Measuring Cooperative Gameplay Pacing in World of Warcraft

Martin Ashton
School of Computer Science
McGill University
Montréal, Canada
martin.ashton@mail.mcgill.ca

Clark Verbrugge
School of Computer Science
McGill University
Montréal, Canada
clump@cs.mcgill.ca

ABSTRACT

Designing video game scenarios that will stimulate the player with an engaging and properly paced level of difficulty is a non-trivial issue, one which can fundamentally impact the playability and popularity of a game. World of Warcraft, like many MMORPGs, suffers noticeably from the less challenging pacing of its later-game scenarios compared to its earlier-game content. To examine this observation in detail, a World of Warcraft client-side plugin was created to record data about the players' progress throughout a cooperative scenario, including health, power, map position, class, and role. This data was analyzed to measure the pacing of each session. The results showed a drop in difficulty between late-game level 80 five-person group content and level 70 five-person group content. Using this basic metric to quantify the level of difficulty is a step forward in designing scalable and adaptable scenarios that can continue to challenge players of all experience levels.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics—*product metrics*; K.8.0 [**Personal Computing**]: General—*games*

General Terms

Measurement, Experimentation, Human Factors

Keywords

social games, tools, World of Warcraft, computer games

1. INTRODUCTION

In all games, the quality of player experience is directly influenced by the level of challenge provided by the gameplay. A low level of challenge results in a trivial and uninteresting game, while a continuous and excessively high level of difficulty will result in many players abandoning the game as being too difficult. Appropriate game *pace* has thus been recognized as an important factor in ensuring players are continuously engaged without being overwhelmed [15]. Games such as Left 4 Dead even include an active monitoring component that uses a combination of player measures to determine difficulty, and thus controls pace in order to match player ability [4].

In this work, we investigate techniques for measuring pace within a cooperative multiplayer environment. We focus on World of Warcraft (WoW) as a contemporary and popular game where player enjoyment is directly affected by the degree of difficulty experienced. As a multiplayer Role-Playing Game (RPG) with a relatively large variety of player statistics, WoW provides a rich context for measuring player experience. Player (team) health, level of cooperation and the importance of battle strategy are all potential measures of difficulty.

Using the WoW plug-in interface, we thus develop a basic tool for gathering player data during gameplay. We explore two fundamental metrics, one inspired by Left 4 Dead using team health, and the other a novel measure that focuses directly on the team experience, measuring cohesiveness of the group during adventuring. The latter uses a distance-based metric that encodes the "amount" of strategy employed by the team, and thus the degree to which cooperation is important to successful gameplay.

We apply our metrics to multiple instances of a 5-person cooperative "dungeon." Results based on a cooperative team that is well-matched to the dungeon level show clear differences in both measures in comparison to results where player characters are over-levelled, or too powerful. In the former case, our metrics show that difficulty level is relatively even and continuous, promoting tight teamwork. However, in the latter, difficulty is notably lower, with player characters requiring a reduced level of cohesion to complete the same content. Our results demonstrate that difficulty and pacing are clearly exposed by these basic measures. This paper therefore proposes simple and effective methods of evaluating gameplay with respect to long-term replayability as players increase in power in a persistent game world context.

Specific contributions of this work include:

- The description and development of a non-trivial and non-invasive snaphot and monitoring plugin to extract WoW player and gameplay data client-side. Our design allows for the collection of a variety of player metrics without requiring game modifications.
- Based on our plugin we develop and evaluate two interesting metrics for understanding the extent of dif-

FDG'11 June 29-July 1, 2011, Bordeaux, France
Copyright 2011 ACM 978-1-4503-0804-5/11/06 ...$10.00.

ficulty and cooperation in cooperative, Player-versus-Environment (PvE) multiplayer game-play; the first related to player health, and a second that attempts to measure the degree of team strategy and cooperation.

- We show data based on gathering and analysis of multiple gameplays, illustrating player behaviour and demonstrating the effectiveness and appropriateness of our measurement techniques.

In the next section, we describe basic background on WoW essential to understanding our study. Section 3 describes our plugin design and the specific measurements we perform. Quantitative results and discussion are presented in section 4, followed by conclusions and future work.

2. BACKGROUND

This section provides background information on World of Warcraft, including the character leveling system, the distinction between classes and roles, and the progression of the dungeon reward system. We also present the plugin capabilities of the World of Warcraft game client, used as a means to collect gameplay data.

2.1 World of Warcraft

World of Warcraft (WoW) is often described as a modern successor of the multi-user dungeon genre. It is currently one of the most popular Massively Multiplayer Online Role Playing Games (MMORPG), having exceeded a user base of 12 million subscribers in October 2010 [2]. WoW has undergone a drastic number of changes since its inception in 2004 including three expansions, the latest of which is *World of Warcraft: Cataclysm*.

Leveling. The character leveling system in World of Warcraft closely follows the leveling systems of most RPGs. Characters start at level 1 and increase in level by gaining experience from completing quests and defeating non-player characters such as monsters, dragons, wizards, etc. The latest WoW expansion raised the level cap from 80 to 85. Note that the experiments described in Section 4 were performed in the previous expansion, *Wrath of the Lich King,* during which the level cap was set to 80. When a character gains a level, stats are incremented. Character stats include stamina, dexterity, strength, intellect, etc. The growth of these stats is prioritized according to the character's class. During the leveling process, the character is also attributed talent points, which can be spent to obtain new spells and abilities.

Classes. During the character creation process, the player must select one of ten classes: Warrior, Druid, Shaman, Hunter, Paladin, Death Knight, Rogue, Mage, Warlock, and Priest. These classes follow the archetypal descriptions found in most RPGs - Warriors are melee-centric and heavily armored; Rogues rely on stealth; Mages cast destructive spells; Priests heal their teammates, etc. The majority of these classes have a unique mechanic or resource system which differentiates their gameplay. For example, Warriors use a *rage* system which accumulates over the course of a battle, allowing them to perform specific abilities. By contrast, most spell-casters use *mana* to cast spells from a distance.

Roles. A character's talent point distribution dictates his or her optimal role in a group setting. Each class has three unique *talent trees* in which talent points can be spent. Deep talents within these trees are more powerful, but also have a higher number of prerequisites. Every class has at least one talent tree which focuses on increasing combat efficiency. Warlocks, Rogues, Mages, and Hunters are pure *damage-dealers.* Warriors, Paladins, Death Knights and Druids can invest points in one of their respective trees to optimize their *tanking* ability, allowing them to mitigate incoming damage. Paladins, Druids, Shamans, and Priests each have at least one *healing* tree, which allows them to heal group members more efficiently. As such, a character can fulfill one of three roles in a group setting: tank, healer, or damage-dealer.

Cooperative gameplay. An in-game matchmaking system facilitates group formation by combining characters according to their role. Character grouping is first prioritized according to level, then according to equipment quality. One tank, one healer and three damage-dealers are bound to an instanced dungeon, in which the group members must cooperate to defeat non-player controlled characters (NPCs). These player-versus-environment (PvE) scenarios usually take 25 minutes to complete, after which the group members are rewarded with experience, items and *dungeon points* which can be traded with vendors for more valuable equipment. Dungeons have a minimum level requirement, and can be repeated an arbitrary number of times. The matchmaking system will ensure that characters are placed in a suitable dungeon, in which the NPCs are roughly the same level as the group members. The environmental layout and events do not change between instances of the same dungeon. In most cases, enemy NPC placement and attributes also remain the same, although a small handful of dungeons spawn enemies pseudo-randomly.

When a character reaches the level cap, his or her stats can only increase by obtaining better equipment. As a level-capped character's stats increase, so does its ability to fulfill roles in more challenging dungeons, such as the "heroic" versions of previously visited dungeons. Equipment gathered from these heroic-difficulty dungeons are then used to participate in longer, more intricate raid instances. These instances can only be completed by groups of 10 to 25 players. As such, the 5-player dungeons act as stepping stones into more demanding raid environments.

One notable issue which affects the replayability of these "stepping stone" dungeons is that their content does not scale dynamically with the stats of the group members. Regardless of the players' equipment quality and stats, the strength, health, attack types, layout and number of enemy NPCs remains fixed according to the original intent of the level designers.

To maximize game content replayability, the dungeon point currency system acts as an incentive for characters with comparatively higher quality equipment to revisit the 5-person dungeons, and help less advanced characters progress through the game. However, considering the static difficulty and eventual predictability of these dungeons, anecdotal player experience suggests that the gameplay becomes increasingly trivial as the character's stats grow: player health pools increase, tanks mitigate more damage, healers heal for higher amounts, and damage-dealers increase their damage throughput. Eventually, the trivial nature of higher-level player interaction with the game is perceived as mundane, or a "grind": a task whose time investment and stimulation level is unfitting with respect to the reward's perceived value.

2.2 WoW AddOns

World of Warcraft is based on a client-server architecture, wherein a game client communicates with a server to update its view of the game world. The World of Warcraft game client supports the use of *AddOns* (plug-ins) to customize the layout and functionality of its graphical user interface (GUI). WoW AddOns are typically written in the Lua programming language [8][16].

The World of Warcraft API offers an extensive array of functions ranging from character inventory management to Mac-exclusive iTunes playback control [3]. Note that there are no functions available to automate character interactions with the game world. As such, spell-casting and character movement can only be performed as a result of the player's direct input via the mouse and keyboard.

The game client contains a Lua interpreter which supports a subset of the language's libraries. Within this Lua sandbox, there is no way to directly access the underlying file system, nor is there a way to directly invoke any of the operating system's functions. Data can nonetheless be saved and loaded by means of a designated Lua save file, the name of which can be specified in the AddOn's meta-data. AddOn data is saved as Lua code, which can be run in a standalone Lua interpreter to recreate the serialized data structures outside of the World of Warcraft client [8].

3. METHOD

In this section, we outline the framework used to perform our experiments, as well as the metrics applied to our data. Figure 1 depicts the architecture of our framework. The Orunj AddOn, presented in the figure as the "Lua Script", runs in the Lua sandbox of the World of Warcraft game client. The AddOn records gameplay data by polling the game client for information about its view of the game state. Recorded sessions are automatically serialized by the game client into a Lua file. Offline scripts are used to analyze this data, namely by computing for each timestep in a session: (1) the total amount of health lost in the group, and (2) the minimum enclosing circle around the group. The analyzed data is exported into flat text files to facilitate plotting. It is also exported into the *.love* file format so it can be retroactively visualized through the LÖVE framework [10], a Lua-based 2D graphics library.

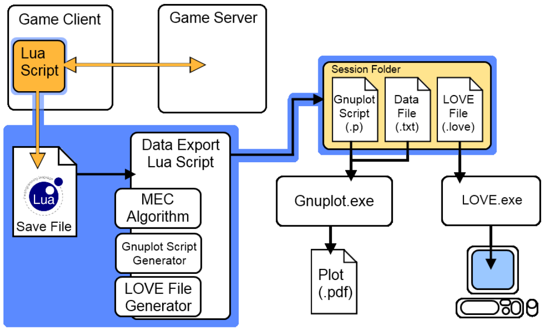


Figure 1: Orunj system architecture.

3.1 Orunj AddOn

The Orunj AddOn begins its data recording procedure by instantiating required data-structures for the new recording session. An initial snapshot of the current group members is taken, including their names, their maximum health, and their maximum power (power here depends on the class' resource system such as rage, mana, runic power, energy, or focus). The talent point distribution of each character is also recorded to infer its role in the group as either a tank, healer, or a damage-dealer.

The default sampling rate is set to one sample per second to minimize memory consumption. A sample point contains each group members' health, power, and map position as an (x, y) coordinate. The map position is normalized to a unit square and represents the position of the character with respect to the 2D overhead map of the environment. Note that full 3D coordinates of the players are not accessible through the WoW API. In most cases, however, the z-coordinate of players is not significant to gameplay.

The majority of the WoW API function calls return a small subset of the local client's data, as opposed to querying the server directly. An exception to this flow is the initial character inspection process, which requires that the server be notified of which character needs inspection in order to push the appropriate data to the game client. The act of recording character health, power, and map position is thus non-invasive with respect to the server and other players, and does not perturb player actions or experience. When a session has been recorded, the player must log out of the game world to ensure that the recorded data is written to the Orunj AddOn's designated save file.

The recorded session's datastructures are loaded outside the WoW game client by means of an external Lua interpreter. Lua scripts are used to compute metrics on the collected gameplay data such as overall health loss per unit time, and the progression of the minimum enclosing circle around the group members based on their 2D map position.

3.2 Measuring Player Intensity

The first metric applied to the recorded data is *player intensity*, which is measured in proportion to the amount of health lost per unit time. To succeed in a dungeon, group members must cooperate to defeat enemy NPCs and avoid being killed. Similar to the *emotional intensity* metric used in *Left 4 Dead* [4], when a group member is damaged, the *player intensity* of the group is incremented proportionally. For each sample in a recorded session, player intensity is computed by summing the total amount of health lost among each group member, divided by the total health of the group. This yields a normalized value between 0 and 1 for each sample, which allows for comparisons of player intensity metrics between groups.

Periods of high intensity correspond to moments in which group members sustain a large amount of damage (i.e. health loss) per unit time. For a group to succeed, its players must meet some baseline level of involvement with the game to cooperate and survive moments of high intensity. As intensity increases, so should the level of involvement required by each player to stay alive.

This level of involvement contributes to the gameplay experience by promoting the player's focus and motivation. If the amount of intensity is insufficient, the scenario becomes trivial and the player may lose interest. It is worth noting, however, that other game genres such as puzzle games or social games may benefit from a more relaxed pacing. In such cases, player intensity may not be an appropriate metric to gauge player involvement.

3.3 Minimum Enclosing Circle Progression

The second metric applied to the collected data is the progression of the minimum enclosing circle (MEC) around the group. The MEC algorithm [14] takes as input the 2D map position of each character in the group, and returns the (x,y) coordinates of the MEC's center, as well as its radius.

To justify the relevance of the MEC metric, we first present a set of key observations in relation to WoW's gameplay mechanics:

- Spell-casting relies on the character's distance and line of sight from its target. Ranged damage-dealers must generally stay within 35 yards of their target, while healers can heal from a distance of up to 40 yards. Melee classes must stay within melee range of their target.

- Most spell-casting requires that the character be stationary. Melee classes maximize their damage throughput when they are standing behind their target, to reduce the chance of parries or blocks. As such, the tank should avoid moving the enemy NPC's unnecessarily.

- Healers and damage-dealers tend to stay behind the tank as it leads the group through the dungeon; the tank is generally the first group member to engage in battle.

Cooperative gameplay pacing in WoW can be measured by comparing the MEC of the group at each timestep. To complete a dungeon, a group must defeat the enemy NPCs along a (usually) linear path towards the dungeon's final boss. The speed at which a group progresses through the dungeon's environment can therefore be used as an indication of the dungeon's pacing.

In light of the WoW gameplay mechanics listed above, the position of the MEC's center should vary minimally during a battle, and should change as the group moves to the next battle. As a rough estimator, the overlap of the MECs around the same region is generally indicative of a prolonged battle, suggesting a higher level of difficulty. It is therefore expected that an easy scenario would be characterized by a highly mobile MEC center; the group would spend less time *engaged* in battle, and more time *moving* towards the final boss.

4. EXPERIMENTAL RESULTS

In this section, we describe the experiments performed in World of Warcraft's dungeon matchmaking system. We also analyze our results and discuss their relevance towards measuring pacing and perceived intensity in WoW.

4.1 Experimental Context

Three characters were used to perform the experiments reported in this paper, namely: (1) "Appuls", a level 80 Druid with high-quality equipment obtained from 10 and-25 person raids, (2) "Orunjs", a level 80 Death Knight (not to be confused with the name of our *Orunj* WoW AddOn), with comparatively lower quality equipment gathered from quests and 5 person heroic dungeons, and (3) "Aid", a level 70 Priest equipped with questing items. Appuls and Aid had their talent point allocation optimized for healing, whereas Orunjs was optimized for tanking.

We examined the results from the regular and heroic versions of the "Utgarde Keep" dungeon (see Figure 2). Aid was used to collect the results of the regular version of Utgarde Keep - the regular version of Utgarde Keep is the first *Wrath of the Lich King* dungeon available at level 70. Appuls and Orunjs were used to collect data from the heroic version of Utgarde Keep, accessible at level 80.

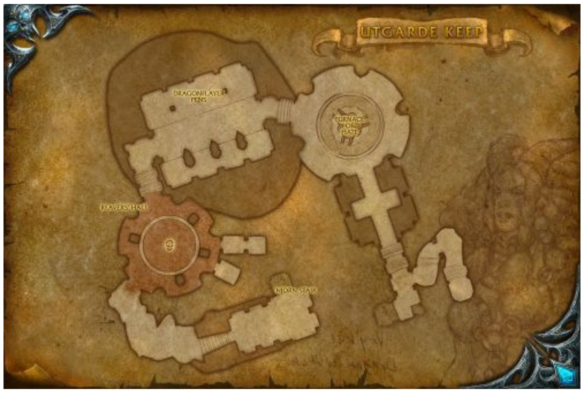


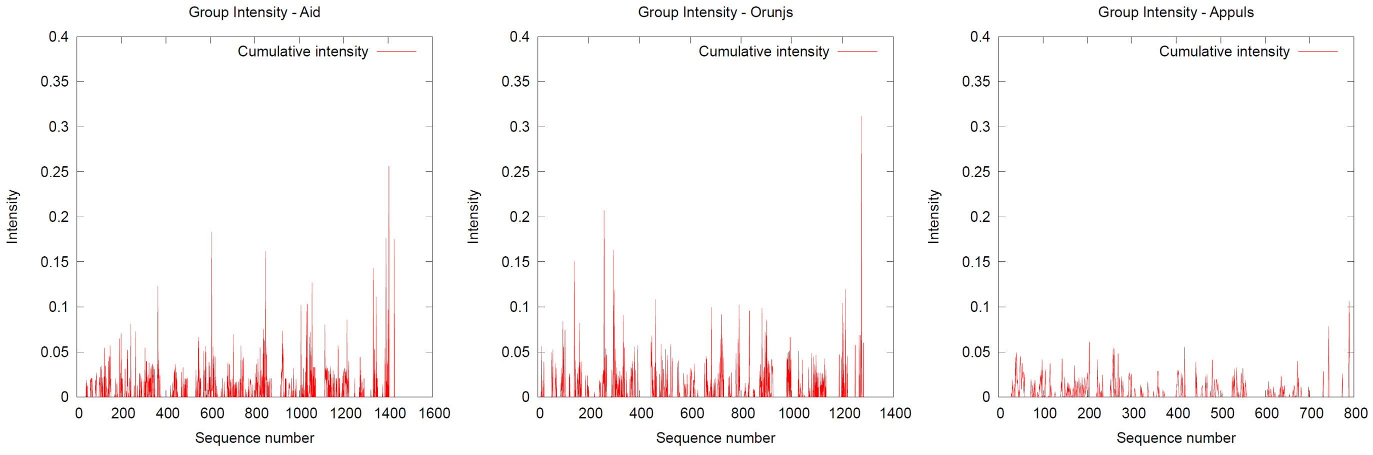
Figure 2: First floor of Utgarde Keep. Players enter the dungeon on the bottom right, and proceed to the second floor of the dungeon, following a horseshoe-shaped path.

Recall that the dungeon matchmaking system prioritizes group formation according to level, then equipment quality. Aid's groups consisted of characters ranging from level 69 to 72. Appuls' and Orunjs' groups consisted of level 80 characters, though Appuls' group members usually had higher stats, imparted by higher-quality items.

4.2 Intensity Results

Aid. The regular version of Utgarde Keep was designed and balanced for characters between levels 69 and 72. The data collected with Aid therefore serves as a basis of comparison for the intended pace and intensity of this dungeon. Figure 3 (a) illustrates the player intensity for a playthrough of the regular version of Utgarde Keep. Observe that the player intensity ramps up during each battle, and then quickly diminishes as the enemy NPCs are defeated. The periods of 0 intensity correspond to group movement towards the next battle. It is worth noting that the intensity peaks at the end of the playthrough, when the group must defeat the last boss of the dungeon. This intuitively serves as a culminating point for player intensity, in which the group can lose upwards of 25% of its maximum health from a single attack.

Orunjs. Like most heroic dungeons in *Wrath of the Lich King*, the heroic mode of Utgarde Keep was balanced to challenge characters like Orunjs, who had just attained level 80 and who were primarily equipped with items obtained by leveling. Orunjs was often paired with similarly equipped players by the dungeon matchmaking system. Thus, the player intensity in Figure 3 (b) shows that Orunjs' session through the heroic version of Utgarde Keep matches the level of intensity recorded by Aid's group in Figure 3 (a).



(a)

(b)

(c)

Figure 3: *Player intensity* in regular Utgarde Keep. Each plot represents one recorded session. The sequence number along the x-axis represents the elapsed time in seconds. The player intensity is defined as the cumulative health loss of each group member divided by the total group's health at each timestep. (a) Aid, level 70, quest-reward equipment. This intensity plot serves as a point of comparison for the intended level of difficulty for this dungeon. (b) Orunjs, level 80, dungeon and quest reward equipment. The intensity pattern is similar to the one recorded by Aid. The stats and skill of the participants is fitting of the dungeon's difficulty. (c) Appuls, level 80, end-game raid equipment. The lack of intensity here suggests that the stats and skill of the group members far surpasses the dungeon's difficulty.

Appuls. Figure 3 (c) illustrates the intensity of a heroic Utgarde Keep session recorded by Appuls, a character equipped with raid-quality items. The player intensity in this session is distinctly lower than that of Figures 3 (a) and 3 (b), which supports the intensity metric predictions of Section 3.2. The duration of the session was also substantially lower than the previous sessions: 796 seconds compared to 1429 and 1295 seconds for Aid's and Orunjs' playthroughs respectively.

The higher stats of Appuls and his group members, imparted by their higher-quality raid items, contributed strongly to the reduction in player intensity in heroic Utgarde Keep. In comparison to Orunjs' and Aid's groups, more damage was mitigated, character health pools were larger, and the damage and healing throughputs of each player were greater.

The reduction in player intensity for Appuls' session thus attests to the relevance of the player intensity metric when attempting to quantify the level of difficulty to which the player is subjected.

4.3 Pacing Results

The MEC metric was used to compare gameplay pacing between the sessions recorded by Aid and Appuls. For clarity, we report the pacing results of the first floor of the Utgarde Keep dungeon. Note that the plots in Figures 4 (a) and 4 (b) correspond to the trajectory suggested by the overhead map of the dungeon in Figure 2.

Appuls. Figures 4 (a) and 4 (b) represent the cumulative MEC data of three sessions collected by Aid and Appuls respectively. These plots can be viewed as heatmaps of the average amount of time spent by each group in a specific area of the dungeon. The regions of the plots darkened by the superimposition of MECs correspond to areas in which the groups spent more time. The lighter regions of the plots denote areas of group movement towards the next battle.

The MEC heatmaps recorded by Aid in Figure 4 (a) for regular Utgarde Keep are noticeably denser than the ones recorded by Appuls in Figure 4 (b) for heroic Utgarde Keep. The difference in these heatmaps suggests that the groups in which Aid participated spent more time in each battle, and therefore followed a more progressive pace than Appuls' groups, whose individual battles were much shorter in duration.

This disparity in pacing can be explained by the static difficulty of the dungeons in WoW. As characters obtain better equipment from (1) the dungeon point system, or (2) the 10 and-25 person raids, they become notably more efficient at completing 5-person dungeons such as heroic Utgarde Keep. The trivialization of this content may, in the long run, cause players to become apathetic to the gameplay which would subsequently hinder the game's replayability.

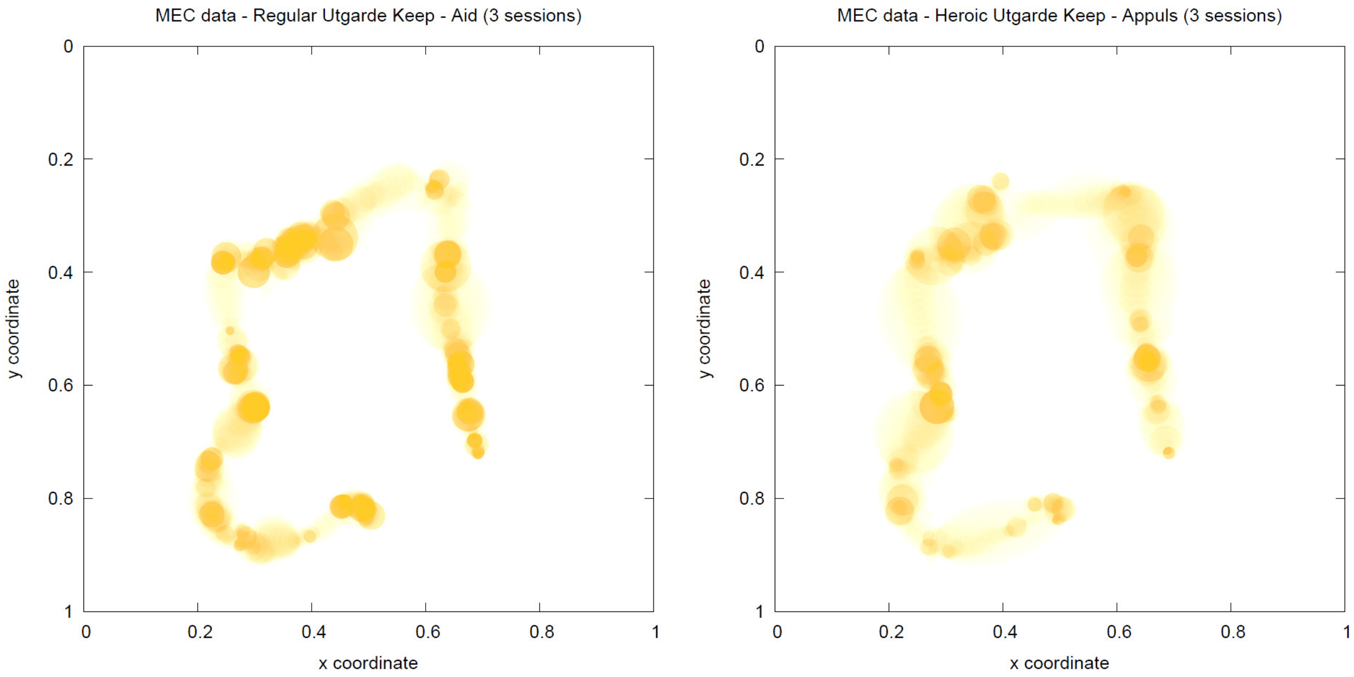
5. RELATED WORK

In this section, we provide related work pertaining to World of Warcraft, player behaviour analysis, and gameplay metrics.

World of Warcraft has been the focus of a variety of experiments in the domain of games research. Several works have offered insight into the social dynamics in WoW [5][13].

Interviews and chat log analyses were used in [1] to qualitatively survey group cooperation in 5-person instances (dungeons). The findings of this work coincide closely with the results of our gameplay metrics. Namely, compelling instances provide a sufficient level of threat to engender social interactions among the players with the aim to coordinate, strategize, and gain control over the encounters.

Ducheneaut et al. reported their findings on census data gathered over 223 043 players, spanning several WoW realms (servers) in [7]. Notable gameplay trends emerged from this



(a)

(b)

Figure 4: (a) Cumulative MEC data of 3 sessions in regular Utgarde Keep (first floor) recorded by Aid. Darker regions indicate areas in which groups spent more time battling enemy NPCs. (b) Cumulative MEC data of 3 sessions in heroic Utgarde Keep (first floor) recorded by Appuls. The lighter regions of this plot allude to a much quicker pace.

data including time spent at each level, class popularity, and distribution of player types according to faction. WoW has evolved substantially since this data was collected, and new census data of the playerbase could yield interesting insights into the development of the game in terms of user appeal and game design.

The World of Warcraft Armory website aggregates historical World of Warcraft gameplay data for characters including achievements, acquired items, and miscellaneous activities. These publically available records were explored in [9] by means of a web crawler. Statistics from this dataset revealed several interesting trends including the average time required for each class to get to level 80, as well as the most popular items.

Player movement in World of Warcraft was investigated in a player-versus-player (PvP) setting in [12]. Network packet inspection sent from the game server to the client was used to collect positional data of players in the environment. The data analyzed in this experiment isolated several important environmental hotspots which corresponded to gameplay objectives. Player movement patterns also suggested three distinct types of player behaviour: wanderers, patrollers and guards.

Gameplay metrics have been used in a variety of other genres such as adventure games and first-person-shooters to evaluate level design and environment layouts [6]. Supervised AI learning techniques were applied to such metrics in *Tomb Raider Unlimited* to predict when a player will quit, or how long he/she will take to complete the game [11].

6. CONCLUSIONS AND FUTURE WORK

MMORPGs rely on continuing and long-term player participation, typically fostered by a growth or enhancement system for player avatars. Ensuring the game remains relevant to players despite their increase in abilities and resources is thus important to player retention. Unchecked or unmatched by sufficiently abundant new and higher-level content, however, such "power-creep" has potential to trivialize game challenges, making the game experience uninteresting. Avatar level restrictions are often applied to avoid this, but especially in the context of cooperative multiplayer scenarios, difficulty can be seen as a complex function of group balance, resources, and relative player skill.

More direct measures of player difficulty can allow for a more straightforward evaluation of the extent and pacing of challenge, and thus the amount and relative level of player attention. Our results here demonstrate that easily gathered data can successfully summarize group experience, illustrating both the degree of difficulty and the how well it is distributed during actual gameplay. Additional data collected with these tools can help us further understand cooperative multiplayer behaviour and reactions to specific game content, as well as in developing adaptive systems that may dynamically adjust to match the group ability.

A variety of future work is possible based on our design. Multiplayer data is necessarily noisy, and although difficult to achieve purely client-side, a larger scale study, evaluating multiple dungeons and more player groupings would improve our analysis and allow for comprehensive and statistically justified determinations. Different metric goals are also of interest; predicting many player behaviours such as amount of play-time [11], degree of group contribution ("free-loading"), etc., has potential to further expose gameplay properties useful to both players and game designers.

7. ACKNOWLEDGEMENTS

This research was supported by the Natural Sciences and Engineering Research Council of Canada. Special thanks are also extended to the players of the *Ruin* battlegroup who were grouped with the characters mentioned in this paper.

8. REFERENCES

[1] S. Bardzell, J. Bardzell, T. Pace, and K. Reed. Blissfully productive: grouping and cooperation in world of warcraft instance runs. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, CSCW '08, pages 357–360, New York, NY, USA, 2008. ACM.

[2] Blizzard Entertainment. World of Warcraft subscriber base reaches 12 million worldwide. `http://us.blizzard.com/en-us/company/press/pressreleases.html?101007`, Oct. 2010.

[3] Blizzard Entertainment. World of Warcraft API. `http://www.wowwiki.com/World_of_Warcraft_API`, Feb. 2011.

[4] M. Booth. AI systems of Left 4 Dead. `http://www.valvesoftware.com/publications/2009/ai_systems_of_l4d_mike_booth.pdf`, 2009.

[5] V. H.-h. Chen, H. B.-L. Duh, and H. Renyi. The changing dynamic of social interaction in World of Warcraft: the impacts of game feature change. In *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*, ACE '08, pages 356–359, New York, NY, USA, 2008. ACM.

[6] A. Drachen and A. Canossa. Towards gameplay analysis via gameplay metrics. In *Proceedings of the 13th International MindTrek Conference: Everyday Life in the Ubiquitous Era*, MindTrek '09, pages 202–209, New York, NY, USA, 2009. ACM.

[7] N. Ducheneaut, N. Yee, E. Nickell, and R. J. Moore. Building an MMO with mass appeal: A look at gameplay in World of Warcraft. *Games and Culture*, 1:281–317, 2006.

[8] R. Ierusalimschy. *Programming in Lua*. Lua.org, 2nd edition, 2006.

[9] C. Lewis and N. Wardrip-Fruin. Mining game statistics from web services: a World of Warcraft armory case study. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, FDG '10, pages 100–107, New York, NY, USA, 2010. ACM.

[10] LÖVE Development Team. LÖVE 2D. `http://love2d.org/`.

[11] T. Mahlmann, A. Drachen, A. Canossa, J. Togelius, and G. N. Yannakakis. Predicting player behavior in Tomb Raider: Underworld. *Proc. IEEE Conference on Computational Intelligence and Games (CIG) 2010*, August 2010.

[12] J. L. Miller and J. Crowcroft. Avatar movement in World of Warcraft battlegrounds. In *Proceedings of the 8th Annual Workshop on Network and Systems Support for Games*, NetGames '09, pages 1:1–1:6, Piscataway, NJ, USA, 2009. IEEE Press.

[13] T. Pace, S. Bardzell, and J. Bardzell. The rogue in the lovely black dress: intimacy in world of warcraft. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 233–242, New York, NY, USA, 2010. ACM.

[14] D. A. Seifert. Minimum enclosing circle in Ruby. `http://www.dseifert.net/code/mec/`, Feb. 2008.

[15] G. Smith, M. Treanor, J. Whitehead, and M. Mateas. Rhythm-based level generation for 2D platformers. In *Proceedings of the 4th International Conference on Foundations of Digital Games*, FDG '09, pages 175–182, New York, NY, USA, 2009. ACM.

[16] J. Whitehead and R. Roe. *World of Warcraft Programming: A Guide and Reference for Creating WoW Addons*. Wiley Publishing, 2nd edition, 2010.