

FRAGMENTVIS: DATA VISUALIZATION-BASED ANALYSIS OF PUZZLE PLAYERS' SOLUTION STRATEGIES

Eros B. D. ^{*}, Kunkli R. 

Abstract In this paper, we present FragmentVis, a browser-based visual analytics system for the retrospective analysis of digital jigsaw puzzle-solving behavior. During play, the system captures piece-level telemetry that describes the manipulation and assembly history of individual pieces. This level of detail may be needed because aggregate measures, such as completion time or success rate, do not show how a player actually constructs a solution. With FragmentVis, we map the recorded data back to the spatial structure of the solved puzzle and present it through coordinated dashboard and inspection views. These views allow users to look back on the solving process and see how individual piece interactions relate to the completed image. We developed the system through an iterative design process informed by two rounds of user evaluation. Our results suggest that image-aligned heatmaps and coordinated dashboard views were perceived as useful and interpretable for strategy reflection and piece-level process inspection.

Keywords: Jigsaw puzzle game, Puzzle solving strategies, Visual analytics, Player behavior analysis, Interactive data visualization, Gameplay telemetry.

AMS Mathematics Subject Classification: 68U05, 68U35.

DOI: 10.32523/2306-6172-2026-14-3-25-45

1 Introduction

In jigsaw puzzles, players gradually transform an initially disordered arrangement of pieces into a single coherent final image. Although jigsaw puzzle solving can be evaluated in both physical and digital settings through outcome indicators such as completion time and success rate, digital implementations make these measures easier to record automatically and consistently. Using such measures, it becomes possible to identify factors that influence solving performance, including, on the one hand, player-related human factors [1] and, on the other hand, variables arising from game design and implementation [2]. Studies of digital puzzle-like games also show that game-embedded support structures can influence players' problem-solving behavior and strategy development [3].

However, when these outcome indicators are measured and evaluated, they do not explain how the solution was constructed or how different players arrived at their results. They provide only an aggregated view of the solving process. Therefore, a process-oriented perspective can be especially relevant for jigsaw puzzle solving, because different influencing factors may lead players to complete the puzzle in different amounts of time and to select pieces in different sequences. Differences may also appear in the solving strategy itself, for example when players focus on salient image regions, postpone ambiguous areas, or assemble local clusters outside the target grid. Here, the advantage of using digital environments becomes apparent again, this time in their ability to record when a player grabbed a particular piece, how it was moved, and which other pieces it was brought close to or attempted to connect with.

^{*}Corresponding Author.

These dense event streams are difficult to interpret directly and therefore require visual abstraction. Information visualization and visual analytics can support this process by transforming low-level interaction records into visual encodings and interactive summaries that make behavioral patterns easier to recognize [4].

Our paper presents FragmentVis, a browser-based visual analytics system for the retrospective exploration of digital jigsaw puzzle-solving behavior. The system records piece-level interaction data during gameplay and transforms these logs into visual representations. A key design principle is that behavioral metrics should be interpretable in relation to the final puzzle image; therefore, several visualizations are displayed as layers over the solved image or as coordinated dashboard views linked to the puzzle structure.

The contributions of this work are as follows:

- Firstly, we present FragmentVis, a browser-based visual analytics system for the retrospective analysis of digital jigsaw puzzle-solving behavior.
- Secondly, we formulate a piece-level telemetry model that provides a formal representation of the puzzle-solving process and serves as the computational basis for the derived analytical metrics.
- Thirdly, we report the results of a two-stage user evaluation in which feedback from the first study directly informed the dashboard-oriented redesign of the system.

The evaluation results suggest that image-aligned heatmaps and coordinated dashboard views were perceived as useful for the retrospective interpretation of puzzle-solving behavior. Participants were able to relate the visual analytics outputs to their own reported strategies, such as edge-first solving, delayed handling of visually ambiguous regions, repeated interaction with specific pieces, and the formation of piece groups outside the target grid. These findings indicate that dashboard-oriented views and coordinated visualizations designed with player needs in mind can make puzzle-solving processes more visible and interpretable, while further controlled studies would be needed to determine whether such feedback leads to measurable changes in puzzle-solving performance.

The paper is organized as follows: Section 2 reviews related work on automatic jigsaw puzzle reconstruction, human puzzle-solving performance, and gameplay visualization. Section 3 describes the system architecture, puzzle-piece generation, collected telemetry, and derived metrics. Section 4 presents the analytical visualizations and interaction mechanisms of the FragmentVis system. Section 5 reports the two-stage user evaluation and the developments introduced between the two rounds. Section 6 concludes the paper, discusses limitations, and outlines directions for future work.

2 Related Work

Research on jigsaw puzzles spans several directions. Early studies can be traced to automatic puzzle reconstruction. In addition, there is research aimed at analyzing human performance and identifying factors that influence the difficulty of the jigsaw puzzle problem. As a third category, we discuss data-driven approaches that support gameplay visualization and reflective interpretation.

2.1 Automatic reconstruction and puzzle formulations

In computer vision and graphics, the jigsaw puzzle problem has long been studied as a reconstruction task in which fragmented content must be reassembled into a coherent whole.

These fragments are often described by their boundary geometry [5, 6] and, in some cases, by visual appearance [7].

Markaki and Panagiotakis [8] provide a broad survey of jigsaw puzzle solving techniques and applications, showing that the field covers not only entertainment-oriented square puzzles but also irregular and apictorial variants, with applications in domains such as archaeology, biology, security, and document reconstruction.

Extending the formal landscape of the problem, Harel et al. [9] introduced a class of pictorial and apictorial polygonal jigsaw puzzles generated from an arbitrary number of crossing cuts, thereby moving beyond the classical square-grid setting.

These studies establish a strong computational foundation for jigsaw research, but their primary objective is the reconstruction of the final arrangement rather than the analysis of how human solvers progressively arrive at that arrangement [8, 9].

2.2 Human performance, difficulty, and player-centered factors

A second relevant line of work focuses on the human factors that shape jigsaw puzzle performance. Hong et al. [1] examined digital jigsaw puzzle solving through the lens of cognitive style and showed that field independence was positively correlated with puzzle score. Moreover, their GridWare-based analysis identified distinct strategy patterns from event sequences generated during play. Their findings indicate that jigsaw solving should be understood not only as a visuospatial task, but also as a temporally unfolding behavioral process in which exploration, revisits, hints, and action sequences all matter.

From a different perspective, Yuan et al. [2] studied jigsaw puzzle difficulty assessment using a deep-learning-based approach and reported that image-level factors such as textureless uniform-color regions and repetitive textures have a measurable effect on completion time. In particular, they observed that repetitive textures tend to increase difficulty, whereas larger uniform-color regions can support faster puzzle completion by facilitating color-based grouping and rapid sequential attempts.

Taken together, these studies demonstrate that jigsaw performance depends both on player-specific cognitive characteristics and on image-specific visual properties. At the same time, they do not provide a dedicated visual analytics framework for inspecting the fine-grained solving trajectory itself at the level of individual piece manipulations [1, 2].

2.3 Gameplay visualization, reflection, and collaborative co-solving

Javvaji et al. [4] argue that many established gameplay visualization techniques become difficult to interpret when confronted with high-dimensional trace data, especially in puzzle-oriented settings, and they propose a knowledge-based data abstraction pipeline combined with interactive visualization to reveal player patterns over time. In the context of their collaborative puzzle game *Daedalus*, they show that abstracted traces can expose interpretable patterns such as early dropout, collaborative problem solving, and brute-force behavior.

Li et al. [10] similarly treat gameplay as an event sequence and demonstrate that behavioral complexity can be used as an informative lens for understanding how player skill develops over time. Closely related to this idea, Maram et al. [11] introduced a spatio-temporal visualization system designed for process reflection, enabling players to inspect strategic events together with board states and trace information. These works shift attention from final outcomes to the temporal structure of play and to the interpretability of player traces. However, they are not centered on classical jigsaw puzzle solving at the level of piece arrangement [4, 10, 11].

A further adjacent body of literature investigates collaborative problem solving and co-solving scenarios in which attention, coordination, and user experience are central. Abeyasinghe et al. [12] proposed a gaze analytics dashboard for distributed eye tracking in collaborative problem solving, highlighting the value of real-time visual summaries for understanding attentional processes during joint tasks. In the specific context of jigsaw puzzles, Choi et al. [13] used immersive virtual reality co-solving tasks to examine how a virtual character’s intelligence and the complexity of the task influence user experience. Their results showed significant effects on perceived collaboration, self-confidence, and perceived contribution, confirming that jigsaw co-solving is a useful experimental setting for studying human-agent interaction [13]. In a related study, Choi et al. [14] investigated the self-correction behavior of an intelligent virtual character during collaborative jigsaw solving and found that increasing either the character’s intelligence or the accuracy of its self-correction improved perceived intelligence and reduced participants’ workload and completion time. Nevertheless, their main focus is on collaborative experience and agent-related perception rather than on a visual analytics tool that supports an individual player’s retrospective inspection of detailed solving behavior [13, 14].

Overall, the reviewed literature indicates that comparatively little attention has been paid to visual analytics tools that allow individual jigsaw players to retrospectively inspect their own solving process at the level of piece-specific interactions. In FragmentVis, we address this gap by focusing on the visual interpretation of human puzzle-solving behavior rather than on automatic puzzle reconstruction or final performance prediction.

3 System design and implementation

3.1 Overview and architecture

FragmentVis is implemented as a browser-based single-page application that combines a p5.js canvas for real-time puzzle interaction with an HTML sidebar for settings and analytics controls. The lightweight Python/Flask backend receives a source image and a corresponding segmentation mask, partitions the image into irregular puzzle pieces, and returns the piece geometry and encoded image data to the client. After puzzle generation, user interaction, rendering, and analytics computation are handled on the frontend in JavaScript, keeping the backend stateless during gameplay.

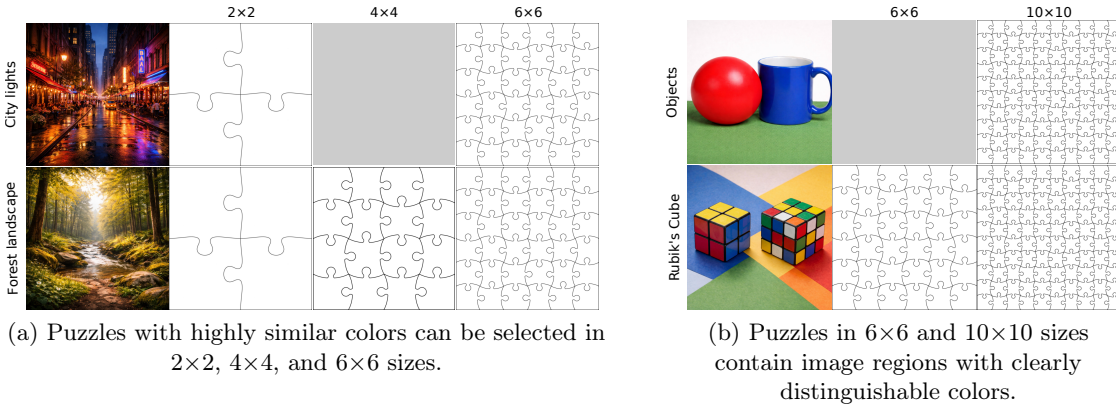


Figure 1: The puzzle images used in the application and their masks. Gray cells indicate image-grid-size combinations that were not included in the evaluated puzzle set.

We generated the puzzle images of the application from text prompts using OpenAI’s

ChatGPT image-generation interface [15] and Google Nano Banana Pro [16], while we created the corresponding segmentation masks manually in Inkscape [17]. The available puzzle sets include visually homogeneous images in smaller grid sizes and images with more clearly separated color regions in larger grid sizes, as illustrated in Fig. 1. This set also reflects the iterative development of our application. The preliminary user evaluation was conducted with an earlier set of puzzle configurations, which included fewer puzzle images (*City lights*, *Forest landscape*) and did not yet contain the 10×10 mask. After the preliminary user test, we extended the puzzle set with additional images and a new 10×10 mask in order to support the visualization of more fine-grained solving patterns.

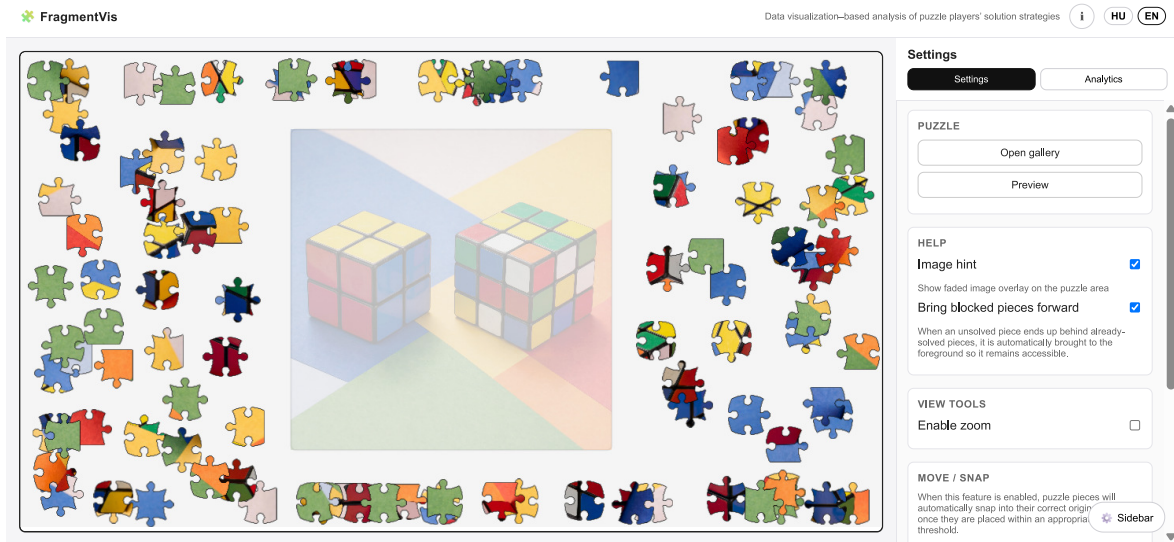


Figure 2: Screenshot from our FragmentVis application. When activated during gameplay, the *Image Hint* function overlays a faint reference image onto the puzzle assembly area, thereby facilitating the player’s puzzle-solving process.

Puzzle-piece generation is performed on the server side using the segmentation mask. The mask is binarized, resized to the image dimensions when necessary, and processed through contour extraction and filtering. The remaining contours are sorted into a grid according to their bounding-box centers. For each detected piece, the backend returns its row and column index, bounding-box metadata, and a transparent PNG crop encoded as a Base64 string. This representation preserves irregular piece silhouettes while retaining the logical grid structure required for placement, snapping, and analytics.

The interface consists of a main puzzle canvas and a right sidebar (see Fig. 2). Before play begins, a gallery wizard lets the user select an image and one of the supported grid sizes. When a puzzle is loaded, pieces are scattered around the target grid, leaving the grid interior clear as a visual reference for placement. During play, the sidebar provides access to the *Settings* panel and the *Analytics* panel. The *Settings* panel exposes controls for snap sensitivity, piece scale, zoom-based inspection, and visual overlay options. The *Analytics* panel contains a view selector that provides access to the currently exposed analytical views. View-specific controls, such as colormap selectors and opacity settings, are displayed where applicable.

At the time of writing, the FragmentVis application is publicly available at <https://fragmentvispuzzle.pythonanywhere.com/>.

3.2 Telemetry model and derived metrics

We integrated the data collection into the interaction loop, so it does not require any explicit user action during gameplay. In FragmentVis, the puzzle pieces are analytical units that connect the fixed geometry of the solved puzzle with the dynamic state of the piece during play. This information is not stored in one monolithic log. Instead, the application maintains live piece-level state on the client side and records selected interaction and position data into session-level telemetry structures.

3.2.1 Piece state and temporal sampling

Let

$$P = \{p_1, \dots, p_n\} \quad (1)$$

denote the set of puzzle pieces in a solving session, where n is the number of pieces generated for the selected puzzle configuration. Each piece p_i has a fixed target position in the solved puzzle grid, represented by its row and column indices, together with the geometric metadata described in Section 3.1.

During gameplay, this fixed representation is connected to a state that changes as the player works with the puzzle. For each piece, the client keeps track of where the piece currently appears on the screen, how it is oriented, whether the piece has already been placed, and whether it is currently connected to other pieces as part of a multi-piece group. At the same time, the system records the direct actions through which the player manipulates the pieces, including grab and rotation events, and periodically samples piece positions to capture their movement over time.

The global snapshot timeline is denoted by

$$S = (s_1, \dots, s_m), \quad (2)$$

where m is the total number of snapshots recorded during the solving session. In the implementation, positional snapshots are collected at approximately 150-millisecond intervals during dragging and once more when the dragged piece or group is released. For $k = 1, \dots, m$, each snapshot s_k is associated with a raw timestamp $\hat{t}_k$, expressed in milliseconds. Let t_0 denote the start time of the solving session. For the analytical metrics, we use the session-relative time

$$t_k = \hat{t}_k - t_0. \quad (3)$$

For a piece p_i , a snapshot stores its sampled center position

$$\mathbf{c}_{i,k} = (x_{i,k}, y_{i,k}) \quad (4)$$

and its current group identifier $g_{i,k}$. In our implementation, this identifier refers to the group object associated with the piece at snapshot s_k . For the analytical metrics, however, only identifiers shared by at least two pieces are interpreted as connected multi-piece groups. The snapshot timeline therefore provides a sampled account of how pieces move and become assembled during play.

3.2.2 Piece-level behavioral metrics

All time-dependent quantities in this subsection are interpreted on the elapsed game-time scale of the solving session.

The placement status of a piece is represented by

$$a_i(t) = \begin{cases} 1, & \text{if } p_i \text{ is placed in its target position at time } t, \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

The placement time of p_i is defined as the earliest elapsed game time at which the piece reaches its target position:

$$T_i^{\text{place}} = \min\{t \mid a_i(t) = 1\}. \quad (6)$$

If a piece is not placed by the end of the session, T_i^{place} is treated as undefined and the piece is shown as unsolved in placement-time visualizations.

Direct manipulation is represented through grab and rotation records. Let B_i denote the number of recorded grab events associated with p_i , and let R_i denote the number of recorded rotation events. Although the client keeps track of the current orientation of each piece during gameplay, the manipulation metric does not reconstruct a full orientation history. Instead, rotations are represented by the number of recorded rotation events. The manipulation frequency of the piece is then defined as

$$M_i = B_i + R_i. \quad (7)$$

This metric summarizes how often the player directly manipulated a piece. In the implementation, grab events store the local click coordinates within the piece, the elapsed game time at grab start, and the drag duration.

Movement is derived from the global snapshot timeline. Let K_i denote the set of valid consecutive snapshot indices for piece p_i , that is, those indices $k \in \{2, \dots, m\}$ for which both sampled center positions $\mathbf{c}_{i,k-1}$ and $\mathbf{c}_{i,k}$ are available in the snapshot timeline. The movement distance of p_i is then computed as the cumulative Euclidean distance between these consecutive sampled positions:

$$D_i = \sum_{k \in K_i} \|\mathbf{c}_{i,k} - \mathbf{c}_{i,k-1}\|_2. \quad (8)$$

In our implementation, this distance is measured in canvas pixels. This metric does not reconstruct the continuous mouse trajectory. It provides a discrete, sample-based approximation of piece displacement. In the case of multi-piece groups, the sampled displacement is attributed to each piece belonging to the moved group, because all group members undergo the same rigid-body translation during the drag operation. Consequently, movement distance reflects the displacement of individual pieces in the recorded snapshots, regardless of whether the pieces were moved separately or as part of an assembled group.

3.2.3 Group formation and assembly metrics

Connection-oriented metrics are derived from changes in group membership over the snapshot timeline. Let H_k denote the set of valid, non-empty group identifiers observed at snapshot s_k . For $h \in H_k$, the piece set associated with this identifier at time t_k is defined as

$$\Gamma(h, t_k) = \{p_j \in P \mid g_{j,k} = h\}. \quad (9)$$

The identifier h is interpreted within snapshot s_k . The set $\Gamma(h, t_k)$ may represent the implementation-level group object associated with one or more pieces, but it is treated as an active multi-piece group in the analytical metrics only if it contains at least two pieces.

The first connection time of a piece p_i is defined as the earliest time at which the piece appears in a multi-piece group:

$$T_i^{\text{conn}} = \min\{t_k \mid g_{i,k} \in H_k \text{ and } |\Gamma(g_{i,k}, t_k)| \geq 2\}. \quad (10)$$

If a piece never becomes part of a multi-piece group, T_i^{conn} is treated as undefined.

Group growth is represented as a pair of time-dependent functions over the snapshot timeline. The set of active multi-piece groups is then defined as

$$\mathcal{G}_k = \{\Gamma(h, t_k) \mid h \in H_k \text{ and } |\Gamma(h, t_k)| \geq 2\}. \quad (11)$$

The number of active multi-piece groups is then

$$N_{\text{groups}}(t_k) = |\mathcal{G}_k|, \quad (12)$$

and the size of the largest active group is

$$L_{\text{max}}(t_k) = \begin{cases} \max_{G \in \mathcal{G}_k} |G|, & \text{if } \mathcal{G}_k \neq \emptyset, \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

Together, $N_{\text{groups}}(t_k)$ and $L_{\text{max}}(t_k)$ describe whether the player created several small assemblies in parallel or gradually consolidated pieces into a dominant group.

For an active group $G \in \mathcal{G}_k$, we treat G as an off-grid assembly if the pieces in G are first observed as a multi-piece group before they are snapped into the solved grid. The off-grid location itself is not reconstructed; instead, the metric records that the involved pieces became grouped before their final placement. The assembly is considered to enter the solution when the grouped pieces are later snapped into their corresponding target positions.

3.3 Computational workflow

FragmentVis follows a computational pipeline that links the puzzle input to the final dashboard views. The process begins with the source image and its segmentation mask, from which the backend generates the puzzle pieces and their geometric representation. During play, user actions and sampled piece positions form a timestamped interaction stream. This stream is then interpreted as piece-level state information, making it possible to follow how each piece changes its role during the session.

On this reconstructed session representation, the analytical layer computes the derived metrics used by the system. Through these metrics, the system describes how the solution gradually takes shape, from early exploratory handling to final placement and, in some cases, the construction of larger assemblies. The final step is spatial re-projection: instead of presenting the results as raw event logs, FragmentVis maps the derived values back onto the solved puzzle geometry and presents them through image-aligned dashboards and inspection views.

4 Analytics visualizations

Our system provides access to dashboard-oriented and task-specific analytical views. At the dashboard level, piece-level measures such as placement timing, manipulation intensity, and movement distance are mapped back onto the solved puzzle layout and complemented with summaries based on piece shape. More focused views allow users to inspect where pieces

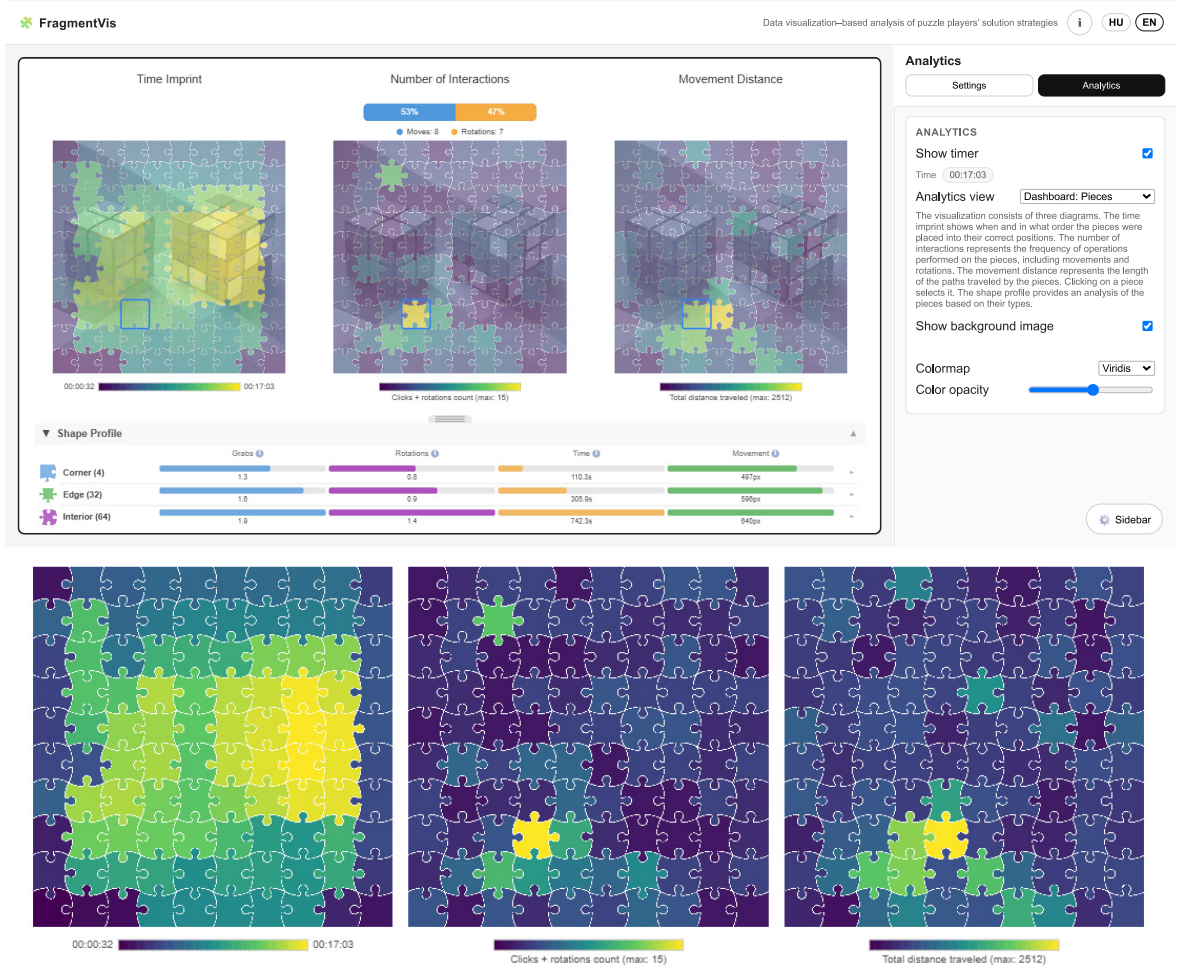


Figure 3: Dashboard view combining three image-aligned analytical encodings: placement time, interaction frequency, and movement distance. The lower shape-profile panel summarizes piece categories and supports the interpretation of pieces with unusually high metric values in relation to piece geometry.

were grabbed, how selected pieces moved over time, and how piece groups emerged during assembly. Unless otherwise stated, the screenshots and interface examples in this section show the revised version of FragmentVIS after the developments introduced between the two user evaluation rounds. The preliminary user evaluation discussed in Section 5.1 was conducted with an earlier version of the system, in which only a smaller set of analytical views was available.

4.1 Dashboard: Pieces

This view provides an image-aligned per-piece dashboard that combines three coordinated analytical encodings: placement time, interaction frequency, and cumulative movement distance (see Fig. 3). In the initial version, the panel design represented pieces as rectangular cells. However, user feedback made it clear that preserving the original piece silhouettes would make the visualizations easier to interpret and improve the overall design. We therefore rendered the pieces in their solved spatial layout using the original mask-derived silhouettes rather than



Figure 4: Second-level drill-down view of the *Shape Profile* component. After selecting the *Edge* category in the panel, this view allows pieces with similar geometric configurations to be compared within the broader edge-piece category.

abstract rectangular cells. To finalize the dashboard, we also introduced a shape-profile panel. The three main diagrams provide an overview of piece-level behavior across the whole puzzle, but we found it useful to offer a more detailed, lower-level summary as well. The shape-profile panel therefore groups pieces according to their structural role in the grid, distinguishing corner, edge, and interior pieces, and summarizes each group through aggregated interaction and timing measures. To make practical use of the available screen space, we designed this panel to be both collapsible and resizable.

4.1.1 Time Imprint

In this part of the dashboard, we render placement time by coloring each puzzle piece in its solved spatial position. The user-selected color scale is normalized between the earliest and latest observed placement times, and unsolved pieces are shown in neutral gray. The view can be understood as a puzzle-specific form of re-spatialization: instead of presenting placement times as a detached log or chart, the temporal attribute is mapped back onto the solved spatial layout of the puzzle [18, 19].

4.1.2 Number of Interactions

In this component, we encode the manipulation frequency of each puzzle piece. The interaction count is computed from the number of recorded grabs and rotations associated with the piece. In the dashboard view, this information is visualized as an image-aligned heatmap and is complemented by a stacked summary bar indicating the relative contribution of grab and rotation events. The view helps identify pieces or regions that required repeated manipulation. Such regions may correspond to visual ambiguity or exploratory manipulation.

4.1.3 Movement Distance

For each piece, we calculated the movement distance by summing the Euclidean distances between consecutive recorded positions. High values therefore indicate pieces that were moved over longer spatial paths, whereas low values indicate pieces that were placed with relatively little repositioning.

4.1.4 Shape Profile

The *Shape Profile* component aggregates pieces according to their structural role in the grid, distinguishing corner, edge, and interior pieces. For each category, we calculated the average number of grabs and rotations, as well as the average placement time and movement distance. The panel also displays representative piece silhouettes, allowing the users to relate quantitative differences to geometric properties of the pieces. It further supports drill-down inspection of shape-signature subgroups, allowing pieces with similar edge configurations to be compared within the broader corner, edge, and interior categories (see Fig. 4). In both the *Shape Profile* panel and its drill-down view, bar lengths are used for relative comparison within each metric column: each column is normalized to the largest average value among the currently displayed piece categories, while the numerical labels show the corresponding absolute average values.

4.2 Grab Points

This view projects recorded grab events onto the puzzle canvas as small circles at the click positions within the corresponding piece. We designed the resulting scatterplot to show whether players typically grabbed pieces near their centers or near their edges.

4.3 Movement Paths

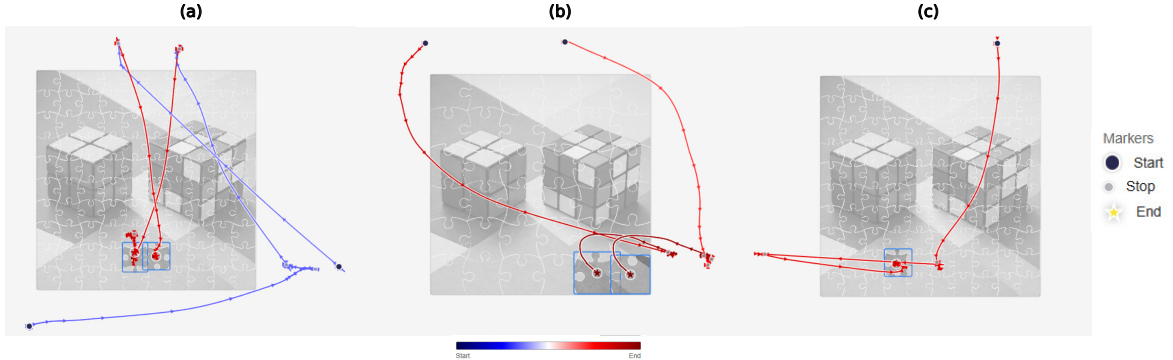


Figure 5: Movement-path visualization of selected puzzle pieces. Trajectories are reconstructed from the global snapshot timeline and encoded with temporal color gradients. We added glyphs to mark start, intermediate states, and final positions. The examples illustrate three trajectory patterns: early grouping of visually similar pieces (a), off-grid assembly followed by joint movement (b), and tentative misplacement followed by later correction (c).

In this view, we visualize sampled piece movements from the global snapshot timeline. Consecutive recorded positions are connected into Catmull-Rom spline curves for readability. We built this view around the discrete samples recorded during piece movement. Path segments are color-coded by normalized game time, and glyphs mark start, intermediate states,



Figure 6: Connection dashboard illustrating an off-grid assembly pattern. The highlighted pieces form an off-grid assembly that was later inserted into the solved layout. After selection, the *Group Growth* chart marks the interval in which the complete group was formed, while the *First Connection* panel shows when pieces first joined a neighboring piece or group.

and final positions. In the initial design, we displayed the movement paths of all pieces at the same time. However, this resulted in visual clutter and made the view difficult to interpret, which was also reflected in the user feedback. In the current version, players can inspect the trajectory of a single selected piece or compare the trajectories of two selected pieces (see Fig. 5).

4.4 Dashboard: Connections

With this dashboard, we summarize puzzle assembly behavior through three coordinated components: *Off-Grid Assemblies*, *First Connection*, and *Group Growth*. Together, they show which pieces formed groups outside the target area, when pieces first joined another piece or group, and how the number and size of multi-piece groups evolved over time (see Fig. 6).

4.4.1 Off-Grid Assemblies

This panel shows pieces that became part of a multi-piece group before the grouped pieces were snapped into their corresponding target positions. The off-grid location itself is not reconstructed; instead, the involved pieces are mapped back to their solved positions and colored by the time they first became part of such an assembly. Selecting an assembly highlights pieces that entered the solution together.

4.4.2 First Connection

This panel encodes, for each piece, the first time it appears in a group of at least two pieces. Selecting a piece highlights the neighboring pieces that belonged to the same group at the moment when the selected piece first became connected (see Fig. 7).

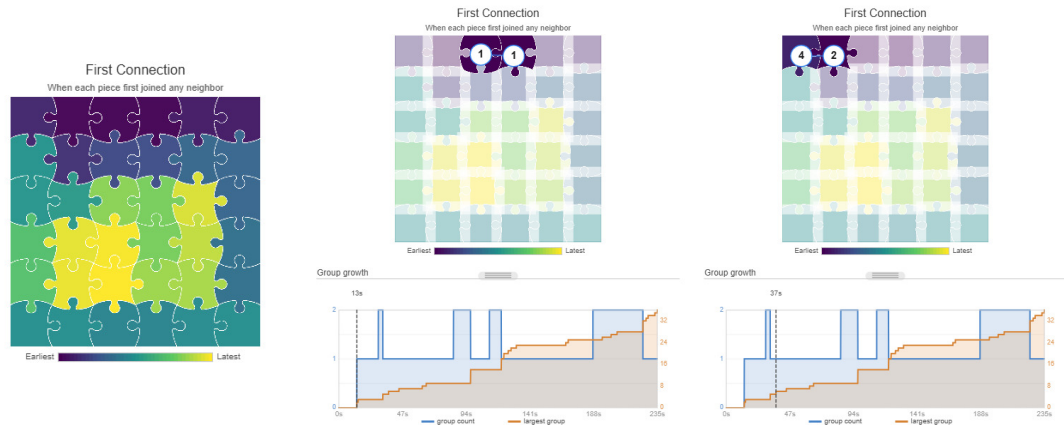


Figure 7: First-connection analysis and group-growth relation. Numbers indicate the order in which pieces became connected. Identical numbers mark the creation of a new group from two separate pieces, reflected by an increase in the group-count curve. Different numbers indicate attachment to an existing group; the largest-group-size curve shows whether this event increased the size of the largest group.

4.4.3 Group Growth

We display the number of active multi-piece groups and the size of the largest group over time in this panel. It supports inspection of whether the solver created several small assemblies in parallel or consolidated pieces into a dominant group. Chart selections filter the *First Connection* panel, while selections from the *First Connection* panel are reflected as temporal markers.

4.5 Statistics

The *Statistics* panel becomes available in the main gallery only after at least one puzzle has been successfully completed. Here, we summarize the ratio of started and completed games using a 100% stacked horizontal bar chart. Solution-time and grab-count statistics are presented in metric grids, while streaks and achievements are shown as badge-style cards. Difficulty-wise summaries are displayed as grouped statistical entries. Completed games are displayed in an interactive timeline, enabling users to compare when each puzzle was played and how long each solving attempt took.

4.6 Interaction design

Beyond passive viewing, the currently available analytical views provide several shared interaction mechanisms where applicable. In the dashboard-oriented views and in the movement-path view, colormap selectors immediately update the corresponding color-encoded elements. Dashboard views also provide controls for displaying the original puzzle image as a contextual background, optionally in grayscale, and for adjusting the opacity of the colored analytical overlay. In the normal puzzle view, the image-hint toggle displays a semi-transparent version of the original image over the target area to support spatial orientation during solving.

The interface also includes interaction controls that affect gameplay and inspection. Zoom and pan can be enabled in the normal puzzle view, while analytical views disable zoom to preserve the fixed spatial layout of the visualizations. Piece scale can be adjusted with a slider; solved pieces are repositioned according to their target locations, and grouped pieces preserve

their relative arrangement under scaling. The snap sensitivity threshold is configurable and is computed relative to a representative piece diagonal. Finally, the bring-forward option prevents solved pieces from blocking access to unsolved pieces by allowing the interaction logic to skip already solved groups during hit detection.

5 Assessment and evaluation

5.1 Preliminary user evaluation

We conducted a preliminary user evaluation of the early version of the application to examine whether the system helps players reflect on their jigsaw-solving strategies and to identify which analytical views were perceived as the most useful. The analysis is based on the merged responses of the Hungarian and English versions of the questionnaire, completed by students from the Faculty of Informatics at the University of Debrecen ($N = 37$).

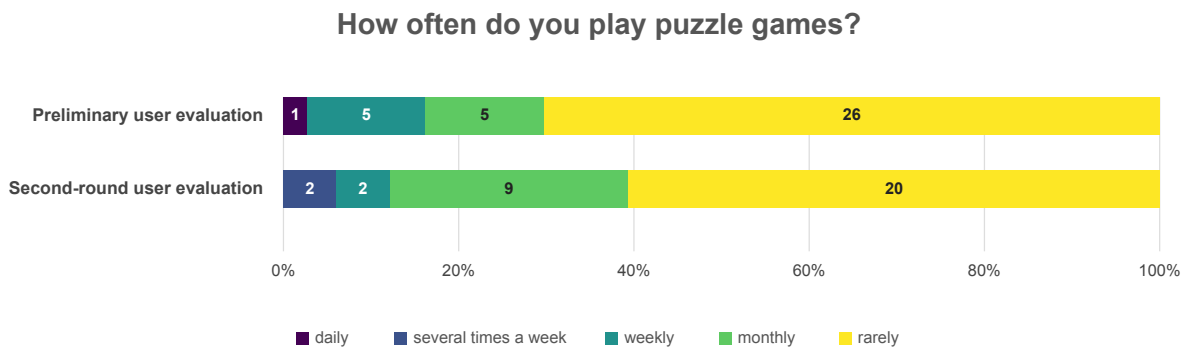


Figure 8: Puzzle-playing frequency of participants.

In terms of player profile, the sample was dominated by occasional puzzle players: 26 participants reported playing rarely, 5 monthly, 5 weekly, and only 1 daily. This suggests that the evaluation primarily captures feedback from typical end users rather than highly experienced or expert puzzle solvers (see Fig. 8).

Participants first tested the application within a 30–40 minute time frame, which was intended to give them enough time to complete at least one puzzle. Afterward, they filled out a questionnaire with predefined questions, which we used to assess their impressions of the system. A total of 51% of participants rated the application with 4 or 5 on a five-point scale in terms of helping them better understand their own solving strategy (see Fig. 11), while 65% gave similarly high ratings for overall satisfaction (see Fig. 13). Usability-related responses were also encouraging: 70% described the system as intuitive or easy to use. According to our survey, 20 out of 37 participants, corresponding to 54%, indicated that they would definitely or probably try to play more consciously in a future session after inspecting the visualizations (see Fig. 10). At the same time, only 30% explicitly stated that they had discovered genuinely new patterns in their own behavior (see Fig. 12). This suggests that, at this stage, the main value of the system lies less in producing radically new insights and more in making existing strategies visible, interpretable, and confirmable.

We also asked participants to rate the usefulness of the visualization components that were available in the application at that stage for analyzing their solving strategy (see Fig. 9). Our aim was to identify which views users considered most helpful for reflection and which components should receive greater emphasis in later versions. Among the visual analytics views, the *Time Imprint* proved to be the most useful, with 76% of respondents rating it

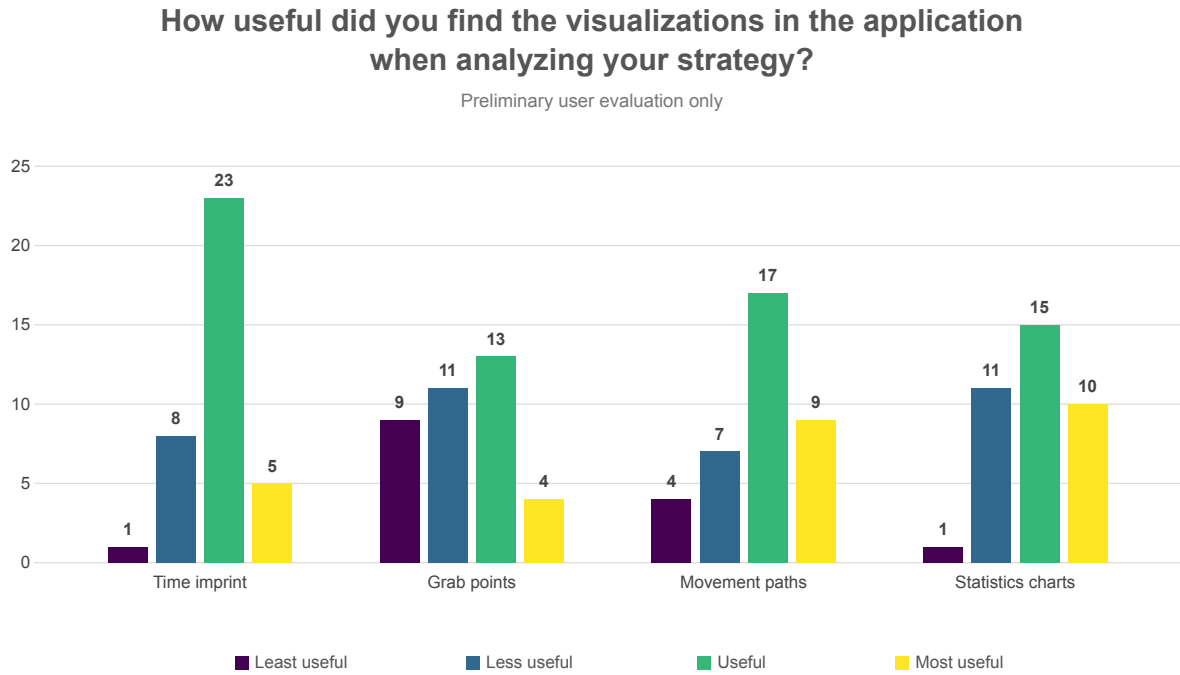


Figure 9: Usefulness ratings of the analytical views in the preliminary evaluation. Regarding this question, the total number of ratings within a usefulness category can differ from $N = 37$, because participants could assign the same usefulness level to multiple diagrams. However, each participant provided exactly one rating for each diagram.

as useful or most useful. This was followed by *Movement Paths* (70%) and the *Statistics* panel (68%), whereas the *Grab Points* view received more moderate evaluations (46%). The open-ended responses provide additional context for our findings. In these responses, users reported observed strategy patterns, such as starting with edge and corner pieces, followed by solving the center later.

Suggestions for future development clustered around five main themes. The most frequent requests concerned the addition of new analytical views and metrics, as well as improvements to the user interface and interaction design (12 mentions each). These were followed by requests related to performance optimization and lag reduction (10 mentions), more puzzle content or customization options (7 mentions), and the correction of snapping/placement issues (6 mentions). Overall, the preliminary evaluation confirmed the viability of the concept and identified performance, interaction stability, and analytical depth as the main areas for improvement.

5.2 Developments following the preliminary user test

Following the preliminary user test, we started an iterative, feedback-driven development phase. Both new features and substantial refinements of existing components were introduced. Our primary objective during this development period was to respond directly to the needs identified in the preliminary user evaluation, especially the demand for deeper analytics, improved usability, greater configurability, and more robust interaction handling.

In this phase, our most important developments can be summarized as follows:

- **Dashboard expansion and new analytical views:** we introduced a multi-heatmap dashboard view combining the *Time Imprint* diagram (this one already existed in the

Do you plan to play more precisely / more consciously next time?

Note: the response scale changed between rounds.

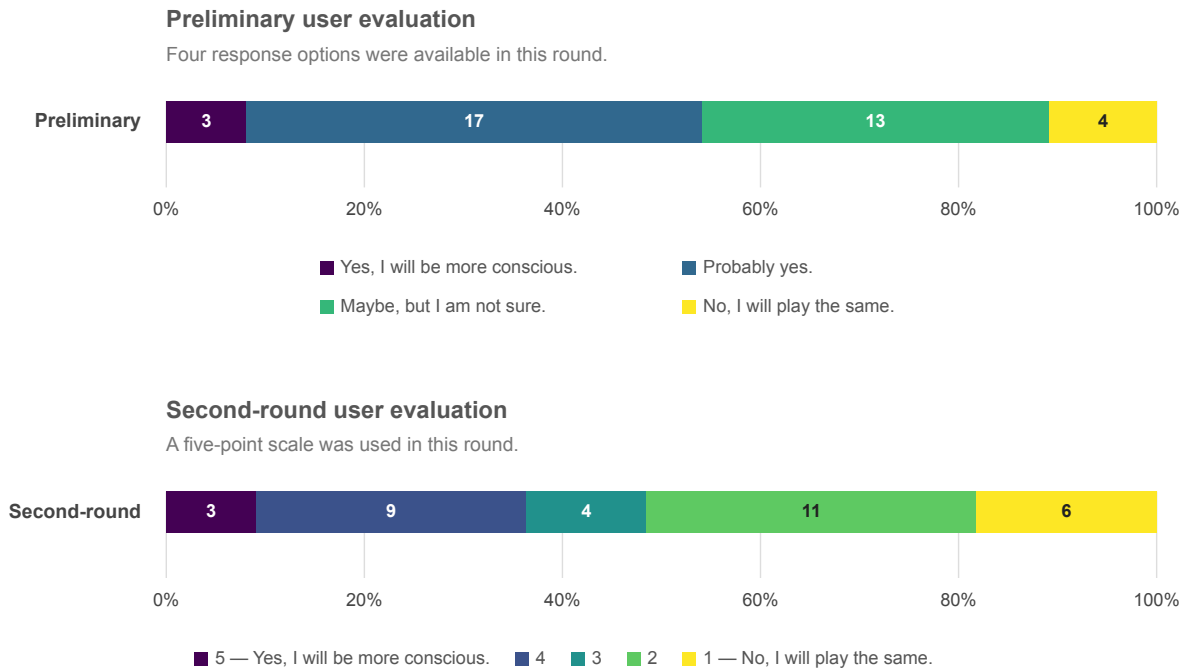


Figure 10: Self-reported intention to play more precisely or more consciously in a future session after inspecting the visualizations. In the preliminary evaluation, 20 out of 37 participants (54%) indicated that they would definitely or probably try to play more consciously. In the second-round evaluation, a five-point scale was introduced to obtain more differentiated and less forced responses. Therefore, the two distributions should be interpreted within their respective rounds rather than as directly comparable response scales. Segment labels indicate absolute response counts.

first phase, but as a standalone visualization) with two new elements, *Number of Interactions* and *Movement Distance*. We also implemented the *Shape Profile* chart.

- **Connection-oriented dashboard:** we designed a second dashboard that includes the *Off-Grid Assemblies* view, the *First Connection* panel, and the *Group Growth* chart. Here, we focused on the connections between individual pieces.
- **Enhanced trajectory analysis:** we extended the *Movement Paths* view with piece-specific trajectory inspection and two-piece comparison. This change reduced visual clutter.
- **Improved interpretability and customization:** we added a faint background-image hint, colormap selector, background-image overlay, color-opacity slider for analytical overlays, and grayscale overlay option.
- **Interface and interaction refinements:** we introduced an updated completion screen and an improved control layer.
- **Content expansion and bug fixing:** we created a new 10×10 puzzle mask, and we included a new image, *Objects*, and an additional Rubik's Cube-themed image. We fixed

the reported snap bug and performance issues, and we implemented a bring-forward option for blocked pieces.

These developments shifted the system from a compact exploratory prototype toward a more complex multi-view visual analytics environment. The added dashboard views and trajectory comparison functions directly addressed the main user requests for richer analytics and more interpretable visual feedback.

5.3 Second-round user evaluation

Once we finished the developments following the preliminary user test, we conducted a second-round user evaluation. The preliminary evaluation included $N = 37$ responses, while the second round was based on $N = 33$ responses. The composition of the two samples remained broadly similar (see Fig. 8): both were dominated by casual puzzle players, with 84% of respondents in the preliminary round and 88% in the second round reporting that they play rarely or monthly. This indicates that the observed differences are unlikely to be driven primarily by major shifts in participant profile. The two evaluation rounds involved disjoint participant groups: no participant took part in both rounds.

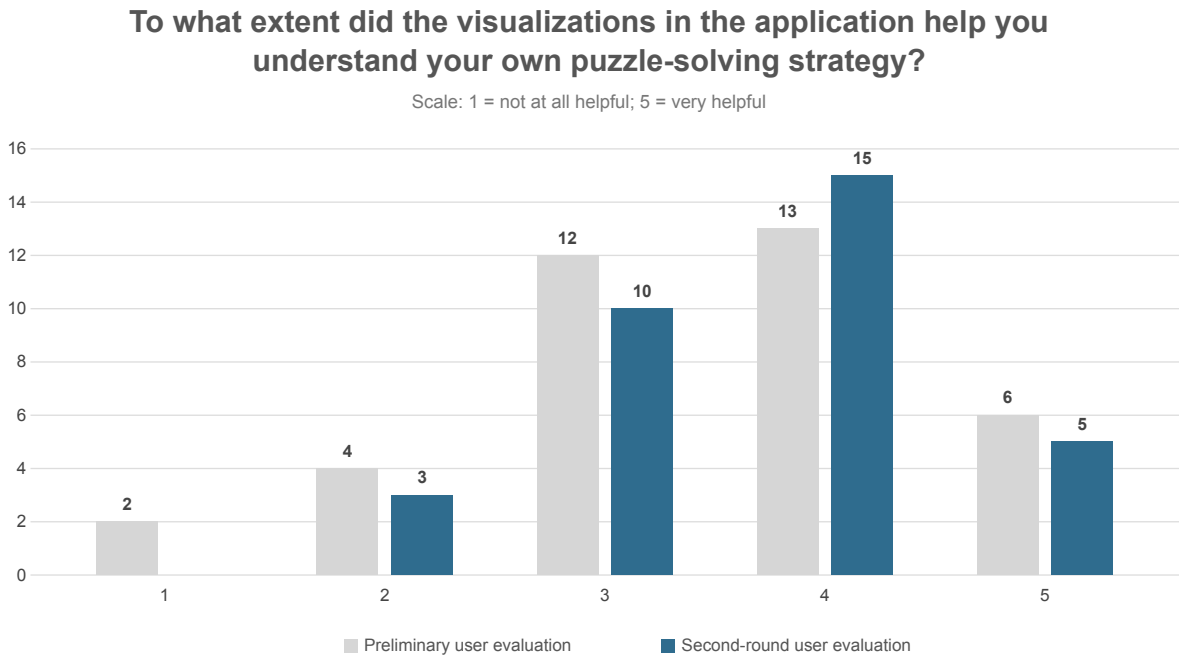


Figure 11: Perceived support for understanding one's solving strategy. Mean ratings: $\bar{x}_{pre} = 3.46$, $\bar{x}_{second} = 3.67$.

The descriptive comparison suggests that the revised version was rated more positively in several respects. The proportion of respondents who gave high ratings for support in understanding their own strategy increased from 51% to 61% as illustrated in Fig. 11. Overall satisfaction ratings also increased, from 65% to 94% (see Fig. 13), while the proportion of users describing the system as intuitive or easy to use increased from 70% to 91%. In contrast, reports of genuinely new self-insights remained essentially unchanged at around 30% (see Fig. 12). This pattern suggests that the revised system was perceived as more usable and more acceptable as a reflective support tool, but the evaluation does not provide evidence

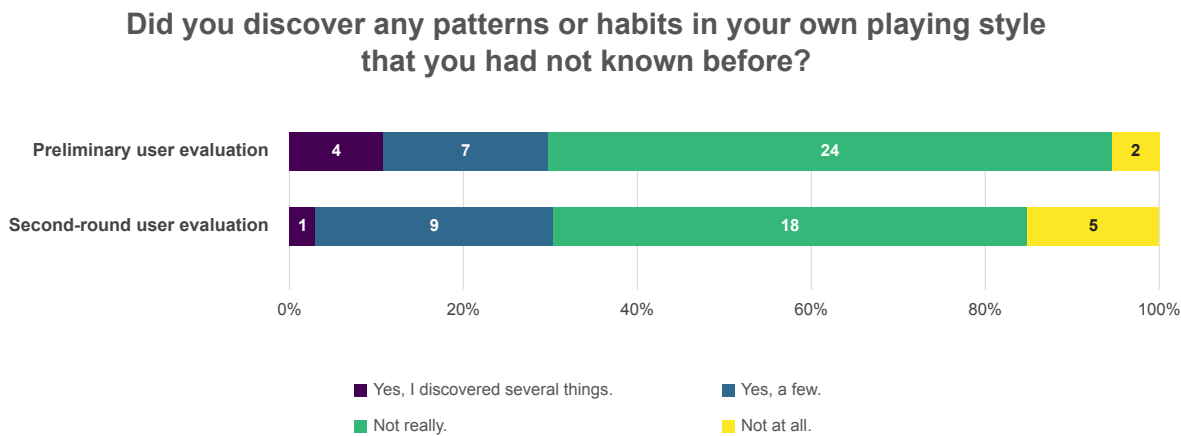


Figure 12: Self-reported discovery of previously unnoticed playing patterns.

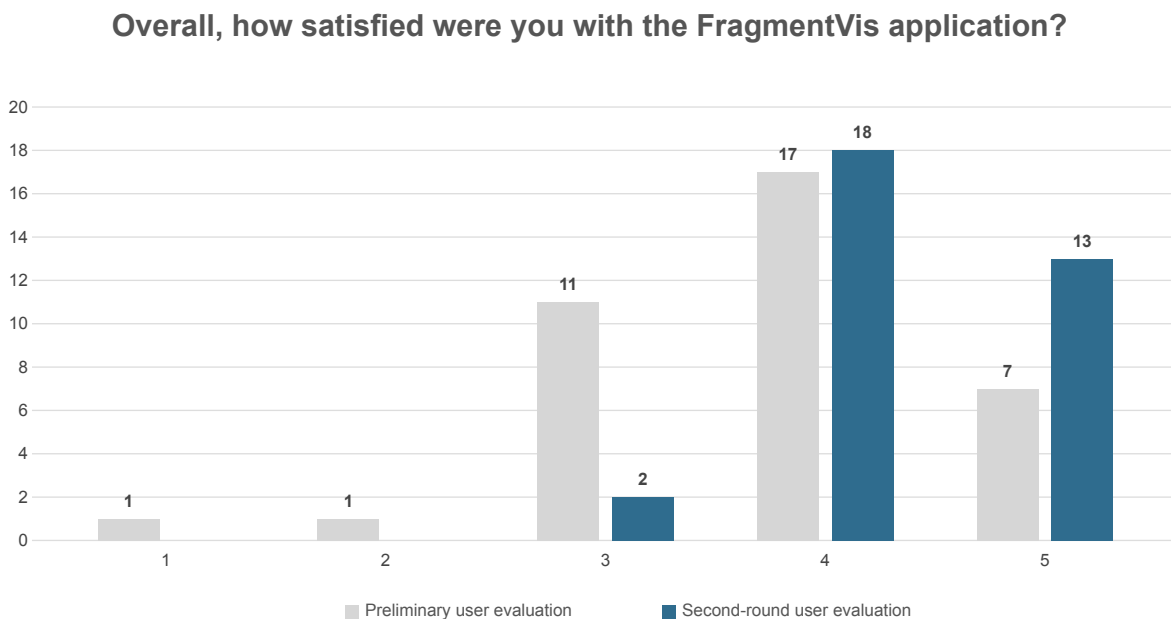


Figure 13: Overall satisfaction with the FragmentVis application.

that it produced substantially more new self-insights or improved objective puzzle-solving performance.

The usefulness rankings in the second round also indicate that the newly introduced dashboard-oriented analytical views were positively received. This time, participants could name the analytical element they considered most useful in a free-text field. The most useful elements were the *Dashboard: Pieces* view (10 mentions), the *Dashboard: Connections* view (7 mentions), the *Time Imprint* view (6 mentions), and the *Movement Paths* view (4 mentions). By contrast, the least useful element remained the *Grab Points* view (19 mentions), followed by *Movement Paths* (5 mentions), *other non-diagram elements* (4 mentions), and *Movement Distance* (3 mentions, as a standalone component from the dashboard). These results indicate that the newly added dashboard-based representations were well received, while the *Grab Points* perspective still appears to provide limited interpretive value for users.

The other open-ended responses further refine this picture. We also asked users about development needs for future versions. These needs concerned interaction, snapping, and

rotation issues (15 thematic mentions, e.g., rotation using keyboard input instead of the mouse wheel), the demand for more content and puzzle variety (14 mentions), user interface, help, and navigation improvements (11 mentions, e.g., configurable background color to improve the visibility of white images), and requests for additional analytics or playback-like functionality (11 mentions). Performance and loading issues were still present, although less dominant (5 mentions). Overall, the second round showed higher perceived satisfaction, usability, and analytical usefulness, especially for the new dashboard views, while the unchanged rate of newly reported self-insights indicates a remaining limitation.

6 Conclusion and future work

The results of the two-stage evaluation indicate that player-informed dashboard views and related visualizations can support the retrospective interpretation of digital jigsaw puzzle-solving behavior. Our findings suggest that directly incorporating players' experiences into the design process may contribute to more favorable perceptions of the system's usability and interpretability, especially when the analytical feedback moves beyond outcome measures and makes the construction of the solution itself more visible. Instead of focusing only on whether a puzzle was completed and how long it took, our proposed approach allows players to revisit the temporal and spatial development of their solution process at the level of individual pieces and local assemblies. In this way, the FragmentVis system can support the identification of strategy-related patterns, such as edge-first solving and the delayed handling of visually ambiguous regions.

The evaluation suggests that visual feedback can be perceived as useful when temporal and behavioral data are mapped back onto the spatial structure of the jigsaw puzzle. This is illustrated by the *Time Imprint* view, which represents the temporal order of piece placement directly on the layout of the completed puzzle and was positively received in the preliminary evaluation. More broadly, the later dashboard-oriented views suggest that spatially embedded visual feedback can make the solving process easier to interpret. By mapping piece-level data onto the completed puzzle image, these views help players compare the displayed patterns with the strategies they reported using during play. At the same time, the relatively stable rate of newly reported self-insights indicates that the system currently functions primarily as a reflective and confirmatory analysis tool rather than as an automatic strategy-discovery method.

These findings should be interpreted in the context of a formative evaluation with mainly casual puzzle players. The results therefore support the perceived usefulness and interpretability of the approach, but they do not yet demonstrate objective improvements in puzzle-solving performance. In addition, the two evaluation rounds were conducted with different versions of the application and partly different questionnaire structures, which limits the strength of direct statistical comparison between the rounds.

Future work will extend FragmentVis in two directions. First, we would like to study physical jigsaw puzzles through sensor-based data collection to capture the steps of the solution process in real-world settings. We also aim to examine differences between problem-solving strategies employed in digital environments and those used in physical contexts. Second, we plan to explore ways to support collaborative puzzle solving in future versions, where multiple players work on the same puzzle. In our opinion, the analytical focus could shift from individual reflection to how players assemble parts of the puzzle in parallel, hand pieces over to one another, and influence each other's solving process.

Acknowledgement

The authors declare that AI-assisted tools, including ChatGPT, were used only as supervised support during the preparation of the manuscript and the development process. In the manuscript, their use was limited to language editing, grammar correction, wording refinement, and the summarization of supporting materials. During development, AI assistance was used in a limited manner to help organize and improve selected parts of the application code. The scientific and technical work itself was conducted by the authors, from the conceptual framing and methodology to the visual analytics design, software design decisions, user evaluation, data analysis, and interpretation of the results. All AI-assisted outputs were reviewed and verified by the authors, who take full responsibility for the manuscript's content, accuracy, originality, and academic integrity.

The authors thank all respondents who participated in the user studies.

References

- [1] J.-C. Hong, M.-Y. Hwang, K.-P. Tam, Y.-H. Lai, and L.-C. Liu, "Effects of cognitive style on digital jigsaw puzzle performance: A GridWare analysis," *Comput. Hum. Behav.*, vol. 28, no. 3, pp. 920–928, 2012, doi: 10.1016/j.chb.2011.12.012.
- [2] Y. Yuan, S. Xu, and S. Lin, "Jigsaw puzzle difficulty assessment and analysis of influencing factors based on deep learning method," *Vis. Comput.*, vol. 40, no. 10, pp. 7133–7145, 2024, doi: 10.1007/s00371-024-03387-2.
- [3] C.-T. Sun, D.-Y. Wang, and H.-L. Chan, "How digital scaffolds in games direct problem-solving behaviors," *Comput. Educ.*, vol. 57, no. 3, pp. 2118–2125, 2011, doi: 10.1016/j.compedu.2011.05.022.
- [4] N. Javvaji, C. Harteveld, and M. Seif El-Nasr, "Understanding player patterns by combining knowledge-based data abstraction with interactive visualization," in *Proc. Annu. Symp. Comput.-Hum. Interact. Play (CHI PLAY)*, 2020, pp. 254–266, doi: 10.1145/3410404.3414257.
- [5] H. Freeman and L. Garder, "Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition," *IEEE Trans. Electron. Comput.*, vol. EC-13, no. 2, pp. 118–127, 1964, doi: 10.1109/PGEC.1964.263781.
- [6] R. W. Webster, P. S. LaFollette, and R. L. Stafford, "Isthmus critical points for solving jigsaw puzzles in computer vision," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 5, pp. 1271–1278, 1991, doi: 10.1109/21.120080.
- [7] D. A. Kosiba, P. M. Devaux, S. Balasubramanian, T. L. Gandhi, and R. Kasturi, "An automatic jigsaw puzzle solver," in *Proc. 12th Int. Conf. Pattern Recognit. (ICPR)*, vol. 1, 1994, pp. 616–618, doi: 10.1109/ICPR.1994.576377.
- [8] S. Markaki and C. Panagiotakis, "Jigsaw puzzle solving techniques and applications: A survey," *Vis. Comput.*, vol. 39, pp. 4405–4421, 2023, doi: 10.1007/s00371-022-02598-9.
- [9] P. Harel, O. I. Shahr, and O. Ben-Shahr, "Pictorial and apictorial polygonal jigsaw puzzles from arbitrary number of crossing cuts," *Int. J. Comput. Vis.*, vol. 132, pp. 3428–3462, 2024, doi: 10.1007/s11263-024-02033-7.
- [10] W. Li, M. Funk, Q. Li, and A. Brombacher, "Visualizing event sequence game data to understand player's skill growth through behavior complexity," *J. Vis.*, vol. 22, no. 4, pp. 833–850, 2019, doi: 10.1007/s12650-019-00566-5.
- [11] S. S. Maram, E. Kleinman, J. Villareale, J. Zhu, and M. Seif El-Nasr, "'Ah! I see' – Facilitating process reflection in gameplay through a novel spatio-temporal visualization system," in *Proc. CHI Conf. Hum. Factors Comput. Syst. (CHI)*, 2024, Art. no. 362, doi: 10.1145/3613904.3642484.
- [12] Y. Abeysinghe, B. Mahanama, G. Jayawardena, M. Sunkara, V. Ashok, and S. Jayarathna, "Gaze analytics dashboard for distributed eye tracking," in *Proc. IEEE 24th Int. Conf. Inf. Reuse Integr. Data Sci. (IRI)*, 2023, pp. 140–145, doi: 10.1109/IRI58017.2023.00031.
- [13] M. Choi, D. Cui, A. Koiliias, and C. Mousas, "The effects of virtual character's intelligence and task's complexity during an immersive jigsaw puzzle co-solving task," in *Proc. 17th ACM SIGGRAPH Conf. Motion, Interact., Games (MIG)*, 2024, Art. no. 12, doi: 10.1145/3677388.3696324.

- [14] M. Choi, S. Guo, A. Koiliyas, M. Volonte, D. Kao, and C. Mousas, "Exploring the effects of self-correction behavior of an intelligent virtual character during a jigsaw puzzle co-solving task," *ACM Trans. Interact. Intell. Syst.*, vol. 14, no. 3, Art. no. 23, 2024, doi: 10.1145/3688006.
- [15] OpenAI, "Images in ChatGPT," OpenAI Help Center. Accessed: Mar. 9, 2026. [Online]. Available: <https://help.openai.com/en/articles/11084440-chatgpt-images-faq>
- [16] Google DeepMind, "Nano Banana Pro: Gemini 3 Pro Image model," 2025. Accessed: Mar. 24, 2026. [Online]. Available: <https://deepmind.google/models/gemini-image/pro/>
- [17] Inkscape Project, *Inkscape: Free and Open Source Vector Graphics Editor*. (2026). Ver. 1.4.3. Accessed: Mar. 9, 2026. [Online]. Available: <https://inkscape.org/>
- [18] A. Abdul-Rahman, S. Walton, K. Bemis, J. Gonnering Lein, K. Coles, D. Silver, and M. Chen, "Re-spatialization of time series plots," *Inf. Vis.*, vol. 16, no. 4, pp. 275–290, 2017, doi: 10.1177/1473871616666395.
- [19] M. Evers, K. Huesmann, and L. Linsen, "Uncertainty-aware visualization of regional time series correlation in spatio-temporal ensembles," *Comput. Graph. Forum*, vol. 40, no. 3, pp. 519–530, 2021, doi: 10.1111/cgf.14326.

Bence Daniel Eros*,
Doctoral School of Informatics,
University of Debrecen,
Kassai út 26.
4028, Debrecen, Hungary;
Faculty of Informatics,
University of Debrecen,
Kassai út 26.
4028, Debrecen, Hungary
Email: eros.bence@inf.unideb.hu

Roland Kunkli,
Faculty of Informatics,
University of Debrecen,
Kassai út 26.
4028, Debrecen, Hungary
Email: kunkli.roland@inf.unideb.hu

Received 29.06.2026, Revised 16.08.2026 Accepted 08.09.2026, Available online 30.09.2026