Playing with Knowledge: Leveraging Visualization Games for Data Validation and Inspiration

F. Becker[1], R. P. Warnking[2], T. Blascheck[1]

[1]University of Stuttgart, Institute for Visualization and Interactive Systems, Germany
[2]Heidelberg University, Medical Faculty Mannheim, Department of Biomedical Informatics, Germany

Abstract

We present an approach to use visualization games for data validation and inspiration in a collaborative coding context. As part of an interactive coding system that lets coders create a tag hierarchy and tag data items, we designed multiple games that support validating that data and exploring it in a novel way. Each game has mechanics inspired by existing games and incorporates visualization and externalization to varying degrees. By playing these games, coders randomly sample the data space to pinpoint problems and find inspiration, like discovering gaps in the data or contemplating novel item-tag combinations. Game results are automatically tracked to let coders analyze their performance and find out in which cases they tend to make mistakes. Coders can also create objection notes at the end of a game to externalize insights which are accessible in other parts of the system. For example, if a coder is convinced that an item should not have a specific tag they were shown in a game, they can create an objection about this issue that all system users can see. Our games can be played with different datasets at `https://arielmant0.github.io/collacode/?tab=games`.

CCS Concepts

• ***Human-centered computing*** → ***Visualization;*** • ***Applied computing*** → *Computer games;*

1. Introduction

When coders tag data during qualitative data analysis [Bry02] (QDA) or social tagging [GLYH10], validating the resulting data can get increasingly difficult if tags are numerous or have complex relationships. Visualization and automated methods can direct coders towards data that may need revisiting. For example, visualizing the distribution of tags or calculating inter-coder agreement [Hal24] can help coders understand their data and diagnose problems. In addition to these methods, we propose games as another part of the toolkit to validate user-created data and to support thinking about the data in new ways. To test this approach, we integrated multiple games into an existing system for collaborative coding. The dataset used for this article comes from a coding process in which three coders tagged the properties of video games [BWBB25] using this coding system [BWB25]. Inspired by existing games like *GeoGuessr*, our games let coders explore and validate their own data in a playful manner, making use of visualizations to varying degrees.

In contrast to conventional data validation workflows, games create additional motivation to engage with the data by crafting challenges that can be completed in a short time. They also relieve coders of the burden of having to find problematic data themselves because they randomly sample the dataset. To make gameplay insights actionable, we automatically track game results and allow coders to directly create data-bound *objection notes* based on the results. Thereby, we prevent losing insights with little effort on the coder's side and simultaneously connect game results to other system components. Our approach overlaps with different topics from games-related research, like gamification, serious games, and gameful design. Which concepts fall under the umbrella of *gamification* [AFF19] is contested, but it broadly considers the use of game elements and characteristics in design contexts outside of games [DDAA15]. Deterding et al. [DDAA15] propose the adoption of *gamefulness* as a complement to *playfulness* [MS22], introducing *gameful design* as "designing for gamefulness, typically by using game design elements." The most basic types of game elements often associated with gameful design include scores, leaderboards, and badges. These elements aim to create extrinsic motivation, but games can also create intrinsic motivation by making players feel accomplished, clever, autonomous, or creative [Cho15]. Serious games "have an explicit and carefully thought-out educational purpose and are not intended to be played primarily for amusement," as so put by Abt [Abt87]. Therefore, serious games are separate from gamification because they are explicitly games, but with the goal to educate rather than to entertain. Another closely related idea are *games with a purpose* (GWAP) [vAD08], which refer to games that have a *serious* purpose other than amusement, but are not limited to education. Although our games can educate players about data characteristics, they do not target education.

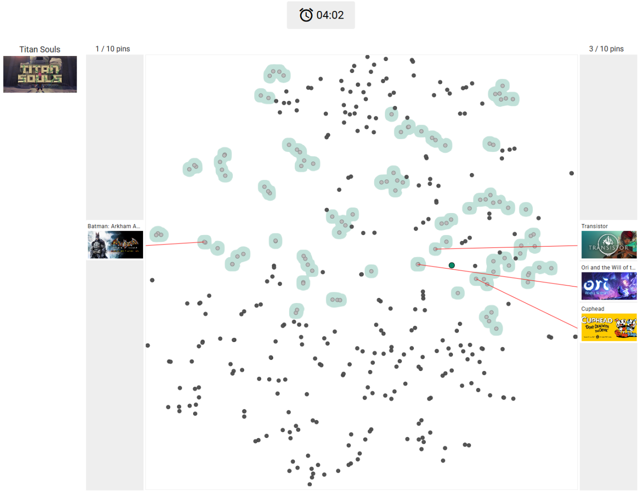

Figure 1: *Example screenshot of the gameplay for* ***Where Am I?*** *showing the target item to place (top left) and the scatter plot depicting other items in the dataset. On both sides of the scatter plot, the player can pin a limited set of items to keep track of relevant item locations. Already visited items in the plot are indicated through background contours and a different fill color.*

We understand our games as GWAP combined with visualization that let coders validate data by uncovering inconsistencies between their mental models and the actual data. They let coders validate data in a more playful and engaging way, randomly sampling the data to find even well-hidden problems or suggest previously unconsidered data combinations. The code for the interactive coding tool, including code for the games described here, is available at `https://github.com/ArielMant0/collacode`.

2. Games

This section details five games we implemented as part of an interactive system for collaborative coding [BWB25]. For each game, we describe its goal, the rules of play, its use of visualization, and how it supports data validation. All games use the currently selected dataset in the system as their data source. Each game has three levels of difficulty (easy, normal, hard) that affect the gameplay in different ways, like changing the time limit, allowing the player to exclude items they do not know, or restricting mechanics.

2.1. Game 1: Where Am I?

GeoGuessr (`https://www.geoguessr.com/`) is a popular game in which players see images of a random place on earth with the goal of identifying the location. The game **Where Am I?** follows the same principle, but adapts it to our specific context. Instead of images, we show a scatter plot of the data, produced from tag assignments with dimensionality reduction algorithms. The goal of the game is to place a target item at its correct location in the scatter plot, using other data points and previous knowledge about tag assignments as a guide.

Game Manual. When the game starts, the player can see the scatter plot and the target item as shown in Figure 1. The player has limited time to choose the location where they suspect the target item to be. When hovering over data points in the scatter plot, two things happen: First, a tooltip shows the names and images of all items around the cursor in the plot. Second, already visited data points are highlighted through semi-transparent contours and different coloring. The player can click on any location in the scatter plot to place their guess, which is locked in when the timer runs out or the player clicks the *submit* button. Upon ending, the game shows the end screen where the correct location of the item is revealed and the pixel distance to the guessed location is shown via text and as a dashed line in the plot. The game is considered to be won when the pixel distance is less than a given threshold, which depends on the scatter plot's size in pixels.

Visualization. During the game, players can see and interact with the scatter plot as described above. The player can pin a limited number of items by right-clicking them in the plot, which displays the item on either the left or the right side of the plot. Pinned items are connected to their respective location in the scatter plot with a red line (cf. Figure 1). In the end screen, players see a bar code visualization for the target item and the guessed item (if guessed incorrectly). A bar code displays all available (leaf) tags from the tag hierarchy in an ordered fashion, coloring those assigned to the respective item in a different color, similar to the bar code visualizations in Figure 4.

Validation. Players use their knowledge about item similarity, derived from tag assignments, to find the correct location in the embedding. Although the chosen dimensionality reduction method can be misleading if players are not familiar with its general properties, they can still validate whether items are placed in the correct neighborhood, comparing their mental model to the embedding.

2.2. Game 2: Who Am I?

The **Who Am I?** game is based on *Guess Who?* and asks the player to guess one item from a random subset of items in the dataset. The goal of the game is to find the correct item within a specified number of questions by asking whether the target item has a specific tag from the tag hierarchy.

Game Manual. The game lets the player ask a limited number of questions to gather information about the target item. Each question allows the player to *ask* about one tag from the tag hierarchy. They do so by clicking on a tag in the treemap (cf. Figure 2) and confirming their choice with the *ask* button. Then, the game gives the player feedback about their guess from three possible options: (green) the target item has the tag, (yellow) the target item has a sibling tag, or (red) the target item does not have the tag or any sibling tag. When the player submits their guess, the game ends. If they guessed the correct item, they won.

Visualization. During gameplay, the player can see the tag hierarchy as a treemap with uniform weights for all leaf nodes. Because the treemap is also used in other parts of the system, players are familiar with its structure. The treemap is used to ask whether the

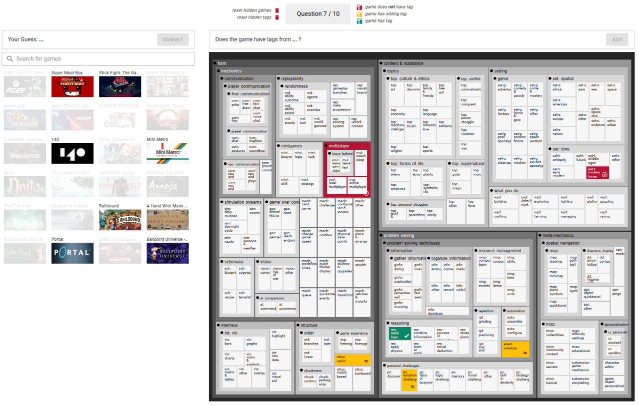

Figure 2: *Example screenshot of the gameplay for* **Who Am I?***, showing the set of possible items (left) and a treemap of the tag hierarchy (right). The player can de-emphasize items and tags to mark them as excluded, which lowers their rendering opacity. The answers to the player's questions are encoded through color in the treemap, described by the legend at the top of the page.*

target item has a given tag and the answer is visualized via one of three colors. Compared to other tree visualizations, a treemap nicely visualizes the feedback semantics due to its nested layout. Both items and tags can be *de-emphasized* by right-clicking them, which reduces their opacity, allowing the player to externalize parts of their mental model. The end screen contains a bar code visualization for both the guessed and correct item, supporting the same features as described for the game **Where Am I?**

Validation. Players need to uncover the correct item from just its tag assignments, thereby validating their understanding of tag assignments for the chosen item subset. The end screen allows players to see all questions and answers, so that they can reflect about their thought process and possible errors.

2.3. Game 3: Trivia

The **Trivia** game is similar to other types of quiz games and consists of multiple questions about items and tags in the dataset. It has a timer for each question, so that players must be quick to answer. The goal of the game is to answer all questions correctly.

Game Manual. For each question, the player is given the question text and a set of possible answers (cf. Figure 3). After choosing an answer, the game gives feedback whether it is correct and then proceeds to the next question after a short waiting period. If the question timer runs out, the game counts this as a wrong answer and proceeds. After the last question, the player is shown a summary of their results in the end screen. The game is considered to be won when all questions are answered correctly. Currently, the game offers four types of questions:

1. Which tag is this item [not] tagged with?
2. Which item is [not] tagged with this tag?
3. Which of these items has the [most | least] tags?
4. Which item does not belong here?

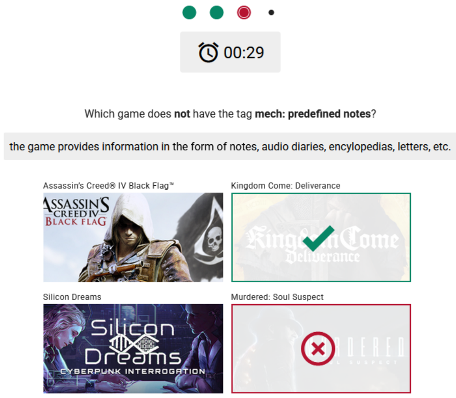

Figure 3: *Example screenshot of the gameplay for* **Trivia***, showing one question type. The dots at the top of the page indicate which question the player is currently at, how they have performed thus far, and how many questions are still to come. After choosing an answer, its correctness is shown via colors and icons.*

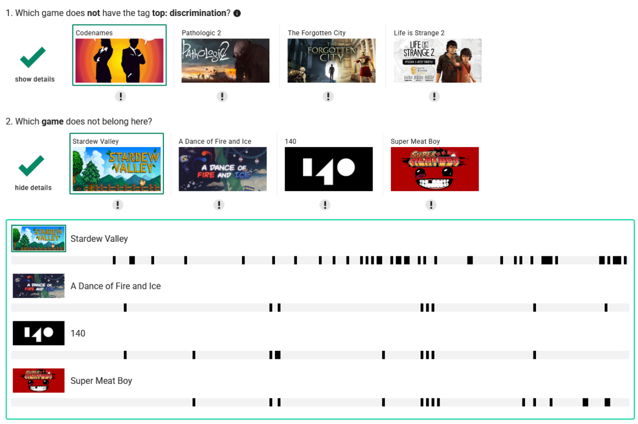

Figure 4: *Partial screenshot for the end screen in* **Trivia***, showing two of the questions with all answer options, the player's chosen answer, and the solution. Below any item or tag, a button with an exclamation mark lets the player create a new objection note. Players can explore tags for any shown item on demand by clicking the "show details" button of a question, which shows interactive bar code visualizations for each items (bottom).*

Visualization. This game does not include any visualizations during gameplay, but the end screen lets the player see bar codes for all question-related items on demand (cf. Figure 4).

Validation. **Trivia** lets the player validate specific tag assignments and test their high-level knowledge about the dataset. Because the game chooses random combinations of items and tags, it can show the player combinations they had not previously considered. This can inspire new ideas or call the existing tagging into question, which can be persisted via objection notes in the end screen.

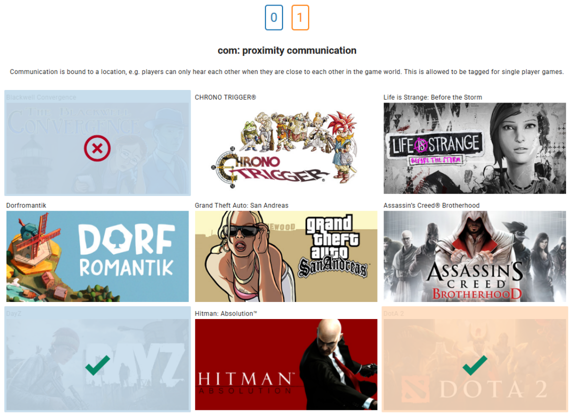

Figure 5: *Example screenshot of the* ***Set*** *game, showing the target tag (top) and the list of possible items (bottom). Items already taken by a player have reduced opacity, an item's color indicates which player took it, and correctness is shown via colored icons.*

Figure 6: *Example screenshot of the gameplay for* ***Matching****, showing items (left) that must be assigned to their respective tags (right). Tags are displayed as a text list and a bar code visualization, which are connected via hover and click interactions.*

2.4. Game 4: Set

Based on one of the **Trivia** questions, we designed a competitive multiplayer game in which up to five players must find all items with a specific tag. The goal of the game is to get the highest score—each correctly chosen item adds a point and each wrongly chosen item deducts one. In our scenario, the server that stores and serves the data also manages a list of active multiplayer lobbies and brokers the initial connection. The game itself is played using peer-to-peer communication, which comes with unique challenges [NPVS07] like managing the game state. We treat the host of a lobby as the game manager, the host makes decisions based on what other players send them and then distributes updates. Using a peer-to-peer system prevents high load on the server and lets us easily add or remove games by only changing frontend code.

Game Manual. The game starts with a countdown, after which all players are given the target tag and a list of items, as shown in Figure 5. When a player clicks on an item, it counts as *taken* by them and is no longer available for other players. As soon as all matching items are taken, the game ends and the end screen is displayed. The player with the highest score wins. Because taking wrong items deducts points, a player can even win with negative or zero points as long as other players have fewer points. A draw does not count as a win.

Visualization. This game does not include any visualizations during gameplay, but the end screen lets players see bar codes for all items on demand, similar to **Trivia** (cf. Figure 4).

Validation. The short duration of the game makes it more suited to testing players' knowledge than validating data. However, the end screen has a similar structure to the **Trivia** game, allowing players to rethink their choices and create objection notes.

2.5. Game 5: Matching

In the **Matching** game, players are tasked with matching a list of items to their respective set of tag assignments in the allotted time.

Game Manual. First, the player is given a list of items and sets of tags, as shown in Figure 6. They can drag items into a slot next to a set of tags to make an assignment. When the timer ends or the player submits their guess, the game proceeds to the end screen, where the player can compare their assignments to the solution. The game is considered to be won when all assignments are correct.

Visualization. The tags for each item are shown not only as a text list but also visualized with a bar code. Players are given several interactions to help them and enable externalization during gameplay. They can hover over a tag (in the bar code or the list) to highlight that tag for all tag sets. In the bar code, this changes the color of the respective bar. In the text list, it makes the respective tag name bold. Clicking on a tag in either representation makes the highlight permanent while right-clicking a tag de-emphasizes it by reducing the opacity in both representations.

Validation. The **Matching** game lets players validate their mental model of the tag assignments for a subset of items. In addition, players can test the tag hierarchy's ability to differentiate items.

3. Data Tracking & Objection Notes

An issue that makes data validation difficult is the question of how to apply coders' insights generated during gameplay. Coders may directly change data they deem incorrect, we largely found ourselves wanting to mark data to revisit it in the future. Especially in a collaborative setting, discussion is usually necessary before making bigger changes to the data. We support coders in making use of their insights in two ways: automated tracking and objection notes.

Automated Tracking. The system automatically tracks game results for items and tags. For example, when giving a guess in **Who Am I?**, the system stores an entry for the target item and game result. Tracking data is available in a dedicated view in the

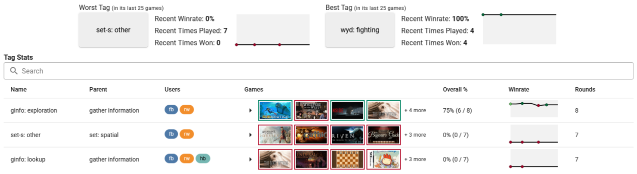

Figure 7: *The tracking view showing recorded game results for tags from the game* ***Trivia****. At the top, it shows details about the best and worst-performing tag. In the table, each row depicts one tag, showing the win rate for that tag and which items were associated with it. The line chart in each row visualizes the win rate over time.*

system, allowing coders to investigate their overall performance, as well as the win rate for items and tags. This view also shows coders the item and tag with the worst win rate. Users may choose between seeing only their own data (if they have an account) or seeing the data for all coders. The system also tracks game results for those without an account, but treats everyone as the same *guest*. The tracking view contains tables with the tracking data for items and tags. Figure 7 depicts tracking data for tags, showing the tag name, parent tag, associated items, overall win rate, win rate over time in a line chart, and number of played rounds. These tables allow coders to analyze their performance over all games and longer time spans. Even if a coder does not immediately identify something as questionable, the tracking view can help them find instances for which they often make mistakes. When multiple coders make mistakes for the same item-tag combination, this could indicate that they misunderstand the tag or that it should not be assigned to the item. This view also supports creating objection notes—explained in the next paragraph—letting coders make their thoughts explicit and accessible to others.

Objection Notes. Our system also supports making insights actionable through insight externalization. At the end of each game, an end screen summarizes the coder's performance for that round (cf. Figure 4). Each item or tag that is depicted in an end screen allows coders to create an *objection note* (cf. Figure 8 a). An objection has an owner and it contains a data reference (item, tag), a suggested action (discuss, add, remove), and an explanation. This allows the system to reference objection notes in other views, like those for tagging or analysis. Objections can be created by clicking on a dedicated button (cf. Figure 8 b & c) or through context menus activated by right-clicking on a tag in a bar code, a data point in the scatter plot, or the small item images. A list of all objections is available in a dedicated view in the system and each item also allows users to view only its related objections. Compared to automated tracking, creating objections is more tedious, but it allows users to explicitly suggest an action and explain their reasoning when they revisit this issue again in the future.

4. Reflection & Conclusion

We designed and implemented multiple visualization games that let coders interact with coded data from another perspective. So far, only the authors played these games to have fun and validate an existing dataset in which video games were tagged. Our main focus

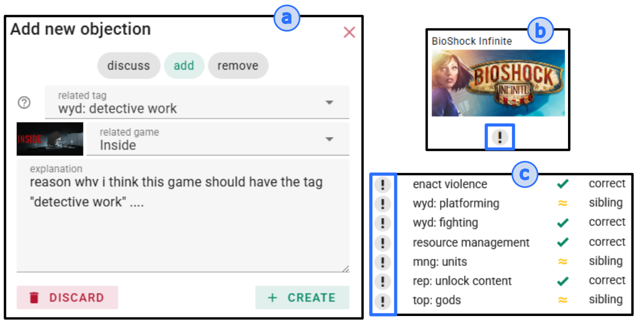

Figure 8: *Examples for a) the dialog to create an objection note, b) an objection button for an item in the end screen of* ***Trivia****, and c) objection buttons for tags in the end screen of* ***Who Am I?****.*

during validation was to find instances where tag assignments are missing, which existing tag assignments we need to discuss again, and to see how informative our tagging can be.

Design Process. To design our games, we started by considering which aspects of the data we could validate, like finding instances where tags are *missing* or where they are used *inconsistently*. Then, we looked at existing games for inspiration, analyzing how they create challenges and how they allow players to overcome these challenges. We considered how games can afford different types of strategies to win, so that we can design games that create different player experiences. For **Where Am I?**, players need to make heavy use of previous knowledge because the game itself does not give them any hints about why items are placed at specific locations. Thereby, it provides a lot of freedom to how players want to approach finding a *good* location. For the game **Who Am I?**, players have more information because they see the complete tag hierarchy in the treemap. Together with the game's feedback about asked tags, players can use their knowledge and the visualized tags to find a strategy that lets them confirm or exclude items. **Trivia** is less complex than the other two games and consequently less about finding a good strategy to win. Instead, it always provides a unique combination of items and tags to ask about and lets players feel accomplished by being knowledgeable. With **Set**, we designed a competitive multiplayer game in which players must be fast and accurate to win. This makes it a skill-based game because players need to quickly comprehend the target tag and item options. Being multiplayer, it adds a social component to a game that makes it more challenging and provides a different kind of motivation. Our games rely heavily on having compact representations for items images. For other cases, designers would need to find more fitting representations and adjust game parameters to accommodate changed needs, like giving players more time to read text passages.

Balancing & Player Experience. An important aspect of game design is *balancing*—that is finding the right balance between difficulty and skill [HBWM*19]. When games are too difficult or too easy, players can lose interest and stop playing [Csi08]. In our context, balancing meant finding the right mechanics and parameters to create a situation that allows coders to use knowledge and develop strategies to overcome meaningful challenges. Balancing for data validation can be used to support different validation goals, for example by explicitly sampling from items with low inter-coder agreement or few tags. A general limitation of our game-based approach to validation is data scale. If the dataset is too small, it can be hard to sample enough data points with the required characteristics, like (not) having a specific tag assignment, thereby making balancing more difficult. Beyond the design of game challenges, we also considered how games improve the player experience through visual design and sound. These aspects are rarely prioritized in visual analytics systems but can be important factors in game design [PJ22]. For sound design, we sourced free sound effects online to accompany players' actions (e.g., clicking on a button or item representation) and communicate game results (e.g., win or lose). Regarding the visual design, each game uses the same icons to give players feedback about their actions and the end screen includes a small animation based on the outcome—as an emotional reward or punishment. Our games use red and green to communicate results, but we use color shades that colorblind players can still differentiate. In addition, we use redundant encodings to ensure clear communication, like combining colors and icons.

Insight Externalization. To take advantage of the insights players generate while playing, we implemented automated tracking and enable annotation via objection notes. The former lets players analyze game performance in relation to parts of the dataset retrospectively and does not require any active efforts while playing. Objection notes present a direct way of externalizing insights that can be referenced in other parts of the coding system. Without these features, we found it cumbersome to use generated insights in a way that does not feel disruptive or tedious. Because the game result depends on the current coding, players might feel frustrated when they have a different opinion about the coding. However, in our experience, being able to create an objection for such cases actually resulted in a positive experience, akin to solving a puzzle.

Conclusion & Future Work. The system in which our games are embedded already contains multiple visualizations to assist coders in understanding and improving data items and tags. However, with increasing size, validating such data is a tedious process. Our games provide coders with another way of looking at their data. Instead of just viewing data items and their tags one after the other, coders are asked to think about specific tag combinations or use their data to make informed decisions. This approach sacrifices claims to completeness but can inspire coders imagination and test their knowledge in a way that conventional interfaces often cannot. Especially in the realm of coding data, validation can become tedious because many things need to be discussed among coders. We see our game-based concept as an *addition* that tackles shortcomings in conventional approaches, in which users struggle to be engaged or come up with new ideas. Future work may explore how this concept can be adapted to other contexts and data constructs as well as in which ways it supports users. Our approach could also enable coders to test their created data in a crowd-sourcing setup. Because the games are short and provide the required context, it should be possible to perform large-scale tests with many people. A different approach we considered was to use games as a means to tag data. There are many challenges that make such a setup more complex than the games we presented here. For example, we discussed different game ideas based on making players define the similarity of an (untagged) item to other (tagged) items. A problem with these ideas is the lack of a ground truth proxy that can be used to give feedback to the player. While such a design might work as an alternative to free tagging or AI-based suggestions, it is closer to gameful or playful design than complete games. It does not provide the same ways of creating motivation that *proper* games often do—due to its lack of feedback—but it is nonetheless an interesting approach to data entry that should be explored in future work.

Acknowledgments

Tanja Blascheck is funded by the European Social Fund and the Ministry of Science, Research and Arts Baden-Württemberg. We thank Hendrik Brückler for our insightful discussions.

References

[Abt87] ABT C. C.: *Serious games*. University Press of America, 1987. 1

[AFF19] ALBERTAZZI D., FERREIRA M. G. G., FORCELLINI F. A.: A wide view on gamification. *Technology, Knowledge and Learning 24*, 2 (2019), 191–202. doi:10.1007/s10758-018-9374-z. 1

[Bry02] BRYMAN A.: *Analyzing Qualitative Data*. Taylor and Francis, Hoboken, 2002. 1

[BWB25] BECKER F., WARNKING R. P., BLASCHECK T.: Seamless collaborative coding with visualization. In *EuroVis 2025 - Short Papers* (2025), Aigner W., Andrienko N., Wang B., (Eds.), The Eurographics Association. 1, 2

[BWBB25] BECKER F., WARNKING R. P., BRÜCKLER H., BLASCHECK T.: Beyond Entertainment: An Investigation of Externalization Design in Video Games. *CGF* (2025). 1

[Cho15] CHOU Y.-K.: *Actionable gamification: Beyond points, badges, and leaderboards*. CreateSpace Independent Publishing, 2015. 1

[Csi08] CSIKSZENTMIHALYI M.: *Flow: The Psychology of Optimal Experience*. Harper Perennial Modern Classics. Harper & Row, 2008. 5

[DDAA15] DICHEV C., DICHEVA D., ANGELOVA G., AGRE G.: From gamification to gameful design and gameful experience in learning. *Cybernetics and Information Technologies 14*, 4 (2015), 80–100. doi:10.1515/cait-2014-0007. 1

[GLYH10] GUPTA M., LI R., YIN Z., HAN J.: Survey on social tagging techniques. *ACM SIGKDD Explorations Newsletter 12*, 1 (2010), 58–72. doi:10.1145/1882471.1882480. 1

[Hal24] HALPIN S. N.: Inter-coder agreement in qualitative coding: Considerations for its use. *American Journal of Qualitative Research 8*, 3 (2024), 23–43. doi:10.29333/ajqr/14887. 1

[HBWM*19] HENDRIX M., BELLAMY-WOOD T., MCKAY S., BLOOM V., DUNWELL I.: Implementing adaptive game difficulty balancing in serious games. *IEEE Trans. Games 11*, 4 (2019), 320–327. doi:10.1109/TG.2018.2791019. 5

[MS22] MASEK L., STENROS J.: The meaning of playfulness: A review of the contemporary definitions of the concept across disciplines. *Eludamos: Journal for Computer Game Culture 12*, 1 (2022), 13–37. doi:10.7557/23.6361. 1

[NPVS07] NEUMANN C., PRIGENT N., VARVELLO M., SUH K.: Challenges in peer-to-peer gaming. *ACM SIGCOMM Comp. Comm. Review 37*, 1 (2007), 79–82. doi:10.1145/1198255.1198269. 4

[PJ22] PICHLMAIR M., JOHANSEN M.: Designing game feel: A survey. *IEEE Trans. Games 14*, 2 (2022), 138–152. doi:10.1109/TG.2021.3072241. 6

[vAD08] VON AHN L., DABBISH L.: Designing games with a purpose. *Commun. ACM 51*, 8 (2008), 58–67. doi:10.1145/1378704.1378719. 1