Modeling Player Decisions in a Supply Chain Game

Yifan Sun*, Chisheng Liang*
Northeastern University
Boston, MA
Email: *{yifansun, cliang, kaeli}@ece.neu.edu

Steven Sutherland†
University of Houston-Clear Lake
Houston, TX
†sutherland@uhcl.edu

Casper Harteveld‡, David Kaeli*
Northeastern University
Boston, MA
‡c.harteveld@neu.edu

***Abstract*—Player decision modeling can provide useful guidance to understand player performance in serious games. However, current player modeling focuses on high-level abstraction of player behavior rather than decision-level player modeling, and is predominantly applied to entertainment games. In this paper, we describe an approach from game design to data mining and data analysis to determine detailed player decision patterns. We illustrate this approach with *VistaLights*, a supply chain game we developed based on a recent oil spill event in Houston. With this game, we set up a within-subjects experiment to study decision making under varying circumstances, specifically to consider whether/how a recommendation system can improve human decisions. Using a series of data analysis techniques we built a coarse-grained decision model as well as a fine-grained model to compare players' actions on the game outcomes. The results confirm the need for decision-level modeling and show an ability of our approach to both identify the good and bad decision patterns among players.**

I. Introduction

Player modeling has become more popular and essential for game design to appeal to a broad audience. Player modeling techniques aim to abstract player behavior patterns and have been successfully applied to game development [1], [2], self-adaptive games [3], [4], and agent design [5]. These techniques would be useful for serious games too; however, thus far applications have been rather limited [6], [7].

Unfortunately, the existing work in player modeling typically classify players based on their high-level behavioral statistics. No matter which machine learning or data mining approaches are used, aggregate properties such as play time or number of actions tend to be used to categorize players into a limited number of classes. Although general clustering based on player features can be useful, this likely does not provide the depth and accuracy needed to understand the dynamics of any game, whether a serious or entertainment game. For example, with this high-level approach it is difficult to determine what the key decisions are that lead to a poor or good performance. Exactly this kind of information is critical for improving the effectiveness of serious games. Therefore, we argue that the existing work needs to be complemented with decision-level modeling and decision-by-decision evaluation. The motivation for this paper is to demonstrate the usefulness of this low-level approach for serious games.

In order to explore how to study human decision-making behaviors in serious games, including how player decisions can be improved by providing help in the form of decision aids, we have developed a supply chain game from scratch called *VistaLights*. In this simplified but realistic simulation game, players manage a port by prioritizing ships and dealing with disruptions, specifically oil spills. The game is inspired by the recent oil spill event at the Port of Houston [8]. Although a simplification of reality, this game provides a complex dynamic decision making environment where optimization techniques cannot find the optimal solution strategy and simple analytical techniques do not reveal why certain players performed as they did. By developing it ourselves, we have complete control over what happens in the game, allowing us to systematically study player decisions. In this paper we report the findings of our initial pilot study, where we set up a within-subjects experiment with three levels that vary in the use of a recommendation system (no recommendation, recommendation, and recommendation with justification).

Our contributions are threefold. First, we illustrate through a detailed description of designing and evaluating *VistaLights* how to develop and study a serious game for decision making, and demonstrate how to assess a well balanced design, which is needed to analyze player performance. Second, we detail a generalized approach not limited to *VistaLights* to understand the impact of player decisions on game outcomes from a high-level and decision-level perspective, and highlight the limitations of clustering techniques. Third, we report our findings from analyzing player behavior in *VistaLights*, including how players engaged with the recommendation systems.

II. Background

A. Serious Games

Serious games, games with a non-entertainment purpose, are increasingly used in various fields, from health to business, to study and improve human behavior [9]. As all games are essentially about making decisions, it is key to identify how players make decisions to increase the effectiveness of serious games. First, by identifying player behavior the design can be adjusted accordingly to maximize the impact the game is attempting to achieve. For example, when players make poor decisions, the game can recognize this and provide personalized feedback or adjustments. Second, players themselves can then identify which types of players they are and how they need to improve their decision making.

Typically, player decisions are evaluated according to a normative model, and then players receive feedback accordingly to improve their behavior. Such evaluations would still benefit from player modeling to be able to personalize the game, and the limited work where player modeling has been applied to serious games has exactly done this, by modeling players according to normative models [6], [7]. However, in complex dynamic games such as *VistaLights* this typical approach will not be sufficient because it is difficult to determine upfront what the key decisions are. But even in simpler games unexpected behavior may happen—actions not identified by the normative models—and identifying how these actions impact the game outcomes will be beneficial. We argue that this type of identification requires a different approach to player modeling, one that considers decision-level analysis.

B. Player Modeling

In terms of player modeling approaches, machine learning and data mining techniques, especially clustering techniques, have been widely accepted [10], [11]. For example, Drachen et al. [1] use Emergent Self-Organizing Maps to cluster high-level player behavior features, such as completion time and number of deaths. The clustering result is used to improve the game and determine whether the player is following the game designer's intention. However, even with advanced machine learning algorithms, modeling human player decisions can be difficult due to the large data dimensions and the uncertainty of human behavior [12]. Other approaches have also been applied to better understand player decisions. Holmgård et al. [2] use generative agents as personas to characterize and discriminate human players. They show that a high-level abstraction of human decisions is possible. In our research, we use similar approaches to cluster human decisions, as well as some novel solutions to identify a more fine-grained analysis, and apply these in the context of a serious game.

III. Design

In this section, we describe the design of *VistaLights*[1], which we use as a research environment to study player decisions and the role of decisions aids. We discuss the context for the development, how to play it, and how the scores are calculated.

A. Context

We modeled the game after the Port of Houston. Based on discussions with stakeholders, we understood that when a disruption occurs, representatives from different industries in addition to several authorities discuss what actions to take. The different industries concern: breakbulk, dry bulk, and liquid bulk. The port authority takes care of container ships but also of special ships such as cruise ships. Actions basically involve prioritizing certain ships and implementing mitigation strategies. The resiliency to bounce back from a disruption is of interest to everyone involved because the port is a shared infrastructure on which everyone's productivity depends.

In building the first game version, we focused on disruptions caused by oil spills. Specifically, we used the oil spill that happened in March 22, 2014 as an example. At that date, a collision occurred between an oil barge and a ship at a critical node in the network (blocking the Houston Ship Channel where all ships must pass to enter and exit the port), causing an almost complete standstill in the port. Interestingly, a cruise ship was required to wait right outside of the port and could not enter due to the spill. The decision was made to let the cruise ship go through the oil spill before cleaning it up. We were also inspired by a decision aid and monitoring system that is used to schedule ships. This system is essentially a combination of a visualization of the entire port as well as priority queue of ships and their characteristics. It formed the basis for the game's interface and gameplay.

[1]For game and source code, see http://hdl.handle.net/2047/D20213074

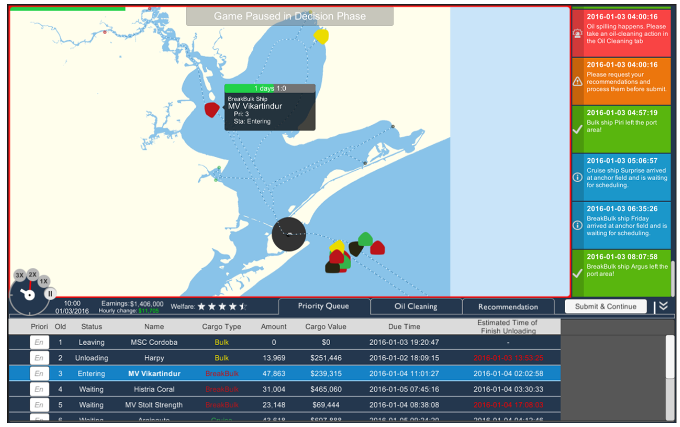


Fig. 1: A screen shot of *VistaLights*. The main screen shows the map of the port with the ships; on the right are the messages; at the bottom is the priority queue and control panel.

B. Gameplay

Like any simulation game, in representing the object of interest we made simplifications. For example, the current version is a single player game where the player single-handedly decides what actions to take. The player's goal is to manage the port by prioritizing ships and dealing with disruptions when they occur. In managing the port there are two benchmarks to consider: earnings and welfare. Earnings is a quantitative economic score based on the efficient use of the port's infrastructure; welfare is a qualitative composite score that considers the environment and reputation.

In the game, players see a map of the Port of Houston with a network of channels that ships use to navigate to the docks to unload their cargo. Players cannot directly control the ships; they can only change the priorities in the priority queue, which lists all of the ships with their names, current status, industry type, cargo amount and value, due time, and estimated time of unloading (Fig. 1). Each ship is assigned a unique priority value and ships with higher priority will be scheduled first. Lower priorities will be scheduled when no conflicts exist. In prioritizing ships, players will need to maximize the occupancy rate of the shipping lanes and docks while minimizing penalties that result from ships being overdue.

Players must further decide how to respond to the oil spill when it occurs. Other than the null-option (leaving the oil spill alone), players have three options to clean up the oil: burning, dispersants, and skimmers. Each option varies in cost, clean up time, impact on traffic, and impact on welfare. Additionally, with burning and skimmers the traffic can only resume after the spill has been cleaned up; with the null-option and dispersants the traffic can continue at a lower speed. Players can postpone their decision to allow time-critical ships to go into port but at the cost of a welfare penalty.

The game is divided into two phases: the decision and simulation phase. During decision phases, players can take actions. Although the simulation time is paused during these phases, players need to submit their decisions within a certain time. During the simulation phases players cannot take action; however, they can retrieve information about the ships that would be useful to make informed decisions during the next decision phase. The game spans several days in the port and decision phases occur every six hours of port operation (i.e., one simulation phase covers six hours of port operation). To help players be aware of what is happening in the port, they receive messages categorized on events throughout (Fig. 1). Players can change the speed of the simulation time during the simulation phases with a control panel.

C. *Score Calculation*

The player's goal is to maximize the economic score without reducing the overall welfare. The economic score is increased by unloading cargo, and calculated by multiplying the amount of cargo by the cargo value. It is decreased by due time penalties if ships fail to unload on time in addition to cargo maintenance costs: the longer the cargo stays on the ship, the higher the cargo maintenance cost will be. The cost for oil cleaning will also be subtracted from the earnings made. We express the economic score as the total earnings generated after a number of days and as the average earnings per hour.

Welfare is a qualitative score with a value between zero and five (represented as stars). It is negatively affected by overdue cruise ships and how the oil spill is handled. When a cruise ship is overdue, it continues to decrease proportionate to the number of passengers until all passengers have left the ship. With the oil spill, it continues to decrease proportionate to the amount of oil until it is cleaned up. An additional penalty is applied for the chosen solution because solutions such as burning and dispersants have further implications for the environment. The welfare score recovers at a constant but slow rate; however, if it becomes zero, players lose the game.

IV. Methods

Our goal for the pilot study with *VistaLights* was to explore decision making in a serious game. For the study we implemented a within-subjects experimental design; however, the scope of this paper is specific to evaluating modeling player behavior within this space, not to evaluate the manipulations themselves, which were intentionally implemented to observe decision making under varying circumstances.

A. *Participants*

Participants were recruited at Northeastern University and University of Houston-Clear Lake. At the first University primarily students in Computer Engineering volunteered to participate (N = 26); at the second University it concerned solely students in Psychology who participated for credit (N = 11). No demographic information was collected.

B. *Materials*

The game *VistaLights* that is provided as the material for the study has four levels. The first level is a tutorial that explains step by step how to play the game with a pop-up screen. The level itself is a short level that ends in two days of simulation time. The other three levels have been designed according to the experimental design described in detail in the following section. These levels end in five days of simulation time. Based on playtests we roughly estimated that it would take an hour to play all levels if players would make use of increasing the simulation speed at times. We set the maximum time during the decision phases at two minutes. In the remainder of the paper we refer to the first level as the tutorial and to the other three levels as Challenge 1, 2, and 3, respectively.

We varied the four levels in terms of a number of level characteristics and manipulations, both which we discuss in detail below. These controlled variation allow us to maintain some consistency over the types of decisions the players must make and to identify how players learn to modify their decisions across the game, while allowing enough variability to isolate the impact of specific variables on player decisions.

1) Level Characteristics: We varied the levels in terms of ships, oil spill location, and goals. First, we populated each level with 30 different ships. We randomized all the ship values between realistic values for one level first. For example, we calculated the arrival time by multiplying a random number between zero and one and multiplying this by three days. For the other levels we kept the same values except for industry type, arrival time, and due time. For those characteristics we calculated a new random value. We made these variations to make sure players experience different scenarios, and are therefore not inclined to take the exact same decisions.

Second, we varied the location of the oil spill between three locations for Challenge 1, 2, and 3; no oil spill occurred during the tutorial. The three locations were chosen such that they would have the same impact on the game; however, it gives players the illusion that there is variation and that they cannot predict what will happen. Every oil spill happens around the same time, after two days, with a few hours difference between each level. Unlike the oil spill, we used the exact same network with the same number and type of docks for all four levels. For each industry type (breakbulk, dry bulk, liquid bulk, and cruise ships) we included two docks and mapped them to how these industry types are located in the Port of Houston.

Third, we varied the earnings and welfare goals between levels. At the start of each level, players receive a message that specify the earnings and welfare targets that they have to obtain. We determined realistic target goals for both the revenue and welfare based on prior playtests. For example, the first level is much harder and so we set the target goals lower than for the subsequent levels.

2) Level Manipulations: Our manipulations pertaining to the levels are related to the provision of a decision aid that provides recommendations regarding how to prioritize the ships. In addition to a level where no decision aid is provided, we settled for this initial study on two variations: a recommendation without and with justification. These manipulations allowed us to explore how players make decisions when confronted with a complex, unfamiliar task under varying circumstances. This manipulation resulted in the following: players received no recommendations for Challenge 1; they received recommendations regarding prioritizing ships with no justification for the recommendation for Challenge 2; and they received recommendations regarding ship prioritization and justifications for those recommendations for Challenge 3.

In both recommendation systems, the player receives up to three recommendations in each decision phase. For each level (other than the tutorial) there are 20 decision phases across the five days. Recommendations are based on the ships that are at the time of the decision phase waiting outside of the port; ships that are moving and unloading are ignored. Both systems will first check if any cruise ship is going to be overdue or is already overdue. Then they will check on overdue or nearly overdue ships. From there recommendations involve prioritizing ships with the highest total cargo value.

Recommendations with suggested priority values are made in the order described above and in the format of "Consider to prioritize ship <ship_name> to priority <x>." We decided for both systems to recommend a specific priority because in that way we can determine whether players comply with the advice. The difference between the two recommendation systems concerns the justification. Justifications are short explanations such as "Because this ship has a high cargo value."

The three recommendations are provided but only after players requested the advice. As players cannot progress without requesting advice, we essentially required them to do this. This seems unnatural but was implemented to ensure that players would consider the recommendations, which is the manipulation they are exposed to, and not play the game without the recommendations provided. Once the recommendations are requested, players then need to accept or reject each one of them before they can progress to the priority queue to make their changes. In this priority queue, they see the old value as well as the suggested value by the recommendation system.

We designed the recommendation system to be imperfect and purposely did not inform players about its logic. For example, players could prioritize moving ships whereas the recommendation system does not include these. Therefore, a better performance is possible by not completely relying on the recommendations.

C. Procedure

We implemented a within-subjects experiment where every participant experiences every condition. There are three conditions: • *Challenge 1* (no recommendation), • *Challenge 2* (recommendation), and • *Challenge 3* (recommendation with justification). The tutorial level was included to make sure that players first learn how to play the game before starting the experiment and to minimize the practice effect from Challenge 1 to Challenge 2. We did not vary in the order of the conditions because seeing the recommendation, and most certainly the recommendation with justification, would likely affect further play. A consideration for just three conditions was a possible fatigue effect. We requested that all players finish the experiment in one session to minimize any possible bias from contextual factors for when and how players engage with the game. Although the within-subjects design creates the possibility of learning effects and behavior constancy, it was necessary to determine whether prior performance in the game predicted future compliance with the recommendations.

The implementation was different at both Universities. At the first University the game was distributed with instructions to play over e-mail. Participants were requested to take an hour and complete all levels in one sitting. At the second University players participated in person in a lab setting. After the facilitator briefed them about the purpose of the study and how to play, they were assigned to a computer where the game was installed. This variation in play context was not intentional. It was pragmatic and based on the infrastructures in place for the researchers involved. For both locations selected participants were contacted for a debriefing interview to understand what strategies they used in the game.

D. Data Analysis

We applied several different data analysis techniques to evaluate player behavior. We first analyzed the distribution of the players' earnings and welfare results to examine if the game is well balanced. If the game is too difficult or too easy, it becomes harder to distinguish players, and what decisions can be considered good or bad. We then examined the role of the recommendation systems by considering the relationship between compliance rates with player performance. We also compared a typical poor and top player from our sample with a hypothetical player who does not take any actions ("No Action Player") and one who complies with everything that the recommendation systems suggest but does nothing else ("Compliance Player"). We performed these analyses to show if the recommendation systems are useful and if players can perform better than the imperfect recommendation systems.

To discover the decision patterns, we performed a coarse-grained analysis and a fine-grained analysis. For the coarse-grained analysis, we linked the players' decisions on the oil cleaning solutions and the ship priorities with their results on earnings and welfare. For the ship priorities we categorized different groups of priority level changes and counted the changes per group. We then performed a clustering analysis and used the resulting clusters of player decisions to predict players' win/lose probability. The clustering was done by first whitening [13] the number of actions of each priority group category, and then using Ward method [14] as the criterion to perform hierarchical clustering [15] analysis.

For each of the above analyses, all 37 players' data are considered; however, for the fine-grained analysis we focused on only two players to illustrate our proposed method of analyzing specific player decisions in serious games. By comparing the cycle-by-cycle decisions of two similar players, we explored how individual decisions impacted the performance trajectory of each player, and tried to infer what strategies the players had used. To complement our strategy inferences, and see if players intentionally took certain actions, we compared this analysis with our interview notes of selected players who articulated their strategies to us.

V. Results

In this section, we discuss first the overall player performance and then the role of the recommendation system, followed by the influence of the oil cleaning decisions and the priority change decisions. Finally, we illustrate the play trajectories of two players and show how making similar decisions can still lead to drastically different results in a dynamic environment such as *VistaLights*.

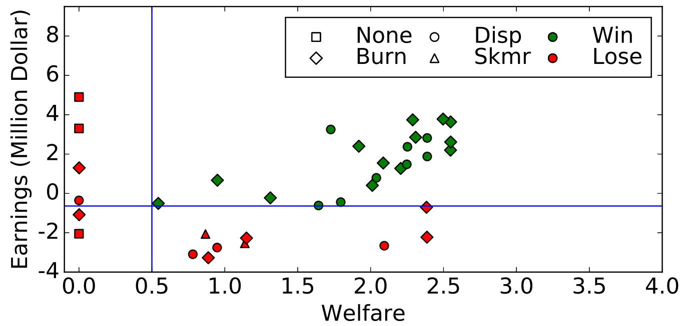
A. Player Performance

The final results on earnings and welfare are depicted for each player in each level in Figure 2. The horizontal lines represent the earnings target in each subfigure; the vertical lines represent the welfare target. The resulting quadrants show if players won the game (upper right quadrant), lost because they failed to meet the earnings goal (lower right quadrant), lost because they failed to meet the welfare goal (upper left quadrant), or lost because they failed to meet both earnings and welfare goals (lower left quadrant). Except for Challenge 3, where few players lost on both earnings and welfare, players were well-distributed over the quadrants, suggesting that the targets were fair and that the game had a reasonable difficulty.

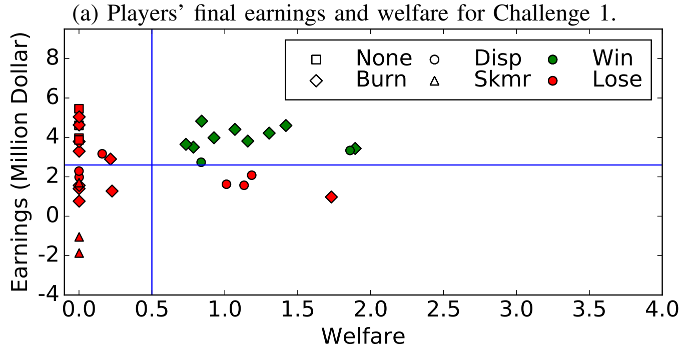
There was no immediate clear pattern for previous player performance predicting future performance. A player who did well in Challenge 1 did not necessarily do well in Challenges 2 and 3. The likelihood that a player got the same result in Challenges 1 and 2, Challenges 1 and 3, and Challenges 2 and 3 were 51%, 62%, 48%, respectively. This finding was also illustrated by the win percentage: the percentages of players who won were 58% (21 of 36), 29% (10 of 34), and 60% (18 of 30) for Challenges 1, 2, and 3, respectively, suggesting that Challenge 2 may have been been more difficult than the other challenges. Of the 30 participants who completed all levels, five players won every challenge and five lost every challenge. Losing two or winning two was split almost equally as well.

B. Recommendation System

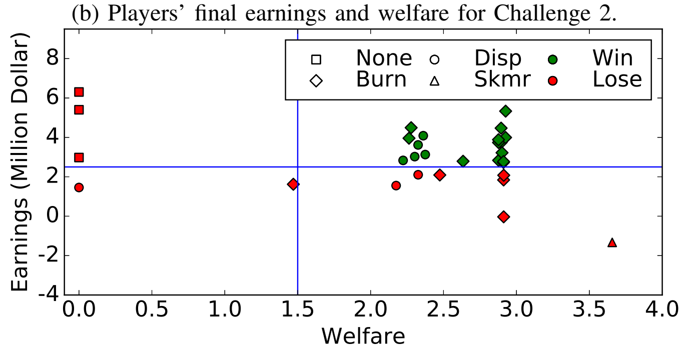
In Challenges 2 and 3, we provided players with an imperfect recommendation system. However, this recommendation system was most certainly beneficial. Table I illustrates the performance of the recommendation system by modeling a player who complies with all provided recommendations and does not do anything else. We compared this performance by modeling a player who does not take any action at all and a typical poor and top player from our sample. This table shows that even the poorest players made some good decisions but



(a) Players' final earnings and welfare for Challenge 1.



(b) Players' final earnings and welfare for Challenge 2.



(c) Players' final earnings and welfare for Challenge 3.

Fig. 2: Distribution of the players' final results. The blue lines are the target earnings and welfare. The shapes represent the oil spill solutions players chose.

that if they only complied with the recommendation system, doing nothing else, their final results would have been better. In fact, if a player simply accepted all the recommendations and chose dispersants or burning as their oil spill solution, they would have exceeded the targets and won both challenges. The table also shows that players can outperform the recommendation system. The system only suggested three priority changes and only for ships that were waiting outside of the port. Therefore, room existed for human decision making to outperform simple recommendation compliance in the game.

In terms of the usage of the recommendation systems, we calculated the rate with which players accepted the recommendations and then the rate that they complied by actually implementing the advice. A strong correlation existed between the acceptance rate and the compliance rate for both Challenge 2, $r = .96$, $p < .001$, and Challenge 3, $r = .93$, $p <$

TABLE I: The performance comparison between a hypothetical player not taking any action, a typical poor player, a hypothetical player who complies with every recommendation and nothing else, and a typical top player. For the hypothetical players we chose dispersants when an oil spilling happens.

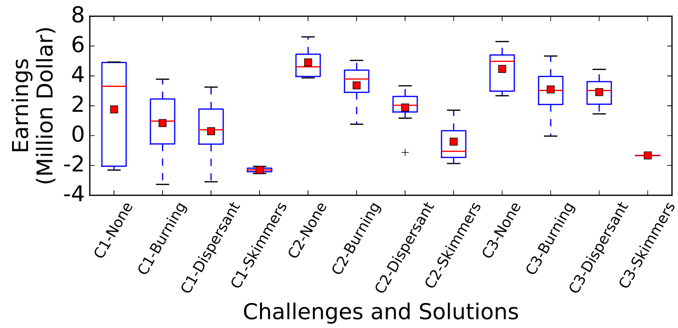
Player	Item	C2	C3
No Action Player	Earnings	-3.89M	-0.76M
	Welfare	0.00	1.98
Poor Player	Earnings	-1.87M	1.45M
	Welfare	0.00	0.00
Targets for Players	Earnings	2.60M	2.50M
	Welfare	0.50	1.50
Compliance Player	Earnings	3.13M	3.15M
	Welfare	1.48	2.30
Top Player	Earnings	4.60M	5.33M
	Welfare	1.42	2.93

.001. Therefore, when players accepted the advice, they also complied by implementing the advice. The actual acceptance rates were similar across challenges: in both challenges, a small majority of the advice was implemented (M_2 = .55, SD_2 = .29; M_3 = .55, SD_3 = .33). The rates were, in fact, similar because reliance on the recommendation system in Challenge 2 was a strong predictor of reliance in Challenge 3, accounting for 81% of the variability (R^2 = .81). Knowing that the recommendations do help, it seems that more players could have benefited from an increased reliance. However, we did not find any relationship in the data between the compliance rates and performance, suggesting that any differences we found between the challenges in terms of performance could not be explained by the use of the recommendation systems. Therefore, other factors determined how well players performed.

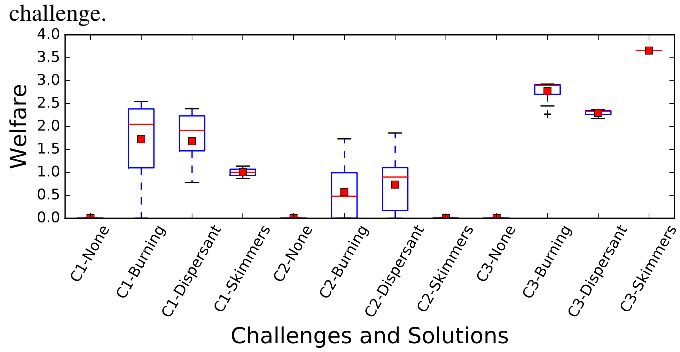
Additionally, performance on previous challenges (earnings, welfare, and whether or not the player won the challenge) did not predict whether players would rely on the recommendations for later challenges. It should be expected that players who had seen that they were unsuccessful would have been more likely to comply with the recommendations, but this was not the case. Because compliance did not predict performance and previous performance did not predict future compliance, despite the fact that simply following the recommendations and doing nothing else would lead to success, we argue that players were not able to effectively decide when the recommendations were beneficial and when they were not. This result may have been due to players not having received immediate feedback about their decisions to comply. The sum of their decisions was reflected as a final score at the end of the challenge, making it difficult to identify which decisions should be changed.

C. Oil Cleaning Decisions

The response to the oil spill was one of the key player decisions, and our results confirmed this. Figure 3 shows the distribution of the earnings and welfare for each challenge by different oil cleaning solutions. The trend of how solutions impacted earnings was clear and similar from challenge to challenge. The null-option did not cost anything and did not



(a) Distribution of earnings of different oil cleaning solutions in each challenge.



(b) Distribution of welfare of different oil cleaning solution in each challenge.

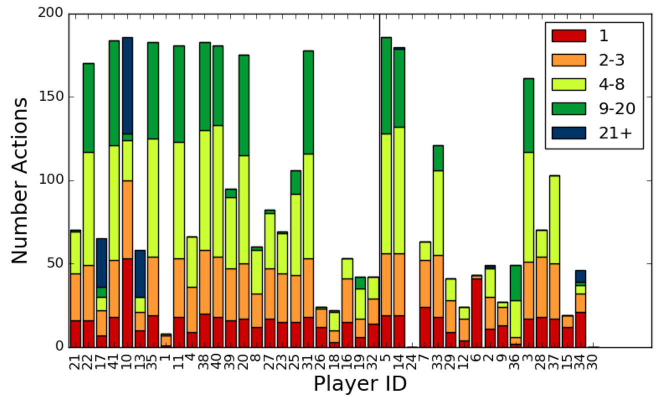
Fig. 3: Oil cleaning solution impact on the final result.

cause traffic to stop. Burning did stop traffic but had the advantage that it cleared the oil relatively quickly. Few players chose skimmers and the figure highlights that it may not have been the best solution for earnings. Its advantages were likely overruled by the penalties for overdue cargo.

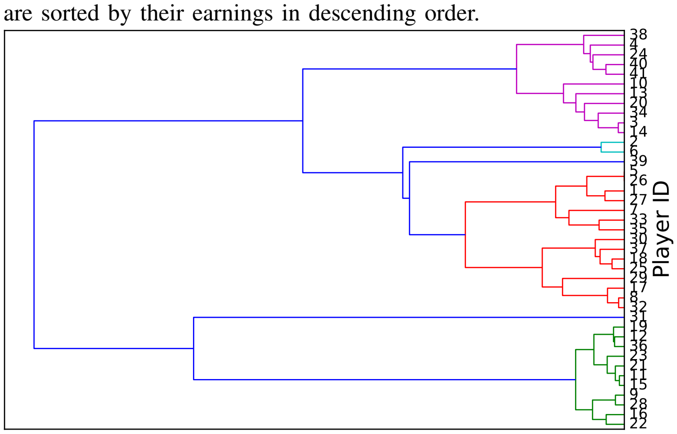
The solutions were also related to welfare. The null-option was a guarantee for losing the game. However, the patterns for the effects across the challenges were dissimilar. Further investigation revealed that this had to do with the arrival of the cruise ships around the oil spill. With Challenge 2, two cruise ships arrived shortly after the oil spill, explaining why so many players lost that challenge due to welfare. In contrast, there was little variation in Challenge 3 because no cruise ship arrived during the oil cleaning period. The variance of burning was also a result of cruise ship scheduling. Those that implemented burning after letting the cruise ships pass first, did better on welfare.

D. Priority Change Decisions

In addition to the one-time oil-cleaning decision, players were tasked with changing the ship priorities. As shown in Figure 4a, we counted the number of priority change actions and what kind of priority changes players made in Challenge 1. We divided priority changes into the following groups: priority 1, priorities 2 and 3, priorities 4 to 8, priorities 9 to 20, and priorities higher than 21. In this figure we further placed those who won to the left side of the vertical line and those who lost to the right side. This figure shows that players who won Challenge 1 made more changes, and that the number of

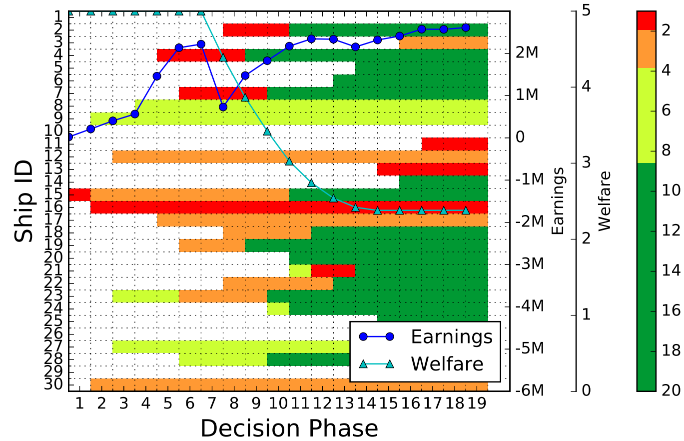


(a) The number of actions for assigning new priorities in Challenge 1, categorized by type of priority change. To the left of the vertical line are the players that won. In both the win and lose groups, players are sorted by their earnings in descending order.

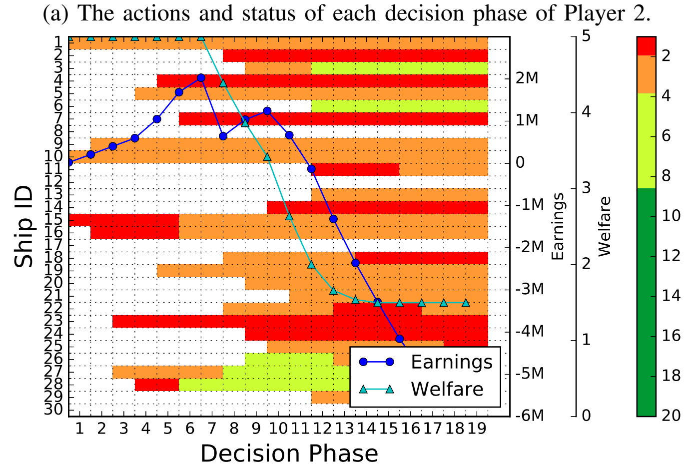


(b) Cluster analysis on the player's priority changes

Fig. 4: Analysis of players decision pattern



(a) The actions and status of each decision phase of Player 2.



(b) The action and the status of each decision phase of Player 34.

Fig. 5: Analysis of individual play trajectories. The trajectory includes the scores on earnings and welfare across the decision phases as well as the priority changes for all 30 ships. The priority changes are color coded with the priority groups.

actions predicted performance earnings with 26% variability (R^2 = .26). This prediction was not evident in Challenges 2 and 3, and the number of actions did not increase due to the recommendation systems. Therefore, it may be that players who put in more effort in Challenge 1 were able to change the priorities of critical ships, whether intentionally or by chance.

To better understand player decision patterns, we conducted a cluster analysis, and the result is depicted in Figure 4b. There were three major clusters, and from the top to bottom, they can be characterized as the medium amount of priority change, the low amount of priority change, and the high amount of priority change. An outlier was player 31 who had a large amount of actions that set priority to 1 and to a value of higher than 20. A related subgroup included Players 2 and 6 who also moved ships to a priority higher than 20. By explicitly moving the ship to the end of the priority queue, they freed up the main channel and let more urgent ships enter.

E. Detailed Play Trajectories

From these results we were unable to get a fine-grained decision evaluation, such as determining if a particular decision was good or bad. For example, Players 12 and 15 made almost the same number of priority changes and they both used dispersants as oil cleaning solutions. However, they ended up with different results. To perform a more fine-grained decision evaluation we selected two seemingly similar players, Players 2 and 34, and analyzed their play trajectories. Their trajectories are illustrated in Figure 5. This figure shows their earnings and welfare over time, in addition to what changes they made to the 30 ships over the 20 decision phases.

According to the earnings curve, there was a turning point at decision Phase 9. After Phase 9, the earnings of Player 2 kept increasing while the earnings of Player 34 sharply plunged until the end of the game. Player 34 must have made some critical decision just before or during this phase that caused the avalanche effect. The first difference was that player 34 moved Ship 3 to Priority 1 at Phase 9 and this ship was a breakbulk ship. When we recreated the game according to the player's log file, we noticed that by that time, the breakbulk docks were already heavily overloaded and Ship 3 was going to be overdue according to the system estimation. During that decision phase, Player 34 made the natural decision to move Ship 3 to a higher priority. As a consequence, ships at breakbulk docks and ships that were already in the port had to wait for Ship 3 to move in. Dock utilization was then significantly reduced by this action, and other ships may have, as a result, become overdue. Similar actions were made by Player 34 repeatedly throughout the game, including prioritizing another breakbulk ship, Ship 26, at Phase 9. We consider such decisions to be bad decisions.

Another key difference between Player 2 and Player 34 was that Player 2 moved Ships 4 and 19 to a low priority. By explicitly moving ships to the end of the priority queue, Player 2 let the ships that had already unloaded wait close to the dock before moving out of the port, which prevented outgoing ships to occupy the channel. We consider such actions as good decisions and Player 34 did not take these.

An interview with Player 2 confirmed that the strategy was intentional. He stated: "I gave lower priority to the unloading ships and higher priority to cruise and cargo with high value." Others also articulated this strategy. However, Player 14 did not use that strategy and still outperformed Player 2 in Challenge 1. From the interview it becomes clear that this player has a well thought out strategy for playing, and is not a top player by chance: "First, I usually set high priority to the overdue cargo or cargo that is going to be due. Second, I watched the docks closely, each color represents a particular type of industry and I always tried to keep the docks busy all the time and if I see any dock is empty, I will give higher priority to a proper ship to enter that dock. Third, I will also keep an eye on the values. The more expensive the ship's cargo, the higher priority the cargo will have." Player 14's strategy of watching the under utilized docks was only used by the high-performing players.

VI. Conclusion

In this paper we presented a supply chain game called *VistaLights* that we developed to model human decisions. The results from a within-subjects experimental pilot study with 37 participants highlighted that this is a fair and valid environment to study decision making: the participants were reasonably well distributed in terms of their performance and participants had to make the right decisions to perform well. Our results illustrate that straightforward analyses do not illuminate what happens in these complex dynamic decision-making environments and that fine-grained decision models are needed in addition to data mining techniques.

Key insights are that certain critical bad decisions can negatively impact the outcomes. Therefore, it is of importance to identify when people are about to make such bad decisions. Likewise, critical good decisions can positively impact the outcomes. Identification and recommendation of such good decisions would help improve the effectiveness of serious games for training, and may even impact the actual workplace. The results also highlight that participants may need to rely more on recommendation systems, even if they are imperfect, especially when their own performance suggests an inability to succeed. Finally, when it comes to the recommendation systems we show that some people are simply more willing to rely on these than others, as illustrated by the only significant predictor of future recommendation compliance having been past compliance. This signifies the importance of player modeling as it identifies individual differences.

Our work can serve as an example of how to design for games to model player decisions and then how to analyze these decisions. We acknowledge there are limitations to our game and our analysis approach. To understand how the recommendation system can help in the decision making process, we plan to perform studies with more players, and compare the performance with additional players that do not have recommendations in any challenge. We plan to expand this pilot work with additional design variations and by developing analytical techniques that will help to analyze fine-grained decisions on a larger scale. The analytical model should also be implemented together with the game to give players real-time guidance as part of the recommendation system.

Acknowledgment

We thank the Greater Houston Port Bureau for their input.

References

[1] A. Drachen, A. Canossa, and G. N. Yannakakis, "Player modeling using self-organization in Tomb Raider: Underworld," in *Computational Intelligence and Games*, 2009, pp. 1–8.

[2] C. Holmgård, A. Liapis, J. Togelius, and G. N. Yannakakis, "Generative agents for player decision modeling in games," *Foundations of Digital Games*, 2014.

[3] O. Missura and T. Gärtner, "Player modeling for intelligent difficulty adjustment," in *Discovery Science*, 2009, pp. 197–211.

[4] J. Togelius, R. De Nardi, and S. M. Lucas, "Making racing fun through player modeling and track evolution," in *SAB Workshop on Adaptive Approaches for Optimizing Player Satisfaction in Computer and Physical Games*, 2006.

[5] P. Spronck and F. den Teuling, "Player modeling in Civilization IV," in *AIIDE*, 2010, pp. 180–185.

[6] S. Göbel, V. Wendel, C. Ritter, and R. Steinmetz, "Personalized, adaptive digital educational games using narrative game-based learning objects," in *International Conference on Technologies for E-Learning and Digital Entertainment*, 2010, pp. 438–445.

[7] Y.-G. Cheong, R. Khaled, C. Grappiolo, J. Campos, C. Martinho, G. P. Ingram, A. Paiva, and G. Yannakakis, "A computational approach towards conflict resolution for serious games," in *Foundations of Digital Games*, 2011, pp. 15–22.

[8] H. Rice, A. Hassan, and L. Olsen, "Ship channel remains closed after oil spill," 2015. [Online]. Available: http://goo.gl/L3nmZm

[9] C. Harteveld, *Triadic game design: Balancing reality, meaning and play*. London, UK: Springer, 2011.

[10] B. G. Weber and M. Mateas, "A data mining approach to strategy prediction," in *Computational Intelligence and Games*, 2009, pp. 140–147.

[11] A. Drachen, C. Thurau, R. Sifa, and C. Bauckhage, "A comparison of methods for player clustering via behavioral telemetry," in *Foundations of Digital Games*, 2013, pp. 245–252.

[12] A. Drachen, R. Sifa, C. Bauckhage, and C. Thurau, "Guns, swords and data: Clustering of player behavior in computer games in the wild," in *Computational Intelligence and Games*, 2012, pp. 163–170.

[13] A. Kessy, A. Lewin, and K. Strimmer, "Optimal whitening and decorrelation," *ArXiv e-prints*, Dec. 2015.

[14] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963.

[15] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.