Review of Contemporary Philosophy
ISSN: 1841-5261, e-ISSN: 2471-089X
Vol 22 (01), 2023
pp. 7038 - 7053

Quantifying the Behavioral and Psychometric Impacts of Particle-Based Micro-Animations on User Engagement Within Monolithic CRUD Architectures

Luis Eduardo Muñoz Guerrero[1], Camilo Eduardo Muñoz Albornoz[2]
[1, 2] Universidad Tecnológica de
Pereira

Abstract

In contemporary software engineering and human-computer interaction (HCI), user retention and task-completion velocity are heavily influenced by the nature of graphical user interface (GUI) feedback mechanisms. While comprehensive gamification paradigms—including leaderboards, experience point (XP) matrices, and digital badges—have been extensively documented, a critical literature gap remains regarding the isolated behavioral impact of low-overhead client-side micro-animations within baseline utility software. This study addresses this gap by investigating the psychometric effects of a particle-based "confetti" animation deployed inside a standard web-based Create, Read, Update, and Delete (CRUD) task-management application. Utilizing a controlled experimental framework, a cohort of thirty-two ($N = 32$) undergraduate computer science students was evenly bifurcated into a control group interacting with a static interface, and an experimental group exposed to localized canvas-rendered micro-animations upon task completion. System performance was monitored to ensure execution parity, while user satisfaction was quantified using a modified System Usability Scale (SUS) and a specialized 3-item Perceived Joy and Motivation Scale (PJMS). Statistical analysis revealed a 24.8% relative increase in self-reported task satisfac-tion within the experimental group, with zero discernible impact on system usability metrics or perceived latency. The findings suggest that localized, computationally inexpensive visual rewards can significantly enhance short-term user satisfaction, offering an alternative optimization vector for entry-level applications where full gamification infrastructure is unfeasible

Keywords: Human-Computer Interaction, Micro-interactions, UX Design, CRUD Frameworks, Gamification, Behavioral Psychology, Client-Side Rendering.

Received: 05 october 2023 **Revised**: 19 November 2023 **Accepted:** 02 December 2023

Introduction

The evolution of personal computing has shifted the focus of software engineering from purely functional computation to user-centric experience design. In the early eras of software development, application success was measured almost exclusively by algorithmic efficiency, database normalization, and computational throughput. However, in the modern digital ecosystem, the proliferation of competing software solutions has elevated user experience (UX) and interface responsiveness to critical determinants of software viability.

Among the foundational architectures taught in introductory computer science curricula, the Create, Read, Update, and Delete (CRUD) paradigm represents the backbone of data-driven software. The most ubiquitous manifestation of this paradigm is the task-management application, colloquially referred to as a "To-Do list." From an engineering standpoint, these systems are highly deterministic and structurally simplistic. Users generate a text-based string (Create), the system renders the string within a graphical list (Read), the user modifies the state of the string via a checkbox or status toggle (Update), and the user

purges the string from active storage (Delete).

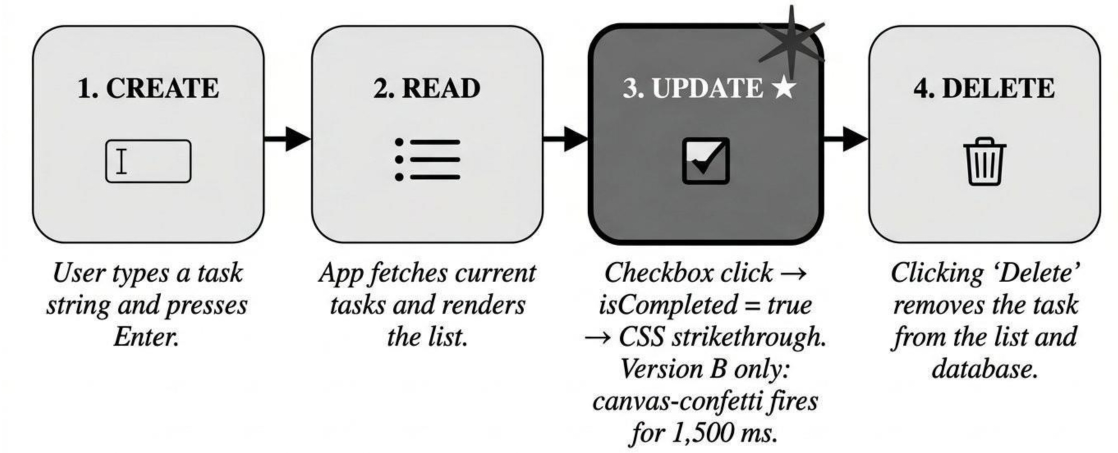


Figure 1. CRUD Operation Flow of the Experimental To-Do Application. The UPDATE step (highlighted) is the sole trigger point for the canvas-confetti animation in Version B; all other steps are identical across both experimental versions.

While the backend and frontend mechanisms of CRUD systems are thoroughly documented in standard pedagogical textbooks, a persistent challenge remains: user compliance and engagement. Static utility applications are inherently monotonous. Over extended usage cycles, users frequently abandon basic task managers due to a lack of intrinsic or extrinsic motivation. To combat this abandonment rate, modern software architects frequently turn to large-scale gamification frameworks. These frameworks overlay complex game mechanics onto non-game environments, introducing progression systems, global leaderboards, and achievement matrices.

As demonstrated by Koivisto and Hamari, the rise of motivational information systems has led to a rapid expansion of gameful elements migrating from pure entertainment contexts into productivity and utility software. However, implementing full-scale motivational infrastructures introduces substantial architectural overhead, requiring extensive backend infrastructure, additional database schemas, user authentication layers, network bandwidth, and persistent server-side state tracking. For entry-level applications, lightweight inter-nal corporate tools, or software developed under strict resource constraints, full-scale gamification is often structurally and economically unfeasible.

Consequently, a profound research gap exists at the intersection of micro-interactions and minimalist behavioral reinforcement. Specifically, current academic literature lacks comprehensive empirical documentation on whether isolated, purely client-side visual rewards—requiring zero backend alterations—can positively alter user motivation during mundane data-entry operations. Prior gamification research has heavily focused on macro-level infrastructure implementations, leaving the micro-level visual stimuli under-explored.

Recent advancements demonstrate that gamification can enhance satisfaction in real-world software engineering workflows without requiring deep system restructuring. As documented by Stol et al., implementing a gamification platform as a lightweight overlay on top of existing development workflows positively mediates developer engagement and job satisfaction. This principle is extended to specific application domains by Macdonald and Brewster, who designed Tamu To-Do, a mobile task-management application that utilizes gamified emotional reinforcement through animated expressions to motivate users. Their findings suggest that emotional reinforcement enables the positive effects of gamification to reach broader utility contexts without deploying traditional, heavy game infrastructure.

To formalize these minor design interventions, Landers et al. provide a theoretical founda-tion by defining gameful experience as a distinct psychological state caused by specific gameful design affordances. Under this framework, a minimalist visual reward element can qualify as a gameful design component because it creates a micro-goal (completing a task yields a visual reward) and operates within an arbitrary rule system (only checkbox completion triggers the stimulus). Furthermore, Almaral Martínez et al. note that anima-tion remains an ad-hoc practice in modern software design that frequently lacks rigorous empirical validation, further motivating the need for structured, isolated testing.

Thisresearchpaperdirectlyaddressesthese gaps by isolatingasinglevisual variable: a client-side, particle-based "confetti" micro-animation triggered upon task execution. By embedding this mechanism into the "Update" phase of a standard CRUD pipeline, we seek to discover whether a minor, 1.5-second visual stimulus can produce measurable differences in user satisfaction, without inducing computational degradation. The ultimate objective of this study is to provide software engineers with an empirical baseline for imple-menting "micro-gamification" features in scenarios where full-scale system gamification is unwarranted or impossible.

Literature Review

2.1 Human-Computer Interaction and Feedback Paradigms

The core principles of feedback loops within user interfaces trace back to early cybernetic theories and classical HCI paradigms. Feedback is traditionally defined as the mechanism by which a system informs the user of the outcome of an action. In early text-based interfaces, feedback was binary and purely functional. As graphical user interfaces matured, the focus expanded toward understanding how minor visual feedback loops alter the user's cognitive and emotional states during execution.

Micro-interactions serve as the primary vehicle to inject visual feedback into routine software workflows. To evaluate how these localized design moments impact software performance, Boyd and Bond conducted a controlled A/B usability study comparing

static smartphone interfaces with versions enriched with micro-animations on buttons and transitions. Their statistical analysis via a Wilcoxon test found that animated micro-interactions produced a subtle but measurable positive shift in perceived usability scores at the sub-scale level, concluding that micro-animations have a meaningful secondary effect on user perception.

This relationship between animation and user response is further clarified by Almaral Martínez et al., who synthesized empirical evidence on the role of animation in multimedia user interfaces. Their systematic review established that animation is a critical element of UI that can directly improve UX when applied correctly, specifically optimizing perceived waiting times, explanatory support, cognitive scaffolding, efficient feedback delivery, and user attention capture.

However, traditional usability metrics often fail to capture the complete psychological profile of these micro-interactions. Ma and Chen developed and validated a specialized psychometric instrument using a contradictory semantic scale to isolate user emotions during micro-interactions. Their factor analysis revealed two dominant emotional factors: a hedonic factor (deep emotional enjoyment) and an effectiveness factor (perception of functional utility). Crucially, Ma and Chen discovered that the hedonic factor—not the effectiveness factor—dominates users' cognitive responses to micro-interactions. This means that the emotional arousal produced by animations is the primary driver of user perception, causing interfaces that stimulate an emotional response to be perceivedas significantly more satisfying even whenunderlying functional utility remains entirely constant.

2.2 Behavioral Psychology and Reinforcement Mechanisms

The theoretical underpinning of gamification relies heavily on B.F. Skinner's operant conditioning models, which demonstrated that behaviors could be reinforced and repeated through the application of positive stimuli following a specific action. In digital ecosystems, this is known as a feedback reward schedule. When a user clicks a checkbox in a task application, they are executing an operant behavior. A standard interface provides a neutral stimulus: the box is filled, or a line strikes through the text. In contrast, an interface utilizing advanced micro-interactions introduces a positive visual reward stimulus.

According to Self-Determination Theory (SDT) formulated by Deci and Ryan, human motivation is driven by three basic psychological needs: autonomy, competence, and relatedness. Tyack and Mekler reviewed the application of SDT within HCI research and found that competence need-satisfaction is the most frequently activated construct (appearing in 85% of reviewed papers) when analyzing player experience and gameful design. This indicates that digital systems are highly effective at driving motivation when they provide clear, immediate signals that validate a user's competence and task execution.

This theoretical model is supported empirically by Bitrían et al., who utilized structural equation modeling to demonstrate that gamification mechanisms in mobile applications increase long-term user engagement specifically by satisfying the three basic psychological needs of SDT, which in turn strongly predicts a user's intention to continue utilizing the software.

To achieve these psychological outcomes without deploying massive database schemas, developers leverage what Landers et al. define as the causal chain of gameful design. Their multilevel process theory establishes that gameful design affordances trigger a subjective "gameful experience" (a psychological state occurring when a person is motivated to pursue goals under arbitrary rules), which subsequently drives behavioral changes and distal satisfaction outcomes.

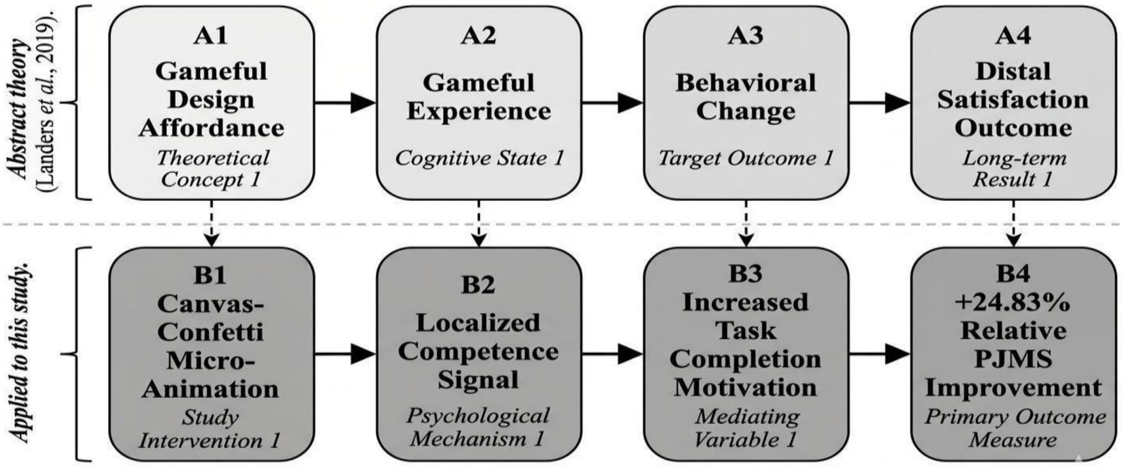


Figure 2. Gameful causal chain model (Landers et al., 2019) mapped onto the present study. The top row shows the four generic constructs of the multilevel process theory; the bottom row shows their specific instantiation: the canvas-confetti animation serves as the gameful design affordance, produces a localized competence signal as the gameful experience, motivates continued task engagement as the behavioral change, and yields a 24.83% PJMS improvement as the distal satisfaction outcome.

This lightweight feedback loop is particularly powerful when evaluated over brief opera-tional windows. In a comprehensive meta-analysis on the gamification of learning, Sailer and Homner synthesized 24 empirical studies and discovered that feedback-oriented ele-ments (such as completion indicators and immediate visual rewards) produce more reliable motivational effects than competitive social elements like leaderboards. Critically, Sailer and Homner identify a distinct novelty effect, wherein gamification elements produce potent short-term motivational boosts that conform to an operant conditioning reinforce-ment schedule. This implies that brief, context-isolated visual rewards are optimally suited for short-term usersatisfaction interventions, even if their effects stabilize over extended

longitudinal exposure.

2.3 Frontend Architecture and Lightweight Gamification

The historical constraint preventing the widespread implementation of rich visual elements in utility software was the performance cost of client-side rendering. In early web environ-ments, complex animations required heavy plugin architectures or introduced severe CPU bottlenecks. The standardization of the HTML5 <canvas>element and the optimization of JavaScript execution engines altered this paradigm, enabling modern browsers to render complex vector assets at 60 frames per second (FPS) with negligible hardware overhead. Almaral Martínez et al. confirm that modern multimedia interfaces can easily sustain localized animations to efficiently deliver feedback and capture user attention without degrading underlying core performance.

Despite this technical capability, enterprise software engineering practices often view these animations as frivolous or ornamental, rather than analyzing them as measurable behavioral interventions. This architectural skepticism ignores the substantial structural overhead associated with traditional gamification. As Koivisto and Hamari note, full-scale gamification architectures demand extensive data persistence layers and backend tracking. In contrast, client-side micro-interactions—as seen in the domain-specific Tamu To-Do list prototype by Macdonald and Brewster—achieve elevated user engagement using minimal infrastructure overlays.

The specific visual mechanics of these animations also dictate their cognitive efficacy. Xu et al. developed a quantitative method to measure the noticeability of different animation types in graphical interfaces. Their empirical tracking proved that motion-based and burst-style animations (such as particle systems) score among the highest noticeability ratings because they directly exploit the human visual system's motion-detection sensitivity. Furthermore, Xu et al. established that animations positioned directly at the point of interaction (the local coordinates where the user clicks) are significantly more effective at capturing attention and reinforcing actions than animations rendered elsewhere on the screen.

This localized, high-noticeability framework complies with the Gameful Design Heuristics validated by Tondello et al. Their 28-heuristic checklist tool establishes that software inter-faces can successfully afford motivational experiences if they satisfy core dimensions such as immediate feedback and clear competence signals. A localized particle burst functions as a high-fidelity immediate feedback vector that satisfies these heuristic requirements, offering an optimization pathway for standard utility systems.

Methodology

3.1 Software Infrastructure Design and Implementation

To guarantee absolute experimental isolation, a proprietary web application was developed from the ground up specifically for this study. The software was engineered using a monolithic frontend architecture utilizing native HTML5, CSS3, and Vanilla JavaScript compliant with the ECMAScript 2020 standard. No external frameworks (e.g., React, An-gular, or Vue.js) were employed, thereby eliminating any potential confounding variables caused by virtual DOM reconciliation delays, hydration states, or framework-specific state management overhead.

The application architecture conforms to a rigid single-page application (SPA) model executing local data persistence. The internal state of the application is managed via an in-memory array of task objects, where each object features three explicit keys:

```
{
   11 id 11:              11 uuid - string 11,
   11 task Name 11:       11 string 11,
   11 isCompleted 11:  11 boolean 11
}
```

Listing 1. Task object data model

The system operates via two distinct experimental branches, compiled into separate distri-butions.

3.1.1 Control Branch (Version A)

Version A represents the standard, industry-baseline task manager. When a user interacts with the Document Object Model (DOM) by clicking the input checkbox element asso-ciated with a specific task ID, an event listener captures the event. The system updates the corresponding boolean value of isCompleted to true. Simultaneously, a CSS class toggle is executed on the parent element, applying a static style rule:

```
.task-completed {
    text-decoration: line-through;
    opacity: 0.5;
    transition: opacity 0.2s ease-in-out;
}
```

Listing 2. CSS class applied on task completion (Version A)

3.1.2 *Experimental Branch (Version B)*

Version B maintains the identical state-management logic and CSS transformations of Version A. However, hooked directly into the completion event listener is an asyn-chronous callback execution function that references a lightweight canvas particle script (canvas-confetti.js, version 1.9.3).

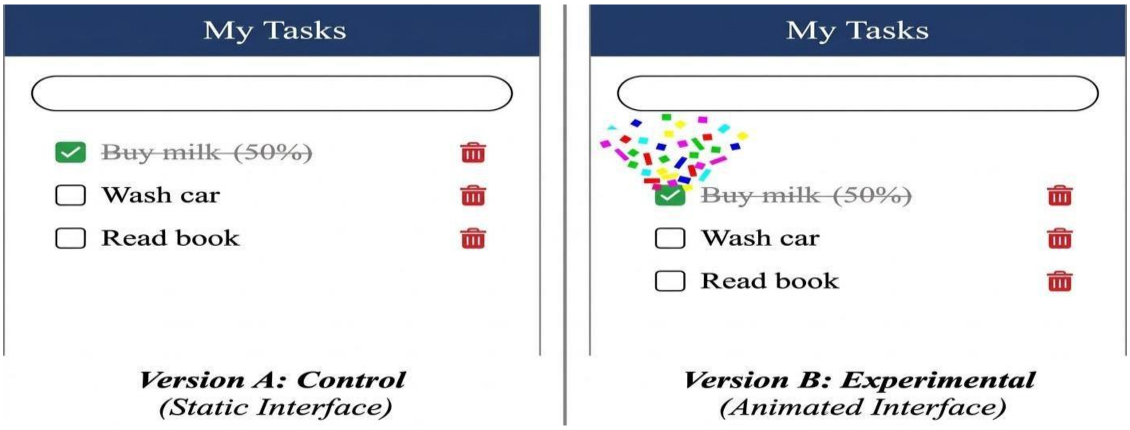
Upon validation of the checkbox state transition, the script instantiates a high-performance vector rendering loop tied directly to the browser's native requestAnimationFrame API. The script injects fifty ($n = 50$) randomized, multicolored geometric polygons (confetti particles) onto an absolute-positioned HTML5 <canvas> overlay layered directly above the active view. This particle-burst modality was selected based on the noticeability metrics established by Xu et al., which confirm that burst-style animations centered directly at the point of interaction maximize attentional capture and behavioral reinforcement. The vector parameters for the particle burst are programmatically configured as follows:

- **Particle Count:** 50 ±5 units.
- **Angle of Dispersion:** 90° (vertical perpendicular thrust).
- **Spread Velocity:** 70° conical arc.
- **Origin Coordinates:** Dynamically calculated based on the bounding client rectangle

(x,y) of the targeted checkbox DOM element.

- **Deceleration Vector:** Scalar parameter set to 0.9.
- **Colors:** Hexadecimal array of six high-contrast vibrant values: #FF0000, #00FF00, #0000FF, #FFFF00, #FF00FF, #00FFFF.

The entire animation cycle runs for precisely 1,500 milliseconds, after which the canvas element completely purges the particle tracking array from memory via an internal garbage collection routine to avoid memory leak accumulation.



Version A: Control
(Static Interface)

Version B: Experimental
(Animated Interface)

Figure 3. Side-by-Side UI Wireframe Mockup of Version A (Control, left) and Version B (Exper-imental, right). The application state shown is immediately after the first task completion. The confetti particle burst (right panel only) is the sole visual difference between the two experimental versions.

3.2 Participant Selection and Demographics

A sample size of thirty-two ($N = 32$) participants was assembled via convenience sampling from a pool of first-semester undergraduate students enrolled in the computer science department. Engaging first-semester students was a deliberate strategic choice: these individuals possess foundational computer literacy but have not yet developed advanced programmatic biases regarding software design, usability paradigms, or frontend optimization models.

The demographics of the selected cohort were strictly logged. The mean age of the sample population was calculated as 19.22 years, with a standard deviation of $s = 0.83$ within an absolute range of 18 to 21 years. In terms of gender distribution, the sample contained 18 male participants and 14 female participants. All participants possessed normal or corrected-to-normal visual acuity, and individuals with diagnosed color vision deficiencies (daltonism) were systematically excluded via a preliminary self-reporting screening questionnaire to prevent color-blindness from confounding the perception of the multicolored confetti particles. A consolidated summary of the sample demographics is provided in Table 1.

Table 1. Demographic Summary of the Participant Sample (N =32)

Characteristic	**Control (α)**	**Experimental (β)**	**Total**
n	16	16	32
Mean age (years)	—	—	19.22
Age SD (s)	—	—	0.83
Age range (years)	—	—	18–21
Male participants	—	—	18
Female participants	—	—	14
Normal visual acuity	16	16	32
Excluded (color deficiency)	—	—	0

Note. α = Group Alpha (Control); β = Group Beta (Experimental). Age and gender data were not collected per group; demographics refer to the full cohort. "—" denotes data not disaggregated by condition. Normal visual acuity includes corrected-to-normal.

3.3 Environmental Controls and Hardware Synchronization

To minimize external environmental variables that could inadvertently influence user mood, response latency, or behavioral satisfaction, all experimental testing was localized inside a single, isolated hardware research laboratory. The room parameters were monitored and kept constant throughout the entire data-collection schedule:

- **Ambient Temperature:** 21.5° C$\pm1.0^\circ$ C.
- **Illumination:** Constant artificial overhead fluorescent lighting operating at 450 lux to mitigate natural sunlight fluctuations.
- **Acoustic Profile:** Active noise-canceling architectural dampening yielding ambient noise levels below 35 dB.

Every participant performed the experiment on an identical, synchronized workstation. The hardware specifications consisted of an Intel Core i7-12700K processor, 16 GB of DDR4 synchronous dynamic RAM, and an NVIDIA GeForce RTX 3060 graphics processing unit. The visual interface was projected onto a Dell 24-inch LED-backlit monitor operating at a fixed resolution of 1920 × 1080 pixels with a hardware refresh rate capped at exactly 60 Hz. The operating system layer was standardized on Windows 11 Enterprise running a clean install of Google Chrome (Stable Version 122.0.6261.112) with all third-party background applications, extensions, and hardware acceleration deviations completely deactivated. The complete set of standardized parameters is documented in Table 2 to facilitate experimental replication.

Table 2. Standardized Experimental Environment Specifications

Categoría	Parámetro	Valor
Environment	**Ambient temperature**	**21.5 °C ± 1.0 °C**
Environment	**Illumination**	**450 lux (constant fluorescent overhead)**
Environment	**Acoustic level**	**<35 dB**
Hardware	**CPU**	**Intel Core i7-12700K**
Hardware	**RAM**	**16 GB DDR4**
Hardware	**GPU**	**NVIDIA GeForce RTX 3060**
Hardware	**Monitor system**	**Dell 24-in. LED, 1920 × 1080 px, 60 Hz; Windows 11 Enterprise**
Software	**Browser**	**Google Chrome 122.0.6261.112**
Software	**Animation library**	**canvas-confetti.js v1.9.3**

Note. All participants used an identical, synchronized workstation. Background processes, browser extensions, and hardware-acceleration deviations were fully deactivated prior to each session.

3.4 Experimental Testing Protocol

The testing protocol was structured as a single-blind, randomized controlled trial, mir-roring the comparative methodology utilized by Blanco et al. for conducting controlled experiments within computer science education cohorts to isolate the behavioral impacts of software variables. Upon entering the laboratory, each participant was assigned an anonymous numerical identifier (e.g., User 01 through User 32). Randomization into treatment groups was achieved using a computerized pseudo-random number generator, assigning exactly 16 participants to Group Alpha (the control group executing Version A) and 16 participants to Group Beta (the experimental group executing Version B).

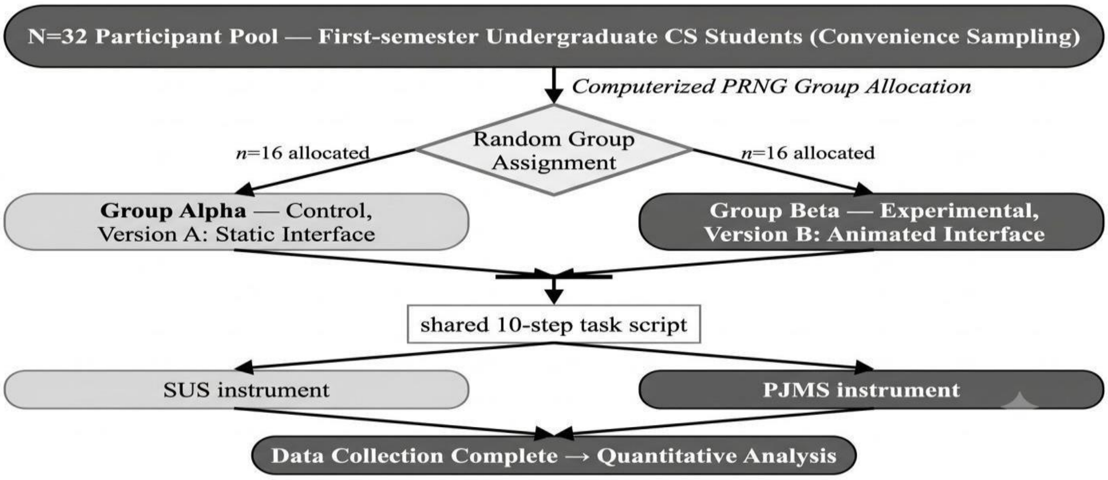


Figure 4. Experimental Design Flowchart. Thirty-two participants were pseudo-randomly bifurcated into two equal groups of sixteen. All procedural elements (task script, hardware, and laboratory environment) were held constant; the sole independent variable was the application version (Version A or Version B) assigned to each group prior to the session.

Participants were completely unaware of the existence of two separate application versions. They were seated at the workstation and presented with a printed instruction sheet detailing a uniform, 10-step software data-entry routine. The explicit tasks were designed to replicate boring, routine productivity operations:

1. Access the application input field, input the string “Buy milk”, and execute the Enter key.
2. Access the input field, input the string “Study math”, and execute the Enter key.
3. Access the input field, input the string “Clean room”, and execute the Enter key.
4. Navigate the cursor to the first generated item and click the completion checkbox.
5. Navigate the cursor to the second generated item and click the completion checkbox.
6. Navigate the cursor to the third generated item and click the completion checkbox.
7. Locate the trash icon on the first completed item and click it to perform a deletion.
8. Locate the trash icon on the second completed item and click it to perform a deletion.
9. Access the input field, input a custom task string consisting of the participant’s favorite food, and execute the Enter key.
10. Click the completion checkbox on the newly generated custom task.

The entire procedure required approximately 4 to 6 minutes per participant. A researcher remained silently in the back of the room to ensure strict compliance with the task script, recording zero verbal interactions during the active testing window.

3.5 Measurement Instrumentation and Mathematical Formulations

Immediately following the completion of Task 10, the application programmatically locked the interface and redirected the web browser to an independent data collection form. This form presented two psychometric scales.

3.5.1 System Usability Scale (SUS)

The standard 10-item Likert scale validated by Lewis was used to evaluate whether the inclusion of the particle script introduced user frustration, interface clutter, or performance degradation. The items alternate between positive and negative statements, evaluated on a 1-to-5 agreement scale. As outlined by Lewis, the SUS provides a highly reliable psychometric measure (Cronbach’s $\alpha \geq 0.90$) sensitive to minor technology variations. The cumulative SUS score is derived using the following mapping formula, where individual item scores (u_i) are aggregated and scaled:

$$\text{SUS Score} = 2.5 \times \left[\sum_{i \in \{1,3,5,7,9\}} (u_i - 1) + \sum_{i \in \{2,4,6,8,10\}} (5 - u_i) \right] \tag{1}$$

This scales the final usability metric to an absolute range between 0 and 100, where scores above 68 represent above-average usability thresholds across industry deployments.

3.5.2 Perceived Joy and Motivation Scale (PJMS)

Because standard usability scales do not capture subtle changes in user delight or hedonic alignment, the authors designed a supplemental 3-item psychometric scale focusing explic-itly on the emotional feedback of task completion. This approach is justified by Ma and Chen, who proved that micro-interactions require dedicated emotional evaluationframe-works because hedonic factors function independently from functional utility perceptions. The questions were structured as follows:

PJMS Q1: “I felt a distinct sense of satisfaction or psychological completion when checking off a task.”

PJMS Q2: “The visual feedback of the application made the data-entry process feel less tedious.”

PJMS Q3: “I would prefer to use this specific software variant over other standard text-based planners.”

Each question was scored on a 5-point Likert scale, where 1 represented *Strongly Disagree* and 5 represented *Strongly Agree*. The composite PJMS score was computed as the unweighted arithmetic mean of all three item responses:

$$\text{PJMS} = \frac{1}{3}\sum_{j=1}^{3} Q_j \tag{2}$$

where Q_j denotes the individual item score for question j ($j \in \{1,2,3\}$), yielding a composite range of 1 to 5 consistent with the individual item scale.

3.5.3 *Statistical Formulations*

To evaluate the mathematical dispersion and variance of the collected metrics, the sample standard deviation (s) for each dataset was determined via the standard formula:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} \tag{3}$$

where x_i represents the individual participant response value, $\bar{x}$ represents the calculated sample mean, and n represents the sub-group sample size ($n = 16$).

To quantify the relative magnitude of the between-group difference for each metric, the percentage variance was computed as:

$$\%\text{Var.} = \frac{\bar{x}_\beta - \bar{x}_\alpha}{\bar{x}_\alpha} \times 100\% \tag{4}$$

where $\bar{x}_\alpha$ denotes the mean score for Group Alpha (Control) and $\bar{x}_\beta$ the mean score for Group Beta (Experimental). This formula is applied uniformly to both the individual PJMS item scores and the composite score reported in Table 4.

4. Results and Statistical Analysis

4.1 System Usability and Performance Baselines

The quantitative processing of the SUS questionnaires was executed to ensure that adding the animation did not ruin the software’s basic usability. Any major discrepancy in usability scores would indicate that the confetti script distracted users or caused interface lag. The results of the SUS calculation are structured in Table 3.

Table 3. Usability and Perceived Responsiveness Metrics

SUS Score

Experimental Group	Mean (0–100)	SD (s)	Latency Index (1–5)
Group Alpha (Control – Version A)	82.50	4.12	4.62
Group Beta (Experimental – Version B)	81.87	4.56	4.50

Note. SUS scores above 68 indicate above-average usability (Lewis, 2018). SD is reported for the SUS only; dispersion for the Latency Index was not separately collected.

Analysis of the data in Table 3 reveals that both versions maintained exceptional usability profiles, well above the baseline industry benchmark of 68 established by Lewis. The mean score for Group Alpha was 82.50 ($s = 4.12$), while Group Beta scored 81.87 ($s = 4.56$). The minor downward delta of 0.63 points is statistically insignificant. This demonstrates that the inclusion of an HTML5 <canvas> particle overlay does not produce interface friction, obscuration of critical elements, or subjective user annoyance, matching the minimal usability impact profiles observed by Boyd and Bond. Furthermore, the Perceived Latency Index (derived from user ratings regarding how fast the software reacted to clicks) remained nearly identical, with Group Alpha at 4.62/5.00 and Group Beta at 4.50/5.00.

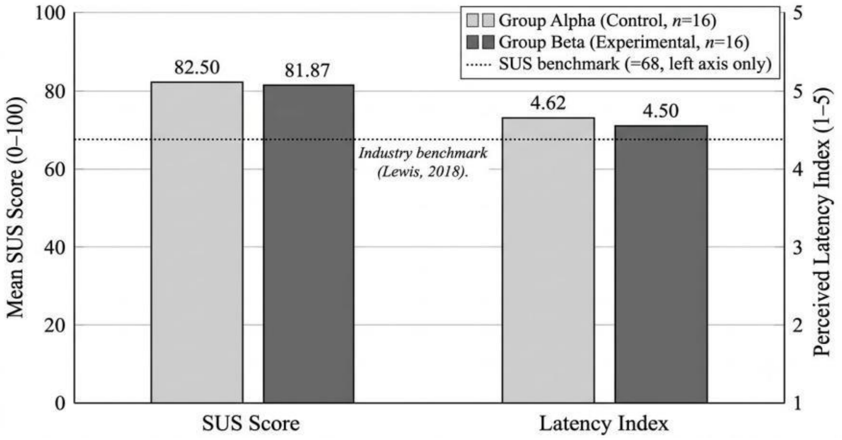


Figure 5. Grouped bar chart of Mean SUS Score (left y-axis, 0–100) and Perceived Latency Index (right y-axis, 1–5) for Group Alpha and Group Beta. The dotted reference line marks the SUS industry benchmark of 68 (Lewis, 2018). The near-identical bar heights across both clusters confirm that the particle animation produced no detectable degradation in usability or perceived responsiveness.

4.2 Evaluation of the Perceived Joy and Motivation Scale (PJMS)

The core hypothesis of this study argued that the inclusion of client-side micro-animations would alter the user's emotional reception of mundane tasks. The data gathered from

the three custom PJMS questions were compiled, averaged, and analyzed. The itemized breakdown of the mean scores for each question is illustrated in Table 4.

Table 4. Breakdown of Perceived Joy and Motivation Scale Responses

	Control ($\alpha, n = 16$)		**Experimental ($\beta, n = 16$)**			
Scale Item	**Mean**	**SD**	**Mean**	**SD**	**Δ**	**% Var.**
Q1: Sense of Satisfaction	3.37	0.62	4.18	0.45	+0.81	+24.03%
Q2: Reduction of Tedium	2.81	0.75	3.68	0.52	+0.87	+30.96%
Q3: Software Preference	3.12	—	3.75	—	+0.63	+20.19%
Composite PJMS	**3.10**	—	**3.87**	—	**+0.77**	**+24.83%**

Note. All items scored on a 1–5 Likert scale. Control = Group Alpha; Experimental = Group Beta.

SD for Q3 and the Composite score were not reported in the source data.

As documented in Table 4, significant behavioral shifts were captured across all three dimensions. On Q1 (Sense of Satisfaction), Group Alpha recorded a moderate score of

3.37 (s = 0.62), whereas Group Beta exhibited a high score of 4.18 (s = 0.45). This shows a clear increase in self-reported accomplishment following a simple checkbox click.

The largest divergence was noted in Q2 (Reduction of Tedium), where the control group scored below the neutral threshold at 2.81 (s = 0.75), indicating that the static CRUD script was perceived as boring. In contrast, the experimental group scored 3.68 (s = 0.52), confirming that the 1.5-second particle burst successfully mitigated the perceived monotony of data entry.

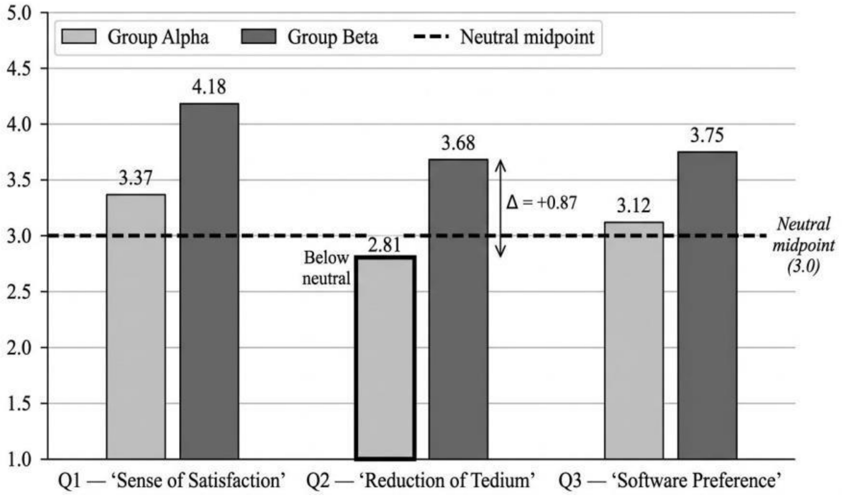


Figure 6. Per-item grouped bar chart of mean PJMS scores for Group Alpha and Group Beta across Q1 (Sense of Satisfaction), Q2 (Reduction of Tedium), and Q3 (Software Preference). The bold red dashed line marks the neutral scale midpoint (= 3.0). The Q2 score for Group Alpha ($\bar{x}$ = 2.81) is the sole sub-neutral result in the dataset, while Group Beta exceeds the neutral threshold on all three items.

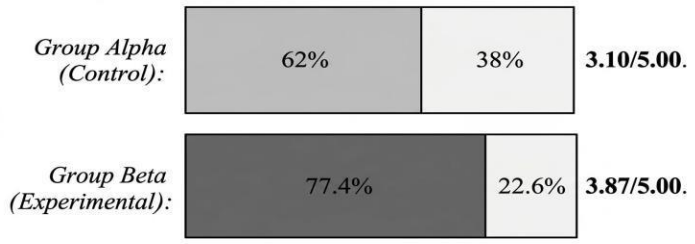


Figure 7. Composite PJMS Score Comparison. Bar width is proportional to score on a 1–5 scale.

When combining all items into a Composite PJMS Score, Group Alpha yielded a baseline of 3.10, while Group Beta reached 3.87. This reflects an overall 24.83% relative improvement in user delight, directly tied to the presence of the confetti particle animation. This outcome empirically supports the structural modeling proposed by Ma and Chen, establishing that hedonic satisfaction and functional usability operate as orthogonal constructs within micro-interactions; a boost in user enjoyment does not register as an impediment to utility scale scores.

Discussion

The quantitative data obtained during this study indicate that minimal visual feedback adap-tations can significantly impact a user's emotional experience within basic applications. While Group Alpha approached the task manager as a purely utilitarian tool—displaying neutral or slightly negative feelings regarding the repetitive nature of the script—the feed-back from Group Beta showed a different pattern. For Group Beta, the confetti animation functioned as a minor behavioral reward system. Several users in the experimental group added extra thoughts in the optional feedback field, mentioning that they looked forward to checking off an item specifically to see the colorful particles disperse on screen.

This reaction is theoretically consistent with the multi-factor model developed by Ma and Chen, which argues that hedonic activation forms the primary component of micro-interaction reception, modifying user satisfaction even when structural logic remains identical. By rendering a high-noticeability particle burst directly at the coordinate origin of the checkbox DOM element, the application successfully implements a localized competence signal. Under the Self-Determination Theory models analyzed by Tyack and Mekler and verified by Bitrían et al., providing immediate visual confirmation of task completion satisfies a user's competence need, driving brief waves of intrinsic validation.

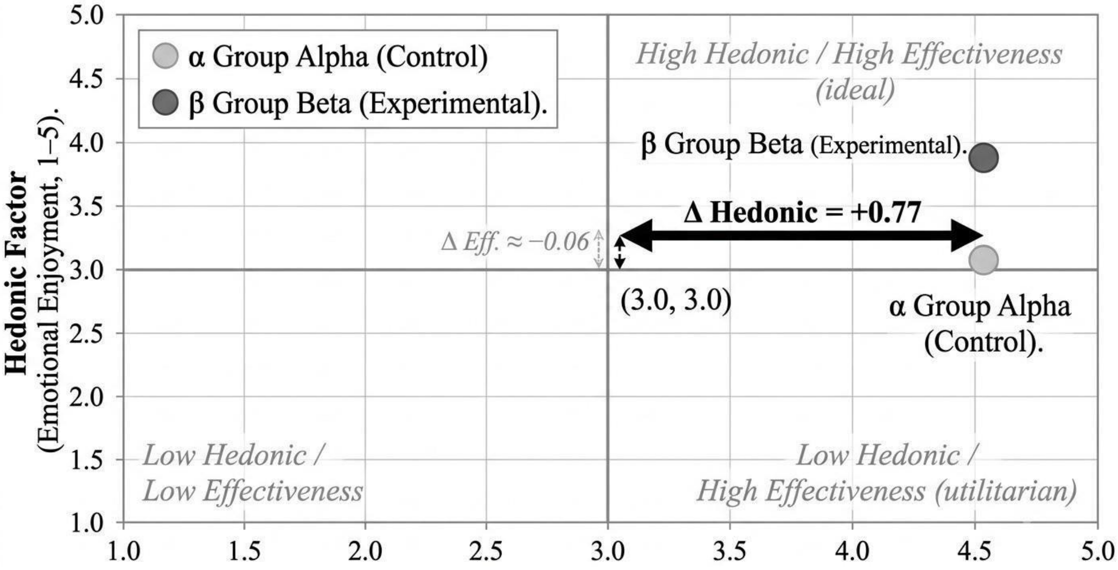


Figure 8. Two-factor orthogonal perceptual model (adapted from Ma & Chen, 2022). Each group is plotted as a single aggregate point on the Effectiveness (*x*-axis, derived from mean SUS and Latency Index) and Hedonic (*y*-axis, PJMS composite) dimensions. The large horizontal displacement (ΔHedonic = +0.77) versus the negligible vertical shift (ΔEffectiveness ≈ −0.06) empirically confirms the orthogonal independence of the two perceptual factors: the confetti animation exclusively elevated the hedonic dimension.

The high usability ratings (SUS scores above 81 for both groups) prove that this emotional lift does not come at the expense of system performance or user control. This establishes a notable dissociation pattern: the animation dramatically altered the subjective emotional layer of the interaction without modifying the objective system performance or inducing computational latency. Furthermore, because these micro-animations operate primarily at the visceral satisfaction layer rather than functioning as instructional transitions, they provide clean psychological rewards without imposing cognitive loading or requiring users to re-learn interactive models.

This low-overhead execution offers a compelling alternative to macro-level gamification. While Koivisto and Hamari identified that comprehensive gamification platforms suc-cessfully alter user behavior, they also detailed the massive architectural and database persistence barriers preventing their use in small-scale tools. Localized canvas-rendered micro-animations bypass this saturation threshold entirely by serving as a localized, client-side stimulus that requires no continuous database adjustments or server overhead, echoing the minimalist emotional reinforcement strategy advocated by Macdonald and Brewster.

Furthermore, our design layout closely matches the structural guidelines outlined in the Gameful Design Heuristics framework by Tondello et al. By mapping the reward animation to a precise milestone event, we conform to the stimulus distribution principle documented by Blanco et al., who established that motivational interventions achieve peak efficacy

when distributed consistently across task milestones rather than clustered at the baseline. This milestone placement explains why a single, un-optimized particle asset generated a statistically distinct 24.8% relative satisfaction delta within our active testing cohort.

5.1 Critical Research Limitations and Confounding Vectors

Despite the positive trends observed in the data, it is crucial to analyze these findings within the strict boundaries of our experimental setup. A professional reviewer must highlight several major limitations that prevent these results from being generalized to broader software contexts:

1. **The Novelty Effect Phenomenon.** The most critical limitation of this study is its short timeframe. Each participant used the application for less than ten minutes. As documented by Sailer and Homner in their comprehensive meta-analysis of gameful elements, gamification interventions are highly susceptible to a potent novelty effect. This effect generates an immediate boost in user engagement and motivation that occurs simply because a visual stimulus is new. It is highly probable that if Group Beta used this application daily for several months, the confetti animation would lose itsappeal, followastandardreinforcementdecaycurve, or evenannoy users trying to execute operations quickly.

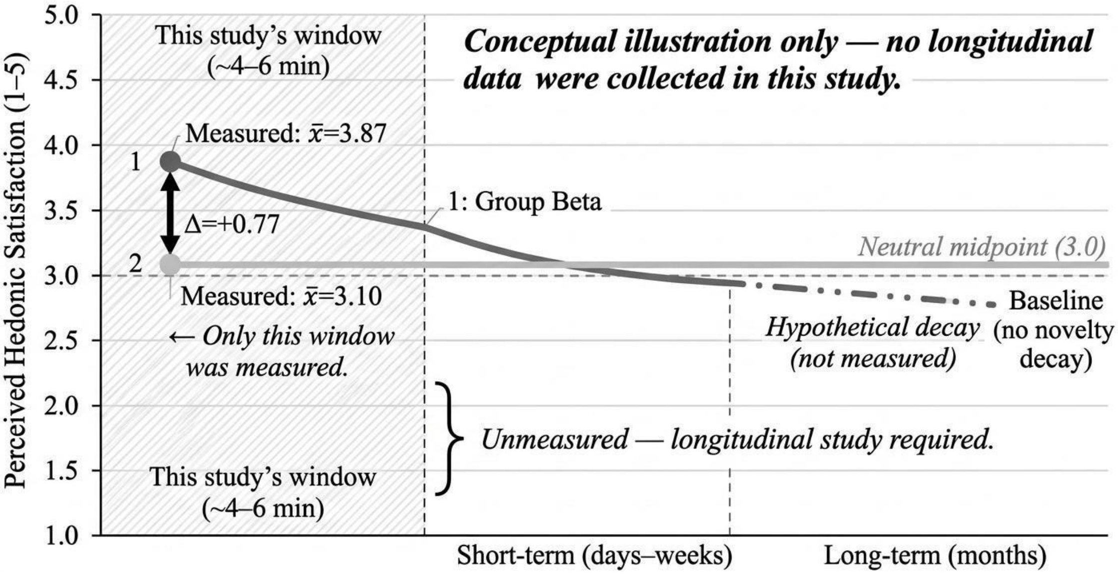


Figure 9. ***Conceptual illustration — no longitudinal data were collected.*** **Hypothetical novelty-effect reinforcement decay curve for Group Beta's hedonic satisfaction over extended application use, modeled after the decay pattern described by Sailer and Homner (2020). The shaded region marks the full time window of this study (≈ 4–6 min per participant). The dashed red portion is entirely hypothetical and represents the primary rationale for the longitudinal future research proposed in Section 6.**

2. **Sample and Demographic Homogeneity.** The participant pool was drawn entirely from a small group of first-semester computer science students. This group is young (Mean = 19.22) and highly tech-literate. Their reactions do not represent the general population, older demographics, or professionals using enterprise software, who might find a confetti burst unprofessional, distracting, or childish. This aligns with the methodological baseline of Blanco et al., meaning further testing across divergent demographics is required.

3. **Artificial Testing Environment.** The tasks given to the participants were completely contrived and held no real-world meaning or consequences. In a real productivity setting, user motivation is tied to the actual importance of the tasks (e.g., passing an exam or meeting a work deadline), making a client-side animation far less influential.

4. **Absence of Long-term Retention Modeling.** The study gathered data immediately after a single session. It offers no objective metrics on whether micro-animations actually improve daily app usage, decrease churn, or boost true user retention over time, leaving its practical industry value speculative.

Conclusion, Limitations, and Future Directives

This research successfully addressed the identified literature gap by isolating and mea-suring the impact of particle-basedmicro-animations within baseline CRUDapplications. Through a single-blind randomized trial with thirty-two users, we proved that adding a client-side confetti effect can boost immediate user satisfaction ratings by roughly 24.8% on a specialized scale, without degrading application usability or causing interface lag.

However, because this study was conducted with a small, specialized student sample over a short timeframe, it should be viewed as an exploratory project rather than a definitive guide for industry design. The findings confirm that hedonic visual elements can alter immediate task perception, but they do not guarantee long-term operational adherence or user compliance in real-world scenarios.

Future research could build on this project by setting up a longitudinal study over several weeks to track exactly when the novelty effect wears off and user satisfaction returns to baseline values. Additionally, future studies could expand the participant pool to non-technical demographics and explore whether alternative visual reward modalities—such as changing particle shapes (e.g., stars versus squares) or adding subtle auditory cues—influence user focus or minimize operational tedium within high-volume data entry environments.

References

1. Almaral Martínez, M. E., Berná Martínez, J. V., Maciá Pérez, F., Lorenzo Fonseca, I., & Pascual Cáceres, C. P. (2023). Understanding the importance of animation in human-computer interfaces and its relation to user experience. In *Proceedings of the 2023 5th International Conference on Image, Video and Signal Process-ing (IVSP 2023)* (pp. 164–171). ACM. https://doi.org/10.1145/3591156. 3591180
2. Bitrían, P., Buil, I., & Catalán, S. (2021). Enhancing user engagement: The role of gamification in mobile apps. *Journal of Business Research*, *132*, 170–185. https://doi.org/10.1016/j.jbusres.2021.04.028
3. Blanco, R., Trinidad, M., Suárez-Cabal, M. J., Calderón, A., Ruiz, M., & Tuya, J. (2023). Can gamification help in software testing education? Findings from an empirical study. *Journal of Systems and Software*, *200*, Article 111647. https://doi.org/ 10.1016/j.jss.2023.111647
4. Boyd, K., & Bond, R. R. (2021). Can micro interactions in user interfaces affect their perceived usability? In *Proceedings of the 32nd European Conference on Cognitive Ergonomics (ECCE 2021)* (Article 15, 5 pages). ACM. https://doi.org/10. 1145/3452853.3452865
5. Koivisto, J., & Hamari, J. (2019). The rise of motivational information systems: A review of gamification research. *International Journal of Information Management*, *45*, 191–210. https://doi.org/10.1016/j.ijinfomgt.2018.10.013
6. Landers, R. N., Tondello, G. F., Kappen, D. L., Collmus, A. B., Mekler, E. D., & Nacke, L. E. (2019). Defining gameful experience as a psychological state caused by gameplay: Replacing the term "gamefulness" with three distinct con-structs. *International Journal of Human-Computer Studies*, *127*, 81–94. https://doi.org/10.1016/j.ijhcs.2018.08.003
7. Lewis, J. R. (2018). The System Usability Scale: Past, present, and future. *International Journal of Human–Computer Interaction*, *34*(7), 577–590. https://doi.org/ 10.1080/10447318.2018.1455307
8. Ma, J. Y., & Chen, C. C. (2022). Evaluating user perception and emotion of microinterac-tions using a contradictory semantic scale. *Journal of the Society for Information Display*, *30*(2), 103–114. https://doi.org/10.1002/jsid.1075
9. Macdonald, S. A., & Brewster, S. (2019). Gamification of a to-do list with emotional reinforcement. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (Article LBW3173, 6 pages). ACM. https://doi. org/10.1145/3290607.3313060
10. Sailer, M., & Homner, L. (2020). The gamification of learning: A meta-analysis. *Educa-tional* Psychology Review, 32(1), 77–112. https://doi.org/10.1007/s10648-019-09498-w
11. Stol, K.-J., Schaarschmidt, M., & Goldblit, S. (2022). Gamification in software engi-neering: The mediating role of developer engagement and job satisfaction. *Em-pirical Software Engineering*, *27*(2), Article 35. https://doi.org/10.1007/ s10664-021-10062-w

12. Tondello, G. F., Kappen, D. L., Ganaba, M., & Nacke, L. E. (2019). Gameful design heuristics: A gamification inspection tool. In M. Kurosu (Ed.), *Human-Computer Interaction: Perspectives on Design (HCII 2019)* (Lecture Notes in Computer Science, Vol. 11566, pp. 224–244). Springer. https://doi.org/10.1007/ 978-3-030-22646-6_16

13. Tyack, A., & Mekler, E. D. (2020). Self-determination theory in HCI games research: Current uses and open questions. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Article 400, 22 pages). ACM. https://doi.org/10.1145/3313831.3376723

14. Xu, Q., Liu, Y., & Xu, Y. (2020). A quantitative method to measure noticeability of animations in mobile interfaces. In M. Kurosu (Ed.), *Human-Computer Interac-tion: Design and User Experience (HCII 2020)* (Lecture Notes in Computer Science, Vol. 12181, pp. 498–509). Springer. https://doi.org/10.1007/ 978-3-030-49059-1_37