Information Visualization Of An Agent-Based Financial System Model

Wei Jiang[ab*], Richard Webber[b] and Ric D. Herbert[a]

[a] School of Design, Communication and Information Technology, The University of Newcastle
University Drive, Callaghan, NSW, 2308, Australia

[b] School of Electrical Engineering and Computer Science, The University of Newcastle
University Drive Callaghan, NSW, 2308, Australia

[*] Corresponding author: jackie4my@hotmail.com

Abstract: This paper considers the application of information visualization techniques to an agent-based model of a financial system. The minority game is a simple agent-based model which can be used to simulate the events in a real-world financial market. To aid understanding of this model, we can apply information visualization techniques. Treemap and sunburst are two such information visualization techniques, which previous research tells us can effectively represent information similar to that generated by the minority game. Another information visualization technique, called logical fisheye-lens, can be used to augment treemap and sunburst, allowing users to magnify areas of interest in these visualizations.

In this paper, treemap and sunburst, both with and without fisheye-lens, are applied to the minority game, and their effectiveness is evaluated. This evaluation is carried out through an analysis of users performing various tasks on (simulated) financial market data using the visualization techniques. A subjective questionnaire is also used to measure the users' impressions of the visualization techniques.

Keywords: Agent model; Minority Game; Treemap; Sunburst; Fisheye-lens.

1. INTRODUCTION

The minority game is a simple agent-based model which can be used to simulate the events in a real-world financial market. Large amounts of data can be produced by the model. As an aid to understanding the model, we use information visualization techniques to display the output of the model. We use two space-filling techniques to visualize the output of the minority game. They are known as Treemap and Sunburst. Further, a focus + context technique known as fisheye-lens is added to the visualizations to help enhance the effects of the displays.

We undertake experiments with the user interfaces to examine how effective the various techniques are to aid users in understanding the output from the model. Our experiments show that the techniques are helpful for users in finding information from the model output and enhance the understanding of the model.

2. BACKGROUND

2.1. Minority Game

The minority game [1] is an abstract model based on the Al-Farol Bar model [2]. The minority game has an odd number of playing agents who, at each time step, choose between two alternatives, which may be represented as '0' or '1'. The agents in the minority side win that round and are rewarded by an increase in score.

The minority game may be used to represent the fundamental mechanism of a real financial system where agents compete. For example, the two alternatives can be used to simulate the real activities of 'buy' or 'sell' and the players may be agents in an equity market. Their score can represent their profits from trading.

Each agent has a time history of whether they were in the minority at each time step. This may be represented as a string of bits. To decide which alternative the agent to choose at a time step, each agent has a set of strategies. Each strategy represents the historic results for the memory of the agent and the corresponding decision for that history. Figure 1 is a sample of strategy for a memory length of 3 time steps.

There are three parameters in the minority game. They are N (the number of agents in the game), M (the length of each agent's memory), and S (the number of the strategies each agent has). Agents cannot learn the strategies from one another, i.e. they can make their decision only depending on the public historic results.

In the minority game used here we introduce an evolutionary mechanism. The agents with the lowest score after a certain number of rounds are removed out of the game and the same numbers of new agents replace the agents who 'died'. The newly-born agents copy the strategies of the best agents but their scores are set to zero. We use this to represent agents who leave the market and are replaced by new agents.

History	Decision
000	1
001	0
010	0
011	1
100	1
101	0
110	1
111	0

Figure 1: A decision strategy (*M*=3)

With the evolutionary mechanism, the agents can be represented in a tree-like structure. The old agents become the ancestor agents of the new agents where the time line defines the different levels. For example, in Figure 2, there are 3 agents in time t1. When evolution occurs, say agent A is the agent with the highest score. It duplicates itself as A* in the next time step t2. Agent B, who has the lowest score, is removed out of the game. To retain the same number of agents in the model, a new agent A1 is created as the child agent of the best agent A. Agent C is neither the best nor the worst one, so it only duplicate itself in time t2.

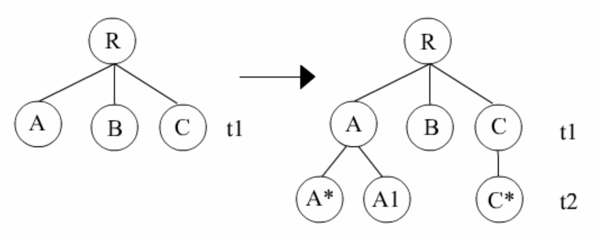


Figure 2: A sample evolutionary tree of minority game

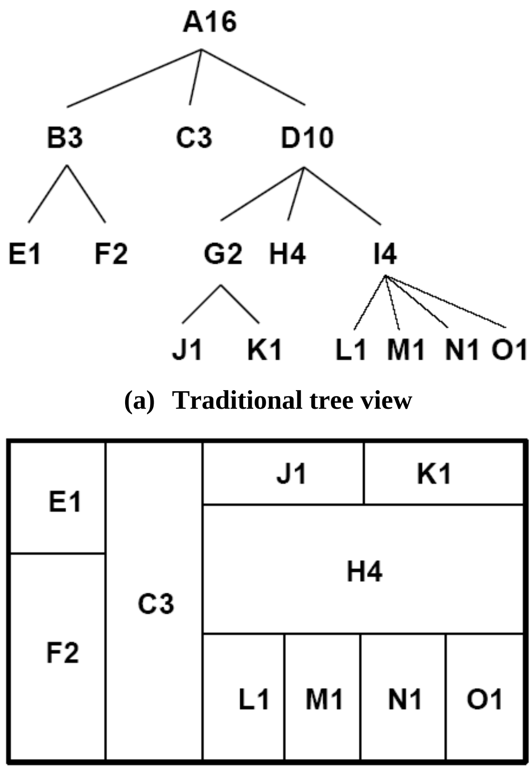
The tree structure can be used to store the information about the game. Each node represents an agent and stores data about the agent (such as their score and strategy). The edges represent the relationship between agents and the levels represent time. The root node is the time before the game starts and the leaf nodes are the final time-step.

2.2. Space-filling techniques

To aid the understanding of the minority game for users, we use information visualization techniques. Treemap and sunburst are two popular space-filling techniques what are often used into visualizing complex tree structures. They have been used, for example, to visualize computer file directory systems.

2.2.1 Treemap

Treemap [3] is a rectangular space-filling technique which separate rectangles into smaller rectangles repeatedly to represent a hierarchical structure. Each rectangle represents a node. Vertical lines and horizontal lines are used alternatively to separate the rectangles to small segments. Figure 3 shows a sample of traditional tree view and its equivalent treemap. In the Figure 3b, the biggest (whole) rectangle represents the root node in Figure 3a. Two vertical lines separate it into three parts which represent the three children node of the root node A16. Then the horizontal lines are used to separate the three children nodes into smaller parts. The procedure is repeated until all the nodes are visualized in the diagram.



(a) Traditional tree view

(b) Treemap display

Figure 3: Traditional tree view and treemap display [7]

When we use treemap to visualize the minority model, each rectangle represents one agent in the model. The sizes and colors of the rectangles can represent the scores and strategies of the agents. The size is based on the proportions of the score of one agent in its sibling nodes. For instance, one agent has a score of 5, while its sibling agent has the score as 10. If the parent agent has a display size of $15cm^2$, then the children will have sizes of respectively 5 cm^2 and $10cm^2$.

Color is used to represent strategy by mapping the similarity in strategy between two agents to a similarity in their color – the more similar the strategies, the closer the colors. This is implemented by using a variation on the 'spring' algorithm [4][5].

The normal treemap has a number of disadvantages. Firstly, the strict alternation between horizontal and vertical strips often leads to rectangles with a high aspect ratio (that is, long and thin or short and wide). This makes it difficult to select a single node in the treemap.

A solution called squarified treemap[6] can be used to solve this problem. It aims to make all the rectangles as close as possible to a square. As the example in Figure 4 shows, the algorithm splits the initial rectangle horizontally first, because the rectangle is wider than it is high. It puts the first agent on the left and sets its aspect ratio as 8/3 (step 1). Next, it puts the second agent above the first one to give a better aspect ratio of 3/2 (step 2). If the third agent is put above the first two agents, then the aspect ratio of 4/1 (step 3) is worse than if the third agent is put to the right of the first two agents, given an aspect ratio of 9/4 (step 4). At this point, the first two rectangles are fixed. The algorithm is used repeatedly to position all children within their parent's rectangle.

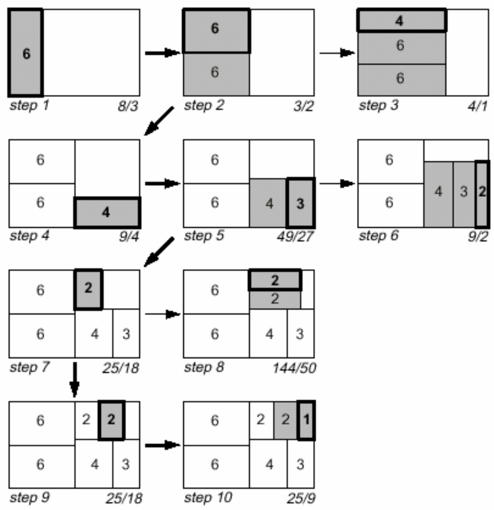


Figure 4: Subdivision algorithm [6]

The second disadvantage is that, since the children nodes supplies the space of their parent node, only leaf nodes are explicitly shown in the display. Users may have difficulty in distinguish which level the node is in.

One solution is the cushion treemap [7]. A cushion treemap draws a curved surface over each rectangle. The shape of this surface is determined by the number and depth of the nodes represented by the rectangle.

First, the algorithm defines a shape for each individual node using Equation 1 below [7].

$$\Delta z(x,y) = 4h(x-x_1)(x_2-x)/(x_2-x_1) \qquad (1)$$

x_1 and x_2 are the left and right boundaries of the rectangle, and h is a coefficient used to indicate the height of the curve, as shown in Figure 5. Similarly, Equation 2 is used to define a shape in terms of the top and bottom boundaries.

$$\Delta z(x,y) = 4h(y-y_1)(y_2-y)/(y_2-y_1) \qquad (2)$$

Equations 1 and 2 determine the colour (intensity) of each pixel z(x,y). When the algorithm draws a pixel on the graph, it combines the values for all the curves at that pixels location, as in Figure 5. Combining surfaces results in various shapes of each grid. The intensity of each cushion pixel can give us more detail about the level of each node than a simple treemap.

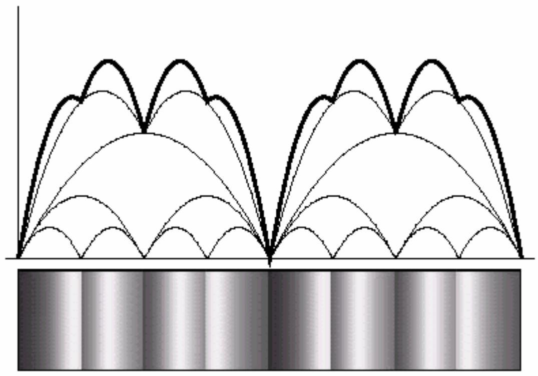


Figure 5: Binary subdivision of interval [7]

Figure 6 shows an example of a treemap that is both squarified and cushioned.

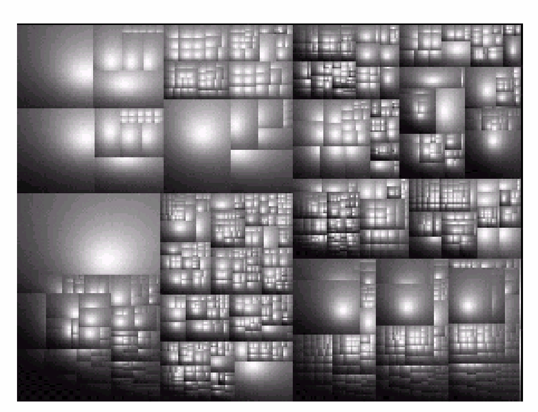


Figure 6: A squarified cushion treemap [6]

2.2.2 Sunburst

Sunburst [8] is a radial space-filling visualisation technique. It represents hierarchical data structures by separating space into a set of homocentric circles. The area between two adjacent circles represents one level in a tree structure. The circle in the centre represents the root node, with the hierarchy moving outward from the centre. Each circle can be separated into segments by radial lines. Each segment represents a node of the hierarchy. Children nodes are drawn within the angle occupied by their parent node. Figure 7 shows a sample diagram of sunburst.

As in a treemap, the size and color of each segment in sunburst also can be used to represent the score and the strategy of one agent in minority game. The circles can represent the levels in the tree data structure which represent time. The advantage of sunburst than treemap is that each node is explicitly shown in the display. Users can search and locate every node on the diagram. The disadvantage of sunburst is that the circles waste space since computer screens are not round. Also, the non-leaf nodes occupy more space than treemap.

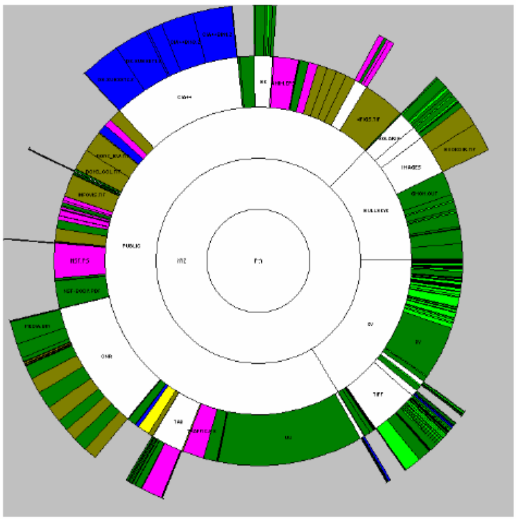


Figure 7: A sunburst [8]

2.3. Fisheye-lens Techniques

Fisheye-lens [9] is a focus + context technique which is used to enhance the effects of the display. There are two kinds of fisheye-lens: geometrical fisheye-lens and logical fisheye-lens. Both of the two kinds of fisheye-lens allow users locate and magnify the focus part without damaging the whole display environment. The difference is that in geometrical fisheye-lens, the magnified part relies on the coordinate within the visual substrate and logical fisheye-lens magnified the part based on the relation of the focus part within the whole structure.

A generic fisheye-lens can be applied to any image (see Figure 8). However, specialized fisheye-lens techniques can also be incorporated directly into treemaps and sunbursts.

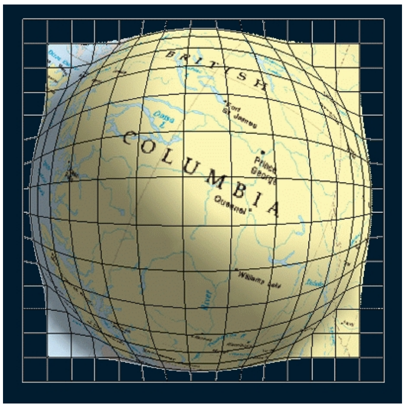


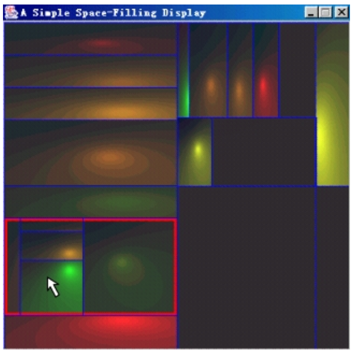
Figure 8: A generic fisheye-lens [10]

3. VISUALIZING THE MINORITY GAME

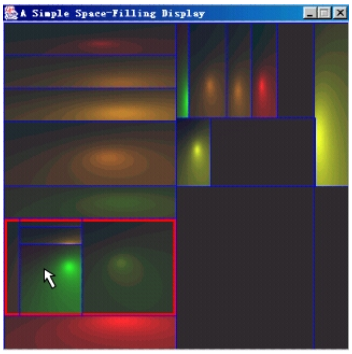
In this section, we build a simple minority model financial minority game model [11], and then visualize the model with treemap and sunburst user-interfaces respectively. Fisheye-lens technique is also used in both displays to enhance the effects.

The parameters are fixed in the model to make it simple. M is fixed as 3 and S is fixed as 1. That means in the model every agent can only remember the 3 previous rounds, and has only one strategy. The strategies are dynamic based on the results of each round. For example, the first corresponding history and decision in Figure 1 is ‘000’ and ‘1’. If the agent wins in the round when the previous three results are 000, the corresponding decision does not change, but if it loses, the decision will be changed from 1 to 0.

When representing the minority game with a treemap, each rectangle represents one agent. The size of a rectangle can represent the total score of an agent – the bigger the rectangle, the larger the score. Color can also be used to represent the strategy set of an agent.



(a) Before magnification

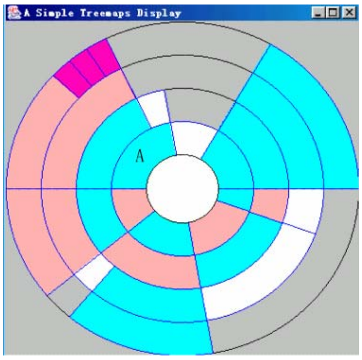


(b) After magnification

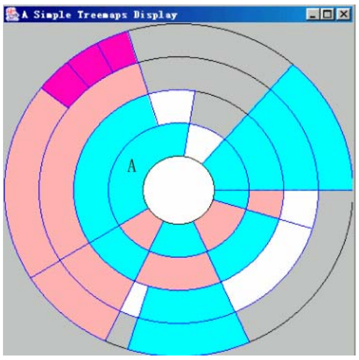
Figure 9: A sample of fisheye-lens use in treemap diagram

In the same manner, the sunburst can visualize the minority model. The segments represent the capital and colors of the segments represent the strategies.

But in both treemap and sunburst, tiny segments can be not easy to distinguish or located by the users. To solve this problem, the fisheye-lens technique is added into the visualizations. A logical fisheye-lens is used in treemap and sunburst respectively. Figure 9 is a sample of fisheye-lens use in treemap. In Figure 9b, users can use mouse to locate and magnify the focused rectangle in Figure 9a. Figure 10 is a sample of fisheye-lens use in sunburst. Similarly, a user has located the mouse cursor in one segment, which has then been magnified. In Figure 10b, the angle of node A is magnified and all its children nodes are magnified proportionally as well.



(a) Before magnification



(b) After magnification

Figure 10: A sample of fisheye-lens use in Sunburst diagram

4. EXPERIMENTS

4.1. Experimental Design

We designed user experiments to measure the performance the interfaces. Squarified cushion treemap and sunburst were used to represent a minority game model. The color represents the current strategy of each agent and the size of each segment in the visualization represents the capital.

Furthermore, fisheye-lens was also used to enhance the effects of the display.

We tested the hypothesis that (1) Treemap is better than sunburst for comparing the sizes of nodes; (2) Sunburst is better than treemap for distinguishing the levels of nodes; and (3) using fisheye-lens techniques will improve the user's ability to find information.

Two data sets were used in the experiments. Eleven agents as small data set and thirty-one as a large data set. Over time both data sets will rapidly produce large quantities of data. The purpose of the use of two data size is to well measure the performance of the technique in visualizing different environments.

Forty participants were invited to the experiment. All the participants were separated into 8 groups where 5 participants in one group. Each group did a different experimental session. These are treemap of small data set without fisheye, treemap of small data set with fisheye, treemap of large data set without fisheye, treemap of large data set with fisheye, sunburst of small data set without fisheye, sunburst of small data set with fisheye, sunburst of large data set without fisheye, sunburst of large data set with fisheye (Figure 11). Each participant was randomly chosen into different group. Each participant can only attend one session.

The participants were asked to finish 7 tasks first and then a subjective questionnaire.

Group	Display	Data	Fisheye	Numbers
1	Treemap	Small	No	5
2	Treemap	Small	Yes	5
3	Treemap	Large	No	5
4	Treemap	Large	Yes	5
5	Sunburst	Small	No	5
6	Sunburst	Small	Yes	5
7	Sunburst	Large	No	5
8	Sunburst	Large	Yes	5

Figure 11: Arrangement of the experiments

4.2. Experimental Procedure

Forty participants volunteered for these experiments. All participants had a similar knowledge of using a computer, but had no experience related to information visualization techniques. The contents of these experiments were new concepts to all participants.

All participants attended a short training session immediately before the task session. The training session included a description of the minority game and the space-filling techniques to be used in the session. The training session included the use of a demo interface program.

Participants where asked to use the interface undertake tasks such as 'which is the agent with the largest score' and 'which agent had a similar strategy to a particular agent'. Participants used a mouse and keyboard to input their answers in the interface which is supplied by the software. The accuracy of answers and time spent for each task were recorded by the system automatically.

The participants also completed a subjective questionnaire recording their opinions of the interface. The subjective questions were designed as Lykert style. Participants simply chose one option among 'strongly agree', 'agree', 'neutral', 'disagree' and 'strongly disagree'.

4.3. Conclusion And Discussion

The results of the experiment partly proved our hypothesis. Firstly, the hypothesis that treemap is better than sunburst for comparing the capitals of agents was validated. It is conjectured that this is due to it being easier to compare the sizes of axis-aligned rectangles than to compare radially-aligned circular segments. The results also show that it is easier to compare the strategies of companies in a treemap than a sunburst. This was unexpected. Perhaps users also find it easier to compare the colors of axis-aligned rectangles than radially-aligned circular segments. This result warrants further investigation.

The results showed that no factors affect the performance of distinguishing the levels of companies within the evolutionary hierarchy. This result is in conflict with that of Stasko et al. [12], who found that sunburst is better at distinguishing levels of a hierarchy than treemap. However, Stasko et al. used a "normal" treemap, whereas we used a squarified cushion treemap. Cushioning is designed to help users distinguish the levels of nodes – it appears that, in this case at least, cushioning is as effective as sunburst for this purpose.

The hypothesis that fisheye-lens techniques make it easier to complete tasks correctly was not statistically validated. In the experiments comparing capitals and strategies of companies, the accuracy of participants without fisheye-lens techniques was generally better than those using fisheye-lens techniques, but those using fisheye-lens tended to answer faster and had more confidence in their abilities to use the visualisation tool. All this supports a conjecture that the participants using fisheye-lens techniques in these experiments had false confidence in their abilities to use the technique.

The subjective questionnaire yielded several interesting results. In generally, participants preferred sunburst to treemap, even though their results in the objective tasks were not as good. This result concurs with those of Stasko et al. They found that users preferred sunburst over treemap, and found it is easier to learn.

Similarly, participants using the large data set were generally more positive than those using the small data set, and those using fisheye-lens were generally more positive than those not using fisheye-lens, both contrary to their results in the objective tasks.

5. CONCLUSION

In this paper we have used information visualization techniques as an aid to the understanding of the minority game as a model of a financial system. We found through experiments that using the techniques in a user interface to the model aided the users in understanding the model and were an aid in interpreting and finding information about the model's output. We find that further research on such interfaces could include more sophisticated treemaps and sunbursts as well as more powerful fish-eye techniques.

REFERENCES

[1] Challet, D & Zhang, YC, 'Emergence of Cooperation and Organization in an Evolutionary Game', *Physica A*, vol. 246, pp. 407–418, 1997

[2] Arthur, WB, 'Inductive Reasoning and Bounded Rationality', *American Economic Review*, vol. 84, no. 2, pp. 406–411, 1994

[3] Shneiderman, B, 'Tree Visualization with Tree-maps: A 2d Space Filling Approach', *ACM Trans. Graphics*, vol. 11, no. 1, pp. 92–99, 1990

[4] Eades, P, 'A Heuristic for Graph Drawing', *Congressus Numerantium*, vol. 42, pp. 149–160, 1984

[5] Kamada, T & Kawai, S, 'An Algorithm for Drawing General Undirected Graphs', *Information Processing Letters*, vol. 1, pp. 7–15, 1989

[6] Bruls, M, Huizing, K, & van Wijk, JJ, 'Squarified Treemaps', *Proc. Joint Eurographics and IEEE TCVG Symp. Visualization (TCVG 2000)*, pp. 33–42, 2000

[7] van Wijk, JJ & van de Wetering, H, 'Cushion Treemaps – Visualization of Hierarchical Information', *Proc. IEEE Symp. Information Visualization (InfoVis'99)*, ed. Wills, G & Keim, D, pp. 73–78, 1999

[8] Stasko, J & Zhang, E, 'Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualization', *Proc. IEEE Symp. Information Visualization,* (9-10 Oct, Salt Lake City), pp. 57–65, 2000

[9] Furnas, GW, 'Generalized Fisheye Views' *Proc. ACM Conf. Human Factors in Computing Systems (CHI '86)*, pp. 16-23, 1986

[10]M. S. T. Carpendale, "Elastic Presentation", Department of Computer Science, University of Calgary, Canada. http://pages.cpsc.ucalgary.ca/~sheelagh/personal/test/

[11] Zhang, YC, 'Modeling Market Mechanism with Evolutionary Games', *Europhysics News*, vol. 29, no. 2, pp. 51, 1998

[12] Stasko, J, Catrambone, R, Guzdial, M & McDonald, K, 'An Evaluation of Space-Filling Information Visualizations for Depicting Hierarchical Structures', Technical Report GIT-GVU-00-03, Graphics, Visualization, and Usability Center, Georgia Institute of Technology, Atlanta, 2000