A Method for Testing Graph Visualizations Using Games

John Bovey and Peter Rodgers[1]
University of Kent, Canterbury, UK

ABSTRACT

We describe a system that enables us to perform exploratory empirical experiments with graph visualization techniques by incorporating them into games that can be played on an internet site, the Graph Games Website (http://www.cs.kent.ac.uk/projects/graphmotion/). We present a general discussion of games as a test-bed for empirical experiments in graph comprehension, and explain why they might, in particular, provide a useful way to do exploratory experiments. We then discuss the requirements for games that can be used and describe some individual games including those we have tried on our own site.

The main part of the paper describes our own experiment in setting up a graph-gaming site where we have carried out tests on the benefits of different graph visualizations. We discuss the design of the site, describe its underlying architecture and present a promising initial trial on movement in graph visualization that gathered the results from over 70,000 played games. We then present some statistical conclusions that can be drawn from the resulting data. Finally, we summarize the lessons that have been learned and discuss ideas for future work.

Keywords: Graph Visualization, Graph Games

1. INTRODUCTION

Many of the information systems that people want to visualize contain graph-like structures – that is, they have objects with connecting links. Examples include social networks, software engineering diagrams and the world wide web. The natural way to visualize systems of this kind is to use graph diagrams in which the objects are positioned on the page and the links are shown as lines joining related pairs of objects. Hand-drawn graph diagrams have a long history, but we are interested in automatically generating diagrams, and in finding ways to make such diagrams clearer and more useful. There are a variety of ways to improve the display of a graph diagram, for instance, applying a graph drawing technique, improving the rendering of the graph on screen, or making graph diagrams more comprehensible by adding movement [1].

Given a range of graph visualization algorithms and techniques, we need ways to evaluate their usability. If we want to determine which of a pair of algorithms produces the best layout, or whether adding motion to a diagram makes it clearer, then we need to do an empirical study. That is, we need to ask people to perform tasks on graph diagrams generated in different ways and measure how well they do. Typically, this is done by assembling a number of paid subjects in a room and asking them to perform the task in a controlled environment [12]. Unfortunately, although this kind of controlled experiment works well when there is a clear hypothesis to test, they are expensive in time and money and so are not very practical for exploratory experiments. For example, we may have a range of different ways to modify a graph layout algorithm and want to determine which, if any, produce an improvement in comprehension. The results of experiments will lead to further experiments based on observations of the data and we are likely to need several iterations of this process. Another factor is that the rigour of exploratory experiments does not need to be as high as for more formal experiments since the goal is to generate hypotheses and direct development rather than clearly demonstrate that a hypothesis is true. Once the development cycle is complete then the final visualization system can be tested within a formal setting to provide better evidence of its effectiveness. Our own solution, described in the remainder of this paper, has been to set up a web site where ordinary internet users can play games on graphs while we monitor the results.

Whilst we have seen no previous work on using graph games on websites to evaluate visualizations in the manner presented here, there is considerable previous work that explores the connection between empirical experimentation and games. Examining player behaviour on existing online games can be used to study real world activities [3]. Various

[1] Further author information: (Send correspondence to John Bovey)
John Bovey: E-mail: J.D.Bovey@kent.ac.uk WWW: http://www.cs.kent.ac.uk/people/staff/jdb/
Peter Rodgers: E-mail: P.J.Rodgers@kent.ac.uk WWW: http://www.cs.kent.ac.uk/people/staff/pjr/

studies on games as an aid to education have been performed [7,11]. In addition, the connection between game theory and empirical studies is long established, e.g. [4,8]. In relation to work that applies the data from games to achieve other goals, the ESP game [6] is a two player task that asks users to identify images using words, the guesses are then used to tag the images for subsequent image database searches.

Although such a website could be used to carry out exploratory experiments into any graph presentation technique (comparing different layout algorithms for example) our interest is in ways to make graphs clearer by adding movement, hence the first study of our Graph Games Website was in this area. There is some experimental evidence that adding small amounts of local motion to graph diagrams can make them easier to comprehend [1] but the experiments were carried out on graphs that had been artificially constructed to include occlusion, either an edge running behind a node or two overlapping nodes. A question that remained open was whether motion could help on graphs that had already been laid out to make them as clear as possible.

The remainder of this paper describes a platform for exploratory graph-comprehension experiments based on a games website, and the results of some initial experiments carried out using it. Section 2 describes the Graph Games Website and the three games that we have implemented. Section 3 describes the variant of phased circular motion that performed best in the experiments while Section 4 describes the experiments themselves and presents their results. The final section, contains some conclusions and plans for the future.

2. A GAME-BASED EXPERIMENTATION WEBSITE

In this section we discuss some of the issues that need to be considered in setting up a graph games website.

Why use Games?

If our goal is to do exploratory experiments into graph comprehension then we need an environment in which a sufficiently large number of people are happy to perform tasks on graph diagrams, and graph games are promising for this purpose. Games have several big advantages as tasks for doing empirical experiments. Firstly, they are self-motivating. If the games are good enough then people will play for pleasure and will genuinely try to win. Secondly, game players do not, on the whole, mind having the results of their games recorded since this would, anyway, be necessary for features such as level-progression and high-score tables. Thirdly, we can assume that playing a game on a graph typically involves similar cognitive understanding of the graph to that needed for other kinds of graph applications. Fourthly, the potential number of subjects for the experiments is very large if the games can be played via the internet. Finally, game players are tolerant of having arbitrary looking changes made to the way games are presented. A person using, say, an email program would object strongly if the font used to display messages changed in a random way. Someone playing treasure-hunt games on graphs would not object if some of the graphs were moving and some were still – it would be seen as part of the game. This suggests that a game playing web site might provide a good platform for exploratory experiments in graph comprehension.

Using a games website to evaluate visualization techniques does also have some drawbacks. One is that we have no control over who is playing the games or how. This means that games that appear to have been played by an individual may have been played by a group gathered around a computer, or a succession of people using the same name. However, we would expect this to be mitigated if a large quantity of data over a large number of subjects is recorded. Another problem is that game players will be a rather biased sample from the population as a whole. It is unlikely that those attracted to playing games on the internet are evenly distributed across the population in terms of age, background or gender. However, other, more formal empirical experiments suffer from similar bias, where often the population is taken from the student body of the institution where the experiments are being run. This being said, if through our exploratory method, we can find visualization techniques that seem to work in games then it is still possible to test them in a more controlled way on subjects more typical of likely end users.

2.2. Requirements for the Platform

A game playing platform for graph experiments has to satisfy requirements from two directions: game playing and experimenting. It needs to provide a game-playing experience that is sufficiently enjoyable and rewarding that enough people will play enough games to provide data, but it also needs to let us vary the parameters that we want to test and capture the results for analysis. On the game playing side, we need to provide games that start off easy but get more difficult. We also need to provide goals, either game-winning or scoring, that are potentially achievable but challenging.

This may seem like an obvious point but it is not trivial to do if the games are generated automatically. Finally, we need to provide motivational trappings, including feedback about how well a player is doing and high score tables. On the experimentation side, we need to be able to vary the experimental parameters independently of the games. In our case, this is not difficult to do since we are interested in the effects of movement on graph comprehension, and movement can be added to a graph game without interfering with the game itself.

2.3. Choice of Game

There is a wide range of games that could potentially be played on graphs but many of them are not suitable for a platform like ours. For example, some well-known games like Go, Checkers and Hex are played on specific graphs and could be modified for more general graphs, but it would be difficult to program a computer opponent or the game would impose symmetry constraints on the graphs. In practice, the requirements of the experimental platform result in some requirements for the graph games themselves, and these are listed below.

1. We need to be able to generate example games automatically in a range of difficulty from easy to impossible.
2. The generated games need to present a predictable target. If they are win-lose games then they should be potentially winnable, though not too easily. If the result of the game is a score then we need a target score that is achievable.
3. For a two-player game, we must be able to program an opponent to play the games at the desired level of difficulty.
4. A game should not place any special requirements on the graph and should be playable on graphs with different layouts, rendering and topological structure (for example: large, small, dense and sparse graphs).
5. The games need to be playable by people who do not know anything about graph theory and graph terminology.

2.4. The Games

Our site currently offers three different games – two variants of the Shannon Switching Game and a treasure-hunt game. The Shannon Switching Game is played on a graph with two pre-selected endpoint nodes. The players alternately choose edges, with one player trying to form a path between the endpoints and the other removing edges from the graph. Because of its asymmetry, this effectively provides two different games for players, the path-blocking game and the path-seeking game.

From our point of view, the Shannon Switching Game has some significant advantages and some defects. Its main advantage is that it can be analysed mathematically – it was proved in [9] that a graph is winnable for the path-seeker, playing second, if and only if it has a subgraph containing both endpoints and two edge-disjoint spanning trees. This is very useful because it enables us to generate random games that can be won by the first player, either blocking or seeking [2]. If we assume that the human player will go first, that, in turn, lets us create games that can won by the human player but are not a walk-over (a bad move will give away the advantage).

The Shannon Switching Game satisfies all our requirements except one. We can choose to ensure all games can be won by the player, hence we need specific graphs, so failing point 4, or we can present arbitrary graphs that we are not sure can be won, so failing point 2. We chose to ensure the games could be won by the players and so the graphs have to contain two edge-disjoint spanning trees, which fixes the ratio of edges to nodes at about two – we cannot use very sparse or very dense graphs.

We have also developed a treasure-hunting game as a third alternative, this allows us to present (nearly) arbitrary graphs, and also permits us to see if a different type of game changes which visualizations are most effective. In this game, some of the graph nodes have treasure on them, and the two players move around the graph following edges and trying to pick up treasure. The players may not move to nodes which the other player is currently visiting. This game does not impose any conditions on the graph other than that it should be connected, but we do need to provide players with a target quantity of treasure so that they can tell whether they are succeeding or not. One way to solve this is to construct an arbitrary graph but then add another two nodes as starting points for the two players. If these additional nodes have edges to the same group of nodes in the main graph then the two players start in symmetric positions and the one who plays first (the human) should be able to pick up at least half the treasure. Figures 1 and 2 show the Shannon Switching and treasure hunt games being played.

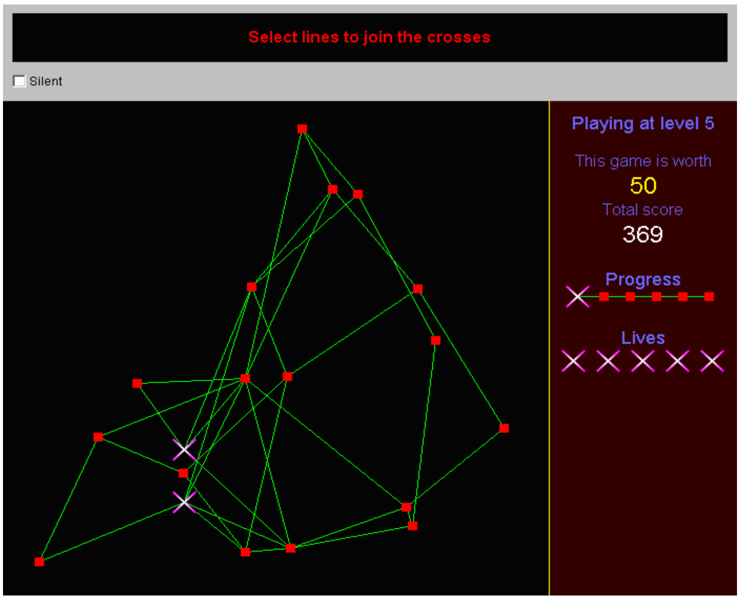

Figure 1. Seeking Variant of Shannon switching game at Level 5

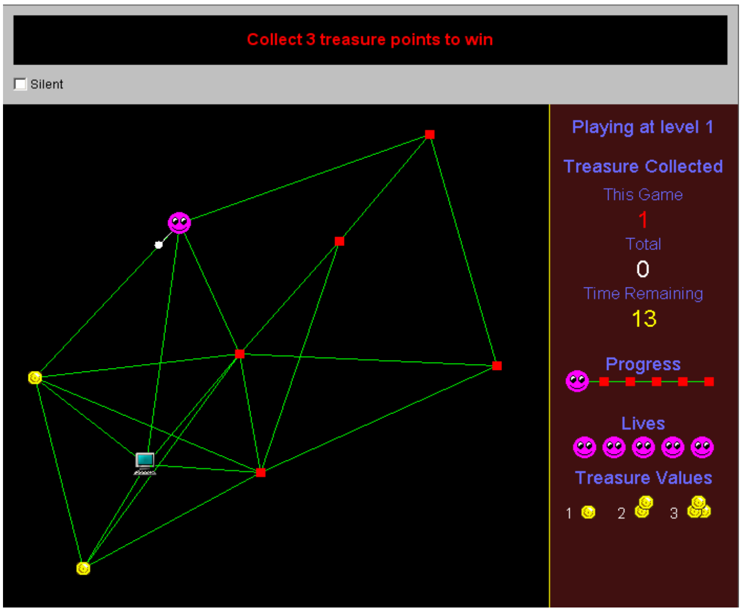

Figure 2. Treasure hunting game at Level 1

To support our assumption that playing games relates closely to real world tasks on graph visualizations, we can examine the cognitive requirements for the three games. The blocking variant of the Shannon Switching Game requires the player to disconnect two nodes in the graph, whilst looking ahead to the edges that the computer might choose to form a path. This inevitably uses general graph analysis in understanding the structure of the diagram as well identifying likely key edges that will form sections of paths. The player can also attempt to win by trying to spot the computer's strategy – and forming a method for foiling it, this requires the player to identify the path, or paths that the computer may be planning to use based on the edges it has already chosen. The seeking variant is similar, however the emphasis is not on working out ways of disconnecting the graph (unless the player is attempting to divine the computers strategy), rather it is on path planning and navigation between two nodes in the graph. We argue that the general graph understanding, identification of likely graph edges that will disconnect the graph and node to node navigation are tasks that users of graph in other application areas are likely to engage in. For the treasure hunt game, the task is slightly different. The player is attempting to find a path through a number of nodes. However, the complication that the computer may not visit a node that the player is currently visiting means that there is a strategic aspect to the game that involves the identification of nodes that are connected to both the node the player is currently visiting and the node the computer is currently visiting. The first task, of finding a route through multiple points relates closely to common navigation tasks on graphs such as the travelling salesman problem. The second task, of identifying close common nodes is perhaps not so widely applicable, but is still a task that requires understanding of the graph.

2.5. The Games Website

First-time visitors to the games site are asked to enter a name. This is not verified and most players use a nickname. It is used in the high-score tables and also allows us to group the games by player when we analyze the results. The user may also, optionally, enter an email address. The email address is used to inform users when new games appear. The player is then shown a summary page with a brief explanation of the games and links to the three games that currently available. Each of these links goes to a page with instructions for playing the game and a high score table. Clicking another link loads a page with the game playing applet. The games themselves are organized into levels of increasing difficulty – a player moves to the next level by winning 5 games at their current level, but a player who fails at 5 games at any level has lost and has to start again at level 1. All the games are time limited. For the Shannon switching games, the winnable score counts down by one point each second and so a player who wins quickly gets a higher score than a player who takes longer. For the treasure collecting game, there is a simple timer that counts down while there is still uncollected treasure in the graph. If the player has not collected half the treasure when the timer reaches zero then they lose that game. This infrastructure of scores, levels and timers is intended to make the games more interesting and to encourage each player to keep playing for as many games as possible.

2.6. Architecture of the Platform

The experimentation platform consists of three main components:

- a java applet to handle the game play;
- a MySQL database with tables of game-graphs (coded in XML), layout styles (also in XML), a table of users and a log table with a history of the games that have been played, how long they took and the result;
- CGI scripts written in PHP that handle requests for new games from the game applet.

When a game finishes, the applet requests a new game from the games URL and, at the same time, reports the result of the completed game. The PHP script logs the results of the completed game. It then selects a game-graph for the next game based on the player's history of wins and losses, and independently selects a layout style at random from the layout table. These are put together in an XML package which is sent to the applet ready for the start of the next game. The applet handles all the game play, displaying the graph, processing moves by the player and generating the opponent's moves. None of the game playing modules make use of the layout information, they purely use the graphs logical structure of nodes and edges, so any difference in play between different layouts are a result of differences in the human player's perception of the graph.

3. PHASED CIRCULAR MOTION

In our initial series of experiments we experimented with a number of different ways to add motion to graphs. The kinds of motion that we tried are listed in Section 4 but the only one that seems to improve comprehension is the variant of phased circular motion described here. In phased circular motion, the nodes move in small circles. An advantage of this kind of motion is that we can use the same size of circle and speed of motion for each node, but make the nodes move relatively to each other by giving different nodes different phases. We can then control the amount of relative motion between a pair of nodes by adjusting their relative phase angles. Two nodes that are 180 degrees out of phase will move a lot relatively to each other whereas two nodes that are only a few degrees out of phase will only have a little relative movement.

Our experiments in [1] showed that there is clear benefit from adding circular movement to graph diagrams that contain occlusion but that adding too much movement can make graphs less comprehensible. The fact that there is a trade-off between the positive and the negative effect of movement suggests that we should make the movement as small as we can (consistent with it working) and we should restrict it to the parts of the graph where it will do most good. In other words, we should try and arrange for potentially occluding lines and nodes to move relatively to each other, while nodes and edges that have no possibility of occlusion remain stationary.

3.1. The Phase Allocation Graph

The technique we used is to assign phases is to construct a new graph (the Phase Allocation Graph) in which the nodes are the same as in the original graph but the edges join pairs of nodes that should move out of phase with one another. This will be achieved if we join each pair of nodes that are very close, and each pair of nodes for which one node is close to an edge that has the other node as an endpoint. This is illustrated in Figure 3 in which the original edges are the solid straight lines and the Phase Allocation Graph edges are the curved broken lines. Node C is linked to nodes B and D because it is very close to the edge joining B and D. Nodes D and E are linked because they are very close to each other. Note that the term *close* here is deliberately imprecise. In practice, *close* means *able to occlude* and depends on the displayed sizes of the nodes and whether they have labels. We should also point out that the Phase Allocation Graph is an abstract graph – there is no need to lay it out or display it.

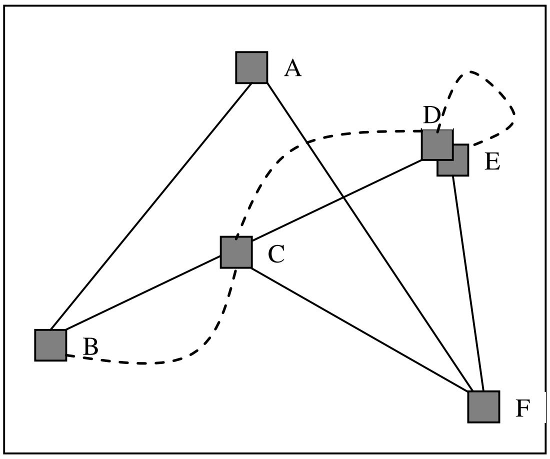

Figure 3. A graph with phase allocation edges

3.2. Allocating Phases

At its simplest, phase allocation is now reduced to the problem of assigning phases to nodes in such a way that no two joined nodes have the same phase. If we are using *n* equally spaced phases then this is just the classical graph colouring problem. One technique is to use a simple heuristic algorithm that starts off with a sequence of phases and a sequence of nodes. It then runs through the nodes in order, assigning to each node, the first phase that has not been assigned to any of its neighbours. There are three additional points worth making.

- Unless the original graph diagram is very complex, this method of phase allocation should need only a small number of different phases. This would make it possible to use a small circle of rotation.
- Since most nodes will get allocated to the first phase we can make these nodes stationary.
- Since the goal is to maximise the relative movement of linked nodes, we can order the sequence of phases to try and achieve this; for example, for 8 phases a sequence like that in Figure 4 gives a good separation.

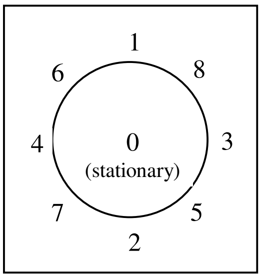

Figure 4. Phase sequence

If this sequence of phases was used to map nodes to phases in Figure 5 then nodes A, B, D and F would be stationary and nodes C and E would be assigned phase 1.

3.3. Nodes with Opposing Edges

Figure 5 shows a typical example of a ambiguity in a graph diagram; the two horizontal edge segments could be connected to the node or they could be parts of the same edge that passes behind the node. In the latter case, the algorithm we have just described would ensure that the occluded edge and the node will have a small amount of relative motion, and so it will be obvious that they are not connected. Unfortunately, though, if the edges do connect to the node then the node could be stationary, which would not give any clear visual clues that the edges belong to the node. A refinement to the algorithm described above is to make sure that all nodes with opposing (and hence ambiguous) edges are moving rather than stationary. This can easily be done by detecting these nodes and skipping the stationary option (phase 0, above) when searching for a phase to allocate to them.

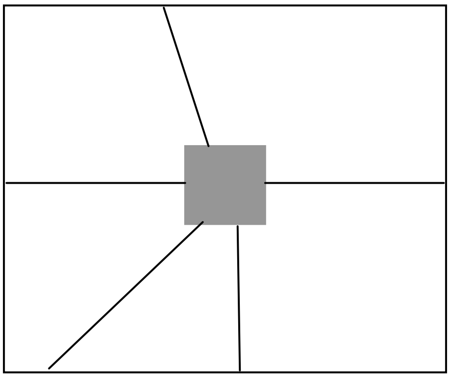

Figure 5. An Ambiguous Node

3.4. Details of the motion used in the experiments

There is almost endless scope for varying the details of phased-circular motion, but for these experiments we used a circle with a diameter of 23 pixels and with 60 points on its circumference. The animation refreshed 25 times a second and so the moving nodes completed each full circle in 2.4 seconds. The nodes themselves were drawn as red squares, 9 pixels across and the edges were displayed using green lines one pixel thick.

4. THE EXPERIMENTS

Our initial experiments had two goals: to determine whether people would be willing to play our graph games in sufficient numbers for us to get useful data, and to try out some kinds of graph motion to see if they provide any benefits. After a short internal trial we made the site accessible on the internet and put announcements on some sites used to promote internet games. There was an initial peak of activity (with nearly 15,000 games played on the busiest day in March 2006) which tailed off to its current (July 2006) average of about 20 games a day, although over the last month the number of games played in a day has varied between 0 and 1000. However, after the initial set of experiments we have not introduced any new games or made any more attempts to advertise the site. The data described in this section was taken on 13 July 2006.

Since the goal of the site is to provide a cost-effective platform for exploring different graph presentation ideas, we tried several different kinds of added motion in the first few weeks of the project. Most of these did not produce any statistically significant improvement in game results (and hence, graph comprehension) but the investigation did eventually produce a graph motion which produced a significant improvement over still graphs – the phased circular motion described in Section 3 above.

4.1 Exploration Method

As we gained data from the games played we modified the movements present in the game, and to a slight extent the way the games were played. We consider that this approach has been effective in guiding our search for an effective movement to help visualization and would expect this exploratory approach to be followed in future investigations.

Layout name	Layout Description	Stage
2D-still	2D layout, no rotation	1,2,3
Phased-circular	2D layout with the phased-circular motion from Section 3	2,3
3D-still	3D layout, no rotation	2,3
3D-small	0.4 radius 3D rotation	1,2,3
3D-medium	0.8 radius 3D rotation	3
3D-large	0.12 radius 3D rotation	3

Table 1. Types of movement applied

We were interested in seeing if movement could help with understanding graph diagrams even if they already had an effective layout algorithm applied, hence all of the variants have first been laid out using a spring-embedded layout algorithm. The 3D variants had a 3D spring embedder applied; the 2D variants had a 2D spring embedder applied. Our investigation had had three stages:

1. There was an initial, internal stage where we advertised the games among computer science students at the University of Kent. This phase used two different layout styles, a 2D layout without movement and an oscillating 2D projection of a 3D layout, (2D-still and 3D-small see Table 1). The results from the first stage were not promising and we hypothesised that this was due to the difficulties in comprehending the 2D projection of the 3D layout because of the naïve projection chosen in this stage.
2. For the second stage the number of visualizations was increased and the web site was publicised on external games listings. We changed the projection of 3D-small to one that removed occlusion. We added a 3D still visualization (3D-still) with a occlusion removing projection so we could have a visualization to compare the 3D moving against. We could be fairly sure that if the 3D moving layout was not an improvement on the 3D still, then it would be unlikely to be better than a 2D layout. We also added the 2D phased circular motion described in the previous section (Phased circular). Despite these changes the 3D moving layout was still not as effective as the 2D still, and was no improvement on the 3D still layout. However, the phased circular motion showed some promising results and we continued it unchanged in Stage 3.
3. For the third stage we concentrated on investigating 3D movement further, in particular we wanted to discover whether any parameter changes could improve the understandibility of graphs visualized this way. Hence, we changed the amount of movement, adding two larger movements, (3D-medium and 3D-large). We retained the previous movement types for comparison, making six in all. We also made a minor change to the presentation of the games in Stage 3 in that the countdown timer did not start until the player's first move rather than starting as soon as the game was presented, as in Stage 2. This change was aimed at making the games more enjoyable to play and does not seem to have had a significant effect on the results. This stage confirmed the results of Stage 2

in that there was no evidence that the moving 3D visualizations helped with understanding of the graphs. However, the phased circular motion appears to have a significant beneficial effect on larger graphs. More detail on the statistics for the successful visualization is given in Section 4.3.

4.2 How Players Used the Site

The numbers of completed games for each stage, is shown in Table 2. There is clear evidence that people will play the games – it contains more than 70000 games, most of them after we got a mention on a popular games site.

	Stage 1	**Stage 2**	**Stage 3**	**All Stages**
Blocking	610	7091	4930	12631
Seeking	2305	21386	19752	43443
Treasure	2114	6289	5654	14057
All Games	5029	34766	30336	70131

Table 2. Number of Games Played in Each Stage

The total number of players who registered for the site was 6385. The largest number of games played by any one player was 802, 1.15% of the total number of games and there were 99 players who completed more than 100 games each, indicating that the data has not be overly affected by a small number of very keen players. Figure 6 gives (on a logarithmic scale) the distribution of the number of games played by visitors to the site. As can be seen, there is quite a rapid fall-off, with a large proportion of visitors playing only a few games. Only 2020 played 10 games or more. Naturally, it would be preferable not to have such a rapid fall-off of players, and we may be able to reduce it if we improve the games. On the other hand, a lot of wastage is probably an inevitable feature of the web environment we are working in. At present, in mid July 2006, there is still some activity on the site, several months after the main publicity for the games, and typically one or two players log on to the games each day.

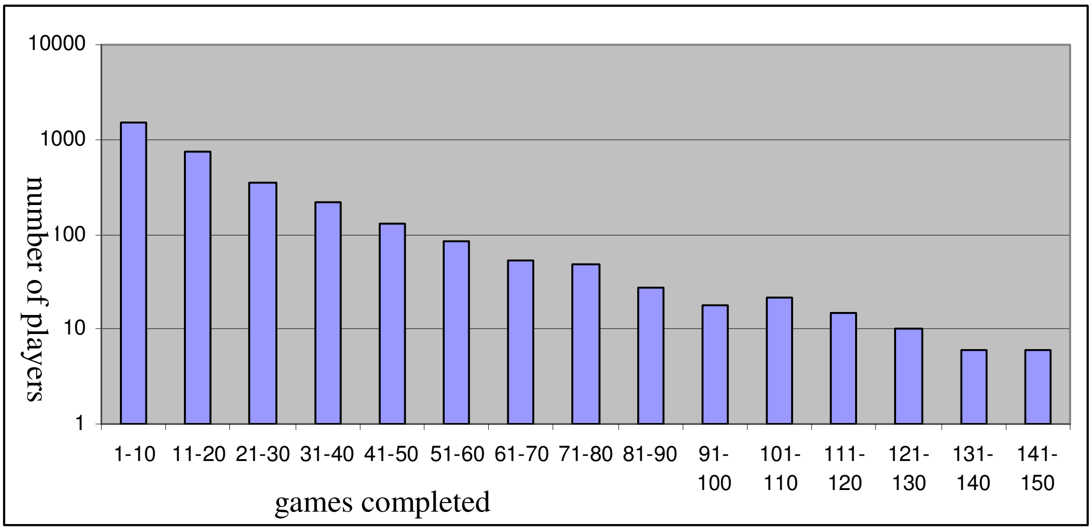

Figure 6. Number of Players and Games

4.3. Experimental Results

The Graph Games Website is designed as an exploratory mechanism, and it is in the nature of this type of investigation that most of the ideas tried do not provide any benefits and can be discarded – this was the case with the 3D movement variants that we tried. However, the phased circular motion did appear to provide statistically significant benefits in some situations, and looks like a good candidate for further investigation. In this section we compare this motion against the 2D still variant. Table 3 shows the numbers of won and lost games for each of the three games at level 4 or above when they were presented without movement and with the added phased-circular movement. The lower levels had smaller graphs, and so the benefit of movement was lesser in those cases.

	Treasure hunt		**Shannon seeking**		**Shannon blocking**	
	still	**moving**	**still**	**moving**	**still**	**moving**
Won	577	568	405	411	161	202
Lost	218	212	286	269	127	107
Win %	72.6%	72.8%	58.6%	60.4%	55.9%	65.4%

Table 3. Won and lost games at level 4 and above

The table shows a striking difference between the three games. The presence of motion seems to have made virtually no difference to the proportion of wins in the treasure hunt game. For the Shannon seeking game (where the player tries to make a path and the computer tries to prevent it) the players have a slightly higher proportion of won games on the moving graphs but the difference is not statistically significant. For the Shannon Blocking game, the difference is larger. If we view the Shannon blocking data as a 2 by 2 contingency table and adopt a null hypothesis that motion has no effect on the outcome of the games, then applying a chi-square test with one degree of freedom and corrected for continuity [10] gives a chi-square value of 5.22 and probability of having such a high proportion of won-moving games of 0.022. This is better than the usual 5% probability used for significance tests and means that we can be fairly confident that players of the Shannon Blocking game are being helped by the presence of motion. Moreover, the improvement in understanding of the graphs comes despite the extra cognitive and motor load required to find and click on moving objects.

The result above is, of course, not conclusive and needs to be confirmed by further experiments, but it does lead us to question why motion helps with one game but not with the others. For the treasure hunt game, the results are not surprising. In its present form, a good playing strategy is to take the shortest route to the largest piece of treasure and this can often be done without needing to understand much about the structure of the graph. It is less clear why there should be such a difference between the two Shannon switching games. We will need to do more experiments to discover whether the results reflect real difference in the cognitive tasks involved and, if so, what those differences are.

5. CONCLUDING REMARKS

The long-term goal of our project is to establish an experimentation games site that runs continuously and can be used to try out new ideas for improving graph visualization. To achieve this we need to maintain a continuous base of players, which means having a collection of games that are good enough and interesting enough to attract new players, as existing players move on to something else. We also need an underlying architecture that will allow new experiments to be plugged into the existing site without disrupting existing experiments or game playing. Although we may not have achieved that yet, we feel that we have achieved a large step towards it. We have shown that graph games can be set up to perform experiments, and that there are plenty of people who will play them by choice. The number of players peaked after the site was advertised and then tailed off but there are signs that it is levelling out. As we develop more games and get better at presenting and advertising them it should be possible to maintain a steady state with several thousand played games each day. It should also be possible to improve the games so that more players continue through to the more complex graphs where layout techniques can make a real difference.

Another reason to develop more games is that different games inevitably test different aspects of graph comprehension. Some games might involve looking for local patterns in graphs whereas other games might involve understanding overall connectivity. It should also be possible to devise games that depend on the player's memory of a graph's structure. The relationship between graph games and other graph uses is not well understood at present and is itself an area that needs more research.

We have also produced some evidence that small amounts of motion can further improve the comprehensibility of graphs that have already been laid out with a spring-embedding layout algorithm. As far as we know, this is a new result but it needs to be confirmed with further experiments, including more formal empirical study. There are considerable differences by game, and there needs to be further analysis to see if the differences are caused by some problems in the game play, the type of players who chose particular games, or whether it relates to the different cognitive tasks required for the games, which makes movement more effective for some tasks than others.

In the near future, when our site is better established and has a stable number of players, we will be looking for collaborators with theories about graph visualization or ideas for graph visualizations that they want to test in this exploratory manner.

REFERENCES

1. John Bovey, Florence Benoy, and Peter Rodgers. Using Games to Investigate Movement for Graph Comprehension.. In Advanced Visual Interfaces: AVI 2004, pp. 71-79. ACM, May 2004
2. John Bovey. Using Games to Do Exploratory Experiments in Graph Comprehension. In Ninth International Conference on Information Visualisation (IV05), pp 335-338. IEEE, July 2005
3. Edward Castronov. On the Research Value of Large Games: Natural Experiments in Norrath and Camelot. Games and Culture, Vol. 1, No. 2, 163-186, 2006.
4. Cunningham, L.B. Baldick, R. Baughman, M.L. An empirical study of applied game theory: transmission constrained Cournot behaviour. IEEE Transactions on Power Systems, Volume: 17, Issue: 1 pp. 166-172. Feb 2002.
5. Peter Eades. A Heuristic for Graph Drawing. Congressus Numerantium 42, 1984.
6. The ESP game. http://www.espgame.org/
7. Hans Gremmen, Jan Potters. Assessing the Efficacy of Gaming in Economic Education. Journal of Economic Education, Vol. 28, No. 4 (Autumn, 1997) , pp. 291-303
8. Robert Kurzban and Daniel Houser. Experiments investigating cooperative types in humans: A complement to evolutionary theory and simulations. PNAS vol. 102 no. 5,. pp 1803-1807. 2005.
9. A. Lehman, A Solution to the Shannon Switching Game. Journal of the Society for Industrial and Applied Mathematics 12, 1964, pp 687-725.
10. Richard Lowry. Concepts and Applications of Inferential Statistics, Chapter 8. http://faculty.vassar.edu/lowry/webtext.html.
11. Leemkuil, H., de Jong, T., Ootes, S. (2000). Review of Educational Use of Games and Simulations. Knowledge Management Interactive Training System, University of Twente. KITS concortium.
12. H.C. Purchase. Which aesthetic has the greatest effect on human understanding?, Proc. of Graph Drawing Symposium, Di Battista, G. (ed), LNCS 1353, 1997.