, OpenAI API). The model returns JSON with fields for name, backstory, driving style, signature move, weakness, and rivalry dynamic.

4. Review and edit. Every field on the card is editable in place, as shown in Figure 3. The user saves the rival or discards it and tries again.

The archetype list, slider set, prompt template, and output fields are also editable through a separate settings page and persisted locally. We use this during development; it is not part of the evaluation.

4. Method

4.1. Participants

Eight participants, five male and three female. Ages: five 13-year-olds and three adults aged 26, 28, and 41 (overall M = 20.0, SD = 10.6). All reported playing racing games at least once a week. Two of the eight had prior experience with character design; the other six did not. Self-reported familiarity with AI tools was mixed (two "very familiar," two "fairly familiar," one "neutral," two reported lower familiarity). The sample is a convenience sample and is not representative of professional designers; it spans adult hobby players and younger players, which is consistent with the tool's intended use.

4.2. Procedure

Each session was about an hour. A short introduction and consent (5 min); a walkthrough of the interface (10 min); an open task in which the participant generated at least three rivals using different archetypes and slider settings, with free access to edit (25 min); SUS and the perception questionnaire (10 min); short open-ended questions in writing (10 min).

4.3. Measures

SUS. The standard 10-item SUS [14], administered on a five-point Likert scale (strongly disagree to strongly agree) and scored 0-100.

Perception questionnaire. Six items on a five-point Likert scale, each a single question rather than a subscale: (Q16) the rival has rich personality traits; (Q17) the rival feels real and trustworthy; (Q18) the rival matches what I set with the sliders; (Q19) the rival gave me ideas I would not have thought of; (Q20) the tool sparked my interest in game character design; (Q21) overall, I am satisfied with the generated rivals. This instrument is exploratory and has not been validated.

Open-ended questions. Nine short written questions covering overall experience, how the participant decided slider values, missing features, the rival that left the strongest impression, surprises or misses, differences from their own imagined designs, usefulness in a real game, what they would change, and who they would recommend the tool to. Responses were collected in writing and analyzed for recurring themes.

DOI: 10.54254/2755-2721/2026.AD33922

5. Results

5.1. SUS

The mean SUS score was 78.1 (SD = 11.9, range 60-97.5). Six of the eight participants scored at or above the 68-point threshold that Bangor et al. [15] associate with above-average usability. The two below-threshold scores came from the two participants who described wanting richer or more comparative features (an adult who wanted a comparison/management view of generated rivals, and one teen who wanted more personality variety). Figure 4 shows the per-participant distribution.

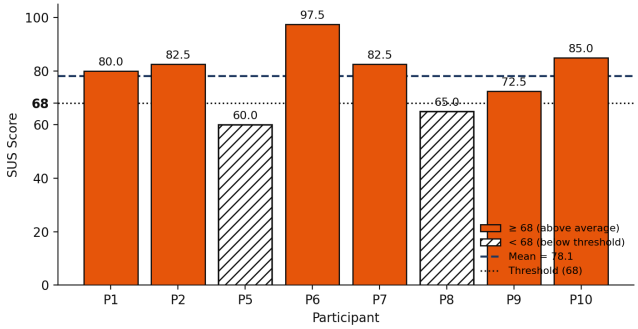

Figure 4. Per-participant SUS scores (N = 8)

The dashed line marks the mean (78.1); the dotted line marks the 68-point above-average threshold [15]. Six of eight participants scored above the threshold; the two below (P5, P8) asked for richer output or more variety.

5.2. Perception questionnaire

Item means on a 1-5 scale are shown in Figure 5. The highest-rated item was alignment with the sliders (Q18: "the rival matches what I set with the sliders," M = 4.38, SD = 0.74), followed by overall satisfaction (Q21, M = 4.25, SD = 0.71) and engagement/interest (Q20, M = 4.12, SD = 0.83). The lowest-rated item was believability (Q17: "the rival feels real and trustworthy," M = 3.25, SD = 0.89). Distinctiveness (Q16, M = 3.88) and novelty (Q19, M = 3.75) sat in between. The direction of these means is consistent across participants: alignment-with-sliders was the highest-rated item for six of eight participants, and believability was the lowest or tied-lowest for six of eight.

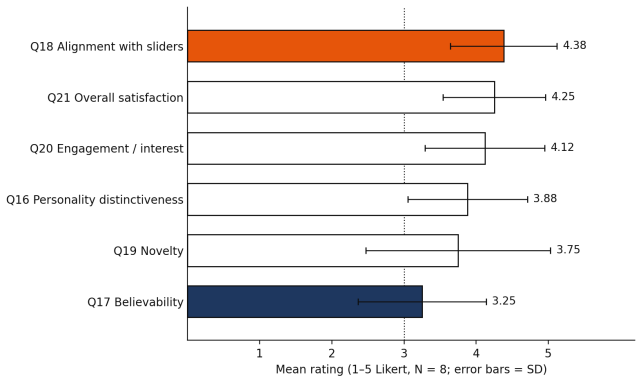

Figure 5. Perception questionnaire item means (1-5 Likert, N = 8)

DOI: 10.54254/2755-2721/2026.AD33922

Error bars are standard deviations. Alignment with the sliders (Q18) was rated highest; believability (Q17) was rated lowest.

5.3. Open-ended responses

Three themes recurred across the open-ended answers.

Theme 1: Users set slider values by taste, not by analysis. When asked how they decided slider values, participants pointed to preference and intuition. P1: "based on my own intuitive preferences." P6: "based on what I usually think." P7: "my own feeling; it fits." P8: "the way I think about opponents; intuitive." P9: "based on my own likes; fits." P10: "just what I was thinking; it fits." This fits the quantitative result that Q18 (alignment with sliders) was the highest-rated item in Figure 5: the sliders map onto what the user was already thinking, so the generated rival lines up with the user's expectation.

Theme 2: The rival people liked best was the one they had shaped themselves. When asked which generated rival left the strongest impression, four of eight participants named the one they had personally customized, explicitly citing their own slider choices as the reason. P6: "the one I generated, because I set the parameters." P7: "the one I made, because it had lots of signature moves." P9: "the one I designed, because I designed it to my own taste." P10: "the one I designed, because it matches what I had in mind." The sliders did more than specify output; they gave participants a sense of authorship over it.

Theme 3: AI drafts are specific but not fully human. Comparing AI drafts with characters they would design themselves, participants returned an ambivalent picture. The drafts were seen as concrete and usable — P2 called AI-defined roles "more specific and more grounded" than her own writing — but also as somewhat flat. P1: "pretty formulaic, conventional." P6: "still lacking a bit of humanity." P7: "not as vivid as a real person; not as cunning as the characters in my imagination." This matches the lowest-rated item in Figure 5 (Q17, believability, M = 3.25): the drafts are workable but not treated as finished characters.

Two secondary findings were worth noting. First, several participants framed the tool as a way to generate rivals they could race against rather than characters they would write into a game. P1: "useful for training." P7: "it can make me better at the game." P10: "yes, helps, because I can practice." Second, the most common feature request across participants was visual or multimedia enrichment of the card: P1 wanted a 3D/visual representation; P2 wanted video or image output; P5 wanted a commentated head-to-head between generated rivals; P7 wanted an animation at generation time.

6. Discussion

6.1. Sliders as a shared language between user and model

The central finding across our data is that the sliders work as a shared language rather than as parameter dials. On the quantitative side, alignment with the sliders (Q18) was the top-rated perception item in Figure 5, and in the "which rival made the strongest impression" question four of eight participants picked the one they had personally tuned, naming their own slider choices as the reason. On the open-ended side, every participant described their decision process as taste-driven ("based on my own intuitive preferences," "my own feeling," "the way I think about opponents"), which only works if the controls actually carry meaning a person would use about another person. The sliders do two jobs at once: they are concrete enough for the LLM to condition on, and human

DOI: 10.54254/2755-2721/2026.AD33922

enough for the user to form a view of the rival before the model writes anything. In Lankoski's [1] vocabulary, this is the recognition-and-allegiance step happening on the specification, not on the finished character. A free-form prompt does not provide that step, which is why the draft-only baseline reads to users as "formulaic" (P1) or "lacking humanity" (P6).

6.2. The gap between "grounded" and "human"

The believability item (Q17, M = 3.25) was the lowest-rated in Figure 5, and the open-ended answers give the reason. Participants read the AI drafts as concrete and easy to picture — P2 said they were "more specific and more grounded" than her own attempts — but also as formulaic and missing humanity. This gap is not a prompt-engineering problem. It is a reason to design the tool as a draft-and-edit loop rather than a one-shot generator: the model fills in the specific details quickly, and the user adds the humanity during the edit step. The card-with-editable-fields design in Figure 3 supports this, but our data suggest the edit step deserves more interface attention than we gave it. Enriched output (visual representations, animated reveal, head-to-head commentary), which was the most common feature request across participants, points in the same direction: the draft should give the user more to react to.

6.3. Sliders as prompt scaffolding, not numeric dials

Watching participants choose slider values, small numerical changes rarely changed the generated output in a visible way. What changed the output was the archetype, the description next to each slider (Figure 2), and the free-text note. The 0-100 value acted as a shorthand for "more of this kind of person." Under this reading, the slider labels and the plain-language descriptions carry most of the weight; the numerical axes carry very little. The design implication is that in LLM-backed character tools, the useful controls are the ones a person would use to describe a person, not the ones that are easy to encode as numbers. This is consistent with the conditional-generation line [6].

6.4. Limitations

Eight participants is appropriate for a formative usability read but not enough for generalization. The sample spans ages 13-41, with five teenagers and three adults; the teen subgroup drove several of the strongest results on self-authorship, and results may differ in a designer-focused sample. The perception questionnaire is unvalidated and should be treated as descriptive. We tested a single model (GPT-4o) and did not compare across models. Finally, we evaluated the design experience, not downstream play: whether one of these rivals actually makes a race more fun in-game is a separate question.

7. Conclusion

In traditional racing games, rivals are fixed at ship time; players race against whatever the designers authored. LLMs make per-player rival generation feasible, but the bottleneck shifts: the model needs a way to be told who to write, and raw prompting does not give the user a grip on the result. RivalCraft treats this as a shared-language problem. Rival personality is exposed as six named sliders that are, on the same screen, human enough for a user to relate to and structured enough for the LLM to condition on. With eight participants, the tool scored 78.1 on SUS. Participants rated alignment with the sliders highest (M = 4.4/5) and believability lowest (M = 3.25/5), and in open-ended answers consistently preferred the rival they had themselves shaped, while describing AI-only

DOI: 10.54254/2755-2721/2026.AD33922

drafts as "formulaic" or "lacking humanity." Two points follow. First, the sliders are doing translation work, not parameter work: they give the user and the model a common vocabulary for specifying a person, and that is what makes the output feel personal. Second, the right target for a tool like this is not a better one-shot draft; it is a draft-and-edit loop, with richer output to react to, that lets the user close the gap between "grounded" and "human."

References

[1] P. Lankoski, "Player character engagement in computer games, " Games and Culture, vol. 6, no. 4, pp. 291-311, 2011.
[2] G. Todd, S. Earle, M. U. Nasir, M. C. Green, and J. Togelius, "Level generation through large language models, " in Proc. 18th Int. Conf. Foundations of Digital Games (FDG '23), 2023.
[3] S. Sudhakaran, M. González-Duque, C. Glanois, M. Freiberger, E. Najarro, and S. Risi, "MarioGPT: Open-ended Text2Level generation through large language models, " in Advances in Neural Information Processing Systems 36 (NeurIPS), 2023.
[4] S. Värtinen, P. Hämäläinen, and C. Guckelsberger, "Generating role-playing game quests with GPT language models, " IEEE Transactions on Games, vol. 16, no. 1, pp. 127-139, 2024.
[5] J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative agents: Interactive simulacra of human behavior, " in Proc. 36th ACM Symp. User Interface Software and Technology (UIST '23), 2023, pp. 1-22.
[6] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, "CTRL: A conditional transformer language model for controllable generation, " arXiv: 1909.05858, 2019.
[7] G. N. Yannakakis, A. Liapis, and C. Alexopoulos, "Mixed-initiative co-creativity, " in Proc. 9th Int. Conf. Foundations of Digital Games (FDG '14), 2014.
[8] S. Alali, "AI development for racing games, " Bachelor's thesis, South-Eastern Finland Univ. of Applied Sciences (XAMK), 2025.
[9] C. V. Samak, T. V. Samak, and S. Kandhasamy, "Autonomous racing using a hybrid imitation-reinforcement learning architecture, " arXiv: 2110.05437, 2021.
[10] A. Summerville, S. Snodgrass, M. Guzdial, C. Holmgård, A. K. Hoover, A. Isaksen, A. Nealen, and J. Togelius, "Procedural content generation via machine learning (PCGML), " IEEE Transactions on Games, vol. 10, no. 3, pp. 257-270, 2018.
[11] A. Yuan, A. Coenen, E. Reif, and D. Ippolito, "Wordcraft: Story writing with large language models, " in Proc. 27th Int. Conf. Intelligent User Interfaces (IUI '22), 2022.
[12] J. J. Y. Chung, W. Kim, K. M. Yoo, H. Lee, E. Adar, and M. Chang, "TaleBrush: Sketching stories with generative pretrained language models, " in Proc. 2022 CHI Conf. Human Factors in Computing Systems (CHI '22), 2022.
[13] M. Shanahan, K. McDonell, and L. Reynolds, "Role play with large language models, " Nature, vol. 623, no. 7987, pp. 493-498, 2023.
[14] J. Brooke, "SUS: A 'quick and dirty' usability scale, " in Usability Evaluation in Industry, P. W. Jordan, B. Thomas, B. A. Weerdmeester, and I. L. McClelland, Eds. London, U.K.: Taylor & Francis, 1996, pp. 189-194.
[15] A. Bangor, P. Kortum, and J. Miller, "Determining what individual SUS scores mean: Adding an adjective rating scale, " Journal of Usability Studies, vol. 4, no. 3, pp. 114-123, 2009.