Unsupervised Modelling of Player Style with LDA

Jeremy Gow, Robin Baumgarten, Paul Cairns, Simon Colton and Paul Miller

***Abstract*—Computational analysis of player style has significant potential for video game design: it can provide insights into player behaviour, as well as the means to dynamically adapt a game to each individual's style of play. To realise this potential, computational methods need to go beyond considerations of challenge and ability and account for aesthetic aspects of player style. We describe here a semi-automatic unsupervised learning approach to modelling player style using multi-class Linear Discriminant Analysis (LDA). We argue that this approach is widely applicable for modelling player style in a wide range of games, including commercial applications, and illustrate it with two case studies: the first for a novel arcade game called *Snakeotron*, the second for *Rogue Trooper*, a modern commercial third-person shooter video game.**

***Index Terms*—Video games, player style, player types, log analysis, LDA, k-means clustering, adaptive games.**

I. Introduction

UNDERSTANDING player style is an essential part of video game design. In all but the simplest of games, there is potential for players to find different ways to play. These styles can be an expression of many factors, such as strategy, ability, mood, taste, experience, personality or culture. Knowledge about player style can be critical in developing and refining a game design, particularly if it is based on observations of real players of the game in question [1], [2]. With this knowledge, designers are free to embrace a range of styles, or reject some and focus on specific groups of players.

Computational methods have a central role to play in making this knowledge available to designers. Many game companies, from console to independent developers, are now collecting quantitative play data on an unprecedented scale, during playtesting and after release via game telemetry. There are many opportunities here for Artificial Intelligence research to contribute to game design, including the modelling of player style, classification of style according to existing models, and adaptation to individual styles.

Because challenge and ability are central to video games, there is a temptation for computational approaches to reduce style to task performance, and adaptation to "dynamic difficulty adjustment". While these approaches are valid, they are only part of a picture of style which includes less tangible factors such as taste, experience and culture. In general, machine analysis of player style is, in part, a form of computational aesthetics. Methods need to account for variations based on the sensory and emotional values of players and adapt in ways that are sensitive to the aesthetics of game design.

In this paper we describe the use of Linear Discriminant Analysis (LDA) [3] for unsupervised generation from log data of traits describing player style, and discuss its potential for informing design. An individual player is represented by a group of automatically extracted segments of log data, to which a number of feature metrics are applied. The resulting high-dimensional space is then reduced using LDA to a low-dimensional representation which maximises separation between log extracts from different players and minimises separation between extracts from the same player. The dimensions (discriminant functions) of this LDA space are interpreted as traits which can give insights into player style, and can support further computational analysis, e.g. clustering of players into types.

LDA is a supervised method — we employ individual players as class labels for log data extracts — but here we are using it to generate a player model in an unsupervised fashion, i.e. no explicit information is supplied about player style. We propose that, compared to applicable unsupervised techniques (e.g. PCA [4], self-organising maps [5]), one advantage of using LDA in this context is that it emphasises differences between players, and is less influenced by other sources of regularities, e.g. differences between distinct combat scenarios. As an approach which does not make use of existing models or classifications, we claim it is well suited to studying the aesthetic aspects of player style, which might be difficult to anticipate for specific games, even in established genres.

We explore the use of LDA in studies with two games. To introduce our approach we describe an analysis of *Snakeotron*, a novel arcade-style game (Section III). The generated player style model distinguishes input rate, wall avoidance, steering style and risk. In the second study, we look at more complex gameplay in *Rogue Trooper*, a modern commercial third-person shooter video game (Section IV). Our method identifies combat dynamism, cautiousness and ammunition management as three player style traits. We compare these result to some external player metrics, contrast with PCA, and build on the model to generate a player style typology with k-means clustering. The feature data and scripts used in our LDA and cluster analyses are publicly available[1].

II. Background

Research on models of video game players is often traced back to Bartle's work on Multi-User Dungeons [6], which later became known as the "Achiever, Explorer, Socializer, Killer" typology. A series of player typologies followed, inspired by both video game theory and empirical studies, e.g. Bateman &

Gow, Baumgarten and Colton are with the Computational Creativity Group, Department of Computing, Imperial College London, UK. Email j.gow, robin.baumgarten06 or sgc@doc.ic.ac.uk or visit ccg.doc.ic.ac.uk.

Cairns (pcairns@cs.york.ac.uk) is with the Human Computer Interaction Research Group, Department of Computer Science, University of York, UK.

Miller (paulm@ieee.org) was formerly with Rebellion Developments Ltd, Oxford, UK.

[1] http://ccg.doc.ic.ac.uk/data/

Boon's "Conqueror, Manager, Wanderer, Participant" [2]. This research is intimately connected to the culture of game design, which is continually developing its own design-oriented theories, e.g. hardcore versus casual gamers. Recent work by Bateman et al. [7] gives a good overview of the area and argues that, as the field matures, it is following a general trend in the psychology of personality, where type-based theories are being superseded by more nuanced trait-based theories.

A. Gameplay-Based Player Models

Player models vary considerably in their nature and application. Smith et al. [8] provide a taxonomy of player models in terms of four facets: *scope* (individual-class-universal-hypothetical), *purpose* (generative-descriptive), *domain* (action-reaction) and *source* (induced-interpreted-analytic-synthetic). In this paper, we focus on relating descriptions of players to in-game actions. Yannakakis & Togelius refer to this as *gameplay-based* player experience modelling [9], and note differences in how players are described. We distinguish here between gameplay-based models of *behaviour*, *personality* and *experience*, although in practice the distinction may not be clear cut.

1) Behavioural models: These simply describe players in terms of their in-game activity. For example, Drachen et al. [10] applied emergent self-organising maps (ESOMs) to game data from *Tomb Raider: Underworld* to create a player typology. The study is notable for its scale and use of natural data: 1365 complete playthroughs of the game on XBox LIVE were collected remotely via the EIDOS metrics suite. Only six features were logged: completion time, use of help-on-demand, total deaths, and deaths caused by opponents, the environment and falling. Their ESOM analysis suggested four types of *Tomb Raider* player, named Veterans, Solvers, Pacifists and Runners.

More detailed records of in-game activity can also be used to build behavioural models. In an earlier study of LDA, Baumgarten conducted a study of 245 *Pac-Man* players, recording every player and ghost movement made during the game. Applying LDA to log data features showed the most distinguishing features were related to key press frequency, understanding of the game rules, and optimising play for high scores [11]. Thawonmas et al. used similarly detailed movement logs to cluster players based on the locations visited within a game [12]. In general, behavioural models can be used to reason about player experience, although this depends on an interpretation (which may be empirically-based) which maps patterns of behaviour to patterns of experience. Although generating behavioural models directly from data also has the potential to reveal significant patterns that have not been captured in any pre-existing conceptualisation of experience — an approach we explore in this paper.

2) Personality models: In contrast, some approaches use in-game activity to describe players in terms of some pre-existing general theory of player personality (a *model-based* approach [9]). The theory should make claims beyond the game being studied, but also be valid and relevant within that specific gaming context. Data on player personality is required to construct and/or evaluate the model. Reasoning about player experience requires some interpretation of how player personality affects player experience. For example, in a study of *Pac-Man* players, Cowley [13] used decision trees to classify Bateman and Boon's 'Conqueror type' players from log data, based on a typology which itself was inspired by the Myers Briggs Type Indicator. Lankveld et al. [14] looked at how game log data from players of *Neverwinter Nights* could be used to estimate players' personalities within the influential trait-based OCEAN model. Similarly, Yee and colleagues were able to estimate players' OCEAN profiles from detailed online statistics about their World of Warcraft gameplay [15].

3) Experience models: A third approach to gameplay-based player models is to directly relate specific player experiences to in-game activity without a mediating model (a *model-free* approach [9]). In this case, some predefined conceptualisation of experience is required which may be inferred from log data. As with personality models, this approach requires external data on player experience. For example, applying an artificial neural network to game log data, Pedersen et al. [16] directly modelled the player experiences of fun, challenge and frustration in a clone of *Super Mario Bros*, as reported by players via pairwise comparison. In contrast to behavioural or personality models, no further interpretation is required to reason about player experience.

B. Adaptive Games

In this paper we look at constructing descriptions of player style from data, and one possible application of this is adapting games to player style. Indeed, any form of adaptation implies a gameplay-based player model, even if it is not made explicit, and so we briefly survey the area here. Relating player style and in-game actions can support the development of adaptive games.

Various simple forms of adaptation have appeared in games for decades, but over the last few years there has been increased interest within academia. Thue et al. used choices in an interactive story to model players according to a typology of narrative-based styles of play: Fighters, Power Gamers, Tacticians, Storytellers and Method Actors. The model is then used to dynamically select content as the story develops.

Yannakakis and colleagues have been particularly active in the area of adaptive games. For example, in [17] an artificial neural network was applied to survey and log data to generate a model of player enjoyment of an augmented reality game for children. This was then used to adapt the game during play.

Beyond variation within a game, some authors have looked at automatically creating novel game content tailored to players. For instance, building on the player model developed in [16], Shaker et al. automatically generate platform game levels [18]. Hastings et al. log weapon use, and employ the data to dynamically evolve novel particle system-based weapons adapted to the player [19].

Live adaptation in commercial games typically performs difficulty scaling, but recently more diverse adaptive mechanisms have appeared. Here we mention three recent examples:

a) Left 4 Dead and *Left 4 Dead 2* (Valve Corporation): These are examples of adaptation by dynamically changing opponent appearance frequency. Depending on the accuracy, health and rate of progress of a players' party, groups of enemies (several types of zombies) appear in more or less dense clusters, boss enemies are triggered. *Left 4 Dead 2* also modifies terrain features in certain levels, such as the path through a graveyard, according to similar player and party metrics.

b) Darkspore (Electronic Arts/Maxis Software): A so-called 'AI Director' is used to adjust game features such as level selection, spawn types, spawn placement, spawn amount, enemy health and damage, and music [20]. These features were adjusted according to the rate of progress and performance of the player.

c) Silent Hill (Konami/Climax Studios): Psychological profiling of the player is integrated into the game [21]. The main antagonist is a psychologist 'testing' the player, and the designers took inspiration from the psychological literature to adjust aspects of the game, based on the player profile.

These and other examples indicate that there is growing interest in player modelling and adaptive games within industry, and that companies are experimenting with a more personalised gaming experience.

C. Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a statistical data mining technique that distinguishes between c classes of objects in an N-dimensional feature space by computing a series of $k \leq (N-1)$ linear discriminants — linear combinations of features which represent directions in the feature space — whose values can be used to characterise the classes [3]. LDA is similar to Principal Components Analysis (PCA) [4] in that it selects directions which maximise feature variance, and so describe the "most important" variations in the data. Unlike PCA, in LDA the class labels are also used: it finds directions which best separates the class means relative to the sum of the class variances along that direction. It maximises the ratio of *between-class scatter* to *within-class scatter*. Intuitively, it finds lower dimensional descriptions of the data which push the class members together, and pulls members of different classes apart.

Formally, we have a set of c classes $C_1, \ldots, C_c$, where class C_i is represented by a sample of n_i objects (N-dimensional feature vectors). Denoting the sample mean for class C_i by $\boldsymbol{\mu}_i$ and the total sample mean by $\boldsymbol{\mu}$, we define the between-class and within-class scatter matrices S_B and S_W as:

$$S_B = \sum_{i=1}^{c} n_i(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T$$

$$S_W = \sum_{i=1}^{c} \sum_{\boldsymbol{x} \in C_i} (\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^T$$

LDA attempts to find a matrix A which maximises the following objective function

$$\frac{det(A^T S_B A)}{det(A^T S_W A)}$$

The column vectors of A correspond to the required $N-1$ linear discriminants. In practice, these can be computed by finding solutions to the following generalised eigenvalue problem using standard methods:

$$(S_B - \lambda S_W)\boldsymbol{x} = \boldsymbol{0}$$

The k linear discriminants correspond to the $k \leq (N-1)$ eigenvectors ordered by eigenvalue. The discriminants can be used to classify new objects or, as in this paper, for dimension reduction, where the first $k' \leq k$ discriminants are used to project the data to a reduced k'-dimensional space which can describe important variations between classes. In this paper, we use the R package MASS [22] to compute discriminants.

The design of LDA makes two assumptions to guarantee the optimality of discrimnants: 1) *multivariate normality*, that any linear combination of features is normally distributed; 2) *homoscedasticity*, that the classes have equal covariance matrices. Despite the risk of suboptimal results, LDA has been widely used for classification and dimension reduction in contexts where these assumptions are violated. The effects of non-normality e.g. [23], [24] and unequal covariance matrices e.g. [25] are complex and have been widely studied. Ashikaga & Chang [26] argue that similarity in the shape of class distributions is more important than normality. It is beyond the scope of this paper to survey this complex area, except to note that 1) it is generally accepted that small deviations are not problematic, and 2) more serious deviations should make the researcher cautious in interpreting and applying results, and the results should be checked with other research methods.

In this paper, we use LDA to generate hypotheses about player style, and advocate the use of LDA in conjunction with other game design methods to ensure validity.

III. Study 1: Snakeotron

Arcade-style action games are suitable candidates for studying player style because of limited input dimensions, a limited state space, and ease of modification for data collection. To investigate learning about player style from gameplay data, we have created a new arcade-style action game inspired by existing arcade games such as *Snake* and *Tron*. These games are similar in that the player-controlled character draws a trail behind him that he is not allowed to cross, and ideas from both are incorporated into *Snakeotron*. It has a pool of game rules that can be added to or substituted to form a customisable game.

Here we look at whether LDA be used to identify important variations in player style in Snakeotron. We use LDA to analyse player data from an internet-based study with *Snakeotron* using pre-defined sets of rules. We also compare the results to an earlier survey where participants played *Pac-Man*.

In future work, we intend to adapt the game rules to different player styles. The ultimate goal is to create an adaptive *Snakeotron* that automatically selects a subset of game rules deemed optimal for the current player's style of play. A simple arcade game was chosen partly because the rules are relatively easy to modify while retaining playability.

Snakeotron is a 2D arcade action game where the goal is to collect all the green balls on the screen. Players can

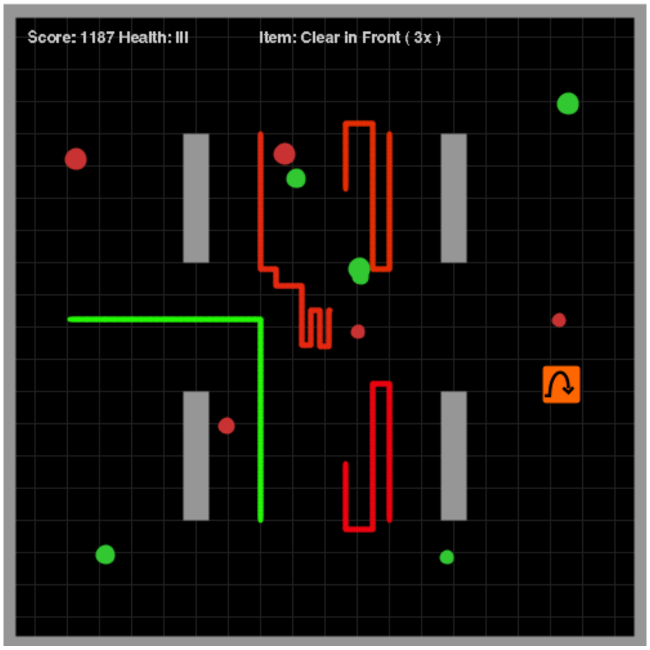

Fig. 1. A screenshot of *Snakeotron* with the player character (green line, bottom left), AI controlled opponent characters (red lines), collectable targets (green balls and orange square), and obstacles (red balls and grey walls). The goal of the game is to collect all green balls without losing by touching obstacles or opponents.

collect power-ups to alter the behaviour of their character, and should avoid hitting any other obstacles such as other computer controlled snakes, red balls and walls. An average match (i.e., a single play-through where the player collects all the green balls within a given rule set, or uses up all their lives trying to do so) lasts 30–90 seconds, depending on the given subset of rules, which are described below. The game has been implemented in Processing, connected to a MySQL database for data capture[2]. A screenshot of a typical game scene is shown in Figure 1.

A. Pilot Study

We conducted a pilot study with *Snakeotron*, involving nine participants, in order to establish basic parameters for the main studies. Feedback from the pilot study participants led us to tweak some parameters, in particular we shortened the overall survey duration to 8 minutes and the individual match duration to 30–90 seconds, clarified the post-match questionnaire, and improved the introductory help screen. A neutral option for a comparative question in the questionnaire (see also subsection III-B) was requested, however, after careful deliberation, we decided not to provide a neutral option in order to force a decision. We argue that the results would even out when a large part of the users were undecided over a choice of games.

B. Data Capture

Snakeotron was set up to capture all player actions, so that a full replay of the match was possible. Additionally, cumulative metrics were recorded to simplify data analysis. The selection of these metrics is related to metrics that have been established in previous surveys [11], [27] to allow for comparable classification. An internet-based study was conducted to measure player participation, engagement and behaviour. To facilitate this, a website was set up to introduce the participant to the game and explain the survey structure. Each player was asked to play a sequence of games, i.e., matches of *Snakeotron* with a subset of game rules. The chosen subset of game rules is different for each match for a given player, but the same for different players. In other words, the nth match in the sequence that any participant plays always has the same set of rules, but no match in the sequence has the same set of rules.

Each participant was asked to play at least 10 matches in at most 7 minutes. These values were established experimentally in the pilot study mentioned in subsection III-A. After each match, the player was asked to evaluate their enjoyment with two questions:

- *How much fun did you have?* Rated from 'very little' to 'a lot' in 6 steps. This question attempts to glean an absolute view on the enjoyment the player experiences.
- *Which of these two previous games did you enjoy more?* Given with a screenshot of the last two matches and a choice of 'a little more fun' 'more fun' and 'a lot more fun' for both matches (i.e., 6 choices), the player is asked to quantify his enjoyment in a relative manner.

When the end of the survey was reached, the player was given the option to keep on playing in order to provide more data points. All information the player provided was anonymised and remotely stored in a MySQL database.

The game survey involved 215 unique players with 1450 matches. After completion of the survey, we selected a subset of players according to minimum requirements: 1) Three or more complete matches were played; 2) Match duration of 10 seconds or more. Less time than this indicates deliberate losing or no input, i.e., the character only goes straight ahead and loses; 3) No missing data due to connectivity issues or closed sessions. This resulted in 74 complete data sets of players with 3 or more matches each.

C. Log Features

From the recorded survey game data, we computed 16 log features that summarise each match. The metrics cover all aspects of the game, and were collected during each match and transmitted to database:

1) Total time: The total time of the match in seconds.
2) Score: The accumulated score, as a sum of points awarded / subtracted for game events.
3) Distance: Total distance travelled during the match.
4) Walls hit: The number of turns where a static grey wall has been hit.
5) Opponent hit: The number of turns where a red opponent tail has been hit.
6) Owntail hit: The number of turns where the player hit his own green tail.
7) Other hit: The number of turns where another obstacle has been hit. In the current version of the game, these are only red balls.

[2]Available at http://aipanic.com/game2/index_no_fb.html (March 2012)

8) Turns cornering: The number of turns spent cornering (i.e., where a the press of a direction key has led to a change of course of the player character).
9) Low corner distance: The average distance to an obstacle that a player would have hit if he had kept going in a straight line, when he is turning. This is only counted if that distance is lower than $1/10$th of the game world width, in our implementation, this is 50 pixels.
10) Low corner count: The number of turns where a low corner distance is recorded.
11) Right corners: The number of turns the player spent going in a right corner.
12) Left corners: The number of turns the player spent going in a left corner.
13) Turns with useless keys: The number of turns the player spent pressing a direction key that did not lead to a change of direction of the player character. This happens if the character is already travelling in the direction of the direction key.
14) Keys pressed: The number of turns the player spent pressing down a key. Repeat keystrokes that might happen from keeping the key pressed down for an extended period of time are not counted.
15) Right turns minus left turns: A composite metric where we subtract the number of left turns from the number of right turns. The pilot study revealed that players tend to prefer one direction over the other and go in a spiral pattern. This feature was introduced to see if this would be an important factor in player classification.
16) Score per distance: A composite metric that records the score per travelled player distance.

Note that a (game) *turn* is an atomic time step in the game, and that there are 30 turns per second.

D. Applying LDA

We applied LDA to the *Snakeotron* match data, labelling each match with the identity of the player. Hence each player has a distinct class of feature vectors describing their matches. As covered in Section II-C, LDA uses a linear base transformation to minimise in-between class variance and maximise between-class variance. In other words, we are attempting to highlight stylistic commonalities between individual player's matches and differences between matches from different players.

To maintain comparability of the data vectors of each match, the features were made time-independent by dividing each value by the total match time. This allows easier comparison of very short and more extended matches, which are bound to happen with differing rule sets. The feature data collected in the *Snakeotron* survey is not multivariate normal nor homoscedastic. As discussed in II-C, the application of LDA on this data can still provide useful results. All features were found to have a roughly log-normal distribution.

In Figure 2, we present a plot of the first two dimensions of the LDA-transformed space. The weights of the features in the first dimensions of the LDA space indicate more important features that determine the behaviour of a player and how it differs from other players. One positive side-effect of this method is that unimportant features are penalised automatically. Furthermore, it is not harmful if features are correlated, which is likely to happen as we look at rates of player actions such as turns per minute and keystrokes per minute.

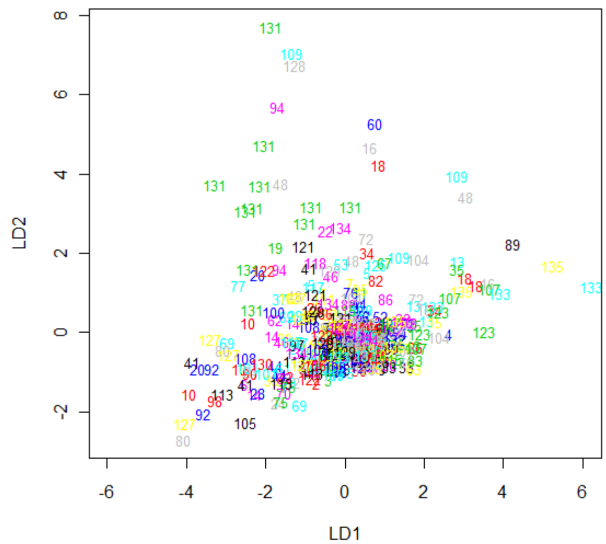

Fig. 2. *Snakeotron*: The first two LDA dimensions showing all matches of all players (number and colour indicates player).

TABLE I
INFLUENTIAL COEFFICIENTS FOR LD1-4 IN SNAKEOTRON

LD	Trace	Positive coefficients	Negative coefficients
1	0.18	Low corner dist (0.86) Keys pressed (1.01)	Turns with useless keys (-0.82) Low corner count (-0.69) Score / distance (-0.52)
2	0.16	Walls hit (1.15) Score (0.55) Keys pressed (0.55)	Score / distance (-0.52) Low corner dist (-0.51)
3	0.11		Turns with useless keys (-0.74) Turns (-0.64) Keys pressed (-0.56) Walls hit (-0.54) Score (-0.51)
4	0.08		Score (-1.1) Opponents hit (-0.85) Walls hit (-0.85) Turns (-0.78) Low corner count (-0.54)

E. Interpretation

To understand what a discriminant represents, we look at the linear coefficient for each feature. Table I shows the most influential coefficients of the first four discriminants. Interpretation depends on the relative values of the coefficients, and here we report features over a 0.5 threshold.

The weights of the first dimension (LD1) are dominated by the frequency of keys being pressed and frequency of turns with useless keys. This means that the rate of physical interaction with the computer is one of the most important factors to classify players. The number and average distance of low corners also have a high weight in LD1. As described in subsection III-C, *low corners* signifies events where an obstacle is nearby and the player decides to turn. Summing up and taking the sign of the weights into account, a high value for LD1 indicates a player that likes to keep their distance from obstacles (high low corner distance and not a lot of low corners in total), which also leads to longer paths and a low score per distance. To achieve this, the player corners a lot.

The second dimension LD2 is mostly determined by the amount of walls hit. Hitting a lot of walls is the most common way to lose the game (other ways to lose include hitting computer controlled snakes or red balls), and thus is an indicator of skill in *Snakeotron*. A high weight for low cornering distance ties in with hitting walls, because if this distance is low, the player is spending more time in close proximity to walls. Other important weights for LD2 are score per distance and – with an opposite sign – score per time. This will separate players who travel a long distance to score high from those who achieve high scores in a short distance. As the player is penalised for hitting walls, more wall impacts will lead to a lower score per distance. Overall, a high value in LD2 means that the player fails to avoid walls frequently, and thus receives a lower score per distance.

A high value in LD3 indicates that the player did not do much, i.e., not many normal or ineffective key presses, thus not many turns, not many walls hit and a low score. All these values are relative to the time played, which means that the match was likely stretched out by careful yet sparse steering commands.

A low value in the fourth dimension LD4 means a high score, a lot of opponents and walls hit, and a lot of normal and low distance turns. This indicates a successful yet risky playing style, where the player accepts damage to reach the goal (of collecting all the green balls).

Considering this, we can say that linear discriminant analysis of the playing data for *Snakeotron* indicates that interaction intensity metrics (frequency of key presses) are most likely to characterize player behaviour, followed by playing skill at wall avoidance and turning style (high / low turn frequency).

F. Discussion

Applying LDA to *Snakeotron* player data has identified some meaningful variations in player style. The findings here are reinforced when we compare the results of a previous study of *Pac-Man* [11]. In that study, we took the data of 245 players with 5 matches of *Pac-Man* each and ran a linear discriminant analysis on similar metrics as described above. We reported a comparable interpretation of the most significant metrics to classify player behaviour: physical interaction with the game interface, and secondly main game ideas. Furthermore, we can also see parallels when identifying risky versus risk-averse playing styles. In *Snakeotron*, this is indicated by the fourth LD dimension (accepting damage for a higher score).

In future work, the above *Snakeotron* analysis will form the basis of a fully adaptive game, where the selection of a rule set that is custom tailored for a player is based on a classification of playing data by LDA. As the games become more complex, we can also see challenges for automated data analysis, namely:

- A diverse rule set: Unlike *Pac-Man*, *Snakeotron* has a more diverse game environment between each match, which entails a larger variation in playing behaviour. We expect this variation to increase with more complex games that allow the player more freedom, both in control and in gameplay decisions.
- Unknown core game idea: *Pac-Man* is one of the most iconic arcade games and very well known. Any learning affects are going to be smaller, while these are likely to be large with novel game concepts.

In *Snakeotron*, we've already seen a less clear-cut classification that required a somewhat more complicated interpretation than with the Pac-Man studies. We are currently planning more studies with *Snakeotron* and another game that will tackle these issues.

Fig. 3. A typical screenshot from the first level of *Rogue Trooper*. The player controls Rogue (centre foreground). On-screen displays show nearby NPCs (bottom left) and ammunition and health (bottom right).

IV. Study 2: Rogue Trooper

Although applying LDA to log data from arcade games like *Snakeotron* can give us insights into player style, a key question for wider commercial application is: does this approach generalise to more complex gameplay? To show that it does, we describe the application of LDA to log data from *Rogue Trooper* (Eidos/Rebellion, 2006), a modern commercial third-person shooter video game.

In *Rogue Trooper*, the player must navigate a soldier through a complex 3D landscape and defeat a series of enemy soldiers in combat. Figure 3 shows a typical screenshot from the first level. On release, the game was well received by critics, and noted for its relatively complex and flexible gameplay involving a variety of combat and stealth elements [28].

Compared to arcade games, this kind of rich gaming environment allows and expects players to engage in a far more complex range of interconnected and overlapping activities.

While studying *Rogue Trooper*, amongst other activities, we observed players preparing for and engaging in combat, checking combat had finished, abandoning combat, navigating to and exploring locations, getting lost, retracing their steps, investigating the controls and game mechanics, even admiring the scenery.

Considering the potential for diverse activities within a level, it is clear that any metric — say, mean ammunition level — applied to the log data for an entire level will be significantly affected by the exact sequence of activities undertaken. When it comes to comparing different level logs and different players, the utility of such general metrics is therefore severely limited, as it is very unlikely that a similar series of activities are being compared.

Hence a key challenge to applying machine learning techniques to complex gameplay is to identify which sections of each log can be extracted and meaningfully compared using metrics. Below we look at extracting combat log data to model player style, and describe a general method for extracting combat data from shooter game logs (Section IV-B). This allows us to apply LDA to *Rogue Trooper* log data (Sections IV-C to IV-D), and then to group the players using k-means clustering (Section IV-G). First, we describe how the log data was obtained.

A. Data Capture

In collaboration with Rebellion Developments Ltd, the developers of *Rogue Trooper*, the PC version of the game was customised so that it recorded log data to a local XML file. The game logged a single game frame every 0.2 seconds, which in our study corresponded to roughly one frame in every seven. For each logged frame, various data about the player character (PC) and every other currently spawned entity (NPCs) within a set distance was recorded: the entity's position and orientation in 3 dimensions, health, selected weapon and weapon state (firing, reloading etc.). Other aspects of the PC's state were recorded, such as ammunition level and current animation, along with the current AI state for NPCs. The game also logged certain events independently of the regularly timed frame: PC actions such as grenade or health pack use, sources of damage to the PC, objective completion, and interruptions to gameplay such as cut scenes, pop up messages and player-controlled pauses.

To collect the log data for this study, 32 participants were asked to play the first level of *Rogue Trooper* in return for a small monetary reward. The level consists of a fairly linear series of combats designed as an introduction to the game controls and mechanics. It introduces the use of the rifle, sniper rifle and grenades, and the use of cover.

Participants were required to play for a minimum of 20 minutes, but could complete the level more quickly or choose to play for longer. As well as being logged by the game, screen video was captured using FRAPS[3]. The first player (P1) piloted the data capture setup, and three players (P2, P4 and P5) were asked to play the first level twice — resulting in 35 level logs. Apart from the pilot, players were additionally asked to rate their overall immersion in the game on a scale from 1 to 10 (taken from [29]). Further questionnaires and interviews on player experience were also completed by players, but as the focus of this paper is on log data analysis, we do not report them here.

The participants were predominantly male (88%), between the ages of 18 and 35 (91%), had been playing video games for more than 5 years (88%) and played at least once a month (75%). The majority completed the level (81%), with 6 players giving up part way through. Overall, they completed over 10 hours of logged gameplay in 35 level attempts, with the average attempt taking 18 minutes. XML log data was imported into the R statistical computing environment for analysis [30].

B. Extracting Combat Data

After data capture, the next stage was to identify those sections of the game logs which corresponded to player combat. As discussed above, our aim was to allow meaningful comparison between players by extracting sections which represented a single activity and excluding other activities as much as possible. We focused on combat, as it is the most important activity in *Rogue Trooper*. Further analysis could include other activities, such as navigation around the level [12]. For other games and genres, activities such as problem solving may be more relevant for modelling player style.

In *Rogue Trooper*, as is often the case in the shooter genre, the player encounters a series of NPC groups which need to be engaged in combat in order to progress. We can therefore define each individual combat as an interaction with a known group of NPCs during a single PC life. If the player engages a given NPC group during a given life, then we can use the NPCs' AI state to identify an 'active period' within the log data for that life. All combat data is contained within the active periods for the various NPC groups.

For a given NPC group and active period, we can identify the start of combat as the moment the NPCs first fire on, or receive damage from, the player. Hence for each NPC in the group, we define α as the first frame in which their weapon is fired, and β as the first frame they receive damage. We then define α (resp. β) for the entire group as the earliest α (resp. β) frame for the group members. Combat between the player and the NPC group is identified as starting at whichever frame α or β is the earliest. We assume that the end of combat is the end of the NPC group's current active period, which indicates the death of the PC, the death of the entire NPC group, or the completion or quitting of the level.

Some features have different interpretations depending on whether the PC survives or dies. For instance, considering combat time, a quick success is very different from a quick death. Hence we restricted our analysis to combats in which the PC survived.

Based on our observations of players, we know that players tend to continue engaging a group of NPCs until the PC or the group is dead. However, this is not always the case. It is possible for a combat to start and then be abandoned when the PC moves away from the NPC group. This could be because

[3]http://www.fraps.com/

the player explicitly chose to end the combat, or because they were not aware that it had started. Such *abandoned combats* distort the feature data, and need to be excluded where possible. One sign of abandonment is that both the PC and at least one NPC survive, and we excluded these combats. We attempted to identify further cases by investigating outlier values and screen capture videos.

An alternative approach would be to identify abandoned combats by looking for periods without weapon use, and increasing distance from and lack of orientation toward the NPC group. However, from observing players, we estimate only a fraction of combats are abandoned, and it is not clear whether satisfactory criteria could be framed to exclude only abandoned combats.

Applying the α-β model to our 35 level logs gives 550 individual combats. In 88 the PC died, in a further 31 at least one NPC survived, and one further abandoned combat was discovered from reviewing the study data. The remaining 430 combats were used to compute the feature data.

C. Combat Features

To compare combats across players, a number of metrics for the combat log data were devised. The process of designing a feature set initially involved reviewing the screen capture video data and observing how individual players varied in their combat behaviour. Visualisation of the log data was then used to relate behaviour to measurable features. The resulting list was then pruned of redundant features — those correlated very highly with another feature — and of features with extremely low variance, which are not useful for distinguishing players. This produced a list of 21 combat features.

The feature definitions rely on the concept of the "first weapon use" (α frame) and "first damage received" (β frame), as described in section IV-B. Either frame may be undefined for a given NPC or NPC group. Given an single NPC's active period which ends at frame ω, we define four values for that NPC:

- postfire — elapsed time after weapon use, measured from α to ω. If α is undefined (NPC did not fire) this is zero.
- postdam — elapsed time after damage, from β to ω. If β is undefined (NPC was not injured) this is also undefined.
- npc.fire — the proportion of the frames α to ω inclusive in which the NPC was firing.
- final.dist — the distance this NPC was from the PC at ω.

Using these values, we can now define the 21 combat features. For an NPC group that has an active period from γ to ω, and possibly frames α and β (defined as the earliest such frames for its group members) we define six features: :

1) prefire — elapsed time before weapon use, from γ to α. If α is undefined (no NPC fired) then this is taken to be the entire active period γ to ω.
2) npc.postfire — elapsed time per NPC after first weapon use, defined as the time from α to ω divided by the NPC group size. If α is undefined (no NPC fired), then this is undefined.
3) npc.lead — elapsed time from attacking to receiving damage, measured from α to β. This may be negative or undefined.
4) mean.postdam — mean postdam value for the NPCs.
5) mean.npc.fire — mean npc.fire value for the NPCs.
6) mean.final.dist — mean final.dist for the NPCs.

The remaining combat features are based on the PC's log data during the same active period γ to ω. Most measure the frames from α to ω, as α is more likely to be defined than β (it is more likely that at least one NPC fired than at least one was injured). For the period α to ω we define:

7) firing — the proportion of logged frames in which the PC was firing.
8) pistol— the proportion in which the pistol was selected as the current weapon.
9) sniping — the proportion in which the sniper scope was in use.
10) move — the proportion in which the PC was moving.
11) cover — the proportion in which the PC had taken cover.
12) gren.rate — the number of grenades used per minute.
13) mean.aim — the elapsed time per grenade spent in grenade aiming mode.
14) dist.rate — the distance covered per second.
15) turn.rate — radians turned per second around the yaw axis.
16) area.rate — the area of the convex hull enclosing the PC's path, divided by the elapsed time.
17) zero.ammo — the proportion of logged frames in which the PC has no ammo.
18) mean.ammo — the mean PC ammunition level.
19) pre.ammo — the pre-combat ammunition level, defined by the frame before α.
20) mean.health — the mean PC health level.
21) dam.rate — the health points lost per second.

All time-based metrics were adjusted to exclude interruptions, i.e. time the player spent watching cut scenes, pausing the game or reading pop-up messages.

D. Applying LDA

Applying the 21 features to all 430 combats gives 9030 feature values. Of these, 14 (0.16%) were undefined because of undefined α or β frames. As standard LDA does not accept missing values, these were replaced with zero. Another 7 values (0.08%) were identified as extreme outliers. These were capped at the upper end of their 'typical' range, to prevent them having an undue influence on the analysis.

Following the method illustrated in the *Snakeotron* case study, our aim was to represent each player with a group of combats in the high-dimensional feature space, then reduce this via LDA to a low-dimensional space. The LDA space attempts to separate different players' combats and bring together combats by the same player. Recall that we are using a supervised learning technique with a set of class labels (one class per player) to perform unsupervised learning of a set of player style traits.

Including all the successful combat instances for each player meant the classes were quite unbalanced — between 9 and 36 combats per player classes. Unbalanced group sizes do not necessarily have a negative affect on the performance of LDA [31]. However, the excess combats correspond to parts of the level where a player has died and then had to complete a combat again. Hence certain combat scenarios will be over-represented, biasing the analysis to the style of play they induce in that player. To balance the number of combats per player, we represented repeated combats between a player and NPC group with a 'mean combat instance' based on the mean of their feature values. This resulted in between 9 and 12 mean combats per player. This feature data was scaled and centred to standard scores before LDA was applied.

It is important to note that the *Rogue Trooper* feature data is multivariate nonnormal and heteroscedastic (i.e. player covariance matrices are unequal), and so does not conform to the assumptions of LDA. As discussed in Section II-C, LDA can still provide useful results in such circumstances. None of the features are univariate normal, and sample distributions vary considerably, but are largely similar across players. Many features are zero-inflated, corresponding to situations where, for example, the PC can either remain still (zero value), or move a distance according to a particular distribution (non-zero value). Some features are closer to log normal than normal, although following Chinganda & Subrahmaniam [32] we chose not to log transform, as this does not achieve normality.

We assessed the player covariance matrices by comparing their log determinants, a measure of total variance. Because some were singular, we ignored some 'low variance' features which had only zero values for some players (pistol, sniping, zero.ammo, cover, gren.rate, mean.aim). For the remaining 15 features, 21 players had very similar log determinants, with the remaining 11 being split between two groups of similar log determinants. This suggests three subgroups of players with roughly equal covariance matrices, with further inequality being contributed by the six 'low variance' features.

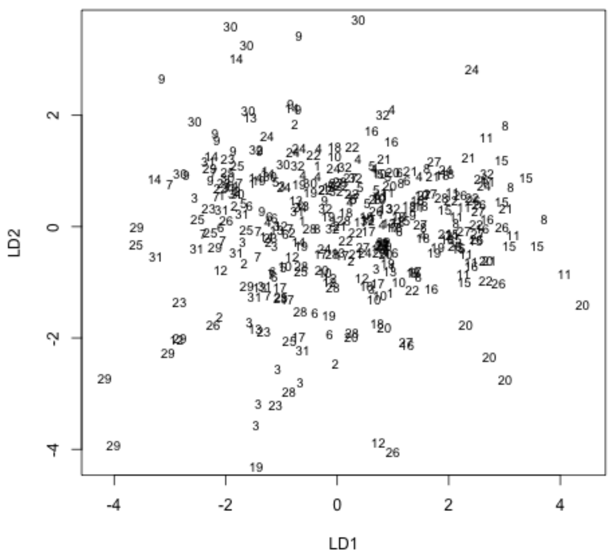

Fig. 4. Mean combats in the first two dimensions of the LDA space. Note that the number indicates the player, with each of the player's data points representing the mean feature values for all their combats versus a specific NPC group.

E. Interpretation

Figure 4 shows the mean combats for each player plotted using the first two linear discriminants LD1 and LD2. Each number indicates a player, and each data point represents the mean feature values for all the combats that player fought against a particular NPC group. Figure 5 shows the player centroids (mean positions) in the LD1-LD2 space, defined as the mean feature values across all NPC groups they defeated.

Table II gives the proportion of trace for the first three discriminants, i.e. the proportion of between-player variance accounted by this dimension. LD1 accounts for 40% of the variance, the next two for about 10% each, and subsequent discriminants for 7% or less each. By just retaining the first three discriminants, we can account for about 60% of the variance between players.

As with *Snakeotron*, to interpret the LDA results, we examine the coefficients for the standardised features. The most influential feature coefficients for the first 3 discriminants are given in Table II. Because of the smaller sample and more complex data, coefficients are smaller than for Snakeotron, and we use a lower 0.3 threshold. Higher discriminants (LD4+) contribute less and are harder to interpret.

The first discriminant (LD1) is highly positively influenced by PC movement and use of cover. Scoring high on LD1

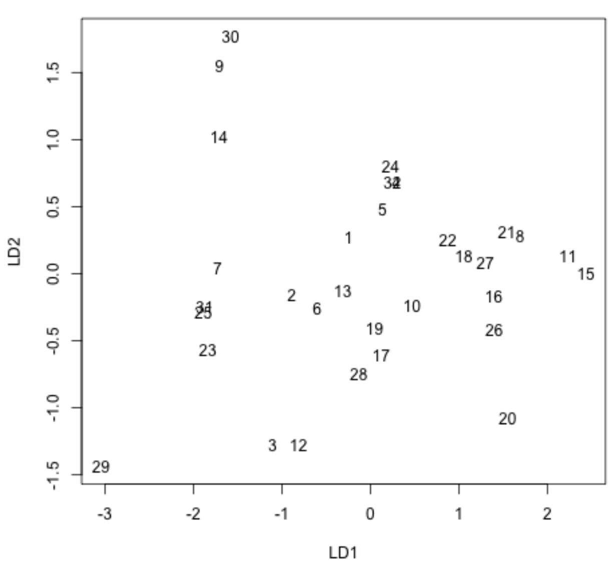

Fig. 5. Player centroids (mean position) in the LDA space

TABLE II
INFLUENTIAL COEFFICIENTS FOR LD1-LD3 IN *Rogue Trooper*

LD	Trace	Positive coefficients	Negative coefficients
1	0.40	move (1.37) cover (1.16) sniping (0.38) firing (0.38) npc.lead (0.33) mean.health (0.31)	npc.postfire (-0.33)
2	0.11	mean.health (0.96) mean.postdam (0.50) firing (0.39) mean.aim (0.39) mean.final.dist (0.39) prefire (0.31)	npc.postfire (-0.52) sniping (-0.39) dist.rate (-0.33) move (-0.30)
3	0.10	pistol (0.47) npc.postfire (0.40) move.(0.33)	mean.ammo (-0.63) area.rate (-0.43) sniping (-0.38)

may indicate that the player is *actively defensive* in combats, moving and taking cover, whereas low LD1 players tend to be more static and exposed. High LD1 scores may also be due to high weapon use (firing and sniping). This suggests that LD1 distinguishes 'combat dynamism': active attackers/defenders (high LD1) versus more passive players (low LD1).

The coefficients tell us which features *may* contribute to a discriminant value, and hence factors that can lead to a high or low value. We can also look at correlations between features and discriminants to see which are *always* associated with a high or low value. LD1 is moderately correlated ($\rho = 0.60$) with the move feature, indicating movement is a fairly good way to distinguish high and low players.

LD2 is highly positively influenced by mean.health, and also correlates with it ($\rho = 0.65$) . High LD2 can also be the result of a high mean.postdam and/or a low npc.postfire: high LD2 players take longer to finish off individual NPCs, but take less time overall. Conversely, low LD2 players take longer to deal with a group, but individual NPCs don't survive damaged for as long. This could indicate combat strategy: engage-the-group (high LD2) versus engage-single-NPC (low LD2). This combination of health and strategy suggests LD2 is contrasting more cautious players (high LD2) with more reckless ones (low LD2). This is consistent with the other influential coefficients: reckless players move more (move and dist.rate), while cautious ones fight at a distance (mean.final.dist) and take longer to engage the enemy (prefire) and aim grenades (mean.aim).

LD3 can be interpreted in terms of ammunition management: it is negatively influenced by, and negatively correlated ($\rho = -0.68$) with, mean ammunition and weakly negatively correlated ($\rho = -0.54$) with the ammunition level at the start of combat (pre.ammo). It is also influenced by weapon use: positively by use of the pistol and negatively by the sniper rifle. This link between ammunition and weapon choice may be explained by an not uncommon situation in the first level, where the PC ran out of ammunition and the player switched to the pistol, which has infinite ammunition. Many players did not realise more ammunition was easily available from an allied NPC, and continued to use the pistol through parts of the level often associated with the sniper rifle. This is essentially a lack of player knowledge that could be mitigated by a minor redesign, e.g. making the ammunition source more obvious. The discriminant is contrasting those players who often had low levels of ammunition (high LD3) and, perhaps through this common error, favoured the pistol (high LD3), with players who maintained higher ammunition levels, and who avoided the error (low LD3).

In summary, the first three discriminants provide us with a model of player style which distinguishes between active and passive players (LD1), cautious and reckless players (LD2), and ammunition management (LD3). We hypothesise that combat dynamism (LD1) is a measure of experience in shooter games — constant movement and use of cover are often signs of an experienced player. We propose dynamism is a measure of style which, although it may be related to combat performance, is distinct from it. An active/'experienced' style of play does not guarantee success.

LD3 highlights a flaw in the level design related to player awareness of the game mechanics. This is of less relevance to player style, but shows the utility of this approach in a game design context.

Returning to Figure 5, we can see that players are distributed in a roughly triangular area: in general, the variation in LD2 increases as LD1 increases. In the light of our interpretation, this suggests a further hypothesis that players who actively attack and defend (which we propose are experts in this genre) are more consistent in their level of caution, whereas the more passive players (novices) are more likely to be very reckless or very cautious players.

F. Discriminant Validity

The validity of the *Rogue Trooper* LDA (and our interpretation) can be tested by comparing the discriminants with external metrics of player behaviour and experience. Here we briefly look at some end-of-level statistics, player demographics and a single retrospective question on immersion [29]. The relationships we identified are reported below, and are consistent with the interpretation given above. Note, however, that a *post hoc* search for relationships is highly likely to identify non-significant effects. Testing these relationships with additional data gathering would provide better evidence for our model and interpretation.

d) Level statistics: We examined a number of end-of-level statistics logged by the game, some of which are presented to the player so they can get feedback on their performance: time played, deaths, shots fired, hits, damage taken, damage inflicted, head shots and tank shots. The latter two measure hits to enemy soldiers' head or oxygen tank, both of which cause instant death and are therefore highly valued. Deaths were normalised to deaths per hour, and head and tank shots were normalised to the number of enemy NPCs killed, i.e. the probability of a head or tank kill. All other metrics were normalised to per minute measures. We also looked at player accuracy (hits per shot) and efficiency (damage inflicted per damage taken).

LD1 correlates negatively with time played ($\rho = -0.78$), indicating the more active players complete the level more

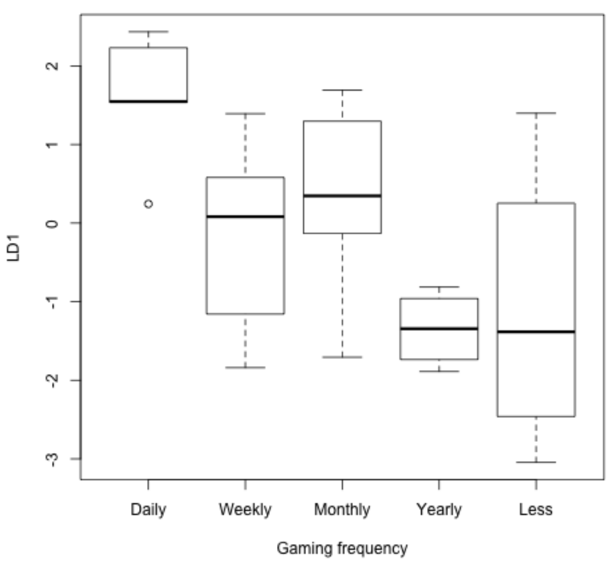

Fig. 6. LD1 by player's gaming frequency. The box-and-whisker plot displays the median value as a thick black line and the interquartile range (IQR) — the middle 50% — as a box. The 'whiskers' extend to the most extreme data point that is within 1.5 IQR. Beyond that, outliers are plotted individually.

quickly than passive players. LD1 is positively correlated with some measures of being active in combat: shots per minute (0.69), hits per minute (0.80), accuracy (0.46), and damage taken (0.58) and inflicted per minute (0.80). The association with taking damage, and the weakness of the association with accuracy, suggests that LD1 is measuring combat activity rather than skill.

In contrast, LD2 is only correlated with efficiency (0.64). This is consistent with the discriminant measuring player cautiousness, as cautious player are likely to take less damage for each enemy they kill.

LD3 is not notably correlated with any of the level statistics. None of the level statistics are directly related to mean ammunition levels, so this fits with our interpretation.

e) Player demographics: Some basic demographic data was collected about each player: age (7 levels), gender, length of gaming experience (5 levels), gaming frequency (5 levels), and a list of their favourite games. Using genre classification data from Wikipedia and Metacritic, these games were classified as either *shooter* or *non-shooter*. Players who chose at least one shooter game were classified as shooter players. Gender was ignored, as only four female players took part.

We carried out a visual inspection of the distribution of the first 3 discriminants across subgroups. Player LD1, LD2 and LD3 are roughly normally distributed, but variance was not equal across subgroups. No notable differences in means were observed for age and experience.

Figure 6 shows LD1 for 5 levels of gaming frequency. More frequent gamers tend to have higher LD1. The differences in group means were significant at the 0.01 level (ANOVA, $p = 0.005$). For genre preference, shooter players had a higher mean LD1 (1.18) compared to non-shooter players (-0.87), which was found to be significant (Welch's t-test, $p < 0.001$). These results support our hypothesis that high LD1 players have higher expertise in shooter combat, although further data collection is needed to provide more substantial evidence.

f) Self-rated immersion: Immediately after play, players were asked to self-rate their immersion in the game on a scale of 1 to 10 (after [29]). No correlations were found between the discriminants and immersion scores. Taken at face value, this is not surprising. Any relationships between our style metrics and player experience are unlikely to be simple correlations.

G. Player Clustering

Both game designers and adaptive games can use a low-dimensional LDA space directly as a model of player style, but in some situations it is more convenient to reduce this continuous representation to a discrete one. Designers may want a typology of players to inspire their work, or the developers of an adaptive game may want a discrete classification to trigger alternate game content. In such cases, we can apply clustering techniques such as k-means to group players into a set of player types. For simplicity, we restrict our cluster analysis to the first two discriminants LD1 and LD2.

To apply k-means, a decision on the number of clusters (k) needs to be made, either by an analyst or by automatic methods. One method is to examine the within cluster sum-of-squares (WSS) for different values of k, a measure of variance within all the groups, with lower values being preferred. Figure 7 shows average WSS over 100 attempts to find k clusters of players in the LD1-LD2 space. The smooth decrease in WSS as k increases shows there is no obvious choice of k for this data, although a slight kink at $k = 2$ suggests splitting players into two types. However, for $k \leq 3$ the cluster solutions merely differentiate players along the LD1 axis. Instead, with the aim of generating an interesting typology, we choose $k = 4$, as it is the lowest k solution which takes account of LD2.

Figure 8 shows a 4 cluster solution using k-means. This is the lowest WSS solution generated in 10,000 runs of k-means ($k = 4$), and was found in 94% of runs. To help characterise the player types, we have named the clusters **Hyperactive** (highly active), **Normal** (moderately active and cautious), **Timid** (passive and cautious) and **Naive** (passive and reckless). The names serve as a stereotype for each class and, without further validation, should not be taken too literally as an expression of player experience.

The clusters can be characterised in terms of the external metrics discussed in Section IV-F. Table III gives typical relative values for the end-of-level statistics and self-rated immersion. The typical level statistics for a group can be predicated from the correlation with each discriminant and the group's location in the discriminant space.

However, examining the player clusters does suggest a novel relationship between immersion and the discriminant values. Figure 9 shows immersion scores by player type. It appears that Normal and Naive players experience medium/high levels of immersion. In contrast, Hyperactive players scored a much wider range of values, from high to low immersion, and Timid players scored uniformly highly. One must be careful not to

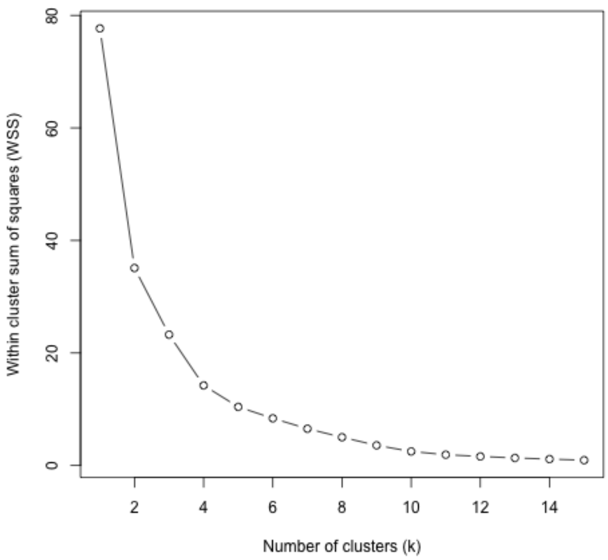

Fig. 7. Within cluster sum of squares by number of clusters k, for *Rogue Trooper* player centroids in LD1-LD2 space.

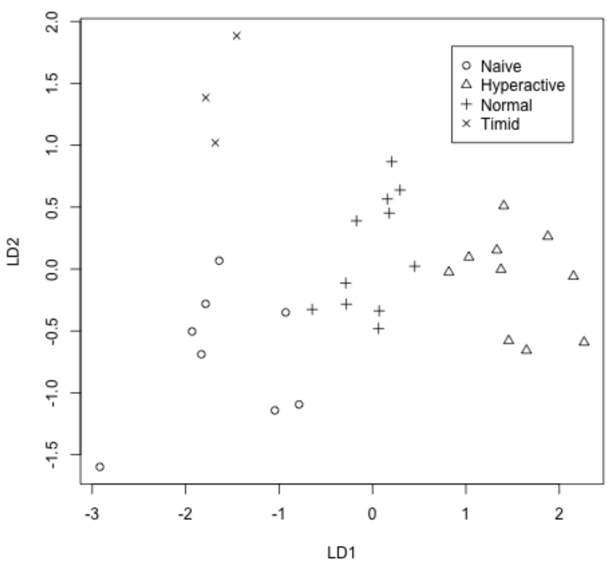

Fig. 8. k=4 cluster solution for *Rogue Trooper* player centroids.

TABLE III
TYPICAL CHARACTERISTICS OF *Rogue Trooper* PLAYER TYPES

Metric	Hyperactive	Normal	Naive	Timid
Time	Low	Low	High	High
Shots/min	High	High	Low	Low
Hits/min	High	High	Low	Low
Accuracy	High	High	High-Low	High-Low
Taken/min	High	High	Medium	Low
Inflicted/min	Med-High	Medium	Low	Low
Efficiency	Medium	Medium	Low	High
Immersion	Low-High	Medium	Med-High	High

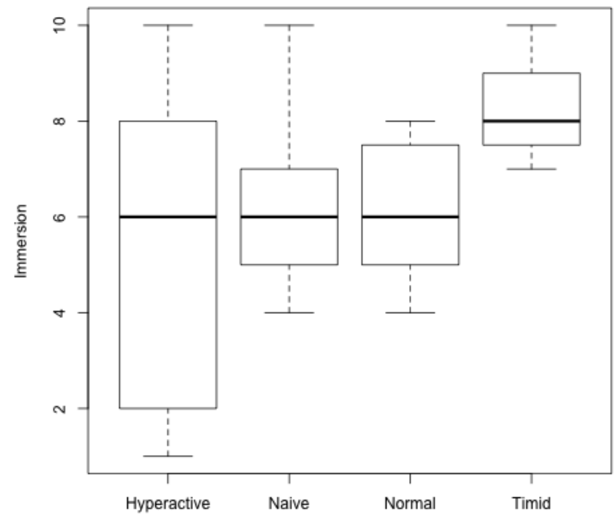

Fig. 9. Self-rated immersion for first level of *Rogue Trooper*, by player type.

over-generalise from this data (there are only 3 players in the Timid group), but this does suggest some hypotheses.

The first hypothesis is that, as we have proposed that high LD1 players are experts in the shooter genre, Hyperactive players may have strong opinions about games in that genre, and if the game is not their taste, they may be more likely to be disengaged and to express this opinion. A second hypothesis is that Timid players, who are playing quite statically and cautiously, may be very focused on the PC's safety and avoiding danger. This would be consistent with a high level of cognitive immersion in the game.

H. Comparison with PCA

Grouping log data extracts by player allows us to apply LDA to identify the dimensions of greatest variation between players. The central idea of LDA here is that it is maximising variance between combats from different players, whilst minimising variance between combats from the same player (as described in Section II-C). A natural comparison to make is with PCA, which chooses dimensions of maximum variance without reference to class labels. What kind of model of *Rogue Trooper* combat does this give us? And do we gain anything by considering individual players?

Table IV shows the influential coefficients for the first three principal components for our mean combat data. Briefly, we interpret PC1 as describing movement in combat, PC2 is combat success (quick, high health versus long, low health), and PC3 is combat distance. PC1 and PC3 are not correlated with any external player metrics. PC2 is weakly correlated with time taken (0.55), and weakly negatively correlated with shots (-0.62) and hits (-0.65) per minute, damage inflicted per minute (-0.53) and efficiency (-0.51). This lends weight to the interpretation of PC2 as combat success. It also suggests PC2 is related to the first two linear discriminants, and

TABLE IV
INFLUENTIAL PCA COEFFICIENTS FOR *Rogue Trooper*

PC	Trace	Positive coefficients	Negative coefficients
1	0.15	move (0.43) dist.rate (0.40) area.rate (0.39) turn.rate (0.34)	
2	0.15	npc.postfire (0.45) mean.postdam (0.38) prefire (0.31)	firing (-0.31) mean.health (-0.30)
3	0.09	sniping (0.52) mean.final.dist (0.52)	

correlations confirm weak negative associations (LD1, -0.50; LD2, -0.59). LD1 is also moderately correlated with PC1 (0.64). The models are clearly related, but quite distinct.

Comparing the first two dimensions of each reduced combat space, the ratio of between-player distance to within-player distance is higher for LDA (1.47) than for PCA (1.06). One would expect this, given the purpose of LDA is to maximise this ratio. However, considering the different 13 combat scenarios in level 1, the ratio of between-scenario distance to within-scenario distance is lower for LDA (1.02) than for PCA (1.30). PCA is separating the different sections of the level more than it is players, and the reduced feature space is more influenced by *game content* than player style. We would argue that the PCA dimensions of combat movement, success and distance are traits of particular combat scenarios, whereas LDA's combat activity, cautiousness and ammunition management are more descriptive of player style.

V. DISCUSSION

The *Snakeotron* and *Rogue Trooper* studies show how LDA can be used to generate quantitative models of player style from log data, by reducing the dimensionality of a feature space for game logs (or log extracts) grouped by player. The reduced space can interpreted as a trait-based model of player style and can be used by designers to yield insights for novel and complex forms of gameplay, either directly or from a player typology generated by clustering.

The studies have suggested a number of hypotheses about player style in these games which merit further research, and may well generalise to others in the same genre. The *Rogue Trooper* study showed that LDA could highlight distinctions in style like Active-Passive and Reckless-Cautious that seem to be distinct from task performance. This suggests that it is well suited to modelling aesthetic aspects of player style that go beyond ability or strategy.

Player style models, like those presented here, could be used as a basis for adaptive content. Developers could either employing the discriminant functions directly for measuring/classifying players, or by using the generated player type labels as training data for some other classification technique. In future work, we will look at using the models of style generated to help estimate player experience from log data, and to realise live adaptation, either directly from style models or as part of a larger player modelling system.

A. Alternatives to LDA

Unsupervised projection of game log feature data to a lower-dimensional space could be performed by a variety of techniques. Unsupervised learning techniques such as Principal Components Analysis (PCA) [4] or self-organising maps [5] would produce models of gameplay of a similar kind to the ones presented here. Multidimensional Scaling has been used in this context to reduce player movement data and identify distinct navigation behaviours [12].

However, an advantage of using a supervised projection method, like LDA, in this context is that the projection can better model differences between players. In Section IV-H we presented a brief comparison of PCA and LDA for the *Rogue Trooper* which illustrates this point: the PCA model characterises combat scenarios better than it does players, whereas LDA is designed to separate classes — in this case, players — in the projection. When using unsupervised techniques to model players, the feature data should be designed to remove other sources of variation. For instance, one could compare repeated plays of the same content, or a single set of feature values per player, as in [10]. LDA can cope with multiple data points per player covering diverse content, which makes it a more flexible approach.

LDA has the advantage of being widely studied and applied, and relatively easy to apply and interpret. However, discriminant analysis offers a range of supervised projection methods beyond LDA, e.g. quadratic DA, mixture DA, LDA with ranks. Many of these have been designed to remove, or to be more robust toward, LDA's assumptions of normality and/or homoscedasticity [24], making them good candidates for player modelling. There is great potential to explore the use of, and perform comparisons between, a range of supervised and unsupervised methods in future work on player modelling.

B. Use in Game Design

Our approach to player modelling is general enough to be applied in almost any gaming context where log data can be obtained. For any new application, the main challenges would be determining appropriate sections of log data to extract, and designing appropriate features for the analysis. Future work with shooter games could focus on improved identification of the start and end points, e.g. based on visual contact and determining player goals.

The process of modelling gameplay, along with the interpretation of results, introduces the possibility of bias in the player style models formed. However, they could provide valuable insights for designers, and the method itself does not carry any preconceptions about style, making it well suited for investigating styles influenced by aesthetic considerations that may themselves be hard to model.

We would also argue that this approach is suitable for adoption by commercial games companies. It employs standard statistical and machine learning techniques, widely available in a number of software implementations, both free (e.g. R) and commercial (e.g. SPSS). Applying LDA to log data can certainly involve some interpretation and decision making, but if properly supported by software tools, this requires only a little technical knowledge, which could be supplemented by diagnostic assistance and visualisation built into those tools.

Learned player style models could play a number of roles in game design. By describing what styles of play are actually emerging from the game mechanics and content, they can inspire game designers to create static or dynamic content that is specifically aimed at providing a satisfying experience for a given style. Conversely, a designer can deliberately work against a particular style, either to temporarily encourage a different style and therefore a more varied experience, or to permanently discourage certain styles of play.

What constitutes an appropriate response to a player style depends on the designer's goals and their understanding of the corresponding player experience, either from their own expertise or player testing. However, returning to our *Rogue Trooper* typology, we can imagine some possible responses to the styles we found: Hyperactive players might be served more complex combat scenarios to maintain their interest, or stealth tasks that require more passive play. Timid players could be allowed larger pauses between periods of high activity, or in-game instructions could encourage them to be less cautious and passive. Naive players could be allowed a 'grace' period of reduced danger at the start of combat to compensate for their lack of caution, or they could be given additional assistance with situational awareness.

Acknowledgments

This work was funded by EPSRC grant TS/G002835/1 *CADGame: Computer-Aided Game Design*, in partnership with the Technology Strategy Board, and EPSRC Leadership Fellowship EP/J004049/1. We would like to thank Kevin Bezant, Steve Burge and Euan Carmichael at Rebellion Developments for their insights into game design and for their work developing the log system for *Rogue Trooper*, and the anonymous reviewers for their insightful comments.

References

[1] K. Isbister and N. Schaffer, *Game Usability: Advice from the Experts for Advancing the Player Experience*. Morgan Kaufmann, 2008.
[2] C. Bateman and R. Boon, *21st Century Game Design*. Boston: Charles River, 2005.
[3] G. J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*. Wiley Interscience, 2004.
[4] I. T. Jolliffe, *Principal Component Analysis*. Springer, 1986.
[5] T. Kohonen, *Self-Organizing Maps*, 3rd ed. Springer, 2001.
[6] R. Bartle, "Hearts, clubs, diamonds, spades: Players who suit MUDs," *Journal of MUD Research*, vol. 1, no. 1, 1996.
[7] C. Bateman, R. Lowenhaupt, and L. E. Nacke, "Player typology in theory and practice," in *Proc. DiGRA Conference*, 2011.
[8] A. M. Smith, C. Lewis, K. Hullett, G. Smith, and A. Sullivan, "An inclusive taxonomy of player modeling," University of California, Santa Cruz, Tech. Rep. UCSC-SOE-11-13, 2011.
[9] G. Yannakakis and J. Togelius, "Experience-driven procedural content generation," *IEEE Trans. on Affective Computing*, vol. 2, no. 3, pp. 147–161, 2011.
[10] A. Drachen, A. Canossa, and G. Yannakakis, "Player modeling using self-organization in Tomb Raider: Underworld," in *Proc. IEEE Symp. on Comp. Intelligence & Games (CIG)*, 2009, pp. 1–8.
[11] R. Baumgarten, "Towards automatic player behaviour characterisation using multiclass LDA," in *Proc. AISB Symp. on AI & Games*, 2010.
[12] R. Thawonmas, M. Kurashige, and K.-T. Chen, "Detection of landmarks for clustering of online-game players," *Int. Journal of Virtual Reality*, vol. 6, no. 3, pp. 11–16, 2007.
[13] B. Cowley, "Player profiling and modelling in computer and video games," Ph.D. dissertation, University of Ulster, 2009.
[14] G. van Lankveld, P. Spronck, J. van den Herik, and A. Arntz, "Games as personality profiling tools," in *Proc. IEEE Symp. on Comp. Intelligence & Games (CIG)*, 2011, pp. 197–202.
[15] N. Yee, N. Ducheneaut, L. Nelson, and P. Likarish, "Introverted elves & conscientious gnomes: The expression of personality in world of warcraft," in *Proc. Int. Conf. on Human Factors in Computing Systems (CHI)*, 2011, pp. 753–762.
[16] C. Pedersen, J. Togelius, and G. Yannakakis, "Modeling player experience in Super Mario Bros," in *Proc. IEEE Symp. on Comp. Intelligence & Games (CIG)*, 2009, pp. 132–139.
[17] G. Yannakakis and J. Hallam, "Real-time game adaptation for optimizing player satisfaction," *IEEE Trans. on Comp. Intelligence & AI in Games*, vol. 1, pp. 121–133, 2009.
[18] N. Shaker, G. Yannakakis, and J. Togelius, "Towards automatic personalized content generation for platform games," in *Proc. AI & Interactive Digital Entertainment (AIIDE)*, 2010.
[19] E. Hastings, R. Guha, and K. Stanley, "Evolving content in the galactic arms race video game," in *Proc. IEEE Symp. on Comp. Intelligence & Games (CIG)*, 2009.
[20] D. Kline and L. McHugh, "Darkspore AI post mortem," Invited talk at AIIDE 2011, available from http://www.aiide.org/aiide11/Darkspore.zip.
[21] G. Mountain, "Psychology profiling in Silent Hill: Shattered Memories," Invited talk at Paris Game/AI Conference, 2010, available from http://gameaiconf.com/?p=141.
[22] W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S*, 4th ed. New York: Springer, 2002.
[23] P. A. Lachenbruch, C. Sneeringer, and L. T. Revo, "Robustness of the linear and quadratic discriminant function to certain types of non-normality," *Communications in Statistics*, vol. 1, no. 1, pp. 39–56, 1973.
[24] J. R. Rausch and K. Kelley, "A comparison of linear and mixture models for discriminant analysis under nonnormality," *Behaviour Research Methods*, vol. 41, no. 1, pp. 85–98, 2009.
[25] S. Marks and O. J. Dunn, "Discriminant functions when covariance matrices are unequal," *Journal of the American Stat. Assoc.*, vol. 69, no. 346, pp. 555–559, 1974.
[26] T. Ashikaga and P. C. Change, "Robustness of Fisher's linear discriminant function under two-component mixed normal models," *Journal of the American Stat. Assoc.*, vol. 76, no. 375, pp. 676–680, 1981.
[27] R. Baumgarten and E. Cabañes Martínez, "The symbiosis of AI and computer games: Two examples of games as a bidirectional interactive system," in *Proc. IADIS Int. Conf. on Game & Entertainment Technologies*, 2010.
[28] "Rogue Trooper critic reviews for PC," Metacritic.com, http://www.metacritic.com/game/pc/rogue-trooper/critic-reviews.
[29] C. Jennett, "Investigating real world dissociation in computer game immersion," Ph.D. dissertation, University College, London, 2009.
[30] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2011. [Online]. Available: http://www.R-project.org/
[31] J.-H. Xue and D. Titterington, "Do unbalanced data have a negative effect on LDA?" *Pattern Recognition*, vol. 41, pp. 1558–1571, 2008.
[32] E. F. Chinganda and K. Subrahmaniam, "Robustness of the linear discriminant function to nonnormality: Johnson's system," *Journal of Statistical Planning and Inference*, vol. 3, pp. 69–77, 1979.