Gameplay Analysis through State Projection

Erik Andersen[1], Yun-En Liu[1], Ethan Apter[1]
François Boucher-Genesse[2], and Zoran Popović[1]

[1]Center for Game Science
Department of Computer Science & Engineering
University of Washington
{eland,yunliu,eapter,zoran}@cs.washington.edu

[2]Department of Education
Université du Québec à Montréal
francoisbg@gmail.com

ABSTRACT
Analysis of gameplay data is crucial for evaluating design decisions and refining a game experience. However, identifying player strategies and finding areas of confusion is difficult because a designer may not know what queries to ask or what patterns to look for in the data. To make this task easier, we present Playtracer, a method for visually analyzing play traces that is independent of a specific game's structure. Playtracer applies multidimensional scaling to cluster players and game states, providing a detailed visual representation of the paths the players take through a game. We evaluate our method by analyzing an educational puzzle game and highlighting common hypotheses, pitfalls, confusing elements, and anomalies. Our results suggest that Playtracer can be an effective tool for game analysis and improvement.

Categories and Subject Descriptors
K.8.0 [**Personal Computing**]: General – Games

Keywords
games, data visualization, playtesting

1. INTRODUCTION
Analysis of gameplay data is an important component of the game design process. Playtesting helps game designers know what players are doing in the game and whether or not this behavior is expected [9]. These insights help game designers evaluate design decisions and iteratively improve the game. However, thorough evaluation of a game's structure is challenging because it often requires finding patterns in high-dimensional data from many players. Game companies spend a lot of time and money on playtesting, resulting in a need for data analysis methods that are efficient, powerful, and easy to use.

Recently, methods for automatic logging and game instrumentation have seen a surge in popularity [16]. These methods have made it feasible to collect detailed gameplay data from many players, and conduct quantitative and empirical analyses of this data. Examples of statistical analysis of gameplay include heat maps [1, 19] and quantitative visualizations coupled with video [16]. These techniques are useful for providing answers to specific queries the designers might have, such as determining the causes of death in a level of a first-person shooter game.

Most games encourage the player to acquire certain skills that are necessary to play the game. In order to evaluate whether or not players are mastering these concepts, it is crucial to classify the strategies that players are trying. Furthermore, it is important to examine when and how players become confused. Ideally, when players encounter a new concept, they will struggle with it for a bit before mastering it and moving forwards. Of particular interest to designers of educational games, confusion has been shown to be an important step along the path to deep learning [4]. However, only certain types of confusion are useful for learning. Confusion due to poor game design or unintuitive elements is problematic. Since game designers may not know what patterns to look for, it can be difficult to formulate queries to examine confusion and player strategies.

We present a new tool for exploring how large groups of players move through the space of a game. In games where the player moves through a virtual environment, it is often possible to visualize players' gameplay in relation to the environment itself. In many other kinds of games, however, player movement through the game space is abstract and must be shown in a more general way. In this paper, we attempt to visualize transitions through game states in a manner that is independent of the structure of the game itself. To do this, we apply Classical Multidimensional Scaling [3, 17] to project any game space onto two dimensions, giving a detailed view of how a group of players approach a particular level. Our tool, Playtracer, helps to show the common ways that players succeed and fail, identify pitfalls and anomalies, and track how a particular player progresses through multiple levels. We provide a simple interface to allow the user to adapt the visualization to explore the game space in detail. We evaluated Playtracer by using it to analyze an educational game that we developed, Space Rescue. We were able to identify common points of confusion in our game quickly and easily, and we also saw some unexpected approaches to puzzles. These results suggest that our tool should be useful for iterative game development and refinement.

FDG 2010, June 19-21, Monterey, CA, USA

2. RELATED WORK

Game designers have long employed playtesters for the purpose of game refinement [9]. Traditional methods include observational studies of players [10], videotaping players, asking players to talk out loud while they are playing, and question-and-answer sessions [1]. These commonly used methods provide insight into how players react to the game and high-level information on the strengths and weaknesses of a particular level. Although direct interaction with playtesters provides useful information, these methods suffer from a few key limitations. First, players may not be able to articulate their impressions accurately and may ignore important design flaws, resulting in incomplete data. Second, since the designers must spend time interviewing each playtester individually, the time required to execute these kinds of playtests does not scale well to large groups of playtesters.

In order to gather data from large numbers of players, and conduct empirical analyses of this data, game designers and researchers have experimented with statistical techniques. Such approaches include that of Kennerly [15], who explains how game designers can apply data mining to analyze how players in an MMORPG acquire experience. Ducheneaut et al. [8] track 220,000 World of Warcraft players in order to gather statistics about how many hours players spend in the game and how quickly they advance in level. Tychsen et al. [22] suggest recording game metrics to identify different play-styles and play-personas. DeRosa [5] describes how BioWare used statistics on playtesters in order to examine where players spend time and what special powers they use. Romero [19] studies how players improve through repeated attempts in the racing game Forza Motorsport 2. Drachen et al. [7] use emergent self-organizing maps to identify player types in Tomb Raider: Underworld. Statistical techniques can provide useful information, but they are typically designed to answer a specific query, and may be difficult to interpret.

Another method of gathering user data in the field of Human-Computer Interaction is to collect and analyze user interface events captured during interaction with a system [11]. One system that builds on and extends this model is Microsoft's TRUE system [16], which combines extensive behavioral instrumentation with attitudinal, demographic, and contextual data. Designers can formulate queries to extract information from this data and visualize their results alongside video playback of important points. This methodology can provide valuable insight into why users behave in the way that is observed. Our work expands on this method by providing tools to explore user behavior without asking specific queries, allowing designers to discover patterns they may not have thought to look for.

Current data mining approaches generally condense play information into characteristics or features, which are then analyzed for patterns. This works best when the designer has specific questions that the data can answer. It is more difficult to formulate queries when one is not sure what patterns are in the data. For example, it is difficult to detect player confusion by looking at game metrics, even when it would be obvious by watching a video of the gameplay. Humans are good at recognizing these kinds of patterns visually, and so it is easier to identify complex patterns by including a person in the analysis [13]. This is the premise of visual data mining, which has been been applied to many kinds of data [14]. It is particularly useful when one does not know beforehand what patterns one will find, such as in games.

Visual data mining has been applied before to analyze game player data. While developing Halo 3, Bungie and Microsoft used heat maps to determine common places of player death in order to find the most difficult parts of a level [21, 19]. This data was used to modify the topography of the environment and strength of enemies in order to minimize unfairness and frustration. Chittaro et al. [2] use heatmaps that track what players look at and where they spend their time in order to identify poor environment layout and player personalities. Others have attempted to show movement through a virtual environment to analyze the flow of battle [12], identify basic player behaviors [6], and find landmarks with multidimensional scaling [20]. All of these methods make use of the fact that the player is present in a virtual environment. To generalize to a broader class of games, we propose a visual tool suitable for any game with a concept of state.

3. PLAYTRACER

A natural way to think of player movement in arbitrary games is the path they take between different game states. A play trace is then a path that a player takes in this high-dimensional game state space. In order to visualize these paths, we use Classical Multidimensional Scaling (CMDS) [3, 17] to represent observed game states in two dimensions. CMDS takes an input matrix that specifies the distance between every pair of states and outputs a set of points, which are positioned to minimize a loss function on all interpoint distances. Therefore, the transformation will place states that are similar close together and states that are dissimilar far apart, making it easy to see the similarity of states that are visited by many players. We used the MDSJ library for Java [18].

Different metrics for calculating distances between states will result in different configurations of points after CMDS is applied. In general, the distance metric should be different depending on the type of game. Additionally, the distance metric can be adjusted depending on what features of the game the designer wishes to analyze or what features he or she wants the state graph to have. For example, if the distance metric has a component that compares how many steps it takes to reach a goal state, then it will naturally cause goal states to cluster together. States from which it is difficult to reach a goal state will appear far away. This allows the designer to identify players who are not making progress and investigate why they are having trouble.

An example of Playtracer's output can be seen in Figure 1, which shows the state space for one level of an educational game we have been developing. Playtracer takes in a list of all of the states that the player visited and a distance metric that calculates the distance between states, and creates a graph where the states are vertices and player movements are directed edges. Here, the yellow state is the start state and the green state is the goal state. To identify which states are most commonly visited, the size of a state is proportional to how many players reach that state.

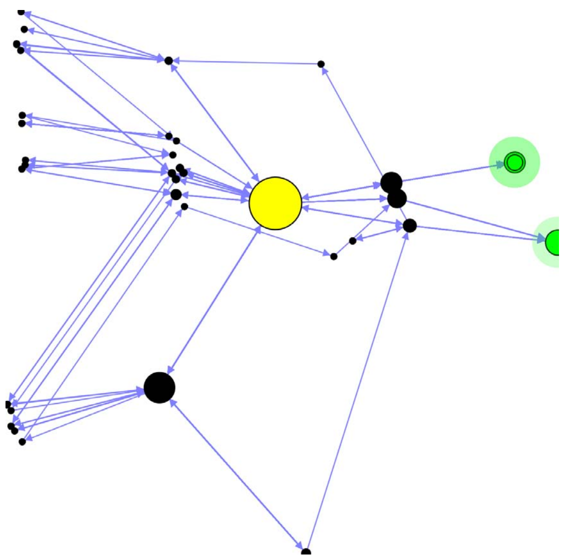


Figure 1: State visualization: Circles are game states; their size is proportional to how many players reach that state. Player paths between states are edges. Classical Multidimensional Scaling is applied to reduce game states to two dimensions and cause similar states to be drawn close together. The yellow vertex is the start state and green vertices are goal states.

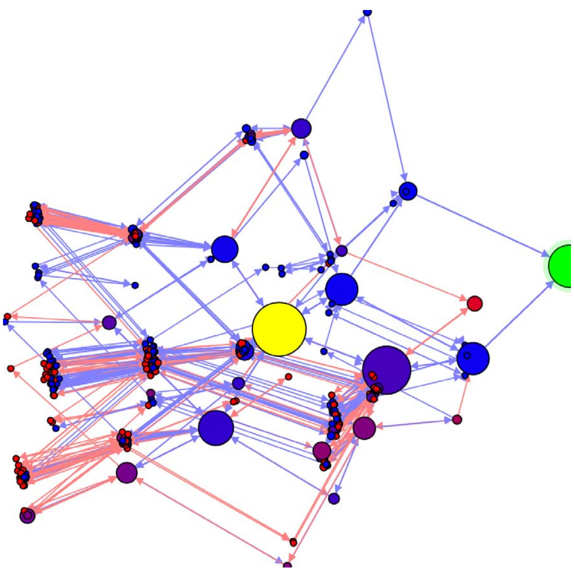


Figure 2: An alternative color scheme. Blue and red lines represent moves made by players that won and lost, respectively. States are shaded between blue and red depending on the probability that a player who reached that state completed the level successfully.

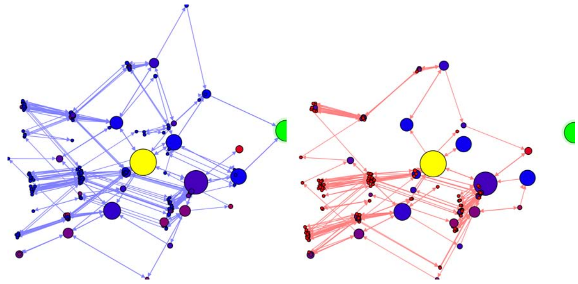


Figure 3: Viewing just winners (left) and losers (right) from the same state graph. Comparing these visualizations can show if winners and losers behave similarly.

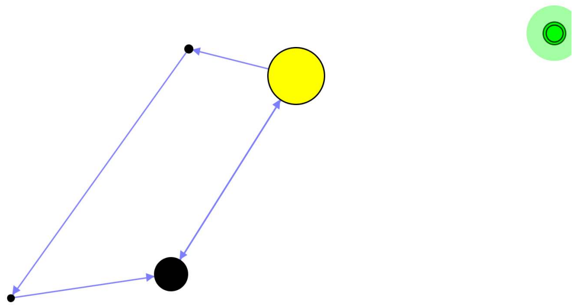


Figure 4: Viewing a cycle in the game state graph. Cycles correspond to failed hypotheses, as players make a move but return to where they started.

One useful way to analyze games is to find areas where a large proportion of players fail. If 90% of players who reach a similar set of states give up, then the designer should focus his or her attention on those states and the paths leading to it. To this end, Playtracer has an alternate color scheme, shown in Figure 2. Several players gave up on one or more attempts and are represented in red. Players that finished, on the other hand, are represented in blue. In addition, states are colored to reflect the probability that a player who ended up in that state went on to finish the level; bluer states are ones mostly visited by winning players, while redder states are ones mostly visited by losing players. Large red-hued states are of particular interest as they represent states visited by many people who mostly failed.

We include two additional features that might prove useful. One is to view only players that won or only players that lost in order to quickly see the differences between them, as shown in Figure 3. The second feature is that we can easily identify path cycles and show or hide them at will. Cycles, like in Figure 4, represent failed player hypotheses: the player started at a state, then went to several other states before returning. Viewing only cycles shows common ideas that players tried but later decided against, providing insight into how players think about the game strategy. Stripping cycles from the displayed paths, on the other hand, reveals the backbone of the path the player took to his or her final state; in essence, it is the last approach that players decided upon for that level.

Figure 5: The educational game Space Rescue. The goal is to split lasers into fractional pieces and redirect them to satisfy targets scattered on the screen. The game teaches the player about partitions of a whole and fractional division.

4. EVALUATION

In this section we describe the results of applying our method to a game.

4.1 Space Rescue

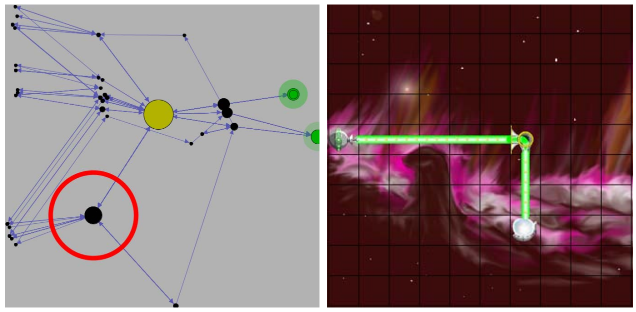
We evaluated our tool on Space Rescue, an educational game that we designed to teach fractions. This is a grid-based puzzle game in which players must split and redirect lasers to targets. In order to reach the targets, the player must correctly place a variety of pieces onto the grid; some of these pieces redirect lasers and others split lasers equally into two or three smaller lasers. All of the pieces have an entrance, indicated by a large funnel, and one or more exits, indicated by smaller nozzles. Targets also have entrances, indicated by a gray tube, as well as a label identifying what fraction of the original laser they require. The game ends when all of the targets have been satisfied. An example of a level of this game can be seen in Figure 5.

In order to visualize players' moves as they attempt to solve puzzles, we define the game state for Space Rescue to be a set of tuples with the form: (piece, piece coordinate), with any missing piece assumed to be in the side container. The distance metric has two equally weighted components: the number of piece pickups or placements required to change one state into the other, and the number of piece pickups or placements required to reach any observed goal state. This metric produces a visualization that shows paths to the goal as explained in Figure 6. Applying CMDS then causes states that contain the same piece in the same grid location to cluster. It has a similar effect on states which are the same number of actions away from a goal state.

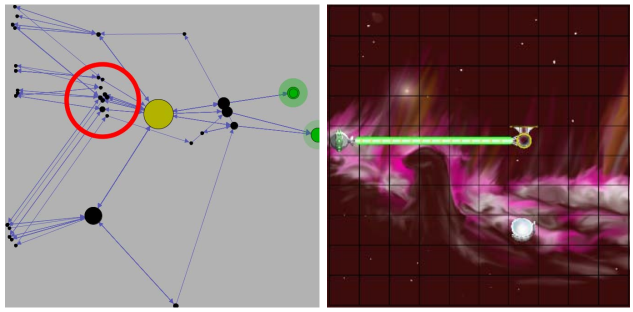
4.2 Analysis of Space Rescue

4.2.1 Level 2

We begin our analysis with level 2 of Space Rescue, seen in Figure 7(a) and solved in Figure 7(b). This level requires the player to place two pieces to redirect the laser to the target. There are multiple solutions because the pieces can be placed in any of the right three columns as long as they align with



(a) Incorrect target entrance



(b) Incorrect piece entrance

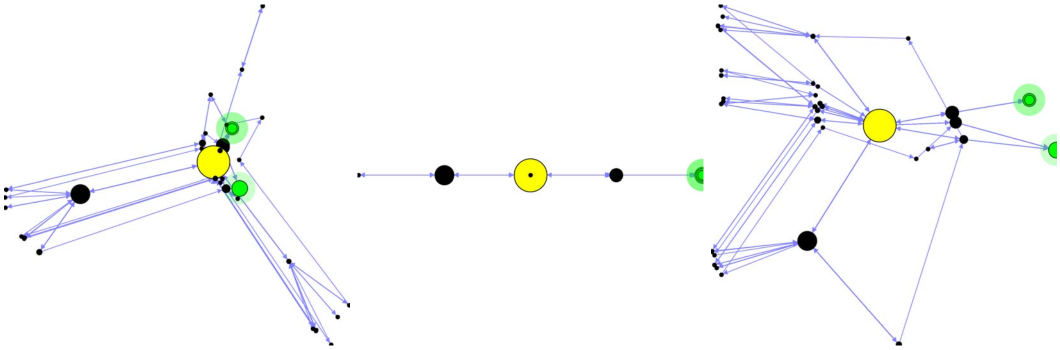
Figure 8: Two clusters in level 2, and their corresponding game states in Space Rescue. They are notable in the visualization as they lie farther from the goal states and include many players. They represent common classes of mistakes made on this level.

each other and are in the correct rows. The CMDS visualization of this level can be seen in Figure 7(c). 35 play traces are shown for this level. A glance at 7(c) shows that many of the players (19 out of 35) went through a state on the lower left that is further away from the goal states. This state is shown in the visualization and the game in Figure 8. Many players tried to hit the target from the top, even though a tutorial message explained that the laser must hit the target from the right-hand side. Further evaluation is necessary to determine why players are becoming confused here, which could be due to poor level design, ambiguous art, unclear tutorials, or simply that restricting the entrances of the targets is a difficult game mechanic that takes several levels to learn.

A cluster of points can be seen in the upper left of Figure 8(b). Examination of these states shows that roughly 20% of the players tried to place a piece with the entrance in the wrong place into the path of the laser, blocking the laser. The presence of this cluster could be evidence that the artwork for the pieces is unclear or that this game mechanic is not explained adequately. We show these two examples in order to highlight how a game designer can use our tool to look for clusters or large states in order to locate areas of possible confusion quickly.

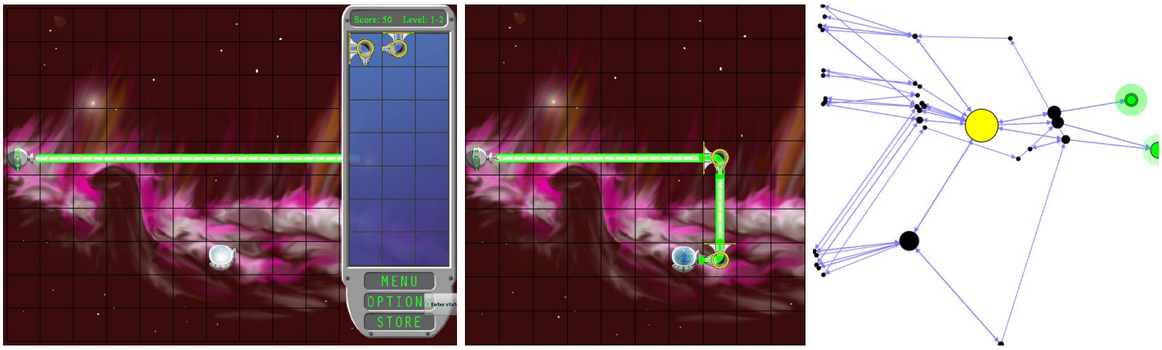
4.2.2 Level 4

The fourth level of Space Rescue is shown in Figure 9(a). This is the first level to require both splitting and redirecting the laser, as shown in Figure 9(b). The CMDS visualization is shown in Figure 9(c). The presence of many red arrows



(a) Inter-state distance only (b) Distance to goal only (c) Weighting both inter-state distance and distance to the goal equally

Figure 6: Weighting between distance metrics. 6(a) shows the result of using distances between states as a distance metric. This shows the true distances between states, but makes it difficult to see how players approach the goal states. Using only the distance from the goal as in 6(b) produces a straight line, but combining both metrics as in 6(c) yields a graph that shows paths towards and away from the goal. Combining these metrics seems the most useful for exploring how players move through the space of the game.



(a) Level 2 (b) One solution (c) Visualization

Figure 7: Level 2 of Space Rescue. Players must bend the laser twice because the target's entrance is on the right-hand side. The visualization in 7(c) calls attention to two clusters in the upper left and lower left that are further away from the goal than the start state. These clusters are examined further in Figure 8 and are shown to be examples of player confusion.

(a) Level 4 (b) Solution (c) Visualization

Figure 9: Level 4 of Space Rescue. This level requires both splitting the laser into halves and bending them to the targets. Even though this level only requires the user to manipulate three pieces, the state space in 9(c) is cluttered, indicating that players tried many different combinations of pieces. The red arrows indicate that several players failed. The visualization shows clusters in the upper left, middle left, lower left, and lower right that are examined further in Figure 10 and are shown to represent player strategies.

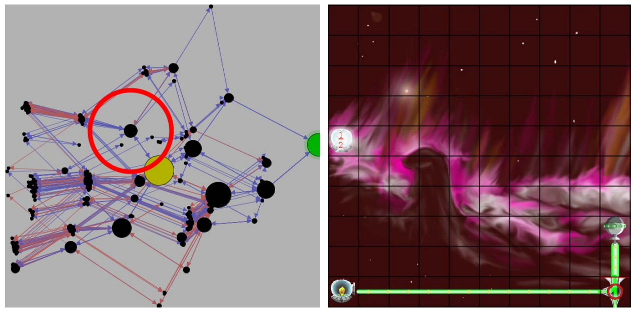
indicates that several players failed this level. However, the red and blue arrows are mixed together, indicating that both successful and unsuccessful players followed similar paths. Despite the fact that this level only has three pieces, the state space is quite cluttered, indicating that players tried many different combinations of pieces. This suggests that this level may have been particularly confusing.

We can also see three major clusters on the left-hand side of Figure 9(c). These clusters are further away from the goal than the start state, indicating that the players moved away from the solution to reach them. A fourth major cluster can be seen in the lower right, on the path to the goal. An example of a state from each of these clusters can be seen in Figure 10. Clusters 10(a), 10(b), and 10(d) represent logical hypotheses. Of these hypotheses, 10(a) and 10(b) are incorrect, but 10(d) is correct and is along the most commonly-taken path to the goal. Cluster 10(c), on the other hand, is a collection of states where pieces have been placed with no effect on the lasers. Even in this more complex example, our visualization is able to draw the user's attention to these clusters, providing clues as to why players seemed to struggle with this level. Furthermore, this example shows how our tool can be used to discover common player strategies.

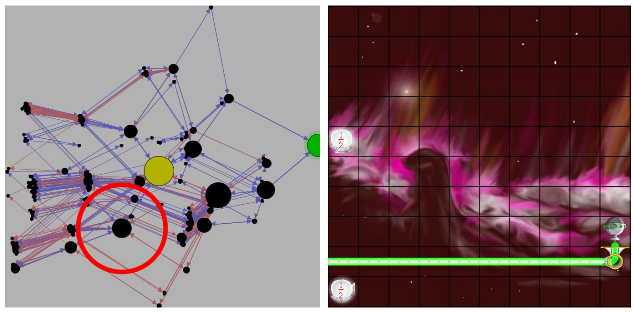
Figure 11 shows a visualization of the same level with the states colored based on the probability of losing after entering that state. Looking at this view, one can see a reddish-colored state that is close to the goal. Examination of this state shows that the two elbow pieces are in the correct place but the splitter has not yet been placed. Even though this state is a single move away from victory, most of the players who entered this state eventually failed. This example is further evidence that this level might have been confusing and premature. Most likely, the level should be simplified so only one bend is required, and the laser and targets are closer together. This state is an example of the kinds of anomalies our system can help a game designer detect.

4.2.3 Level 5

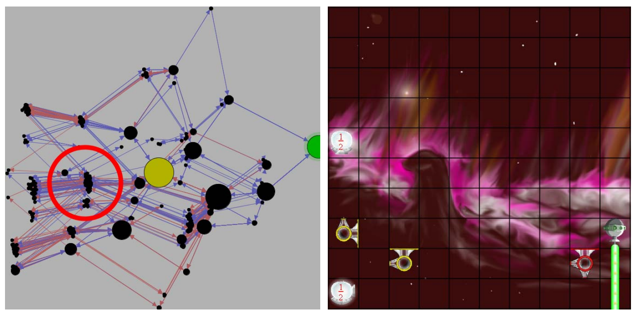
Level 5 (Figure 12(a)) requires the player to rotate a laser counterclockwise to reach the target, but is only given three pieces that all rotate the laser clockwise. Therefore, the player must place all three of these pieces so that the laser bends all the way around. From the visualization (Figure 12(c)) we can see that roughly 20% of the players go into the cluster visible in the upper left. This cluster corresponds to game states where the player incorrectly tries to use a clockwise-rotating piece as a counter-clockwise-rotating piece, even though the pieces cannot function in this way (Figure 13). Our visualization quickly draws attention to this area of confusion. In fact, the visualization shows that players do not make the same mistake with the other incorrect piece. One possibility is that the laser does not show direction clearly enough, so that players cannot tell if it is traveling from bottom to top or top to bottom: in that case, placing a piece that accepts the laser from above would be logical. The other possibility is that players care about the piece's exit direction and not its entrance direction. The target is to the left, so a player's instinct may be to place the piece that points left. In either case, using Playtracer reveals a cluster of similar problem states that the game designer can investigate further through traditional playtesting.



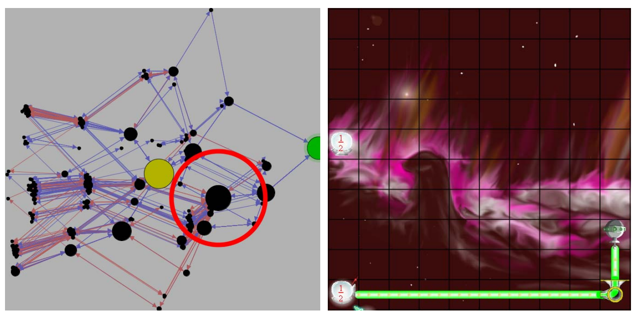
(a)



(b)

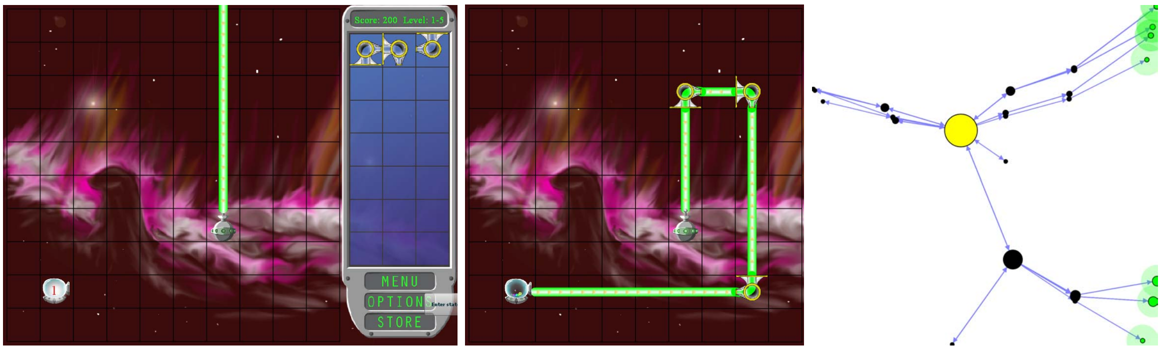


(c)



(d)

Figure 10: Some interesting clusters in level 4. 10(a) is a greedy strategy that satisfies one target but makes the other impossible to satisfy. 10(b) is a strategy that tries to direct the full laser near the targets. 10(c) is one of a set of states of complete confusion, where no piece does anything. 10(d) is a correct move. These clusters provide insight into player strategies.



(a) Level 5 (b) One solution (c) Visualization

Figure 12: Level 5 of the educational game Space Rescue. The goal of this level is to rotate the laser clockwise three times. The cluster in the upper left of 12(c) is examined further in Figure 13 and is shown to represent a state of confusion where players try to turn the laser counterclockwise instead of clockwise.

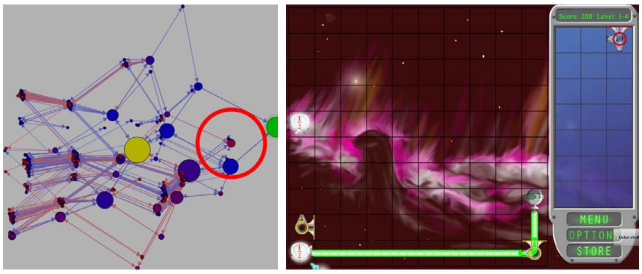


Figure 11: A deadly state in level 4. The red shading on this state indicates that most players who enter it eventually give up. Strikingly, the level is close to the goal - only one more move is required to win - but no player found the winning move. It is likely that this level is presented too early, as players still struggle with the spatial aspects of the game.

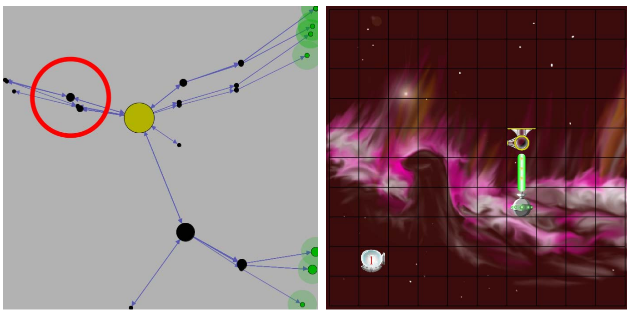


Figure 13: The main cluster of confusion in level 5 and the class of states it represents. The piece cannot be used in this way; the players most likely cannot judge the direction of the laser or are ignoring which side is the piece's entrance.

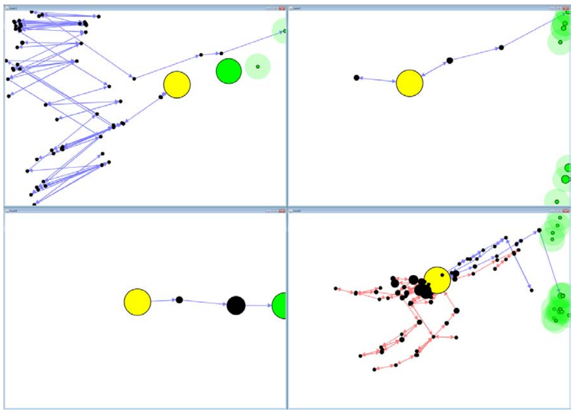


Figure 14: A single player tracked across multiple levels (3, 5, 8, 9). The player completes levels 5 and 8 easily, but struggles with 3 and 9, suggesting that the level ordering is not optimal. Closer examination shows that 3 and 9 require both splitting and bending, while the intervening levels require only one or the other; for this player, difficulty may come from combining both mechanics.

4.2.4 Multiple levels

When only viewing a single level, it is not always clear what a particular player is thinking. We can use Playtracer to track a single player or multiple players across several levels, giving a long-term view of how players play the game. Figure 14 shows an example of a player who stood out because he made many unnecessary moves in the earlier levels. Viewing his play traces, we can see that he quickly completes levels 5 and 8 but stumbles on levels 3 and 9. Taking into account the configurations of these levels, we can hypothesize about his behavior. A likely possibility is that he can solve levels that require only splitting or only redirecting lasers, but has difficulties when these game mechanics are combined. By comparing graphs from different levels, the game designer can reason about what players find difficult and adjust the game to maintain a steady difficulty progression.

5. LIMITATIONS AND FUTURE WORK

The usefulness of the visualization is highly dependent upon the chosen state distance metric. This provides some flexibility because game designers can examine different aspects of a game by varying this metric. However, poor metrics will not provide accurate visualizations, and some games are hard to design distance metrics for. For example, in a first-person shooter, a player with a rocket launcher might have an advantage over a player with a pistol, but it is not clear exactly how far apart they are in the game's state space. Another limitation of Playtracer is that the visualization tends to become cluttered as states and transitions increase. In these cases, traditional methods and statistical tools may be more effective. One area of future work is to make the visualization more manageable and robust by clustering similar states and paths and removing outliers. A second area of future work is to evaluate when game designers would prefer the state projection approach and to classify the games for which it is the most effective.

We plan to extend Playtracer in two additional ways. First, we hope to aid in the exploration of temporal patterns that are embedded in play data. For example, a game designer may want to observe patterns in the times at which players reach a particular state in the game. Our current system is primarily intended for viewing spatial relationships between game states and is unable to support visualization for temporal queries. This could be accomplished by adding a third dimension to the display and is an area of future work. Second, we believe that Playtracer could be useful for evaluating prototypes in iterative game design and refinement. We hope to explore the potential for using our tool to identify a weak point, fix the component, redeploy the game, and then show clearly that the weakness has been removed.

6. CONCLUSION

We have presented a way to project game states for gameplay visualization that helps a designer view how large groups of players move through a game. Our method visualizes game traces in a way that is independent of a game's structure as long as the distance metric between states is appropriate. We showed how this method can be used to analyze a game, highlight areas of confusion, provide insight into players' strategies, and identify portions of the game that need refinement. We have provided some initial insight into how to specify the distances between game states in order to maximize the usefulness for visual data mining, but more exploration is needed. Furthermore, we have only scratched the surface of how to use patterns of state transitions to differentiate between types of confusion and classify player strategies, and this remains a major problem to be solved.

7. ACKNOWLEDGMENTS

We acknowledge Brian Britigan, Seth Cooper, Dennis Doan, Happy Dong, Justin Irwen, Marianne Lee, Emma Lynch, and Jennifer Yang for their assistance in building Space Rescue. We also thank David Niemi and Ellen Clark of K12 for distributing Space Rescue. This work was supported by an NSF Graduate Fellowship, NSF grant IIS0811902, DARPA grant N00173-08-1-G025, the University of Washington Animation Research Labs, Adobe, Intel, and Microsoft.

8. REFERENCES

[1] M. Ambinder. Valve's approach to playtesting: The application of empiricism. Game Developer's Conference, Mar. 2009.

[2] L. Chittaro and L. Ieronutti. A visual tool for tracing users' behavior in virtual environments. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 40–47, New York, NY, USA, 2004. ACM.

[3] R. F. Cox and M. A. A. Cox. *Multidimensional Scaling.* Chapman and Hall, 1994.

[4] S. D. Craig, A. C. Graesser, J. Sullins, and B. Gholson. Affect and learning: An exploratory look into the role of affect in learning with autotutor. *Journal of Educational Media*, 29:241–250, 2004.

[5] P. DeRosa. Tracking player feedback to improve game design. *Gamasutra*, 2007.

[6] P. N. Dixit and G. M. Youngblood. Understanding playtest data through visual data mining in interactive 3d environments. In *12th International Conference on Computer Games: AI, Animation, Mobile, Interactive Multimedia and Serious Games (CGAMES)*, 2008.

[7] A. Drachen, A. Canossa, and G. N. Yannakakis. Player modeling using self-organization in Tomb Raider: Underworld. In *CIG'09: Proceedings of the 5th international conference on Computational Intelligence and Games*, pages 1–8, Piscataway, NJ, USA, 2009. IEEE Press.

[8] N. Ducheneaut, N. Yee, E. Nickell, and R. J. Moore. Building an MMO with mass appeal: A look at gameplay in World of Warcraft. *Game Developer Magazine*, 1:281, 2006.

[9] T. Fullerton. *Game Design Workshop.* Morgan Kaufmann, 2nd edition, 2008.

[10] B. Fulton. Beyond psychological theory: Getting data that improve games. *Game Developer's Conference 2002 Proceedings*, 2002.

[11] D. Hilbert and D. Redmiles. Extracting usability information from user interface events. *ACM Computing Surveys*, 32(4):384–421, 2000.

[12] N. Hoobler, G. Humphreys, and M. Agrawala. Visualizing competitive behaviors in multi-user virtual environments. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 163–170, Washington, DC, USA, 2004. IEEE Computer Society.

[13] D. Keim, W. Müller, and H. Schumann. Visual data mining. Eurographics 2002, 2002.

[14] D. A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 7(1), 2002.

[15] D. Kennerly. Better game design through data mining. *Gamasutra*, 2003.

[16] J. H. Kim, D. V. Gunn, E. Schuh, B. Phillips, R. J. Pagulayan, and D. Wixon. Tracking real-time user experience (TRUE): a comprehensive instrumentation solution for complex systems. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 443–452, New York, NY, USA, 2008. ACM.

[17] J. Kruskal and M. Wish. *Multidimensional Scaling.* Sage Publications, 1978.

[18] U. of Konstanz Algorithmics Group. MDSJ: Java library for multidimensional scaling (version 0.2). http://www.inf.uni-konstanz.de/algo/software/mdsj/, 2009.

[19] R. Romero. Successful instrumentation: Tracking attitudes and behviors to improve games. Game Developer's Conference, Feb. 2008.

[20] R. Thawonmas, M. Kurashige, and K.-T. Chen. Detection of landmarks for clustering of online-game players. *International Journal of Virtual Reality*, 6(3):11–16, 2007.

[21] C. Thompson. Halo 3: How microsoft labs invented a new science of play. *Wired*, 2007.

[22] A. Tychsen and A. Canossa. Defining personas in games using metrics. *Proceedings of the 2008 Conference on Future Play: Research, Play, Share*, pages 73–80, 2008.