Visualisation of the Minority Game using a mod

Stewart G. Heckenberg[a], Ric D. Herbert[b] and Richard Webber[a]

[a]Faculty of Engineering and Built Environment
[b]Faculty of Science and Information Technology
The University of Newcastle, Newcastle, NSW 2308, Australia

Abstract

This paper explores visualisation of a simplified model of a financial market, known as the Minority Game, using a computer game modification (mod) as a medium. The purpose of this particular work is to ascertain whether game engine capabilities have some benefit to visualisation of financial markets through investigation of the above application, and to discuss the possibilities for future research into visualisations of this kind.

Keywords: Visualisation, Minority Game, Mod.

1 Introduction

Visualisation is as much about user perception as it is about the fundamentals of charting or graph theory. People have a predisposition for recognising the basic properties and principles of the physical world, thus it would seem that representing data via physical analogies could only make visualisation more intuitive (Shneiderman 2003). Rather than have to explain that feature X of a visualisation has a certain meaning and why, the use of colour or size or movement or any other physical attribute can reduce the perceptual bandwidth for the user of that visualisation, because the user is dealing with concepts they already understand. Computer games offer an alternative platform for visualisation. The interaction and graphics capabilities provided by the *Unreal Engine* that enable gamers (users of computer games) to immerse themselves in a virtual world are the basis for our visualisation of the Minority Game.

The structure of this paper is as follows. Section 2 will describe what a mod is, and provide some background as to the application of computer games to scientific and information visualisation. Section 3 will explain what the Minority Game is and present some visualisations of it along with details of how the visualisations present various state information about the model. Section 4 will discuss the mapping of features of the Minority Game to the computer game Unreal Tournament 2003, provide rationale for the utilisation of a game engine as a visualisation medium, and detail our current state of development. The final section will outline our conclusions from this visualisation as well as suggest the possible direction of future research.

2 Mods

Mods, or computer game modifications, are well known in the PC gaming community. The current proliferation and popularity of mods has come about due to commercial game developers realising that game players will hack their favourite games to provide themselves with better gameplay experiences, and rather than prosecution, promotion is turning out to be a better alternative because user-created content is extending the longevity of many game titles (Kushner 2003).

This wave of creativity was originally inspired by titles such as *NetHack* and *Doom* (and its predecessor, *Castle Wolfenstein*, which dates as far back as 1982 on the Apple II). *Doom* came on the scene around the same time as the Internet began its surge of popularity — "From minor hacking and simple maps to 'total conversions' and even commercial releases, the PC modding scene has grown enormously since the early '90s." (Equip 2003). This widespread use of the Internet, allowed collaboration of geographically scattered mod development team members. Not only is collaboration via the net a big drawcard, but mod development "can be faster, more experimental and more creative than traditional game development" (Cleveland 2001) due to the fact that the development is starting at a higher level, utilising a pre-existing game engine.

Nowadays game engines, such as Epic's *Unreal Engine* (Epic 2003*a*, Epic 2003*b*) are incorporating the laws of physics and cutting-edge graphics to create virtual environments, making games that are more realistic , and which offer richer gameplay experiences (Hecker 1996). Putting the creative power of developers in the hands of users allows them to create their own worlds, and experiment with the fully interactive spatial dynamics provided with by these physics-based, 3D graphics engines. Besides games, these engines can be utilised as visualisation mediums. "Mods have graduated from Internet servers to being an integrated part of publishers' marketing strategies and shop shelves worldwide. But where do they go next?" (Edge 2003). Visualisation, that's where!

2.1 Visualisation using a Mod

Mods allow a paradigm shift in visualisation away from the more traditional direct representations of entities, relationships, and the attributes of each, and allow for a more artistic and creative visualisation process. The greatest advantage of leveraging mods for visualisation is the inherent interactivity of gaming. That being said, games are a medium akin to film and literature (Hecker 2002), but are only in the early stage of development as an art. Programming is necessary to know how to design good games because algorithms, especially problem-solving algorithms, are the key to good games — not the graphics (Hecker 2002). Moreover content is the important aspect of any good game, and communicating a message or set of information to users. The focus should be on the relaying of information to players to allow them to work out in-game problems. With mod development, graphics and other features have already been taken care of, and mod developers are free to concentrate on providing more intuitive user experiences and methods of communicating information to players (the "visualisation forest" if you will), instead of being bogged down in technical aspects of game development (the "trees").

Game developers are pushing the boundaries of computer graphics and are now more influential than the visualisation community with regards to advances and standards in graphics technology (Rhyne 2000). Games are already visualisations by virtue of the fact they are graphical representations of both abstract and concrete data. They attempt to communicate their message to users in a quick and concise fashion, especially because of the need to allow interaction in a timely manner — usually in real time. Games are more than just visualisations because the majority of titles are mass-marketed commercial endeavours and as such require a number of additional features to go with their "user-friendly" interfaces, such as sound and "playability" (interactivity), in order to sell.

Our left brain processes symbolic information and our right brain processes visual information. The important thing for any visualisation, including games, is to make sure the visualisation communicates the message you want it to (Blinn 1991). A game such as *Unreal Tournament 2003* could be said to affect both sides of our brain, in that it has a symbolic heads-up display, and also a visual representation of a virtual world. This virtual world provides a familiar interface with which users can interact. The Gestalt principle of 'Law of Familiarity' — "things are more likely to form groups if the groups appear familiar or meaningful" (Nesbitt & Friedrich 2002) — can be carried out with an *Unreal Tournament 2003* mod by using avatars to represent agents in a simulation (of the Minority Game, for example) moving about in a space resembling buildings and nature — familiar concepts of the everyday world.

The impetus for implementing a visualisation with a mod is that it allows exploration of visualisation as a "conceptual framework...based on human perception", with particular attention paid to semiotics (the study of how symbols convey meaning) and J.J. Gibson's 'Affordance Theory', where action is suggested via direct perception as opposed to learning of symbolism (Ware 2000). Traditional design-based visualisations (such as the Spring Graph in section 2.1.1) rely on defining entities, relationships and attributes which are arbitrary symbols (those which require learning). Besides these, visualisation via a 3D virtual environment can make use of sensory symbols (those that do not require learning), by providing an immersive experience that takes advantage of Affordances, e.g. "perceivable possibilities for action", "physical properties of the environment that we directly perceive", and "[perception of] surfaces for walking, handles for pulling, space for navigating, tools for manipulating, and so on" (Ware 2000). Thus a mod can provide a perception-based approach. Using *Unreal Tournament 2003* makes it relatively simple to create new objects that can interact with the player and other game objects — an important feature for any game (Devine 2002), and any visualisation for that matter, and therefore makes this visualisation easier to implement than starting from scratch using more primitive (and again, traditional) graphical components and tools. Interactivity is a key component of many systems that are subject to visualisation, with one such type of system being financial market simulations. We have chosen to visualise a simple financial market simulation known as the Minority Game.

3 The Minority Game

The Minority Game (Challet & Zhang 1997, Johnson, Jarvis, Jonson, Cheung, Kwong & Hui 1998, Manuca, Li, Riolo & Savit 2000) is a simplified model of the adaptive inductive-reasoning system in the El Farol Bar problem (Arthur 1994) that simulates agent behaviour in a financial market. It is a game with an odd number of players consisting of several rounds where each round players choose one of two possible options. The two options could be zero or one, A or B, buy or sell, or any two things indicative of a competitive situation.

Each round of the game results in a majority of players choosing one of the two possible options. Those players who did not choose the option which the majority of players chose are therefore in the minority and are deemed the winners of that particular round. Note there is no minority if all players choose the same option. Each player has a score, and their score increases when they win a round by being in the minority, hence the name the Minority Game.

The number of winning players each round can range from 0 to $\frac{(n-1)}{2}$ for a game involving n (where n is an odd number) players. Each of the players has the ability to remember which of the two available options was the winning option (i.e. the option chosen by the minority of players) for each of the previous m number of rounds. This m represents the memory size of each player.

In addition to each player being able to remember the previous m winning options, they also have a set of s strategies, each of which determines what option they should choose for the current round based on the history of the previous m rounds. Given that each player must choose from 2 possible options each round, for the previous m rounds there are thus 2^m possible winning histories, and therefore each strategy has a size of 2^m. For example, given 0 and 1 as the two possible options a player can choose for any given turn, and assuming $m = 3$, there are 8 possible histories: 000 through 111 in binary notation. The *actual* winning history is known as the signal. The length of the signal can be greater than m, however only the m previous results are relevant to the players. There is a variation on the Minority Game where each player can have a different memory size (Zhang 1998), however in this paper we shall be concentrating on players who differ only in their strategies.

Like players, strategies themselves also accumulate points; if any of a player's strategies would enable the player to win the current round, the scores of these "winning" strategies increase. If one of these successful strategies is the player's current strategy — the current strategy being that with the highest score among the player's set of strategies — the score of the player increases also. Sometimes several strategies have equal best scores. Ties are broken by randomly selecting one of these to be the current strategy.

Strategies are central to the behavioural dynamic exhibited by the Minority Game. The group behaviour of the system due to the use of strategies and the scoring system is of interest to mathematicians, physicists and psychologists alike, and there have been several papers (Challet & Zhang 1998, Savit, Manuca & Riolo 1997) explaining the mathematical whys and wherefores of the Minority Game. Topics such as Nash equilibria (Challet, Marsili &

Zecchina 2000), which is beyond the scope of this paper, are also related to the Minority Game in that available information is minimized by the agents — they keep their strategies secret and aren't aware of what choices other agents are going to make, nor what individual agents chose in previous rounds, only the history of what the overall winning choice was.

The Minority Game can be seen to be a fairly simplified version of a financial market. Instead of stockbrokers or investors we have players and instead of choosing buy or sell each turn, the players choose 0 or 1, or choose to be on the red team or the blue team. Being in the minority when buying or selling is a good thing because the majority of agents involved in the system are doing the opposite thing, and thus it is a model of supply and demand, where the minority wins (this is a grossly simplified model, and we acknowledge that actual financial markets are far more complex). Visualising this model, and creating "fun" visualisations by means of providing interactivity, is what our research is about.

3.1 Visualisations of the Minority Game

Visualisations of the Minority Game in the papers mentioned in the previous section were various plots showing which of the two possible options agents chose over time, detailing fluctuations in minority population when different values were used for the memory size and other variables. Here we look at a Damien Challet's applet visualisation of the Minority Game, the online "Interactive Minority Game", and our own visualisation that uses a Spring Graph to represent agent properties and performance.

3.1.1 Challet's Applet Visualisation

The *Minority Game web page* (Challet 2003) has two applets that visualise aspects of the Minority Game. According to the web page: "[the applets are] aimed at giving a good perception of what dynamically happens in [the Minority Game]. Dynamics of macroscopic quantities, namely the attendance, the fluctuations and the available information Dynamics of microscopic quantities, i.e. the dynamical use of strategies [are shown]".

The first applet (see Figure 1) shows the dynamics of the macroscopic quantities, such as the attendance; how many agents choose each of the two possible options at a given time, the fluctuation of this attendance, and a measure of the available information. The information is displayed via three plots, one for each quantity over time (the horizontal axis). The first is a plot of $A(t)$, "the difference between the number of people taking one action and those choosing the other one". The next plot is of H/N, "the available information, and is minimized by the agents...When $2^M/N < 0.3374...$ and S=2, you will see that H=0". The third plot is σ^2/N, where σ^2 measures the fluctuations of $[A(t)]$ and also represents how well the agents behave. If every agent behaves randomly, $\sigma^2/N = 1$, thus whenever $\sigma^2/N < 1$, the agents are reputed to cooperate. As it appears, the agents do not minimize this quantity". There are drop-down boxes for uses to choose N (number of agents), M (agent memory size) and S (number of agent strategies), and a slider to control the Speed (time taken for each round or step) of the simulation. There is also a Pause and a Reset button.

The second applet (see Figure 2) shows the dynamics of the microscopic quantities, namely what players chose which of the two possible choices (A or B) at any given moment along with the total for each choice. The information is displayed via a grid where

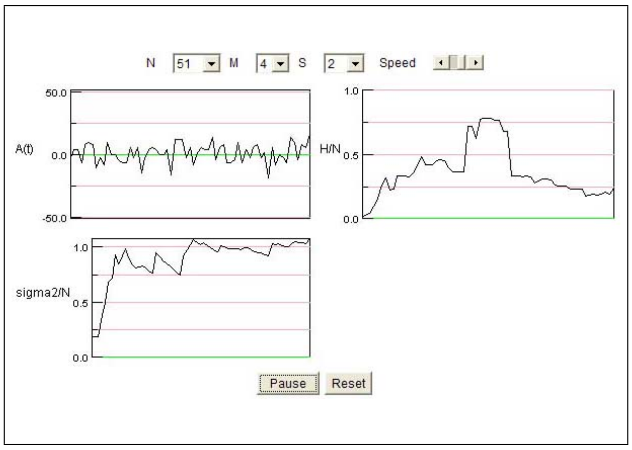
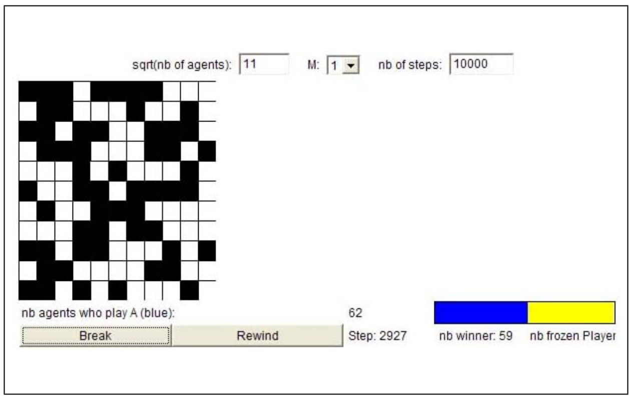

N 51 M 4 S 2

Figure 1: Macroscopic quantities of the Minority Game (Challet 2003)

each square represents one of the $N = l * l$ agents. "Each agent has two strategies, one black and one white. During the game you can observe which strategy everyone is using. If the color of one agent turns into red, she is frozen, i.e. she uses only one of her strategies". There is a sliding bar representing the ratio of each of the two possible choices; "the number of agents who choose A (blue) versus the number of agents who choose B (yellow)". There is a box for users to enter l, the square-root of the number of agents (i.e. entering 11 creates an 11 by 11 square grid for 121 players). There is a drop-down box where users can choose M (agent memory size), and a box where users may enter the number of steps (or rounds) over which the simulation should run. There is also a Play/Break button to Start/Pause the simulation and a Rewind button to reset the simulation when new values are chosen for the variables.


sqrt(nb of agents): 11 M: 1 nb of steps: 10000

nb agents who play A (blue): 62

Step: 2927 nb winner: 59 nb frozen Player

Figure 2: Microscopic quantities of the Minority Game (Challet 2003)

Both the applets above use traditional visualisation tools, namely time-based plots, grids and bars, and this is generally how much of the information to do with simulations of adaptive inductive-reasoning systems, such as financial markets, is displayed in the media and visualisation software.

In this visualisation, information about the Minority Game is displayed concisely, and there are no extraneous ambient visual artifacts. There is also limited interactivity between the Minority Game agents and the user — the user controls simulation variables, but is separate from the simulation underlying the visualisation, and does not participate in the Minority Game.

3.1.2 Our Spring Graph Visualisation

A graph can be used to represent players or agents in the Minority Game and their relationships to one another. A spring graph involves laying out a graph using springs, whereby pairs of adjacent vertices (those connected by an edge) attract each other while pairs of non-adjacent vertices (those not connected by an edge) repel each other (Eades 1984). Here is a point-form summary of the features of, and rules governing, our implementation of the Spring Graph layout to model the Minority Game:

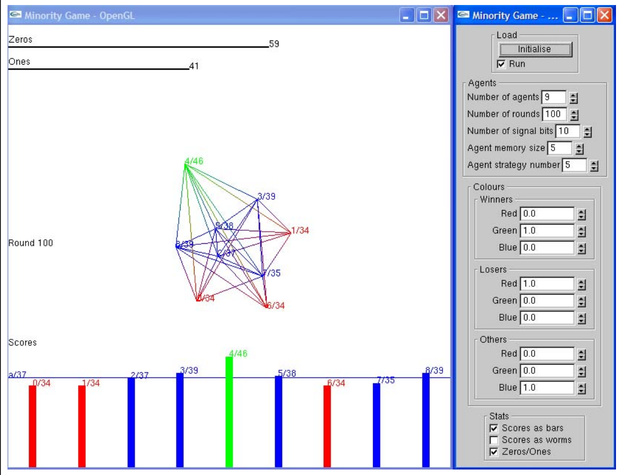
- each vertex in the graph represents a player
- player vertices with the highest score are coloured green
- player vertices with the lowest score are coloured red
- player vertices with neither the highest nor lowest score are coloured blue
- edges between player vertices represent the relationship between player scores
- the existence of an edge uv means player v has a greater score than player u
- edges exist between all players except between players with equal scores
- each end of an edge is coloured the same as the player at that end of the edge
- players u and v attract one another with force inversely proportional to their score difference
- players with equal score repel one another with a finite constant force
- forces acting on any particular player vertex are cumulative
- edge length, player position, and the forces acting on players are co-dependent
- each player vertex is placed randomly prior to the beginning of the game
- player movement is determined by their initial position and game performance

Along with the spring graph, information about the number of players who chose which option, along with the score of each player, is shown via bars and/or worms. The number of rounds played so far is also shown. The screenshot (see Figure 3) is of a sample run of the Minority Game with 9 agents, each with 5 memory bits and 5 strategies, over 100 rounds. The GUI shows the quantities for each of these variable.

This visualisation uses concepts of the physical world, such as springs, to represent data, however, like Challet's applets, user interaction with the simulation agents is limited.

3.1.3 The Interactive Minority Game

The *Minority Game web page* has a link to an external site where you can "Play against inductive agents in the Interactive Minority Game", however at the time of writing, this site was unavailable. The authors of the Interactive Minority Game were kind enough to direct us to a draft of their upcoming paper (Laureti, Ruch, Wakeling & Zhang 2004) which sheds some light on the visualisation methods involved. The user is given the choice of viewing the history via either a "price-chart viewpoint", which is a time-based



Load
[✓] Run

Agents
Number of agents 9
Number of rounds 100
Number of signal bits 10
Agent memory size 5
Agent strategy number 5

Colours
Winners
Red 0.0
Green 1.0
Blue 0.0
Losers
Red 1.0
Green 0.0
Blue 0.0
Others
Red 0.0
Green 0.0
Blue 1.0

Stats
[✓] Scores as bars
[ ] Scores as worms
[✓] Zeros/Ones

Figure 3: Visualisation of the Minority Game using a Spring Graph

plot of the market "price", or a "binary viewpoint", which is simply a binary string showing whether the minority of agents bought or sold. The user must then choose whether to buy or sell during the next turn, and is an agent of the game themselves. This is somewhat how our follow-up visualisation to the Spring Graph works, however instead of a web-based interface, we've chosen to use a mod of the computer game *Unreal Tournament 2003*.

As alluded to in the Introduction, we want to combine both physical analogies and interaction to create visualisations, and this can be done using mods as an alternative visualisation medium.

4 Visualisation with Unreal Tournament 2003

Unreal Tournament 2003 is a popular computer game by Epic (Epic 2003*b*). The game's genre is First Person Shooter, as the basic object of the game is to shoot things (usually humanoids) while viewing the world from a first-person perspective, as seen through the eyes of your chosen avatar. Like the "Interactive Minority Game" mentioned above, *Unreal Tournament 2003* allows human players to compete against computer AI. Also, there is the ability to play versus other human players via the Internet by connecting to servers running the game. These servers can run without human players being connected, and as such are populated with computer AI players, or "bots". This ability for the game to go on as different players connect and disconnect suits a simulation of a financial market in which companies and investors come and go over time — one of the reasons that inspired us to implement a visualisation on this alternative platform.

Unreal Tournament 2003 is very mod-friendly. Epic provide the source for the game, written in UnrealScript, which is much like Java and C++, and this in turn runs on the underlying engine or virtual machine (written in C/C++) (Epic 2003*a*). The language is very well suited to programming game logic; "State machines...are particularly important in game programming" (Hecker & Simpson 2002). "Before UnrealScript, states have not been supported at the language level, requiring developers to create C/C++ "switch" statements based on the object's state. Such code was difficult to write and update. UnrealScript supports states at the language level." (Epic 2003*c*). This, and the fact that mod development is completely object-oriented with *Unreal Tournament 2003* meant that porting the original Minority Game implementation (see our Spring Graph visualisation above) from C++ to UnrealScript was straightforward.

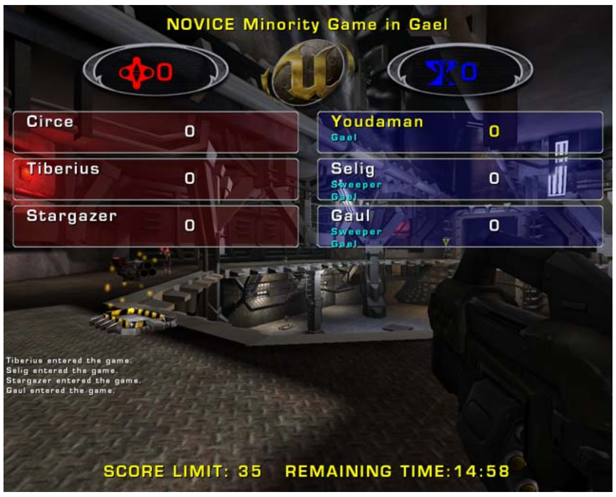


Figure 4: Minority Game mod screenshot showing the standard *Unreal Tournament 2003* team scoreboard GUI overlay. Teams start with an equal number of players as the scoreboard indicates, and neither team is the minority.

Below is a point-form outline of how the Minority Game translates into an *Unreal Tournament 2003* mod at the time of writing:

- *Unreal Tournament 2003* is a First Person Shooter (FPS) and thus agents are modelled as gun-toting avatars. This aspect has nothing to do with the Minority Game itself as such, but it provides an interactive means of agent interaction that's familiar to anyone who plays computer games, and has room for ideas, such as players shooting coins, etc., as well as attracting users to participate in a financial market simulation unknowingly — they just think they're having fun playing a game!
- Instead of agents choosing 0 or 1, they choose red or blue, as there are two teams, red and blue. Computer controlled agents (known in game parlance as Bots) make their choice according to the original rules of the Minority Game and their strategies (as outlined in section 3). Human controlled agents (simply known as Players) are shown a GUI with two buttons labelled Red and Blue and can choose their team, thus defining their own strategy — a new addition to our Minority Game due to the interactive nature of the mod.
- The two teams start the game with an even number of agents (see Figure 4), and as soon as the teams become unbalanced (due to agents choosing a team, see Figure 6) the 'fun' begins. Those agents on the smaller team (the minority) are given some advantage over the other team. For example, they may be given a powerup of some kind such as increased firepower or more health/shields.
- Analogies are drawn from financial markets could be that the concept of frag count (how many kills an agent has) is a measure of 'market share', the type of powerup the minority team receive is their 'market leverage', and so on.

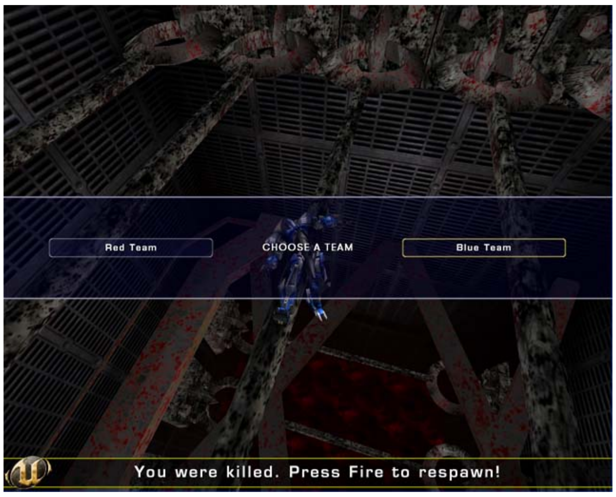


Figure 5: Minority Game mod screenshot showing our custom "Choose a team" GUI overlay that allows human players to change their team once their avatar has been killed.

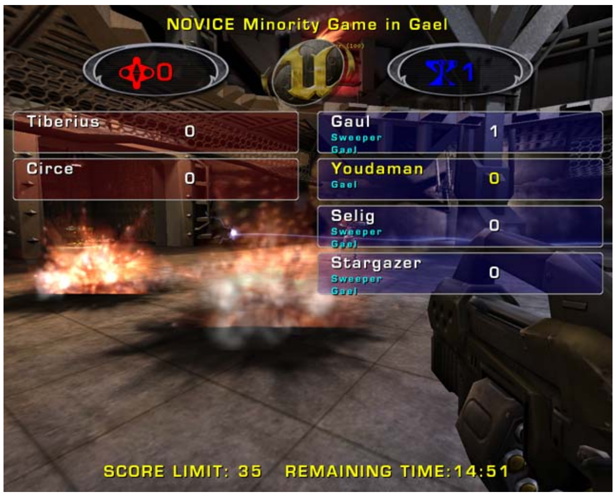


Figure 6: Minority Game mod screenshot again showing the standard *Unreal Tournament 2003* team scoreboard GUI overlay. Note that the computer-controlled player "Stargazer" has swapped teams, thus making their original team the minority.

At this stage, the actual Minority Game logic is only one part of the mod implementation — determining which of the two teams an agent is on. The extension of this into real-time advantages/disadvantages in a 3D environment is, the current motivation, and offers the opportunity to investigate the usefulness of the mod as a tool to educate users about stock market behaviour.

This is where development is up to at the time of writing. The initial hurdle of understanding how to create mods for *Unreal Tournament 2003* has been passed, and what now remains is to improve mapping of the features of the Minority Game model to the mod. This could involve things such as new textures, player models, and maps. Perhaps some augmentation of the AI could be involved, to make agents swarm or cluster together (using *Unreal Tournament 2003* Squads, for example) to represent agents who work together to gain advantage in a market.

Currently, the only feature directly applied to the mod from the Minority Game model is the ability for agents to choose teams based on their private set of strategies. We have also enabled human interaction by providing a custom "Choose a team" GUI overlay (see Figure 5) which appears when human-controlled characters are killed. In addition to the points above, possibilities for the mod include:

- instead of players shooting guns, players could throw stocks or money at one another, so instead of killing your trading with or investing in other agents/companies.
- instead of players having an inventory of weapons, they have a stock portfolio
- the multi-player/network aspect of the game allows for distribution of the mod over the Internet and use by geographically separate users
- the fact that the mod can run continuously on a server with or without human interaction means that the ongoing behaviour of the simulation can be studied
- Minority Games within Minority Games could be implemented, resembling local markets that work within a global market
- player skins — instead of looking like sci-fi characters, perhaps players could look more business-like, with bowler hats and pinstripe suits.
- new maps/levels/arenas — perhaps recreating an office building or even a stock market floor as the playing environment.

These are a few of the ideas we've come up with in our discussions. The point to any of the above features, or others, being implemented is to try and make the visualisation fun to play and immerse players in the Minority Game. We can't say whether our visualisation is "better" or "worse" than others — our research is more about introducing the use of mods as an alternative visualisation platform, and exploring their effectiveness.

5 Conclusions and Future Work

Initially our research was focussed on traditional means of visualisation, augmented with things like springs, however our focus changed to the use of alternative platforms, such as game engines, in the hope that they will help provide more intuitive and familiar interfaces, and also be more technically superior with regards to graphics, physics, sound, and so on. Harnessing these game engine capabilities will allow us to focus on content that makes the concepts of the Minority Game, and financial markets in general, intuitive and more easily perceived (directly, as opposed to needing an explanation) by users through interactivity.

Our future research will concentrate on computer game modification to visualise financial market data. We hope to augment our mod visualisation using better physical analogies (such as coins instead of bullets, for example) that are more familiar and pertinent to the financial market modelled by the Minority Game, and that are more expressive than simple lines and dots. We believe the field of computer gaming has a lot to offer with regards to visualisation methods of this kind — we have shown that using a computer game engine as a medium for visualisations is possible, however it raises the question: How can the capabilities of computer game engines be used to advantage and what gains may be achieved for visualisation? This is an open question we intend to address in future research.

References

Arthur, W. B. (1994), 'Inductive Reasoning and Bounded Rationality (The El Farol Problem)', *American Economic Review (Papers and Proceedings)* **84**, 406–411.

Blinn, J. F. (1991), 'Visualization'. Retrieved November 28, 2003, from http://www.research.microsoft.com/~blinn/VISUAL.HTM.

Challet, D. (2003), 'The Minority Game's Web Page'. Retrieved November 28, 2003, from http://www.unifr.ch/econophysics/minority/.

Challet, D., Marsili, M. & Zecchina, R. (2000), 'Statistical Mechanics of Systems with Heterogeneous Agents: Minority Games', *Physical Review Letters* **84**, 1824–1827.

Challet, D. & Zhang, Y.-C. (1997), *Physica A: Statistical and Theoretical Physics* **246**(3-4), 407–418.

Challet, D. & Zhang, Y.-C. (1998), *Physica A: Statistical and Theoretical Physics* **256**(3-4), 514–532.

Cleveland, C. (2001), 'The Past, Present, and Future of PC Mod Development', *Game Developer* **8**(2), 46.

Devine, G. J. (2002), 'Game Design Lessons from Real Life: Game Object Interactions', *Game Developer* **9**(9), 36–40.

Eades, P. (1984), 'A Heuristic for Graph Drawing', *Congressus Numerantium* **42**, 149–160.

Edge (2003), 'The Modern Age', *Edge* **126**(The Mod Scene: What Happens When Gamers Build Games?), 58–67.

Epic (2003*a*), 'Unreal Developer Network Website'. Retrieved November 28, 2003, from http://udn.epicgames.com/.

Epic (2003*b*), 'Unreal Tournament 2003 Website'. Retrieved November 28, 2003, from http://www.unrealtournament2003.com/.

Epic (2003*c*), 'UnrealScript Language Reference'. Retrieved November 28, 2003, from http://udn.epicgames.com/pub/Technical/UnrealScriptReference/.

Equip (2003), 'Building Blocks', *Edge presents Equip* **8**(The Insider's Guide to the Future of PC), 64–75.

Hecker, C. (1996), 'Behind the Screen: Physics, The Next Frontier', *Game Developer* .

Hecker, C. (2002), 'Art, Game Design, Programming, and Technology', *Game Developer* **9**(5), 56–55.

Hecker, C. & Simpson, Z. B. (2002), 'State Machine A.K.A. (Non) Deterministic Finite State Machine, Finite State Automata, Flow Chart', *Game Developer* **8**(1), 8.

Johnson, N., Jarvis, S., Jonson, R., Cheung, P., Kwong, Y. & Hui, P. (1998), *Physica A: Statistical Mechanics and its Applications* **258**(1-2), 230–236.

Kushner, D. (2003), 'It's a Mod, Mod World', *IEEE Spectrum* **40**(2), 56–57.

Laureti, P., Ruch, P., Wakeling, J. & Zhang, Y.-C. (2004), *Physica A: Statistical Mechanics and its Applications* **331**(3-4), 651–659.

Manuca, R., Li, Y., Riolo, R. & Savit, R. (2000), *Physica A: Statistical Mechanics and its Applications* **282**(3-4), 559–608.

Nesbitt, K. V. & Friedrich, C. (2002), 'Applying Gestalt Principles to Animated Visualizations of Network Data', p. 737.

Rhyne, T.-M. (2000), 'Entertainment Computing: Computer Games' Influence on Scientific and Information Visualization', *Computer* **33**(12), 154–156.

Savit, R., Manuca, R. & Riolo, R. (1997), 'Adaptive Competition, Market Efficiency, Phase Transitions and Spin-Glasses'. Retrieved November 28, 2003, from http://arxiv.org/abs/adap-org/9712006.

Shneiderman, B. (2003), 'Why Not Make Interfaces Better than 3D Reality?', *IEEE Computer Graphics and Applications* **23**(6), 12–15.

Ware, C. (2000), Foundation for a Science of Data Visualization, *in* D. D. Cerra, ed., 'Information Visualization : Perception for Design', Morgan Kaufmann Publishers, San Francisco, chapter 1, pp. 1–33.

Zhang, Y.-C. (1998), 'Modeling Market Mechanism with Evolutionary Games', *Europhysics News* .