IMET - International Conference on Interactive Media, Smart Systems and Emerging Technologies (2023)
N. Pelechano, F. Liarokapis, D. Rohmer, and A. Asadipour (Editors)

Short Paper

Investigating Crowdsourced Help Facilities for Enhancing User Guidance

Sooraj K Babu, Tobias Brandner, Samuel Truman, Sebastian von Mammen

Games Engineering, Julius-Maximilians University, Würzburg, Germany

Abstract

We present two help facilities aimed at addressing the challenges faced by users new to a software. Software documentation should ideally familiarise with the overall functionality, navigate through the concrete workflows, and opportunities for customisation. Large degrees of freedom in the use and vast basic scopes, however, often make it infeasible to convey this information upfront. Rather, it needs to be broken into helpful bits and pieces which are delivered at appropriate times. In order to address this challenge, we have designed two concrete help facilities, i.e. tooltips and tips-of-the-day, to feature crowdsourced information. In this paper, we present the results of a formative study to implement early proofs-of-concept for the open-source game authoring platform Godot. To support further research, we have made the code base publicly available on GitHub.

CCS Concepts

• ***Human-centered computing*** → *User studies;* ***User centered design;***

1. Introduction

The development of complex games from scratch can be a difficult and time-consuming task. Game authoring platforms, like Unreal Engine and Unity3D, have emerged aiming to provide a more manageable and feasible solution to address this challenge [Sch17]. These platforms enable the creation of diverse applications across various domains, including entertainment, education, and research [Haa14]. However, while an experienced user might find the creation process easy, a novice typically needs support to learn the tools. Therefore, help facilities play an essential role in teaching and guiding users to gain a positive learning experience [CSKFMR87]. Help facilities can be divided into text-based help, video help, and interactive help. [Kao20] reviewed 85 game authoring platforms and almost 90% of them offered text documentation, about 50% video tutorials, and only 20% interactive tutorials. [KKW17] pointed out that providing traditional help facilities such as text-based documentation is laborious, especially when it comes to maintaining them. They observed that tutorials are frequently produced by users rather than developers and disseminated through blogs or social media platforms, also pointed out in [Wyr14], and well-documented by numerous tutorial videos on the use of authoring platforms on YouTube [KABAAR20]. This kind of content can become competitive and official, i.e. Professionally Generated Content [Kim12]. Epic Games, the creators of Unreal Engine, outsourced their video tutorial creation to content creators on YouTube [Gam22]. Considering these facts, we started exploring user-generated content as an additional help resource: Individual users may, for instance, provide videos, tutorials, and articles that explain certain features of authoring platforms [Wyr14]. In addition, we use crowdsourcing to obtain user-generated content and maintain it. In software documentation, crowdsourcing has been shown to improve documentation quality and coverage [MCHJ17]. For example, [PTGS12] investigated crowdsourced documentation in the context of online question-and-answer forums like StackOverflow. The crowd generated a rich content source by voting on individual contributions. For example, 87% of the Android authoring platform was covered by code examples and discussions from over 35,000 developers and viewed over 70 million times. Or in other words, crowdsourcing can be considered a massive help facility management system, which can split the content generation process into small tasks handled by multiple users [MCHJ17]. This paper introduces two novel help facilities for authoring platforms that utilise user-generated content and crowdsourcing: (1) "The enhanced tooltip" and (2) "The community tip of the day". The primary objective of the proposed assistance facilities is to acquaint users with the functionalities provided by the platform by presenting detailed descriptions as tooltips, guiding users through diverse workflows to accomplish tasks by providing additional resources and finally informing users about the various customisation options available within the platform by explaining their usage. We present the user-centred engineering process of proof-of-concept implementations of both ideas for the Godot Engine. We discuss the findings of early feedback that we have gathered in the process. The designs are applicable to a wide range of authoring and software systems in general, as the contents are only limited by the expertise of the user communities. We conclude this paper with short

DOI: 10.2312/imet.20231252

summaries of our contributions, potential future work in terms of improvements and functional extensions.

2. Concept & Design

The concept of learnability in software design has been studied in the field of Human-Computer Interaction. However, the term learnability has multiple definitions, leading to the need for a taxonomy to classify learning issues effectively. Grossman et al. developed an according taxonomy that categorises learning issues into initial and extended learning issues based on the challenges faced by novice and experienced users, respectively [GFA09]. In their work, they suggested that the well-established help facilities like tooltips and tips-of-the-day address the initial learning issues, i.e. the problem for a novice user to accomplish a particular task within a software efficiently. As a next step, we augmented their base functionality with crowdsourcing, i.e. allowing the community of users to contribute to the interactive documentation and guidance and have them rate and filter the results at the same time. The extended help facilities are useful in familiarising the users with various system components and navigating the users across various workflows of the authoring tool, enabling them to customise various system states. The enhanced tooltip incorporates user-generated content to aid novice users in overcoming initial learning difficulties while using a game-authoring platform. By embedding concise and targeted content within the workflow, users can extend the existing components within the software (e.g., game objects, assets, scripts) with helpful tips, which benefit other users working with them. Similar to the Tool-Clip concept introduced by Grossman et al., [GF10] we enhance our tooltip with the ability to link additional information like blog posts or videos or a video, and a rating system to manage further and improve the tips. For the Community Tip of the Day, we enhanced the traditional tip-of-the-day feature by leveraging crowdsourcing to convey deeper knowledge and fill documentation gaps. The tip-of-the-day typically consists of brief explanations of random platform features provided by the developers. Our proposed enhancement includes two key elements: (a) An improved behaviour similar to the enhanced tooltip, where tips are contextually displayed, and (b) the integration of a crowd voting system using a simple ranking algorithm. This allows platform users to contribute their own tips and rate the tips provided by others. By actively participating in tip creation and rating, users are encouraged to commit to their own tips and provide feedback on the tips shared by the community. This approach aims to harness the collective knowledge of the crowd, providing more comprehensive and relevant information to users while fostering engagement and collaboration within the platform community.

2.1. Human-Centred Design

We followed a user-centred engineering process to develop both tools. In this iterative process, feedback was consistently gathered from a group of experts in the fields of application game design and game development to determine the effectiveness of the gathered requirements, proposed solutions, and implementations [RF14]. We approached domain experts ($n = 10$) from an academic environment, including professors, PhD students, and students with a Human-Computer Interaction background. We solicited weekly feedback sessions. The gathered and refined requirements are summarised in Table 1 emerged. As a result, the enhanced tooltip appears while hovering over an accordingly augmented game object. It disappears after some time or when the mouse pointer leaves the game object's vicinity. The community tip of the day is displayed when the software is started. Tips can be browsed and entered ones are previewed before uploaded. Both help facilities feature embedding of external, linked media and rating facilities alongside accumulated rating values. Any users' inputs are disseminated to all the others and the likelihood of showing a particular tip is linked to its overall approval. The enhanced tooltip is shown in Figure 1, the community tip of the day in Figure 2. For the design and refinement process of the UI mock-ups we relied on Figma. To blend into the target environment Godot, we adopted its default UI colour palette [Mun22]. Commonly known community-based platforms such as StackOverflow inspired the design of the buttons.

Requirement	E	C
Addable to a game object	✓	
Visible when hovered over the object	✓	
Visible when hovered over the tip's GUI	✓	
Disappears when object/tip does not hover over	✓	
Disappears after a certain time period	✓	
Shown in its own screen space		✓
Tips can be cycled through		✓
Shows preview before upload to database		✓
Shows upload date and engine version		✓
Clickable blog and video links	✓	✓
Shows ratings next to like/dislike buttons	✓	✓
Share updated ratings with all users	✓	✓
Calculate relevance based on likes/dislikes	✓	✓

Table 1: *Requirement list for the enhanced tooltip (E) and the community tip of the day (C)*

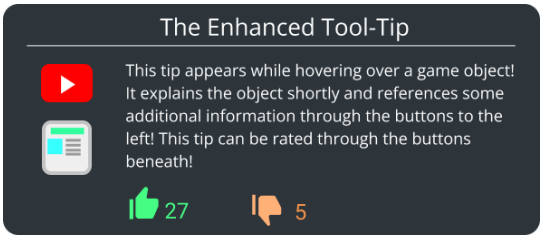

Figure 1: *The final version of the enhanced tooltip. The pop-up window has a title at the top, a description text to the right, a blog/video button to the left, and a like/dislike button with a count at the bottom.*

3. Implementation

The proofs-of-concept were implemented as Editor-plugins for Godot, which allow for extending the UI of the engine, and were written in GDScript [tea22]. GDScript is Godot's scripting language and has a syntax similar to Python. Both plugins communicate with a serverless and self-contained SQLite database. The

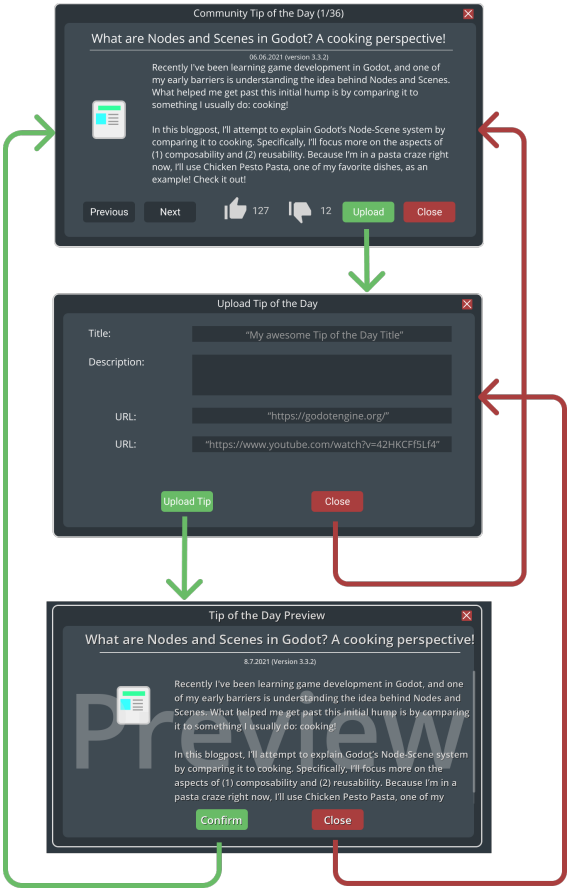

Figure 2: *From top to bottom: The 'Tip', 'Upload' and 'Preview' windows show the interaction cycle for contributing to the community tip of the day.*

database engine is embedded and runs as part of the application, making it an amicable solution for fast prototyping. We used an SQLite plugin for Godot, which allows accessing a *.db* database file. Godot organises its essential data components as nodes. We implemented the enhanced tooltip as a 'custom node plugin'. This lets us easily attach it to game objects and automatically get information about its properties. As a result, once added, the enhanced tooltip's title is automatically inferred from the associated game object, and a corresponding database entry is created. The entry saves an ID for referencing and voting data. The community tip of the day was implemented as a 'main screen plugin', another subtype of Godot's 'editor plugin' which extends the editor's interface and allows the creation of new UIs. For easier access, we extended Godot's tab system and added a custom tab called 'Community tip of the day'. Clicking it would display a tip from a sorted list read from the database. A ranking algorithm sorts the tips with a score calculated based on the following equation:

$$score = \underset{ratio}{(upvote - downvote)} - timeelapsed \qquad (1)$$

Any user should be provided with the most helpful tip. Generally, it is sufficient to calculate a ratio, i.e. the simple difference between the up-and-down votes. But in this case, showcasing the tip with the highest ratio leads to showing the same tips repeatedly and does not give new tips a chance to be appraised by the users. To solve this issue, a *time decay function* is taken into account, subtracting the time since the tip was uploaded in days from the ratio between up-and-down votes. This enables supplying the users with new content and keeping them engaged. Famous social media blog-post platforms, like Reddit or HackerNews, use more complex, exponential time decay functions. Still, the basic principle remains the same [Sat22, LR17]. The code base is publicly available on GitHub - *https://github.com/BrandnerKasper/CrowdbasedHelpFacilites.*

4. Proof-of-Concept Evaluation

In addition to the formative evaluation during the user-centred engineering process, we also conducted a very early summative evaluation of the two proof-of-concept implementations with novice users ($n = 4$), i.e. users that were both new to using the Godot engine and game development as a whole, but had basic programming skills. In the tests, the subjects were asked to familiarise themselves with a basic Godot demo project that aims to teach the users how to use the engine. While doing so, they were encouraged to think aloud to gain insights into the subjects' thought process about the help facilities [BR00]. Afterwards, we conducted semi-structured interviews in which we asked about the two help facilities' usability. We had attached a tooltip to every important object within the game, like the player or the enemy. When the user starts playing, the tooltip over certain game objects appears, displaying a short description of the game objects and referencing their documentation during run time. The participants agreed that it benefited their understanding of Godot's principles and core mechanics. They compared the tips to helpful comments inside a code project, while blog and video buttons saved them from searching for contextual and more in-depth explanations themselves. One known drawback in the proof of concept was the display of tooltips during runtime but not in the edit mode. Still, participants in the study suggested several enhancements. They noted that in complex scenes, it's unclear which game objects have attached tips, recommending visual cues like info icons or sparkle effects. They also proposed displaying tooltips in Edit mode for a more natural experience and deactivating tips once revealed. Despite these shortcomings, participants acknowledged the potential benefits, suggesting that such a feature should be a standard in authoring platforms, particularly for infrequently used tools or UI elements.

The evaluation setup for the community tip of the day was a little more difficult. We decided to test only how users benefit from the tips rather than how they enter them. Since the tips could cover a wide range of topics, two types of tips were chosen. The first type was about the general functionality of the engine, for example, how nodes and scenes work in Godot, explained by an analogy about cooking The second type was about a specific task the engine could fulfil, for instance, how a 2D particle node can be used to visualise an animated torch Evaluating the usefulness of the community tip of the day was not easy for the participants. It depended highly on the content explained in the tips and which tip was shown. Sometimes, the contents of the tips were already known, either in part or entirely. Occasionally, the tips were exciting but not quite helpful in

their current context. Nevertheless, the participants appreciated this help facility and its flexible access, likening it to a video game loading screen. They desired more interactivity, including the ability to comment on tips, bookmark favourites, and sort tips by categories or difficulty levels, such as '2D,' 'Particle System,' or 'Procedural Generation,' and 'beginner,' 'intermediate,' or 'expert.' These additions would allow users to tailor their tip preferences. The participants suggested having a user management system to save tips, follow users, etc., but allowing this was unfortunately out of the scope of this current version. The biggest concern raised by the participants was the ability to misuse the crowdsourcing features. At the moment, users can commit any content they want to, even inappropriate ones. The only means to stop spreading inappropriate content in the current version was by disliking and down-voting it, a relatively ineffective way to handle the problem. An effective solution would require incorporating moderation tools. This is a concern that was considered in the meeting with the experts but was seen as too complex to address for this version.

5. Discussion & Future Work

This paper explored using user-generated content as a help facility and managing it with crowdsourcing. Two concepts and according proof-of-concept implementations were developed and tested: The enhanced tooltip, an extended tooltip referencing additional information, and the community tip of the day, a tip-sharing platform managed by the user base. As the objective of a user-centred engineering process, both ideas underwent multiple iterations. In addition, each proof-of-concept was summatively evaluated. The results showed that both approaches could help to document game authoring platforms and support their use. Yet, numerous concrete improvements and functional extensions were demanded, including more customisation features and better integration into contextual workflows. The presented help facilities aim to serve as a valuable resource that goes beyond the traditional role of assisting users in understanding software. These facilities address topics that are often challenging for conventional help systems to cover comprehensively. Additionally, the concept of incorporating crowdsourcing into user guidance has the potential to be applied to a wider range of user scenarios beyond authoring. The greatest practical concern with respect to proof-of-concept implementations is dealing effectively with inappropriate content. This problem could be solved by moderating and filtering, which would require a more extensive back-end infrastructure, including user account management. At the same time, this would also benefit many of the additional functionalities the users asked for during the summative tests. In addition, The current Community Tip of the Day may favour experienced users over novices. We propose addressing this by sorting tips in the future based on both expertise and knowledge level. Additionally, the paper lacks a robust evaluation; potential improvements include creating a dedicated project for enhanced tooltip assessment, gathering more tips for user evaluation, and considering a shift to a crowdsourced FAQ to aid novice users.

References

[BR00] BOREN T., RAMEY J.: Thinking aloud: Reconciling theory and practice. *IEEE transactions on professional communication 43*, 3 (2000), 261–278. 3

[CSKFMR87] CARROLL J. M., SMITH-KERKER P. L., FORD J. R., MAZUR-RIMETZ S. A.: The minimal manual. *Human-computer interaction 3*, 2 (1987), 123–153. 1

[Gam22] GAMES E.: Unreal online learning - unreal engine, 2022. URL: https://www.unrealengine.com/en-US/onlinelearning-courses. 1

[GF10] GROSSMAN T., FITZMAURICE G.: Toolclips: an investigation of contextual video assistance for functionality understanding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2010), pp. 1515–1524. 2

[GFA09] GROSSMAN T., FITZMAURICE G., ATTAR R.: A survey of software learnability: metrics, methodologies and guidelines. In *Proceedings of the sigchi conference on human factors in computing systems* (2009), pp. 649–658. 2

[Haa14] HAAS J.: A history of the unity game engine. *Diss. WORCESTER POLYTECHNIC INSTITUTE* (2014). 1

[KABAAR20] KADRIU A., ABAZI-BEXHETI L., ABAZI-ALILI H., RAMADANI V.: Investigating trends in learning programming using youtube tutorials. *International Journal of Learning and Change 12*, 2 (2020), 190–208. 1

[Kao20] KAO D.: Exploring help facilities in game-making software. In *International Conference on the Foundations of Digital Games* (2020), pp. 1–14. 1

[Kim12] KIM J.: The institutionalization of youtube: From user-generated content to professionally generated content. *Media, culture & society 34*, 1 (2012), 53–67. 1

[KKW17] KÄFER V., KULESZ D., WAGNER S.: What is the best way for developers to learn new software tools? an empirical comparison between a text and a video tutorial. *arXiv preprint arXiv:1704.00074* (2017). 1

[LR17] LEAVITT A., ROBINSON J. J.: Upvote my news: The practices of peer information aggregation for breaking news on reddit.com. *Proc. ACM Hum.-Comput. Interact. 1*, CSCW (dec 2017). URL: https://doi.org/10.1145/3134700, doi:10.1145/3134700. 3

[MCHJ17] MAO K., CAPRA L., HARMAN M., JIA Y.: A survey of the use of crowdsourcing in software engineering. *Journal of Systems and Software 126* (2017), 57–84. 1

[Mun22] MUNRO L.: The role of color in product design: Ux of color palettes, 2022. URL: https://xd.adobe.com/ideas/principles/web-design/ux-of-color-palettes/. 2

[PTGS12] PARNIN C., TREUDE C., GRAMMEL L., STOREY M.-A.: Crowd documentation: Exploring the coverage and the dynamics of api discussions on stack overflow. *Georgia Institute of Technology, Tech. Rep 11* (2012). 1

[RF14] RICHTER M., FLÜCKIGER M.: User-centred engineering. In *User-Centred Engineering*. Springer, 2014, pp. 11–24. 2

[Sat22] SATOSHI I.: How are popular ranking algorithms such as reddit and hacker news working?, 2022. URL: https://medium.com/jp-tech/how-are-popular-ranking-algorithms-such-as\-reddit-and-hacker-news-working-724e639ed9f7. 3

[Sch17] SCHREIER J.: *Blood, sweat, and pixels: The triumphant, turbulent stories behind how video games are made*. Harper New York, 2017. 1

[tea22] TEAM G. E.: Plugins - godot engine (stable) documentation, 2022. URL: https://docs.godotengine.org/en/stable/tutorials/plugins/index.html. 2

[Wyr14] WYRWOLL C.: User-generated content. In *Social Media*. Springer, 2014, pp. 11–45. 1