Rushes: A Human Preference Dataset for Pluralistic Alignment

Anonymous ACL submission

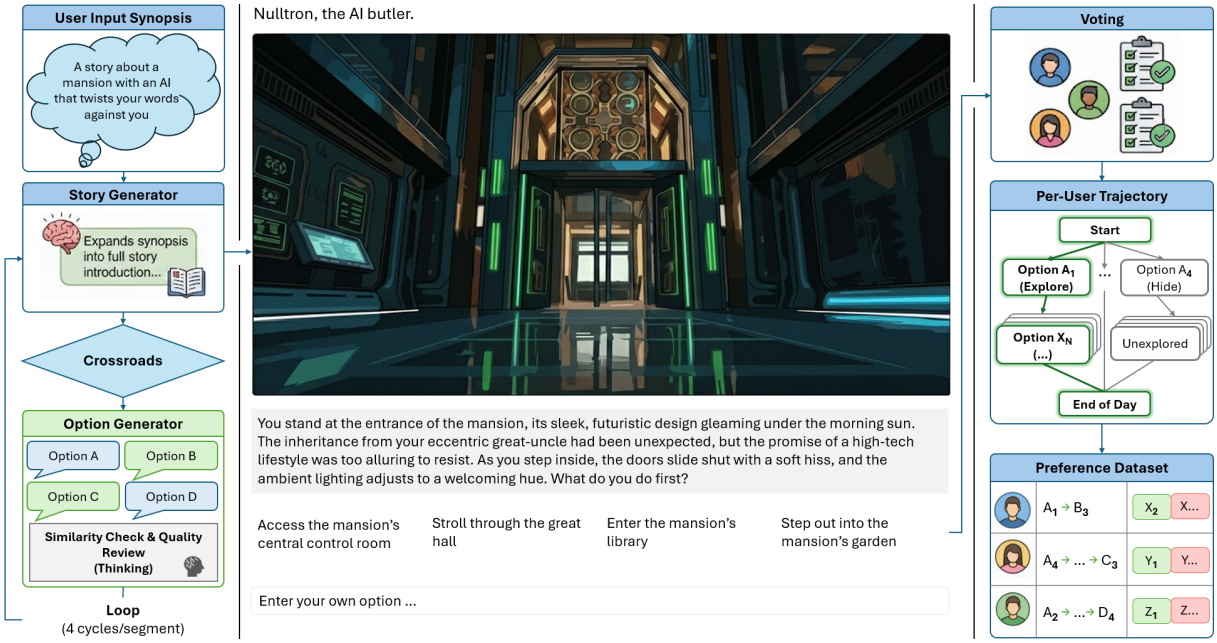

Figure 1: Overview of the *Rushes data collection framework*. **Left:** The generation pipeline expands a user provided synopsis into story segments and iteratively generates distinct options via similarity and quality checks. **Center:** The game interface displays the narrative and visual context, allowing players to interact via predefined choices and submit suggestions to us. **Right:** The data collection mechanism aggregates user votes and individual playthrough trajectories to construct a revealed-preference interaction dataset.

Abstract

We introduce **Rushes**, a dataset and benchmark for studying revealed human engagement preferences in interactive narrative environments. Rushes is collected through a game interface where users interact with AI-generated branching narratives and select one choice from a small, explicit candidate set at each decision point. Each interaction logs the full candidate set, the user's choice, and the evolving narrative context, yielding time-ordered trajectories with persistent user-level identifiers.

Rushes contains **44,226 decision events from 8,167 unique users across six games**, capturing sequential, personalized engagement behavior rather than static judgments. We show that user choices exhibit structured, non-random patterns, quantified by a low choice entropy relative to a uniform baseline.

We position Rushes as a diagnostic benchmark for pluralistic alignment and demonstrate a robust *Engagement Gap*: state-of-the-art LLMs, including GPT-5, fail to outperform simple baselines. While classical Matrix Factorization (SVD) captures measurable personalized signal (37.7%), frontier LLMs (34.23%) struggle to even match the Popularity Baseline (36.4%) on event-level choice prediction. This gap suggests that single, population-level objectives, like those used in modern RLHF, appear insufficient to capture heterogeneous, context-dependent engagement signals. As a result, even highly capable models default to majority preferences rather than adapting to individual trajectories. We release Rushes to support research into pluralistic alignment and sequential decision-making in generative systems.

1 Introduction

Foundational work in Large Language Models remains largely focused on capability and safety, codified by datasets that reward helpful and harmless outputs. In safety, both in research and practice,

the goal is convergence: to minimize harms for all users. Entertainment domains such as games, movies, or books, however, introduce an orthogonal dimension to model development. Here the goal is divergence: to maximize "interestingness" and "fun" for specific individuals. Learning what makes an experience meaningful for individuals across many subjective dimensions will be critical across a broad range of important application domains where there may be multiple targets depending on the user population.

A personalized notion of engagement calls for pluralistic alignment, where models adapt to diverse human values rather than collapsing to a single mean. Current alignment methods, however, often fail to capture this subjectivity. As noted by Ali et al. (2025) , aggregating diverse preferences into a single reward model suppresses minority viewpoints, leading to generic outputs. Furthermore, in these settings, the contextual history plays an important role, as prior user choices significantly influence subsequent model predictions.

To our knowledge, there is no previous large-scale human preference dataset that jointly addresses the challenges of alignment with engagement, sequential decision-making, and personalized modeling. We present Rushes, a dataset and benchmark built around human reactions to AI-generated stimuli: branching narratives that include text, images, video, and audio narration.

The Engagement Gap: Our experiments reveal a critical limitation in current frontier models. When tasked with predicting user choices in Rushes, models like GPT-4o (OpenAI, 2024) and even GPT-5 (OpenAI, 2025) struggle to outperform simple popularity heuristics. This mirrors the *Popularity Bias* in recommender systems but highlights a distinct failure mode in LLMs: they are fine-tuned to be "universally acceptable" rather than "personally compelling." Recent work by Castricato et al. (2025) with the *PERSONA* benchmark has begun to address this using synthetic user proxies. Rushes complements this synthetic approach by providing organic, revealed preferences from real human trajectories, capturing the noisy and implicit nature of true engagement.

Figure 1 shows a screenshot of the interface we built to collect the dataset. We created six games, all AI-generated, for users to play. Each game starts by setting a context, generating a storyline plot, and asking players what should happen next at a branching point in the game. Users select

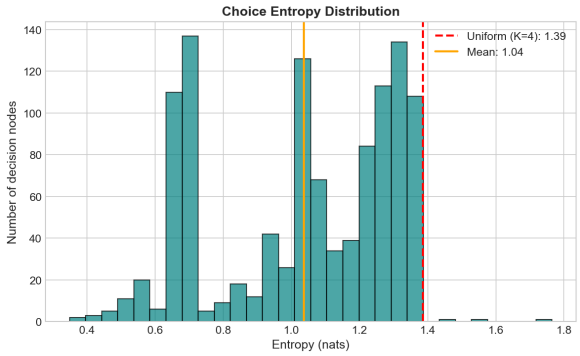

Figure 2: Distribution of user vote entropy across decision points. For a fixed candidate set size of four, observed vote entropy is consistently lower than the uniform baseline, structured, non-random choice behavior. This demonstrates that shared stimuli elicit systematic preferences, establishing the presence of signal necessary for studying engagement and preference diversity, without implying convergence to a single dominant outcome.

from a small set of options (typically four). Each chosen option is then integrated into the story, and continued until the next branching point, until we reach the end of the day, which is set at depth $= 4$. Since these stories are meant to be open-ended, we can continue generating more days until we decide to stop and finish the story. During this daily voting process, we log the chosen option alongside the alternatives, plus some additional metadata.

No payments or instruction-following were required, users participated because the activity itself is enjoyable; yielding preferences that reflect natural, in-situ behavior. We observe that the entropy of user votes is consistently lower than that of a uniform distribution (Figure 2), indicating non-random preferences. This aligns with information-theoretic approaches to narrative evaluation, such as *Fabula Entropy Indexing* (Castricato et al., 2021), which posits that high-quality narratives exhibit low entropy in human question-answering tasks.

The current snapshot of our dataset consists of 44,226 preference votes by 8,167 unique users across the six games. We frame Rushes as a benchmark for predicting personalized user engagement in interactive narratives.

Our contributions are:

1. A method for collecting large-scale human preferences on user engagement of fun and interestingness, that is personalized at a user level;

2. A large-scale dataset of 44,226 preference votes by 8,167 unique users across six narrative-based games (with multimodal content);

3. A comprehensive benchmarking suite that establishes performance baselines using both collaborative filtering and state-of-the-art LLMs, identifying a significant 'Engagement Gap' that challenges current alignment techniques;

4. We are also releasing code/prompts for the generation pipeline as well as the environment itself, which will be publicly released in an archived state;

We hope that our release will support researchers interested in personalized alignment, engagement modeling, and interactive narrative creation. Rushes is designed as a testbed for studying human preferences in open-ended, multimodal environments.

2 Related Work

Interactive Narrative and Storytelling Benchmarks The landscape of interactive narrative benchmarks has expanded significantly in 2025. *TextQuests* (Phan et al., 2025) utilizes classic interactive fiction to benchmark the reasoning and planning capabilities of agents. While *TextQuests* evaluates whether an agent can solve a puzzle (competence), Rushes evaluates whether a model knows what a human wants to happen next (engagement). This distinction is crucial for developing agents that are not just capable, but enjoyable.

Similarly, *What-If* (Huang et al., 2024) and *Narrative Studio* (Ghaffari and Hokamp, 2025) explore the generative mechanics of branching narratives. *Narrative Studio*, for instance, employs Monte Carlo Tree Search (MCTS) to maximize narrative diversity during generation. Rushes complements these system-focused works by providing the data necessary to evaluate the "fun" factor of the resulting generations. While we employ similar semantic diversity checks in our generation pipeline (see Section 3.1) to prevent redundancy, our primary contribution is the capture of revealed human preferences within these diverse structures, rather than the generation method itself.

Personalized Alignment and RLHF Prior work on aligning large language models (LLMs) with human preferences has focused primarily on dimensions such as safety, helpfulness, or factual correctness (Ouyang et al., 2022; Bai et al., 2022). These datasets typically lack the longitudinal user history required for personalization.

Recent works have highlighted the "cold-start" problem in personalized alignment, arguing that static reward models fail to capture evolving user intent. *LiteraryTaste* (Chung et al., 2025) addresses this in the creative writing domain, finding that explicit surveys ("stated preferences") often fail to predict actual choices ("revealed preferences"). Rushes is a pure "revealed preference" engine, capturing user intent through action rather than survey. Furthermore, *LikeBench* (Rahman et al., 2025) attempts to measure "likability" using simulated personas. Rushes advances this by providing trajectories from real humans, whose preferences are often noisier and more context-dependent than simulated agents.

Drama Management and Interactive Narrative Classical *Drama Managers* (DMs) sought to adapt ongoing narratives to user preferences to maximize agency or enjoyment (Yu and Riedl, 2013; Riedl and Bulitko, 2013). However, these systems often relied on handcrafted rules or symbolic planners, making it hard to scale. While recent neural approaches like *AI Dungeon* (Walton, 2019) and *Hierarchical Story Generation* (Fan et al., 2018) demonstrated the potential of open-ended text generation, they often lack the structured, longitudinal preference data necessary for personalized modeling. Rushes modernizes this objective by scaling the environment using LLMs. Unlike classical DMs which operate on restricted state spaces, Rushes leverages the open-ended generation capabilities of frontier models while capturing "revealed preferences" (Chung et al., 2025) at a scale (44k+ interactions) to provide the necessary user models to enable modern, LLM-based Drama Management.

Subjective Evaluation Metrics Measuring "fun" is notoriously difficult for standard reward models. *WritingPreferenceBench* (Ying et al., 2025) demonstrated that sequence-based reward models—the standard for RLHF—achieve only 52.7% accuracy on subjective writing tasks, barely outperforming random chance. This aligns with our finding that neural preference models struggle to beat popularity baselines in Rushes. It suggests that modeling engagement requires architectural innovations, such as the *Generative Reward Models* proposed

Aspect	Rushes (Ours)	RLHF Chat Datasets	SeqRec Datasets
Multi-choice Decision Space ($k > 2$)	**Yes**	Limited (Binary)	Yes
Longitudinal User History	**Yes**	No (Stateless)	Yes
Sequential Context Dependency	**Yes**	Limited	No (ID-based)
Multimodal Assets (Image/Video)	**Yes**	No	Limited
Modeling Objective	**Personalized Engagement**	**Safety & Helpfulness**	**Clicks & Purchase**

Table 1: Comparison of Rushes with prior work. Unlike standard RLHF datasets which focus on safety without longitudinal history, or Sequential Recommendation (SeqRec) datasets that track item IDs rather than narrative context, Rushes combines long-term user trajectories with rich, multimodal interactive narratives.

by Ying et al. (2025), capable of reasoning about style and subtext.

3 Rushes

3.1 Game Generation

3.1.1 Generating branching narrative text

In the current release, all narrative text and decision options are generated using GPT-4o with a temperature of 0.3 and a fixed prompting template (see Appendix A for all prompts). All generated nodes and options are stored prior to gameplay. Each story begins from a high-level synopsis that specifies the intended narrative trajectory, and the generation process recursively expands the story tree to a depth of four, yielding approximately 330 nodes that can be hand reviewed before release.

We selected a depth of four to mirror a concise narrative arc while keeping generation computationally manageable. Each decision node presents four options, a branching factor chosen to balance computational cost with sufficient variance to capture distinct behavioral strategies, such as aggressive, diplomatic, exploratory, or passive choices.

Semantic Diversity Enforcement To prevent the generation of redundant options, a common failure mode in LLM storytelling, we employ a semantic similarity filter. At each decision node, an LLM-based checker compares the candidate option against previous options along the trajectory. If the option is judged too similar (considering action type, complexity, and narrative outcome), it is discarded and regenerated.

Lexical Diversity via Deterministic Paraphrasing To mitigate lexical repetition and prevent users from navigating based on memorized surface text, we generate multiple semantic paraphrases for each option node during the story generation phase. The number of variants generated for a given node is scaled dynamically based on the tree depth and expected player density, ensuring that distinct users are unlikely to encounter identical text even in high-traffic branches (see Appendix A.2 for the scaling derivation). At runtime, we maintain low latency by selecting a variant deterministically using a hash of the user's anonymized ID and the node identifier. This yields a personalized textual experience that preserves the underlying action semantics without requiring expensive real-time generation.

3.1.2 Generating image, audio and video

Each narrative node is paired with multimodal assets to enhance immersion. Image generation is performed using a staged prompt construction pipeline (meta-prompts, similar to Huang et al. (2024)) that improves character consistency and stylistic coherence. Images are then used as input to generative video models to create short clips using *LTX-Video* (HaCohen et al., 2025). Audio narration is synthesized using the *Azure Text-to-Speech* (TTS) API with expressive styles.

3.1.3 Generating narrative continuations

To support multi-session narratives, Rushes enables dynamic story continuation across multiple "days" of gameplay. At the end of each day, we identify all active leaf nodes. We prune the exponential expansion by clustering leaf scenes into four broad narrative categories using an LLM. We then generate custom continuations for each active node that align with these categories.

3.1.4 Quality Control and Responsible AI

All generated content is passed through an automated safety screening pipeline (Azure Content Safety API), with results summarized in Table 2. Across 3,982 screened generations, the system maintained strict safety standards on sensitive dimensions. The vast majority of content was classified as safe (Severity 0) for Hate (99.7%), Sexual (98.5%), and Self-Harm (99.0%).

	Counts by Severity Level					Percentage (%)				
Dimension	**0**	**2**	**4**	**6**	**Total > 0**	**0**	**2**	**4**	**6**	**Total > 0**
Hate	3,969	13	0	0	13	99.67	0.33	0.00	0.00	0.33
Self-Harm	3,943	31	8	0	39	99.02	0.78	0.20	0.00	0.98
Sexual	3,922	52	7	1	60	98.49	1.31	0.18	0.03	1.51
Violence	2,726	1,142	113	1	1,256	68.46	28.68	2.84	0.03	31.54
Any Flag (non-zero)	**2,674**	**1,182**	**124**	**2**	**1,308**	**67.15**	**29.68**	**3.11**	**0.05**	**32.85**

Table 2: **Azure Content Safety Analysis (N=3,982).** Distribution of safety severity scores across four dimensions. Severity levels range from 0 (Safe) to 6 (High). The higher prevalence of low-severity Violence flags reflects the action-adventure nature of the narrative genres.

As expected for a dataset focused on action and adventure genres, the 'Violence' dimension showed higher activity, with 31.5% of generations registering above Severity 0. However, the majority of these (28.7%) fell into low-severity buckets (Severity 2), consistent with standard genre tropes (e.g., sci-fi combat or dramatic tension) rather than graphic or gratuitous violence. Only a small fraction (0.05% overall) reached higher severity levels (Severity 6).

All content was reviewed manually and approved by the authors. The gameplay interface also includes a user-facing report mechanism, however we received no reports from users during our release.

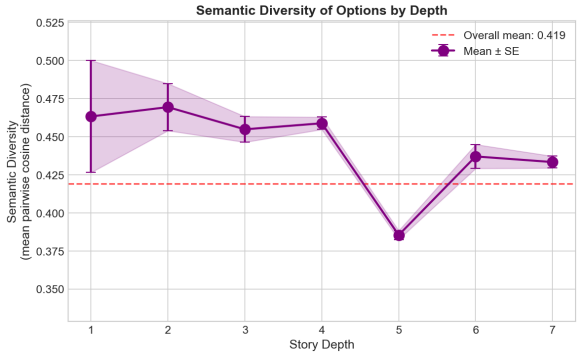

Figure 3: Semantic diversity calculated as pairwise cosine distance, embeddings generated from OpenAI's *text-embedding-3-small* model. Note that $depth = 5$ has a large dip in diversity because of our end-of-day narrative consolidation.

3.2 Analysis of Generated Games

3.2.1 Lexical and Semantic Diversity

We also evaluate diversity of our generated branches and options per level using average cosine distance in sentence embedding space, shown in Figure 3. We notice that the diversity drops around depth $= 5$. This reflects the episodic nature of our generation pipeline: at the end of each day, leaf nodes are clustered into a small set of broad thematic continuations, temporarily consolidating the narrative state. This allows the story to maintain long-term coherence and manage complexity before expanding into new divergent paths in the subsequent day, mirroring the structure of serialized television. We also observe that the variance of the diversity drops over time, which seems to be due to an additive effect of the prompt length reducing variance over time.

3.2.2 Multimodal Asset Evaluation

We observed occasional inconsistencies between text and generated media (image/video), and some users reported that narration quality varied across scenes. Despite these imperfections, the presence of multimodal assets likely increased user immersion, grounding decisions within a coherent narrative world rather than isolated text prompts. This richer context encourages in-world decision-making and reduces superficial text skimming, helping ensure that the collected preferences reflect more genuine engagement. Since the benchmark evaluations in this paper are text-conditioned, we release the accompanying media primarily to preserve the full context under which human preferences were revealed and to support future multimodal modeling work.

3.2.3 Summary

Our goal in generating these narratives was not to use LLMs to break new ground in narrative construction, but rather to construct plausible stimuli that could be easily understood and enjoyed by our players.

These results suggest that users are consistently presented with distinct and non-redundant alternatives at each decision point. This property is critical for preference data collection: if options were trivially similar or repetitive, observed user choices

would be dominated by noise or superficial cues rather than substantive engagement.

Taken together with the low choice entropy observed in user behavior, the generation analysis supports the interpretation that Rushes captures structured, context-dependent decisions rather than arbitrary clicks. This validates the dataset as a suitable testbed for studying revealed preferences, sequential decision-making, and the limits of current alignment methods in interactive generative environments.

3.3 Data Collection

User Recruitment All participants in Rushes were authenticated users recruited through the Xbox Insiders Program. Participation required signing in with verified Xbox credentials, ensuring that each interaction corresponds to a real, persistent user identity rather than anonymous or crowdsourced accounts. Users voluntarily opted into the experience and engaged without financial incentives, reflecting intrinsic motivation and familiarity with interactive gaming environments.

3.3.1 Logging and Schema

Each 'click' generates a vote which is recorded in a standardized schema:

- `user_id` (anonymized identifier);
- `game_id` and `level` (narrative depth);
- `vote` (selected option text) and `other_options` (unselected candidates);
- Metadata: `time_taken_ms`, `user_agent`, `session_depth`.

Each record captures both the decision context and behavioral outcome, allowing reconstruction of complete narrative trajectories.

3.3.2 Dataset Composition

The final dataset comprises 44,226 distinct decision events generated by 8,167 unique users across the six available titles (Table 3). The distribution of user engagement follows a long-tailed pattern typical of gaming environments. While the average participant interacted with 1.4 games , a dedicated core of 195 "power users" engaged with all six narrative environments.

In terms of session length, the average trajectory reached a depth of 5.4 decision points. Since the standard "day" cycle concludes at depth 4, this indicates that a significant portion of users persisted past the initial narrative loop to experience multi-day continuations. The participant pool consists exclusively of authenticated Xbox Insiders, yielding a demographic that is predominantly English-speaking and highly literate in branching game mechanics.

Statistic	Value
Active users	8,167
Total decision events (votes)	44,226
Number of games	6
Average trajectory depth (decisions per playthrough)	5.4
Average games played per user	1.4
Users who played all 6 games	195
Typical candidate set size per decision	4 options

Table 3: Summary statistics for the current Rushes snapshot. Each decision event logs the full candidate set and the user's chosen option, plus metadata (e.g., time taken and session depth).

We also observe that engagement is highly non-uniform. As shown in Figure 2, the entropy of user votes (1.04 nats) is consistently lower than the uniform baseline (1.39 nats), confirming that users are not clicking randomly but are driven by structured, context-dependent narrative preferences.

3.3.3 Preference Transformation and Modeling

To support alignment research, we can also transform the raw interaction logs into training-ready formats. We convert each 'choose 1 of k' decision (where $k = 4$) into $k - 1$ distinct pairwise comparisons, denoted as $(o_{chosen} \succ o_{rejected})$. These pairs formally represent the user's preference for the chosen option over the alternatives, enabling the training of standard Reward Models and Direct Preference Optimization (DPO).

4 Experiments and Results

Task Definition We frame evaluation as event-level text-based candidate choice prediction. At each decision point, the model observes the narrative context, the available candidate options, and the user's interaction history up to the current decision point, and must predict which single option the user selected.

We use Top-1 accuracy because Rushes captures single, irreversible user decisions rather than graded preferences or ranked lists. Pairwise and ranking metrics would inflate performance by decomposing one holistic choice into multiple com-

Baseline	Accuracy [95% CI]	*N* Samples	Description
SVD Collab. Filtering	**0.3773 [0.3669, 0.3878]**	8293	Matrix factorization on user-option interactions
Popularity (Most Freq)	0.3639 [0.3536, 0.3743]	8293	Always select the historically most popular option
GPT-5 w/ History	0.3423 [0.3360, 0.3590]	8293	GPT-5 prompted with user history
SASRec	0.3406 [0.3304, 0.3409]	8293	Self-Attentive Sequential Recommendation
Semantic Classifier	0.3000 [0.2902, 0.3100]	8293	Fine-tuned DeBERTa-v3 on context+option pairs
Random (Uniform)	0.2541 [0.2452, 0.2631]	8293	Uniform selection among 4 available options

Table 4: Main Baseline Accuracy on the Rushes test set with 95% Wilson confidence intervals. All models are evaluated on the same held-out test split.

Depth	Accuracy [95% CI]	*N* Samples
0	0.3048 [0.2728, 0.3390]	735
1	0.3100 [0.2854, 0.3357]	1300
2	0.3740 [0.3706, 0.3993]	1441
3	0.3763 [0.3620, 0.3908]	4361
4	**0.4207 [0.3761, 0.4666]**	454

Table 5: Popularity Baseline accuracy by narrative depth. Accuracy peaks at late levels (4).

History Type	Accuracy [95% CI]	*N* Samples
Same-game history	**0.3886 [0.3775, 0.3999]**	7310
Cross-game history	0.2909 [0.2634, 0.3201]	983
All history (SVD)	0.3773 [0.3669, 0.3878]	8293

Table 6: Impact of history source on prediction accuracy. Same-game history is significantly more predictive than cross-game history.

Method	Sparse Players [95% CI]	Active [95% CI]
Random	0.2540 [0.2450, 0.2632]	0.2491 [0.2008, 0.3045]
Popularity	**0.3523 [0.3390, 0.3660]**	0.3778 [0.3618, 0.3940]
SVD	0.3840 [0.3704, 0.3978]	0.3683 [0.3523, 0.3845]

Table 7: Accuracy stratified by user activity level. "Active" users are those who played 2+ games.

Model	Accuracy [95% CI]	N Samples
GPT-4o (Zero-Shot)	0.3030 [0.2931, 0.3130]	8293
GPT-5 (Zero-Shot)	0.3090 [0.2991, 0.3191]	8293
GPT-4o (w/ History)	0.3390 [0.3288, 0.3493]	8293
GPT-5 (w/ History)	**0.3423 [0.3321, 0.3526]**	8293

Table 8: Impact of Model Scaling and Context. Scaling from GPT-4o to GPT-5 yields marginal gains ($< 1\%$). Adding historical context provides a larger boost ($\approx 4\%$).

parisons, obscuring the true difficulty of predicting the user's committed action.

Evaluation Protocol We evaluate event-level Top-1 choice prediction using a user-stratified chronological split. For each user, interactions are ordered by time, with the first 80% used for training and the remaining 20% held out for testing, ensuring all test decisions occur strictly after the user's training history.

4.1 Main Results

Popularity Bias as a Strong Baseline We observe that SVD (37.73%) slightly outperforms the Popularity Baseline (36.39%). This confirms that Rushes contains structured, personalized signals that distinguish individual users from the aggregate mean. However, frontier LLMs still fail to capture this signal, falling behind both classical collaborative filtering and simple popularity heuristics. This mirrors findings in recommender systems, where popularity bias often overshadows user-specific signals, and in recent creative writing benchmarks such as those described by Ying et al. (2025); Chung et al. (2025). In these subjective domains, standard reward models frequently struggle to decouple 'generic quality' from 'personal appeal,' defaulting to safe, high-probability tokens rather than taking riskier, context-dependent bets.

We further evaluate *SASRec* (Kang and McAuley, 2018) to test if specialized sequential modeling can bridge the engagement gap. *SASRec* achieves 34.06% accuracy, performing on par with the much larger GPT-5 w/ History (34.23%). However, both methods fail to outperform the simple Popularity baseline (36.39%) and trail significantly behind SVD (37.73%). This result suggests that, in the current formulation, sequential attention alone does not outperform simpler identity-based baselines.

4.2 Ablation Studies

To verify that user preferences contain personalized signal beyond global trends, we analyze the limits of the Popularity Baseline. While Popularity achieves 36.3% accuracy, this implies that 63.7% of human choices are idiosyncratic and divergent from

the majority vote. If preferences were purely uniform, popularity would match the random baseline (25%); if they were monolithic, popularity would approach 100%. The large residual gap (36.3% vs 100%) confirms that while users share some common ground (global attractors), the majority of the engagement signal is highly personalized and context-dependent.

4.2.1 Engagement by Narrative Depth

We analyzed the Popularity Baseline's accuracy at different depths of the story tree (Table 5). Accuracy consistently rises as the narrative progresses, peaking at Depth 4 (42.07%). This suggests that as users deepen their engagement with a specific narrative arc, their choices become easier to predict under popularity heuristics.

4.2.2 The Role of History

We evaluated how user history impacts prediction in Table 6.

Same-Game History: When a user has history within the current game, accuracy is 38.86%.

Cross-Game History: When a user has history only from different games, accuracy drops to 29.09%. This significant gap (9.8 points) indicates that preferences are highly context-dependent. A user's preference for "action" in a Sci-Fi game does not perfectly translate to a Mystery game, highlighting the difficulty of transfer learning in narrative engagement.

4.2.3 Active vs. Sparse Players

As shown in Table 7, 'Active' players (returning for 2+ games) remain harder for SVD models to predict (36.83%) compared to new/sparse players (38.40%). Interestingly, the Popularity Baseline performs best on these Active players (37.78%). One possible explanation is that while new users stick to predictable personal patterns (captured by SVD), highly engaged users may be actively 'exploring' the system, making choices that deviate from their own history but aligning with globally 'interesting' content. We leave disentangling exploratory behavior from model limitations to future work.

4.2.4 Frontier Model Scaling: GPT-5 vs. GPT-4o

To assess if reasoning capabilities improve alignment, we evaluated GPT-5 against GPT-4o on the full test set in Table 8. While scaling offers marginal zero-shot gains (GPT-5 30.9% vs. GPT-4o 30.3%), adding user history provides a stronger boost, lifting GPT-5 to 34.23%. Crucially, however, even the most capable model utilized with full context fails to outperform the simple Popularity baseline (36.3%). This reinforces the finding from *WritingPreferenceBench* that simply scaling models or context does not fully bridge the engagement gap. "Fun" is not solely an emergent property of scale; it requires specific alignment with subjective values to capture idiosyncratic preferences that defy population trends.

5 Conclusion

As Large Language Models evolve from passive tools to interactive agents, the ability to model engagement becomes as critical as modeling competence. Rushes shows that organic user choices exhibit structured, non-random patterns that remain difficult for current frontier LLMs to predict under standard training and evaluation paradigms. The significant performance gap between simple personalized history models and state-of-the-art LLMs highlights the difficulty of modeling engagement in sequential narrative settings. Rushes provides a diagnostic benchmark for studying these limitations and for advancing research on pluralistic alignment, where models must adapt to diverse, subjective notions of a meaningful experience rather than converge to population-level averages.

Ethical considerations

Data Provenance and Recruitment Participants were recruited through the Xbox Insiders Program (Public Ring), a platform where users voluntarily opt-in to test pre-release content and experiments. Users were presented with a clear consent page explaining that their anonymized interaction data would be logged for research purposes and potentially released as an open-source dataset. Participation was strictly voluntary, and no financial incentives were provided; users engaged with the system solely for the intrinsic value of the gameplay experience.

Responsible AI and Dual Use We release the Rushes dataset and the associated code to foster research into personalized alignment. However, we acknowledge that methods for optimizing "engagement" can be dual-use, potentially applicable to addictive design patterns or dark patterns in UI/UX. We condemn the use of this dataset for manip-

ulative purposes and urge the community to focus on pluralistic alignment—serving diverse user needs—rather than engagement maximization for its own sake. The release is governed by a license that prohibits malicious use, and no personal identifiable information (PII) is included in the release; all user IDs have been cryptographically hashed.

Limitations

This report relates to Rushes as implemented using GPT-4o. The results shown in the demonstration will differ if other LLMs are used. No claim is made to the superiority of performance of any LLM. Outputs will vary under different temperature settings and with different prompting strategies and formats.

This system relates to games generation only. In principle, the approach taken by Rushes should be extensible to multimodal games generation, particularly those with a visual component, e.g., in a storyboarding application, but that is beyond the scope of this work.

The system is implemented using English-language prompts. It has not been investigated in other languages. Given our observation that Rushes appears to perform better on better documented settings, we expect that some degradation may occur when used with languages other than English.

As we have noted elsewhere, the architecture of this system readily lends itself to iterative editing and reprompting for further exploration of paths. Full implementation of this feature, however, involves application-specific considerations and harm mitigations for public presentation. This must be left for future work.

Given the recruitment platform, the user base is demographically skewed towards gaming-literate populations who are likely comfortable with branching narrative mechanics. Furthermore, as the generated content and interface were presented exclusively in English, the dataset reflects the preferences of English-speaking users, predominantly from regions with high Xbox Insider adoption. Consequently, the engagement patterns observed in Rushes should not be interpreted as a universal baseline for human preference but rather as a specific reflection of this gamer-centric demographic. We explicitly caution against generalizing these findings to non-gaming or non-English speaking populations without further validation.

References

Dalia Ali, Dora Zhao, Allison Koenecke, and Orestis Papakyriakopoulos. 2025. Operationalizing pluralistic values in large language model alignment reveals trade-offs in safety, inclusivity, and model behavior.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR*.

Louis Castricato, Spencer Frazier, Jonathan Balloch, and Mark Riedl. 2021. Fabula entropy indexing: Objective measures of story coherence. In *Proceedings of the Third Workshop on Narrative Understanding*, pages 84–94, Virtual. Association for Computational Linguistics.

Louis Castricato, Nathan Lile, Rafael Rafailov, Jan-Philipp Fränken, and Chelsea Finn. 2025. PERSONA: A reproducible testbed for pluralistic alignment. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 11348–11368, Abu Dhabi, UAE. Association for Computational Linguistics.

John Joon Young Chung, Vishakh Padmakumar, Melissa Roemmele, Yi Wang, Yuqian Sun, Tiffany Wang, Shm Garanganao Almeda, Brett A. Halperin, Yuwen Lu, and Max Kreminski. 2025. Literarytaste: A preference dataset for creative writing personalization.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Parsa Ghaffari and Chris Hokamp. 2025. Narrative studio: Visual narrative exploration using LLMs and Monte Carlo Tree Search.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. 2025. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*.

Runsheng "Anson" Huang, Lara J. Martin, and Chris Callison-Burch. 2024. What-if: Exploring branching narratives by meta-prompting large language models.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. pages 197–206.

OpenAI. 2024. Gpt-4o system card. arXiv preprint, https://arxiv.org/abs/2410.21276. Accessed 2025-09-22.

OpenAI. 2025. Gpt-5 system card. https://openai.com/index/gpt-5-system-card. Accessed: 2025-09-22.

Long Ouyang et al. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Long Phan, Mantas Mazeika, Andy Zou, and Dan Hendrycks. 2025. Textquests: How good are LLMs at text-based video games?

Md Awsafur Rahman, Adam Gabrys, Doug Kang, Jingjing Sun, Tian Tan, and Ashwin Chandramouli. 2025. Likebench: Evaluating subjective likability in LLMs for personalization.

Mark O. Riedl and Vadim Bulitko. 2013. Interactive narrative: An intelligent systems approach. *AI Magazine*, 34(1):67–77.

Nick Walton. 2019. Ai dungeon: Dragon model upgrade. *Aidungeon. io*.

Shuangshuang Ying, Yunwen Li, Xingwei Qu, Xin Li, Sheng Jin, Minghao Liu, Zhoufutu Wen, Xeron Du, Tianyu Zheng, Yichi Zhang, Letian Ni, Yuyang Cheng, Qiguang Chen, Jingzhe Ding, Shengda Long, Wangchunshu Zhou, Jiazhan Feng, Wanjun Zhong, Libo Qin, Ge Zhang, Wenhao Huang, Wanxiang Che, and Chenghua Lin. 2025. Beyond correctness: Evaluating subjective writing preferences across cultures.

Hong Yu and Mark Riedl. 2013. Data-driven personalized drama management. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 9(1):191–197.

A Appendix: Rushes Game Generation Pipeline

A.1 Configuration Parameters

Parameter	Value
LLM Model	GPT-4o
API Version	2024-10-01-preview
Expected Players	5000
Options per Level	4
Maximum Depth	4 levels per day
Speech Service	Azure TTS (Fable HD)
Image Model	FLUX.1 schnell

Table 9: System configuration parameters

A.2 Deriving the paraphrase scaling rule

Rushes pre-generates multiple surface realizations (paraphrases) for each option to reduce repeated wording across users. Let P be the expected number of players for a game, b the branching factor (number of options per node; in our setup $b = 4$), and d the depth index of a decision node (root at $d = 0$).

Assuming players are approximately evenly distributed across branches,[1] the expected number of players who reach a particular node at depth d is:

$$\mathbb{E}[\#\text{players at node depth } d] \approx \frac{P}{b^d}. \quad (1)$$

Each such node presents b options. Under the same uniformity assumption, the expected number of players who consider a particular *option* at that node is therefore:

$$\mathbb{E}[\#\text{players per option at depth } d] \approx \frac{P}{b^{d+1}}. \quad (2)$$

We aim to have enough paraphrase variants so that different players are unlikely to see identical surface text for the same option. Let $K(d)$ denote the total number of surface variants available for an option at depth d (including the original phrasing). A simple sizing rule is to set:

$$K(d) = \left\lceil \frac{P}{b^{d+1}} \right\rceil. \quad (3)$$

Since we store the original phrasing plus $V(d)$ additional paraphrases, we have $K(d) = V(d) + 1$, yielding:

$$V(d) = \left\lceil \frac{P}{b^{d+1}} \right\rceil - 1. \quad (4)$$

[1] This assumption is used only to size the paraphrase budget; the actual distribution may be skewed.

Variant assignment. At interaction time, a single variant is selected deterministically using a hash of (anonymized) `user_id` and the (`node_id`, `option_id`) pair. This provides stable per-user lexical variation without any on-demand generation.

A.3 Main Generation Pipeline

A.4 Story Setup and Theme Generation

A.4.1 Theme Extraction Prompt

A.5 Recursive Story Generation

A.6 Level Generation with Branching

A.7 Option Generation with Variations

A.7.1 Option Creation Prompt

A.8 Similarity Checking for Uniqueness

A.9 Option Expansion for Variation

A.10 Paraphrase Generation for Player Uniqueness

The system generates variations to ensure each player sees unique option text, calculated as:

$$n_{variations} = \left\lceil \sqrt{\frac{P_{expected}}{n_{options}^{depth+1}}} \right\rceil - 1 \quad (5)$$

where $P_{expected}$ is the expected number of players (typically 5000), $n_{options}$ is options per level (4), and $depth$ is the current tree depth.

A.11 Game Continuation Algorithm

For multi-day games, the system continues stories from active player paths:

A.12 Image Prompt Generation

A.12.1 Character Extraction and Management

A.13 Audio Generation with SSML

A.14 Media Generation Pipeline

GenerateNewGame: Create Complete Interactive Narrative

Require: $synopsis$, $game\_name$, $num\_options$, max_depth
Ensure: $game\_uuid$, complete game data

$game\_uuid \leftarrow$ GenerateUUID()
$setup \leftarrow$ LLM + StorySetup($synopsis$)
$theme \leftarrow setup.theme$ {Visual themes for consistency}
$results \leftarrow$ LLM + CreateStory(
    $synopsis$, $num\_options$, max_depth,
    $levels$, $checkpoint\_file$)
SaveToFile($game\_name$, $results.levels$, $theme$)
return $game\_uuid$

LLM Prompt

System Prompt:

```
TASK: Storywriting
INSTRUCTIONS: You are a writer tasked with creating visuals
for a short story based on a provided synopsis. Give the user
a concise but detailed description of the overall art style
of the story and look of the subjects.

For the medium, specify: digital art, illustration, oil painting,
3D rendering, photography, etc.
For the style, specify: impressionist, surrealist, pop art,
realism, fantasy, etc.
For the colors, list the main colors that should be used.
For lighting, specify: natural, artificial, neon, dark, bright, etc.
Include additional details using EXTRA that would help an artist.
Use only keywords and short phrases. End with 'END'.

EXAMPLE:
Synopsis: A detective investigates mysterious disappearances
in dystopian futuristic America.
OUTPUT:
MEDIUM: Digital art
ARTISTIC STYLE: hyperrealistic, fantasy, dark art
COLORS: iridescent gold, deep purple, midnight blue
LIGHTING: studio lighting, shadows at sharp angles
EXTRA: sci-fi elements, neon lighting, retro-futuristic tech
END
```

User Input: Synopsis: {synopsis}
Assistant Output: {theme}

CreateStory: Generate Branching Narrative Tree

Require: $synopsis$, $n\_options$, max_depth, $levels$
Ensure: Complete story tree with multiple paths
System: Set context as game design expert
User: "I want a story about {synopsis}. Begin writing and stop at CROSSROADS."
Assistant: $initial\_story \leftarrow$ LLM.generate(stop="CROSSROADS")
$levels$["start"] $\leftarrow$ {
    dialog: [$initial\_story$],
    depth: 1,
    menu: {buttons: []}
}
$levels \leftarrow$ GenerateLevel(
    $initial\_story$, depth=0, max_depth=max_depth,
    $n\_options$, level_id="start", $checkpoint\_file$)
return $levels$

GenerateLevel: Create Single Story Node with Options

Require: $story$, $depth$, max_depth, $n\_options$, $level\_id$
Ensure: Updated levels with new branches
Create level entry in $levels[level\_id]$ with $story$, $depth$
if $depth \geq max\_depth$ **then**
  return $levels$ {Reached maximum depth}
end if
$n\_variations \leftarrow \lceil \sqrt{EXPECTED\_PLAYERS/(n\_options^{depth+1})} \rceil - 1$
$options \leftarrow$ LLM + CreateOptions($n\_options$, $depth$, $n\_variations$)
$parent\_menu\_texts \leftarrow$ Extract titles from $options$
$seen\_options \leftarrow$ Accumulate seen options for uniqueness checking
for each new_level_id, $option$ in $options$ **do**
  Ensure new_level_id is unique (append counter if needed)
  User: "User chose: {option.action}"
  if $depth = max\_depth - 1$ **then**
    User: "This is the last level. Provide conclusion. ENDSTORY."
  else
    User: "Continue story, stop at next CROSSROADS."
  end if
  Assistant: $option\_story \leftarrow$ LLM.generate(stop=["CROSSROADS", "ENDSTORY"])
  $levels[new\_level\_id] \leftarrow$ Create new level with $option\_story$
  $levels \leftarrow$ GenerateLevel(
    $option\_story$, $depth + 1$, max_depth,
    $n\_options$, new_level_id)
end for
return $levels$

LLM Prompt

User Prompt:

```
Provide {n_options} short and descriptive options on what
the user could do next.

REQUIREMENTS:
- Each choice must be fully ACTIONABLE (not vague or mental)
- Each choice must be narratively and visually engaging
- Each choice must be unique in this level
- Titles must be in lowercase snake_case format

NARRATIVE GUIDANCE:
{depth > 0: "Unfold narrative smoothly while introducing
action-heavy, tense, and cinematic events."
else: "Since we're at the beginning, unfold smoothly with
actionable options without building tension yet."}

OUTPUT FORMAT:
<think>[Your reasoning for each option]</think>
OPTION 1: [title]: [option description]
OPTION 2: [title]: [option description]
...
OPTION {n_options}: [title]: [option description]
ENDOPTIONS
```

CreateOptions: Generate Diverse Action Choices

Require: $n\_options$, $depth$, $enforce\_unique$, $n\_variations$
Ensure: Set of unique, actionable options with variations

$options \leftarrow$ Empty dictionary
while len($options$) $< n\_options$ **do**
  User: Request $n\_options$ using format above
  Assistant: $response \leftarrow$ LLM.generate(stop="ENDOPTIONS")
  $parsed\_options \leftarrow$ ExtractOptions($response$) via regex
  for each $option$ in $parsed\_options$ **do**
    if $enforce\_unique$ **then**
      $is\_similar \leftarrow$ CheckSimilarity($option.text$, $seen\_options$)
      if $is\_similar$ **then**
        Continue {Skip similar option}
      end if
    end if
    if $n\_variations > 0$ **then**
      $expanded \leftarrow$ ExpandOption($option$, $n\_variations$)
      $option.variations \leftarrow expanded.variations$
      $option.details \leftarrow expanded.details$
      $option.outcome \leftarrow expanded.outcome$
    end if
    Add $option$ to $options$
  end for
end while
return $options$

LLM Prompt

System Prompt:

```
You are a story similarity checker.
Task: Determine if the provided option is overly similar to
any previous nodes that have been seen by the user.

Analyze similarity across these dimensions:
- Nature of Action (combat vs. dialogue vs. exploration)
- Complexity (simple vs. multi-step)
- Physicality (physical action vs. mental/social)
- Outcome (consequences and story progression)

OUTPUT FORMAT:
<think>[Your detailed analysis comparing current option
to previous nodes]</think>
<answer>[True or False: True ONLY if current option is
overly similar to a previous node, otherwise False]</answer>
ENDRESPONSE
```

User Input:

```
Previous nodes: {seen_options}
Current option: {current_option_text}
```

Assistant Output: {analysis + answer}

ExpandOption: Generate Detailed Variations

Require: $option, n\_variations$

Ensure: Expanded option with process details and outcome

System: "Generate concrete description of option and outcome."
User: "Option Title: {option[0]}\nOption Action: {option[1]}"
Assistant: $details\_response \leftarrow$ LLM.generate(
    format="DETAILS: Details/Process: ... Immediate Outcome: ...")
$details \leftarrow$ Extract from $details\_response$
$outcome \leftarrow$ Extract from $details\_response$
System: "Generate {n_variations} variations of Details/Process"
    "Keep title, action, outcome same. Vary only process."
Assistant: $variations\_response \leftarrow$ LLM.generate(
    format="VARIATION X: Details/Process: ...")
$variations \leftarrow$ Extract all variations via regex
return {details, outcome, variations}

LLM Prompt

Paraphrase Generation Prompt:

```
TASK: Generate Variations of Options
Given the current option, generate {n_variations} distinct,
actionable variations, keeping the general idea consistent.

RULES:
1. Preserve every piece of context from the original:
   - Character names, locations, roles, relationships
   - Specializations or backstory
2. Each variation must:
   - Begin by restating essential context
   - Offer fresh style or approach in two sentences
   - Avoid repeating exact wording while keeping details
   - Not assume prior knowledge

FORMAT:
VARIATION X:
Option Action: [brief two sentence description]

Stop when you have exactly {n_variations} variations.
Print ENDOPTIONS.
```

ContinueGame: Extend Game from Active Storylines

Require: $game\_uuid$, $current\_day$, $n\_storylines$, $num\_options$
Ensure: New day's story branches

$game\_data \leftarrow$ LoadFromDatabase($game\_uuid$)
$levels \leftarrow$ LoadFromDatabase($game\_uuid$, $current\_day$)
$story\_tree \leftarrow$ GenerateStoryTree($levels$, root="start")
$current\_depth \leftarrow 5$ {End of previous day}
$storylines \leftarrow$ GetActiveStorylines($votes\_db$, $game\_uuid$, $current\_depth$)
$stories \leftarrow$ Map storylines to story text from $story\_tree$
if len($stories$) = 0 **then**
  return {No active players}
end if
$merged \leftarrow$ LLM + MergeOptions(
    $n\_storylines$, $num\_options$, $stories$, $levels$, $current\_depth$)
$next\_day \leftarrow current\_day + 1$
$results \leftarrow$ LLM + ContinueStory(
    $synopsis$, $num\_options$, $current\_depth + 2$, $current\_depth$,
    $merged$, $story\_tree$, $levels$)
SaveToDatabase($game\_uuid$, $next\_day$, $results.levels$)
return $results$

GenerateImagePrompt: Create Stable Diffusion Prompts

Require: $themes$, $caption$, $subjects$
Ensure: Image prompt for scene

System: "Extract characters from text. Convert to snake_case."
    "Use existing names if already in EXISTING SUBJECTS."
User: "EXISTING SUBJECTS: {subjects.keys()}\nINPUT: {caption}"
Assistant: $scene\_subjects \leftarrow$ LLM.generate(format="char1, char2 ENDOUTPUT")
Parse $scene\_subjects$ into list
for each $subject$ in $scene\_subjects$ **do**
  if $subject$ not in $subjects$ **then**
    System: "Create detailed character description."
      "Format: species, gender, age, appearance, clothing, traits"
      "Must be fully clothed and appropriate."
    User: "Create character named: {subject}"
    Assistant: $char\_desc \leftarrow$ LLM.generate(stop="END")
    $subjects[subject] \leftarrow char\_desc$
  end if
end for
System: "Create detailed Stable Diffusion prompt."
    "Third-person, vivid visual details, comma-separated."
    "Match themes: {themes}"
    "AVAILABLE CHARACTERS: {subjects for scene_subjects}"
User: "I want an image about: '{caption}'"
Assistant: $image\_prompt \leftarrow$ LLM.generate()
return $image\_prompt$, $subjects$

LLM Prompt

Audio Prompt Generation:

```
TASK: Audio Synthesis
You are an expert in generating audio prompts for text.
Generate SSML to narrate the scene, including character
dialogue in distinct voices.

- Use voice `en-US-FableMultilingualHD` for narration
- Set mstts:express-as style to `narration-professional`
- Use <prosody> for rate, pitch, volume adjustments
- Use <mstts:express-as> for character roles and styles

OUTPUT FORMAT: <speak>SSML Prompt</speak>

INPUT: {caption}
```

GenerateGameMedia: Create Images and Audio

Require: $game\_uuid$, day
Ensure: Image prompts and audio files

$levels \leftarrow$ LoadFromDatabase($game\_uuid$, day)
$game\_data \leftarrow$ LoadFromDatabase($game\_uuid$)
$images\_data \leftarrow$ LoadFromDatabase($game\_uuid$, day, type="images")
$theme \leftarrow images\_data.theme$
$subjects \leftarrow images\_data.subjects$
$setup \leftarrow$ LLM + StorySetup($synopsis$)
$theme \leftarrow setup.theme$
$subjects \leftarrow \{\}$
$dialog\_texts \leftarrow$ Extract dialog from all levels
for each $level\_id$, $text$ in $dialog\_texts$ **do**
  if $level\_id$ not in $image\_prompts$ **then**
    $prompt \leftarrow$ LLM + GenerateImagePrompt($theme$, $text$, $subjects$)
    $image\_prompts[level\_id] \leftarrow prompt.image\_prompt$
    $subjects \leftarrow prompt.subjects$ {Update character registry}
    SaveCheckpoint($image\_prompts$, $checkpoint\_file$)
  end if
end for
SaveToDatabase($game\_uuid$, day, $image\_prompts$, $subjects$, $theme$)
$audio\_texts \leftarrow$ Extract dialog texts
$job\_id \leftarrow$ "{game_name}-{day}"
GenerateAudioBatch($audio\_texts$, job_id) {Azure TTS}