FACILITATING LEARNING IN A REAL TIME STRATEGY COMPUTER GAME

Penelope Sweetser[1] and Simon Dennis[2]
[1]*School of Information Technology and Electrical Engineering; penny@itee.uq.edu.au*
[2]*Key Centre for Human Factors and Applied Cognitive Psychology, University of Queensland*

Abstract: The aim of this project was to implement a just-in-time hints help system into a real time strategy (RTS) computer game that would deliver information to the user at the time that it would be of the most benefit. The goal of this help system is to improve the user's learning in terms of their rate of learning, retention and avoidance of stagnation. The first stage of this project was implementing a computer game to incorporate four different types of skill that the user must acquire, namely motor, perceptual, declarative knowledge and strategic. Subsequently, the just-in-time hints help system was incorporated into the game to assess the user's knowledge and deliver hints accordingly. The final stage of the project was to test the effectiveness of this help system by conducting two phases of testing. The goal of this testing was to demonstrate an increase in the user's assessment of the helpfulness of the system from phase one to phase two. The results of this testing showed that there was no significant difference in the user's responses in the two phases. However, when the results were analysed with respect to several categories of hints that were identified, it became apparent that patterns in the data were beginning to emerge. The conclusions of the project were that further testing with a larger sample size would be required to provide more reliable results and that factors such as the user's skill level and different types of goals should be taken into account.

Key words: computer games, learning, help system

1. Introduction

Current computer games would benefit from the addition of a help system that provides information to the user at the time that it is most required. Many computer games currently utilise help systems with tutorials

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35660-0_65

R. Nakatsu et al. (eds.), *Entertainment Computing*

or require the user to ask for information. However, tutorials, as used in games such as Lionhead's 'Black & White' and Sierra's 'Zeus', can be rigid and monotonous when the player's main focus is to start playing the game. Furthermore, these tutorials can only take the learning of the user to a certain point and usually only focus on interaction with the game interface. Whereas, skills that are beyond this basic interaction, namely perceptual, declarative knowledge and strategic skills, are often ignored by these tutorials and left to the user to resolve. These shortcomings can cause stagnation in the player's learning and often many elements of the game will go unnoticed.

In contrast, the built-in request-and-retrieve information help systems, as used by games such as Activision's 'Call to Power II' and EA's 'The Sims', require the player to stop playing the game and search for answers in the system. This can seriously detract from the enjoyment of playing the game and results in a very slow learning curve for the user. Furthermore, in order to benefit from this type of help system the player needs first to know that there is information in the system that will help them to achieve their goal.

Currently, the alternative to these methods is that the user fully reads and understands the game manual before commencing play. For most people, on purchasing a new game, this is the last thing that they want to do. Additionally, a common problem that arises in each of the aforementioned help system techniques is that the player is unlikely to retain information that is read from a manual, given in a tutorial or retrieved from a help system. This is due to the fact that in each of these methods a large amount of information is presented to the user in a short period of time and not necessarily put into practice immediately and hence this information is not reinforced in the mind of the user.

On the whole, there are three main problems that can be identified in the current types of help systems that exist in computer games. Specifically, these problems are that the current help systems (i) induce a slow rate of learning in the user, (ii) result in lack of retention of information and (iii) cause stagnation in learning for the user. However, it is proposed in this project that a tool to overcome these problems can be developed by introducing a just-in-time hints help system that monitors the user and provides information at the time that it is most useful.

Previous research which has contributed a great deal to the study of learning in real time computer games is the Learning Strategies Project (LSP), which examined the effects of using different learning strategies to improve learning in a real time computer game (Donchin, 1989; Mane & Donchin, 1989). In particular, one study in the LSP demonstrated a part-task learning strategy that effectively avoids stagnation and increases the user's rate of learning of the game (Frederiksen & White, 1989). In addition to the

LSP, the broader field of training was also relevant to this project. In particular, the literature on increasing retention (Atkinson, 1972; Druckman & Bjork, 1991), avoiding stagnation (Frederiksen & White, 1989) and increasing rate of learning (Patrick, 1992; Druckman & Bjork, 1991) was reviewed. In particular, Atkinson's study of different strategies for learning German vocabulary demonstrated that a response-sensitive learning outperforms other learning strategies. It is projected, that in a similar way, the use of a just-in-time hints help system in this project, as a continuation of the response-sensitive learning strategy, will prove more successful in facilitating learning than the help system techniques that are currently employed in computer games.

2. The Computer Game as a Research Tool

According to Frederiksen and White (1989) there are four different types of skill that a user must acquire in order to master successfully a computer game. These are motor, perceptual, declarative knowledge and strategic skill. In order to acquire motor skill, the user must learn to interact with the game via the keyboard, mouse and interface. In mastering perceptual skill, the user must gain an understanding of the objects they see and the sounds that they hear. As the visual and auditory environment of a computer game is usually very stimulating, it can provide many clues as to what is happening in the game and a great amount of intrinsic feedback. The declarative knowledge type of skill means that the user's knowledge of the game includes a set of factual statements that describe relations and states that exist in the game. Finally, in order to master strategic skill the user must learn to develop and understand strategies that can be employed to win the game.

Consequently, a real time strategy (RTS) game called "Tubby Terror" was developed for this project. This game was designed to incorporate features in each skill type, so that the game would be sufficiently complex to warrant and benefit from the application of the just-in-time hints help system.

3. The Help System

The goal of the just-in-time hints help system is to give the user the information that they require to play the game successfully at the time that it will be of most benefit. The design, implementation and testing of this help system comprises the most important part of this project. The just-in-time hints help system works by monitoring the user's actions during gameplay and making an estimate as to which hint should be given to the user next. The help system can be divided into two major components, the knowledge inference engine and the scheduler, which will be described in this section.

3.1 Knowledge Inference Engine

The purpose of the knowledge inference engine is to estimate the probability that the user already possesses the knowledge that is contained in each particular hint. This is done by feeding a weighted system state, which initially consists of estimated values, into a regression equation. This equation then returns a value between zero and one, which indicates the system's level of confidence in whether the user does already possess this knowledge. Through the experiments that will be completed in this project, the values of the weights will ideally be refined to give a more accurate estimate of the user's knowledge.

3.2 Scheduler

The purpose of the scheduler is to create dynamically a schedule for the delivery of the hints. The priority that is assigned to a hint is based on the minimum time in the game to give the hint, the time since the last hint was given, the output of the knowledge inference engine and whether the hint has already been given. The priority of each hint is recalculated every second and the hint that has the highest (non-zero) priority after the time to wait between hints has elapsed will be delivered to the user.

4. Pilot Testing

Two pilot testing sessions were carried out to provide initial feedback on the game and the help system. This included assessing the playability of the game, revealing any errors and providing useful information about the interaction of the game and the help system. The results from these sessions were used to update the game to remove errors that occurred during play and to make changes that would facilitate testing.

5. Method

The experiment consisted of two phases of testing. In each phase there were six participants from a pool of first year psychology students. Each participant was given an information sheet prior to commencing the experiment and a debrief sheet after finishing the experiment. After reading the information sheet, the participant was presented with the game instructions and given as much time as was required to read the instructions and the opportunity to ask any questions. However, no more information was given in answering the participant's questions than was present on the instruction sheet.

The participant was then asked to play Tubby Terror for a period of fifteen minutes. During this time they were presented with hints aurally. Each time the participant was presented with a hint they were required to respond verbally with a "yes" or a "no" to indicate whether the hint was helpful, where the term 'helpful' meant that they did not already know the information contained in the hint and it helped them to play the game. The participant's responses were recorded. Also, as the game progressed, the system generated a log file of all the events that occurred in the game and output the system state each time a hint was presented to the participant.

After the first phase of testing, the weights in the help system were altered according to the participants' responses and the system output at the time each hint was given. These new weights became the weights applied to the system state in the second phase of testing. The equation used to alter the weights was the single layer perceptron learning rule: $\Delta w_{ij} = \delta(t_j - h_j)s_i$ (Haykin, 1999), where Δw_{ij} is the change in weight, δ is a learning constant, t_j is the participant's response, h_j is the output of the knowledge inference engine for the hint and s_i is the state variable associated with the weight being altered. For each participant that was presented with a particular hint, the change in weights for that hint was calculated using the aforementioned formula and the mean of the change in weights across all participants given this hint was taken.

6. Results and Discussion

The results that were collected included the participants' responses for each hint that was presented, the system state values for each participant at the time that a hint was presented and the system logs that were generated for each participant. Figure 1 shows a summary of the participants' responses for Phase 1 and Phase 2, in terms of the number of 'yes' and 'no' responses from each participant. The success of these phases of testing can be evaluated by taking the mean ratio of 'yes' responses to total responses over all the participants. For Phase 1 this ratio was 0.62, meaning that 62% of responses given were 'yes' responses. For Phase 2 this ratio was 0.54, meaning that 54% of responses given were 'yes' responses.

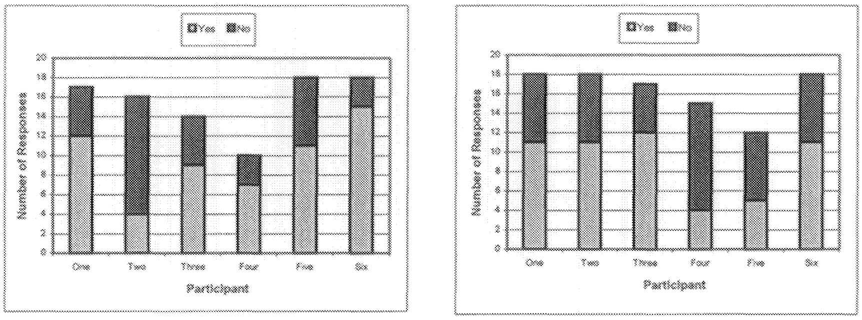

Figure 1. The number of Yes/No responses from each participant in Phase 1 (left) and Phase 2 (right).

Even though the difference in proportions between the Phase 1 and Phase 2 results was not significant (χ^2=1.038,p>.01) there are indications that the just-in-time hints help system is capable of detecting relationships that exist between the state variables and the optimal times for hint presentation. Initially, the results that were obtained were analysed in relation to individual hints. This analysis demonstrated that the just-in-time hints help system was 62% effective on the first trial and only 54% effective on the second trial. The most prominent reason why there was no significant difference between the two phases was that the sample size of subjects that participated in the experiment was very small. Due to this fact, the changes in weights that were calculated were not representative of the population and were therefore not able to be generalised to the testing sample. Therefore, the results have also been grouped into seven categories to enable the analysis to be carried out on larger subsets of data. This data was then analysed to determine the variables that contributed to the user's evaluation of the helpfulness of the hints. The method used to analyse this data was a step-wise linear regression. The results that were obtained from these groups of hints demonstrated that feasible patterns were beginning to emerge from the data.

Another factor that weighed heavily on the results that were obtained in this experiment was the fact that there was a great deal of variance in the skill levels of the subjects that participated. As the help system for this experiment was not designed to take varying skill levels into account there was considerable differences between responses from subjects who could be classed as beginner, intermediate or advanced. These distinctions could be made by assessing each individual's competence with computers in general and the extent of their prior experience with computer games. In short, from the observations that were made, it was apparent that this particular help system was best suited to the intermediate user as the beginner required more coaching on basic interactions and the expert was not interested in following the hints that were presented but rather in exploring the game in their own way.

A related point to the differing skill levels of the subjects is that there were various levels of helpfulness of each particular hint. Some hints were deemed to be unhelpful by each participant that they were presented to in both phases of testing, which seems to suggest that it was not the timing of these hints that was at fault but rather the hints themselves were simply not helpful in the game. Also, other hints were appraised as being helpful by each participant they was presented to in both phases, suggesting that these hints were so helpful that the stage in the game at which they were presented was irrelevant. So, a factor that contributes to a hint being deemed as being helpful, apart from the time at which it is presented, is the overall helpfulness of the hint in relation to the game.

Another concept that needs to be considered, when calculating which hint to deliver next, is not only the player's long term goals but also their short term and immediate goals. That is, the cumulative events that have occurred in the game may cause the help system to deliver a hint that is helpful in achieving the user's long-term goal but the deliverance of this hint at this particular point in time may conflict with their immediate and short-term goals. This in turn will cause the player to respond that this hint was not helpful, which is not true of its timing in the overall game, but is true of its timing in relation to immediate events. In order to combat this problem the help system would need to recognise the difference between these three different types of goals, i.e., long term, short term and immediate.

The recognition that there are different types of goals gives rise to the idea of clustering similar hints into a tree formation rather than using a simple list of hints. This would allow the user to follow one of multiple paths of sub-goals that lead to the overall goal of the game. Accordingly, this would facilitate the support of the user's immediate, short term and long term goals in that depending on which strategy the user is currently employing the appropriate branch of the tree of hints could be identified. In order to do this the system would need to incorporate the user's most recent actions as well as their progress over the entire game. Ideally, this would provide the user with hints that are most relevant to the goals that they are currently trying to achieve.

7. Conclusions

There are four major conclusions that can be drawn from this experiment. These are that (i) there was an insufficient number of participants to sample the population accurately, (ii) the skill level of the subjects needs to be taken into consideration, (iii) the player's immediate and short term as well as long term goals need to be considered and (iv) the hints may be more suitable to being arranged in groups than in a list formation. Taking this into consideration, it still can be seen from the analysis of the groups of hints that likely patterns in the data are beginning to emerge. Furthermore, it is predicted that with a significantly larger sample size the patterns being identified in the data will probably become more apparent and more representative of the population. Therefore, this project paves the way for future work on this topic.

Acknowledgements

Acknowledgement for contribution to this project must go to The Key Centre for Human Factors and Applied Cognitive Psychology and the School of Information Technology and Electrical Engineering at the University of Queensland for their support and the use of their resources. Also, thanks must go to Angela Maguire for her contribution to the experiments run in this project as well as for recording the hints for the game. Furthermore, the expertise in 3D modelling and animation of Peter Surawski was greatly appreciated in this project. Finally, acknowledgement must go to Auran, the creators of the 3D engine that was used to develop the computer game tested in this project.

References

R. Atkinson (1972). Ingredients for a Theory of Instruction, *American Psychologist*, 27: 921-931.

E. Donchin (1989). The Learning Strategies Project: Introductory Remarks, *Acta Psychologica*, 71: 1-15.

D. Druckman & R. A. Bjork (1991). *In the Mind's Eye: Enhancing Human Performance.* Washington D.C.: National Academy Press.

J. Frederiksen & B. White (1989). An Approach to Training Based Upon Principled Task Decomposition, *Acta Psychologica*, 71: 89-146.

S. Haykin (1999). *Neural Networks: A Comprehensive_Foundation*, New Jersey: Prentice Hall.

A. Mane & E. Donchin (1989). The Space Fortress Game, *Acta Psychologica*_71: 17-22.

J. Patrick (1992). *Training: Research and Practice.* London: Academic Press.