Now You Can Compete With Anyone: Balancing Players of Different Skill Levels in a First-Person Shooter Game

Rodrigo Vicencio-Moreira, Regan L. Mandryk, and Carl Gutwin
Department of Computer Science, University of Saskatchewan
Saskatoon, Canada, S7N 5C9
firstname.lastname@usask.ca

ABSTRACT
When player skill levels differ widely in a competitive First-Person Shooter (FPS) game, enjoyment suffers: weaker players become frustrated and stronger players become less engaged. Player balancing techniques attempt to assist the weaker player and make games more competitive, but these techniques have limitations for deployment when skill levels vary substantially. We developed new player balancing schemes to deal with a range of FPS skill difference, and tested these techniques in one-on-one deathmatches using a commercial-quality FPS game developed with the UDK engine. Our results showed that the new balancing schemes are extremely effective at balancing, even for players with large skill differences. Surprisingly, the techniques that were most effective at balancing were also rated as most enjoyable by both players – even though these schemes were the most noticeable. Our study is the first to show that player balancing can work well in realistic FPS games, providing developers with a way to increase the audience for this popular genre. In addition, our results demonstrate the idea that successful balancing is as much about the way the technique is applied as it is about the specific manipulation.

Author Keywords
First-person shooters; game balancing; player balancing

INTRODUCTION
The enjoyment people get from multiplayer games is often reduced when players have widely different skill levels. This problem happens because the difficulty of a multiplayer game is determined largely by the expertise of the other players – and when experts play against novices, the game is either too hard (for the novice) or too easy (for the expert). Player balancing is a solution to this problem; it involves making different adjustments to the game mechanics for different people, in order to help equalize the performance of stronger and weaker players.

Player balancing has been successful in some commercial games (e.g., differential powerups in Mario Kart), and has been shown to be effective in studies of 2D shooting-gallery games (e.g., [6]). However, other than early efforts such as the Fatboy mod for Unreal Tournament, relatively little work has been done on player balancing in the popular and fast-paced first-person shooter (FPS) genre.

Initial work in this area has suggested that effective player balancing in FPS games is more difficult than in other genres. One study tested several aim-assistance techniques in single-player FPS walkthroughs, and showed that it was very difficult to bring the performance of novice players up to the level of experts [24]. There were two reasons given. First, the magnitude of the difference between novices and experts was much larger than expected – meaning that standard balancing techniques did not work well. Second, there are several game factors that determine expertise in an FPS game in addition to accurate aiming (e.g., avatar movement, learning and remembering maps, management of resources), and these other factors made it difficult for a single mechanism to equalize performance. The fact that experts are so much better than novices – in many different aspects of the game – suggests that basic approaches to balancing realistic FPS games are unlikely to work.

To address this problem, we have developed three new player-balancing schemes for FPS games that are better designed to overcome large skill differences than previous techniques. First, we refined existing aim assists (bullet magnetism and area cursor) that showed promise in previous work, to increase the size of the techniques’ effects. Second, we incorporated a new way of dynamically applying the aim assist that maintains the effect for a longer time period. Third, we developed a new assistance technique that manipulates two additional elements of FPS games in addition to aim assistance: a map view that shows the location of the expert to the novice, and different rates of damage for stronger and weaker players.

Our goal in developing these schemes was to handle a large range of skill differences, and be able to balance players in FPS scenarios that have all of the complexity of real games – but without making the techniques seem obvious and unfair to players. We implemented the balancing schemes with the Unreal Development Kit (UDK), and carried out a study in which novice, intermediate, and expert players played a series of one-on-one deathmatches with the different
balancing schemes. Our test environment was similar in look, feel, and performance to a commercial FPS game, making the players feel like they were competing in an off-the-shelf game. We recorded performance (shots, hits, and kills), player experience (ratings of enjoyment, autonomy, competence, and relatedness), and perception of fairness (ratings of how much assistance had been provided to player, and direct ratings of how fair the game was).

Our results show that the balancing schemes worked extremely well. The study provides seven main findings:

- The schemes all showed significant improvements in minimizing score differential compared to the control condition with no assists provided.
- The schemes worked for pairings with very large skill differences (i.e., novice-expert pairings).
- There were significant differences between the balancing schemes in terms of efficacy – the standard aim assists were least effective, and the multi-factor scheme (that combined aim assist with location indicators and differential damage) was the most effective.
- Despite the large degree of assistance sometimes provided by the schemes, perceptibility of the assistance was very low for players who were assisted, and stronger players only noticed the assist for the multi-factor scheme.
- The balancing schemes actually improved the perception of fairness for the weaker player without detracting from the perceived fairness of the stronger player.
- The new balancing schemes improved the weaker player’s perception of their own competence (in line with their improvements in accuracy), but did not affect the stronger player’s feelings of their own competence.
- Finally, although the schemes affected only the weaker player’s performance and perceived competence, experienced enjoyment increased for both the stronger and weaker players in when matches were balanced.

Our study is the first empirical examination of player balancing techniques for realistic competitive scenarios, with human players, in a commercial-quality FPS game. We make five main contributions. We show: that player balancing can be extremely effective (and therefore should be considered by game designers who wish to engage a wider audience); that the way in which balancing is applied is as important than the game mechanics that are being manipulated; that combining game mechanics in a balancing scheme can improve effectiveness; that effective player balancing does not compromise perceptions of fairness for either player; and that although effective player balancing improves only the performance of the weaker player, it leads to greater enjoyment for both, suggesting the importance of the roles of suspense and uncertain outcome in generating positive play experiences.

RELATED WORK

Game Balancing and Player Balancing
There are several types of balancing that can be considered in game design. Traditionally, designers attempt to balance games in terms of difficulty level – that is, a game is balanced when the difficulty level of the game matches the skill level of the player [10]. This keeps players in the “flow zone” where challenge and accomplishment are balanced [23]; games that manage to keep players at an optimal level of challenge are more enjoyable [8, 10, 13, 16].

In multiplayer games, however, players can be at very different skill levels. If a game is balanced in the traditional sense, it means that expert players will always win over novices, and usually by a large margin – which can lead to a poor play experience for both players. Player balancing attempts to address this problem – it attempts to provide assists to weaker players (or detriments to stronger players) in order to provide a more competitive game [1].

Player balancing is valuable because when players are more evenly matched, enjoyment is increased [6]. This result has been demonstrated in several studies – for example, in one study of siblings, the younger child became upset or lost motivation to continue playing if they were never able to perform as well as the older sibling [14]. However, if player balancing techniques are too obvious, both the stronger and the weaker player can feel that the game is unfair [5, 14].

Types of Player Balancing
There are several ways to balance player abilities and competition. Previous game research presents four general categories: difficulty adjustment, matchmaking, asymmetric roles, and skill assistance.

Difficulty Adjustment
Player balancing can be achieved by adjusting the challenge of the game for different players, statically or dynamically.

The static approach allows the player to choose a difficulty level, typically at the game start. The problems are that players need to know their skill level before the game starts, which can lead to situations where the game difficulty does not match the player’s ability [15]. In terms of multiplayer balancing, static difficulty tends to take the form of handicaps. This can involve manipulating the capabilities of units, access to resources, and starting positions, so that no one player has an initial advantage in the game.

Dynamic difficulty adjustments change game difficulty based on player performance [15]. For example, in Mario Kart, players who are farther behind get better powerups to help them catch up. These techniques, however, can be highly obvious to players – when difficulty adjustments are too noticeable, some players can exploit the system better than others [15] and can make players feel cheated [13].

Matchmaking
Matchmaking systems are present in many popular games, such as Dota 2, Starcraft 2, and Halo. Matchmaking systems
try to achieve player balance by matching players of equal skill [21]; however, the system may not always be able to perfectly match players and must compromise based on the available players. It also does not take into account certain conditions that may temporarily change an individual’s performance, such as having a “bad day.”

Asymmetric Roles
In team games, player balancing can occur naturally if players can choose different roles that are better suited for their level of expertise. For example, Team Fortress 2 allows players who do not have good shooting skills to choose a class such as the Medic, where they can be a benefit to their team without needing to shoot accurately. These games must be carefully designed to ensure that every class contributes to the team and no class is considered unnecessary. The downside to this kind of game balancing is the fact that players may not like feeling forced into a certain role, and game balancing issues may arise if not every role is filled.

Skill Assistance and Aim Assistance
Player balancing techniques can differentially assist the core mechanics that are needed to play a multiplayer game – such as steering in driving games. Aiming assistance (algorithmic changes that alter the accuracy of targeting movements) help weaker players to hit their targets more often when they shoot. Aim assistance has been shown to be effective at improving competition in 2D games where players have unequal skill [6]. In addition, the techniques were not obvious to the expert or the novice, and did not make people feel that the game was unfair [6].

A recent study showed that although 3D versions of 2D aiming assists improved in-game performance, real game factors reduced their efficacy. For example, the presence of distractor targets (friendlies), player movement ability, and having enemies shoot back reduced the effectiveness of aim assistance in different ways [24]. The results in this previous paper indicate that use of aiming assistance for game balancing in a 3D FPS may not be effective.

Overall, previous work has shown that “expertise” in a 3D first-person shooter is made up of much more than just aiming ability – there are substantial differences between experts and novices in the ways that players move to evade enemy fire, use cover in the environment, learn the map, and make use of resources such as ammunition and health packs. Aiming and shooting is still important, but it is not the only factor that differentiates experts and novices.

Aim Assistance Techniques
There are several ways to assist targeting actions in pointer-based computer systems, and several of these can be adapted for use in FPS games. In general, all targeting actions are governed by Fitts’ Law (including 3D virtual spaces [4,18]), which states a relationship between the difficulty of targeting and the target’s distance and width [12]. Assistance techniques work either by changing the effective width of a target or by adjusting its distance.

Two main types of solutions that increase target width are Sticky Targets, which reduces the control-to-display ratio of the cursor when it is over the target [9, 26], and Target Gravity, which attracts the cursor to targets using a simulated gravity function [6].

Reducing distance to the target can be carried out in several ways. Existing techniques warp the cursor towards the target based on the user’s initial movement (e.g., Delphian Desktop [3]), or create temporary proxies of targets that are closer to the user’s staring point (e.g., Drag-and-Pop [7]).

Finally, some techniques combine both of these approaches. For example, the Angle Mouse [25] changes the Control to Display (CD) ratio depending on the current phase of pointing, to reduce the distance and increase the width of the target. The CD ratio is lowered when the user is in the ballistic phase (which means less physical movement to get to the target); when the corrective phase is detected (i.e., when small quick movement occurs), CD ratio is increased, making the object bigger in motor space.

Recent research tested the performance of several aiming assistance techniques in single-player walkthroughs in an FPS game [24]. The study showed that several techniques that had previously worked well in 2D were not as effective in the FPS environment. The techniques that worked best (Area Cursor and Bullet Magnetism) were those that did not interfere with the user’s aiming motions – instead, they only affected targeting after the shot. However, this study did not test multiplayer scenarios in a realistic FPS.

Commercial video games use aiming assistance mainly on console systems; PC-based games generally do not employ aiming assistance because mice are more precise than the game controllers used in consoles [20]. Despite their use in games, aiming assistance for multiplayer game balancing is not a central component of commercial games.

TECHNIQUES FOR FPS PLAYER BALANCING
We developed four player-balancing techniques for our realistic evaluation. The techniques vary in three ways:
• The game mechanic that is manipulated (e.g., aiming)
• The manipulation that changes the game mechanic (e.g., aiming can be changed using assistance algorithms that increase the effective width of the target);
• The application method that controls the manipulation, which includes elements such as the game statistic that determines when balancing is needed, the start and end of balancing, and the strength of the manipulation.

Our four techniques cover a range of possibilities for these factors, including different game mechanics, different manipulations, and different application methods. In addition, all our techniques make use of a basic application method that we call levels: the score differential between two players determines a level between 1-10 that is used to determine when to apply assistance, and how strong the effect should be. Larger score differentials (and thus higher levels) lead to increased assistance to the weaker player.
the score difference decreases, the effect tails off until it disappears entirely when scores are equal.

**Area Cursor with Levels**

Area Cursor balancing (Area) uses a 3D area cursor algorithm as the manipulation. With Area Cursors, players essentially fire a larger bullet (effectively making all targets larger). Normally, zero-extent traces are used to determine if the shot has resulted in collision with any enemies; area cursor instead uses a rectangular trace for collision detection. The area cursor algorithm is based on a typical 2D implementation [17] modified to work in 3D [24].

Our implementation in UDK does not change the size of the targeting crosshair, but still changes the size of the bullet trace based on the level (determined by score differential). The size of the rectangular activation area is 
\[10\text{px} + (10\text{px} \times \text{level})\]. As the level increases, less precision is needed in targeting. If there are multiple targets within the activation area, the system chooses the enemy closest to the center of the crosshair. Area cursors have been shown to improve targeting performance for older adults [26] and users with motor impairments [11], and was also effective in initial studies of aim assistance in FPS walkthroughs [24].

**Bullet Magnetism with Levels**

Bullet balancing is an aim-assistance technique that uses the bullet magnetism algorithm from [24] to adjust and improve aiming for weaker players. Bullet magnetism bends the path of a bullet towards any opponents that are within a certain angle of the initial shot, so that some shots hit that would normally have missed. Bullet magnetism essentially increases the width of the targets, making them easier to hit.

In the UDK system, Bullet Magnetism (Bullet) is implemented by adjusting a shot vector towards the first enemy that is within a certain range from the normal bullet path (100 UDK Units \(\times\) level + 100). The higher the level, the farther away the effect begins, and the more correction is applied. The algorithm corrects the shot towards the body of the enemy; if the aim is already on the body, the bullet is corrected towards the head.

**Bullet Magnetism with Levels and Delay**

Delay balancing (Delay) is similar to the Bullet technique, but changes the application method, from Levels to Levels plus Delay. Score differential is still used to determine the initiation and strength of the bullet-magnetism manipulation, but extends the time that balancing is active. Whenever the level of assistance becomes zero (indicating that the scores are now equal), the Delay method retains a level 1 assistance for 30 extra seconds. We developed this condition because during pilot tests we noticed that the novice could often get close to the expert’s score, but could not get ahead because the aim assist had been removed.

**Bullet Magnetism with Levels, Damage and Location**

This hybrid balancing technique (Combo) differs from the other techniques in that it adds two new game mechanics to the technique in addition to bullet magnetism-based aim assistance. First, Combo gives both players an indicator of the opponent’s location, as seen in Figure 1. This indicator can be seen behind walls and can indicate when the opponent is behind the player. Second, Combo modifies the damage done by a hit – both reducing incoming damage and increasing outgoing damage. For example, a player with level 10 assistance only needed two body shots to kill a player, and could withstand 50 shots before being killed. This technique was developed to investigate whether a combination of assists could be more effective than aim assistance alone.

![Figure 1. Game images: (a) bullet bending towards target in Bullet technique; (b) opponent location icon in Combo.](image)

**STUDY DESCRIPTION**

This study extends previous work in aiming assistance in FPS games, which established that aiming assistance can be an effective technique at increasing player performance in a single player 3D FPS scenario [24]. This current study involves matching participants with an opponent in a multiplayer PC FPS game, where participants of differing expertise were matched up and told their goal was to score more kills than the opponent. The goal was to evaluate the effectiveness of the different balancing schemes described in the previous section at balancing gameplay, and to see if the addition of the assistance would increase enjoyment and experience of both parties. Participants played one round with each technique with an additional round as the control (with no assistance). Participants were given an extra round at the start with no assistance for training.

The game the participants played, Mega Robot Shootout, was developed using the Unreal Development Kit (UDK). This UDK game was developed in the UnrealScript language, using Visual Studio 2010 and the NFringe add-on as the IDE. All sessions were played on 64-bit Windows 7 machines.
with comparable Intel processors and Nvidia graphics cards, Razor Imperator mice, and 22-inch LCD monitors with 60 Hertz refresh rates. Each participant was allowed to set custom mouse sensitivity. Logging was done to a Microsoft SQL Server 2008 R2 database.

Participants
We recruited fifteen pairs of participants, and they were compensated with $10. Participants ranged in age from 18-40 (mean 24.7). Participants were asked to fill out a questionnaire about their gaming habits and FPS experience to gauge their experience level and sort them into a “novice”, “intermediate”, or “expert” category. This was verified by watching their performance during the training round. The fifteen pairs consisted of five expert-novice, five expert-intermediate, and five intermediate-novice pairs.

Task
Each session involved one pair of participants (novice-expert, expert-intermediate, and intermediate-novice). Each pair was told that they would be testing different game balancing techniques and that some of the rounds they played may have game balancing enabled. The two players were then instructed to join the server and play a 1-on-1 deathmatch game. At the end of each round, participants filled out a questionnaire for subjective measures. The map was the default UDK “deck” map, which was customized to be smaller to accommodate only two players.

Each session consisted of one training round with no assistance followed by rounds of Bullet Magnetism, Area Cursor, Delay, Combo, and no assistance (order balanced across participant pairs using a Latin Square). Each round lasted 5 minutes except for the training round, which lasted 10. Each session lasted around 45 minutes including survey completion time. After each round, subjective questions were presented to the participants (detailed below).

Players could control their view and aim at opponents using the mouse. Shooting was controlled with the left mouse button and player movement was controlled with the standard WASD control scheme. During the training round, a full explanation of the controls was given to both players.

Both players had access to a minimap at the top right corner of the screen that showed the map of the level, their location (in white), and the location of their opponent (in red). This was meant to help novices, who had difficulty remembering both the map and the other player’s location. There was no way for players to heal themselves.

The players started each round with an assault rifle and a pistol. There were two locations that spawned sniper rifles the players could pick up. The assault rifle has a higher rate of fire than the sniper rifle but does less damage. The pistol is slower and more powerful than the assault rifle, and faster but less powerful than the sniper rifle.

Dependent Measures
To see if gameplay with aiming assistance is more balanced than without, we looked at several dependent measures across three categories.

Performance: Hit Ratio is the number of shots that hit a target out of the total number of shots. Kills is the number of times the player killed their opponent in the match. Score Differential is the difference in the number of kills between the players. Reversals are the number of times in a match that there were changes in which player was in the lead. Outcome refers to which player won the match.

Perception: Participants were asked to rate the amount of My Assistance, and the amount of Opponent Assistance.

Experience: The Fairness of a match was recorded as a simple fair/not fair answer on a questionnaire. The Competence, Autonomy, and Relatedness subscales from the Player Experience of Needs Satisfaction (PENS) scale of player experience [22] and Interest-Enjoyment from the Intrinsic Motivation Inventory (IMI) scale of player motivation [19] were used to gauge user experience.

Data Analyses
We conducted a repeated-measures Multivariate ANOVA (RM-MANOVA) with Balancing Scheme (Control, Area, Bullet, Delay, Combo) as a within-subjects factor and Pairing (Novice-Expert, Novice-Intermediate, Intermediate-Expert) as a between-subjects factor on the dependent measures of Score Differential (with each dyad treated as a single case) and Reversals. We also conducted a RM-MANOVA with Balancing Scheme as a within-subjects factor and Pairing and Expertise (whether the participant was the weaker or stronger player) as between-subjects factors on dependent measures of Performance (Kills, Hit Ratio), Perception (My Assistance, Opponent’s Assistance), and Experience (Competence, Autonomy, Relatedness, Enjoyment), treating each participant as a single case. Type 1 error was prevented by using the Holm-Bonferroni adjustment on all pairwise comparisons with \( \alpha \) set to 0.05; because the Holm-Bonferroni method was used to correct for familywise Type 1 error, individual \( p \) values for pairwise comparisons are not presented, but are less than 0.05 after the correction is applied. Sphericity violations were corrected using the Huynh-Feldt method of adjusting the degrees of freedom.

We also conducted chi-squared tests on the true/false question of “In general, did you find this round to be fair?” and on Outcome for each Balancing Scheme separately.

Our sample was of a reasonable size (N=30) to draw conclusions in a repeated-measures design; however, we report eta-squared values to give an indication of the amount of variance explained by each significant effect.

RESULTS
We present our results by answering questions about player performance, player perception, and player experience.
Performance

Were the games closer with balancing applied?

The RM-MANOVA for score difference showed a main effect of pairing (F_{2,12}=18.4, p<.000, η^2=.75), in which the novice-expert pairing had a greater score differential than the novice-intermediate (p<.000) or intermediate-expert (p<.000) pairings. The was also main effect of Scheme (F_{8,48}=27.7, p<.000, η^2=.70). Pairwise comparisons using the Holm-Bonferroni correction with α=.05 showed that there was a higher score differential in the Control and Area conditions than with the Bullet, Delay, or Combo schemes. Combo was also lower than Bullet. However, the significant interaction between Scheme and pairing (F_{8,48}=5.2, p<.000, η^2=.46) revealed that the differences of Scheme were significant for the novice-expert pairings, but not the intermediate/expert or novice-intermediate pairings as calculated with the Holm-Bonferroni adjustment (Fig. 2).

Did game outcome change as a result of balancing?

Although the balancing schemes helped to balance games through score differential, there was not a significant effect on outcome in terms of who won the match (determined by the relative number of kills when the game ended). Chi-squared tests of who won the game showed significant differences for each condition, indicating that the stronger player won significantly more matches, regardless of balancing scheme (all p<.02); see Table 1.

However, there was a significant effect of Scheme on the number of lead reversals that occurred throughout the match, i.e., when one player took the lead (F_{2,48}=4.63, p=.014, η^2=.28). Pairwise comparisons showed that Combo provided the most lead reversals, but that Bullet and Delay also improved over Control. Although there was also a significant effect of pairing on the number of lead reversals (F_{2,12}=5.4, p=.021, η^2=.48) – in which the intermediate-expert group had more lead reversals than the novice-intermediate or novice-expert group (see Figure 2) – there was no interaction between assist and pairing.

<table>
<thead>
<tr>
<th>Reversals</th>
<th>Control</th>
<th>Area</th>
<th>Bullet</th>
<th>Delay</th>
<th>Combo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weakers</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Strongers</td>
<td>15</td>
<td>13</td>
<td>13</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>Ties</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Χ^2</td>
<td>n/a</td>
<td>19.2</td>
<td>8.1</td>
<td>5.4</td>
<td>8.4</td>
</tr>
<tr>
<td>p</td>
<td>n/a</td>
<td>.000</td>
<td>.005</td>
<td>.020</td>
<td>.015</td>
</tr>
</tbody>
</table>

Table 1. Count of who won the game in each condition and mean (SE) of lead reversals by balancing scheme

Was accuracy affected by the balancing scheme?

To measure accuracy in performance, we looked at the hit ratio and the number of kills. Hit ratio is an individual metric, whereas kills also depends on the evasive abilities of the other player. As expected, there were main effects of expertise on both measures, with the stronger player having more kills (F_{1,24}=24.1, p<.000, η^2=.50) and a higher hit ratio (F_{1,24}=9.2, p<.001, η^2=.38) than the weaker players. There were also main effects of Scheme on both kills (F_{4,96}=13.5, p<.000, η^2=.36) and hit ratio (F_{4,96}=27.7, p<.000, η^2=.70); however, the significant interactions between assist and expertise on both kills (F_{4,96}=22.8, p<.000, η^2=.49) and hit ratio (F_{4,96}=11.9, p<.000, η^2=.33) show that the differences exist mainly for the weaker player who was provided assistance. Specifically, as can be seen in Figure 2, weaker players had a higher hit ratio with Delay than all other conditions and a lower hit ratio with Control than all other conditions, whereas there were no differences for the

Figure 2. Means +/- SE for dependent measures. Charts are split by expertise or pairing if there is a significant interaction.
stronger players between any conditions. In addition, weaker players had fewer kills with Control than with any other type and more kills with Combo than any other scheme (also more with Delay than Area), whereas the stronger players killed less often with Bullet than with Control or Area.

**Perception**

**Did players notice the balancing schemes?**

We asked players to rate the level of assistance that they were provided with on an 11-point scale, where 0 is none and 10 is high. Figure 2 shows that perception was generally rated quite low (between 2 and 4 points on the 11-point scale). Stronger players perceived balancing less than weaker players (F\textsubscript{1,24}=4.8, p=.038, \eta\textsuperscript{2}=17), which makes sense given that they would have rarely received assistance. In addition, there were differences in the perceptibility of schemes (F\textsubscript{4,96}=6.7, p<.000, \eta\textsuperscript{2}=22), in which players perceived the Combo technique more than the other approaches, with no other differences between the techniques. This is also not surprising as the Combo technique provided the visual location indicator that was not present in the other techniques (see Figure 1). There were no effects of pairing or other significant interactions.

We also asked players to rate the level of assistance that they thought their opponent was provided with on the same 11-point scale. A significant three-way interaction between assist, pairing, and expertise (F\textsubscript{8,96}=2.6, p=.014, \eta\textsuperscript{2}=18) shows that for novice-expert pairings only, the stronger player perceived the assistance given to the weaker player more in Combo than in Bullet, Area, or Control and more in Delay than in Area or Control. In the intermediate-expert pairings, the stronger player perceived their opponent’s assistance more in Combo than Control (see Figure 2).

**Were the games perceived as fair?**

We asked players to agree or disagree with the statement, “In general, did you find this round to be fair?” by answering either true or false. A chi-squared test on the frequency of responses split by whether the player was the stronger or weaker player in the dyad showed that the weaker players felt that games were fair only when assistance was applied, whereas the stronger players were divided on whether games were fair, regardless of provided assistance (Table 2).

**Experience**

**Did people experience greater competence with balancing?**

As expected, there was a main effect of expertise on perceived competence (F\textsubscript{1,24}=5.5, p=.027, \eta\textsuperscript{2}=19), in which the stronger player perceived himself or herself as more competent than the weaker player. The RM-MANOVA also showed a main effect of Scheme on competence (F\textsubscript{4,96}=3.8, p=.006, \eta\textsuperscript{2}=14). Figure 2 shows how the measure of perceived competence reflects the inverse of score differential. However, the significant interaction of Scheme with expertise (F\textsubscript{4,96}=6.6, p=.000, \eta\textsuperscript{2}=22) shows that the differences in perceived competence were only changing for the weaker player (see Figure 2). Specifically, the pairwise corrections showed that the weaker player felt more competent after using Combo than any other approach, and more competent after using Bullet than Control, whereas there was no difference in perceived competence for the stronger member of the dyad. There were no effects of pairings or interactions with pairing.

<table>
<thead>
<tr>
<th></th>
<th>Control</th>
<th>Area</th>
<th>Bullet</th>
<th>Delay</th>
<th>Combo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weaker</td>
<td>8/15</td>
<td>11/15</td>
<td>12/15</td>
<td>13/15</td>
<td>13/15</td>
</tr>
<tr>
<td>(\chi^2)</td>
<td>0.67</td>
<td>3.3</td>
<td>5.4</td>
<td>8.1</td>
<td>8.1</td>
</tr>
<tr>
<td>(p)</td>
<td>.796</td>
<td>.071</td>
<td>.020</td>
<td>.005</td>
<td>.005</td>
</tr>
<tr>
<td>Stronger</td>
<td>11/15</td>
<td>9/15</td>
<td>11/15</td>
<td>8/15</td>
<td>10/15</td>
</tr>
<tr>
<td>(\chi^2)</td>
<td>3.3</td>
<td>0.6</td>
<td>3.3</td>
<td>0.67</td>
<td>1.7</td>
</tr>
<tr>
<td>(p)</td>
<td>.071</td>
<td>.439</td>
<td>.071</td>
<td>.796</td>
<td>.197</td>
</tr>
</tbody>
</table>

Table 2. Count of responses of “FAIR” to the question of whether the game was fair, split by expertise.

**Did people experience greater satisfaction of autonomy and relatedness with balancing?**

There were no effects of Scheme, expertise, or pairing on relatedness or autonomy. There were also no interactions between factors on either measure.

**Did the effects from balancing translate into enjoyment?**

There were no effects of pairing or expertise on enjoyment. The RM-MANOVA showed a main effect of Scheme on enjoyment (F\textsubscript{4,48}=3.3, p=.015, \eta\textsuperscript{2}=12), in which the pairwise comparisons showed that the Combo and Delay techniques were considered more enjoyable than Control (Combo was also more enjoyable than Bullet). There was no interaction of Scheme with pairing or expertise, suggesting that all players experienced similar patterns of enjoyment regardless of their expertise within the dyad or the relative expertise of their opponent, i.e., enjoyment increased for both members of the dyad in conditions where assistance was most successful at balancing the score (Combo and Delay).

**Summary of Results**

Our results show that there are differences in the performance, perception, and experience of the various balancing schemes and that these differences often interact with whether the player was the weaker or stronger member of the dyad.

Specifically, although all assist techniques provided value, we can summarize our results with the following findings:

- Combo was best for balancing the games in terms of score differential and the number of lead reversals, although the stronger player still generally won the game;
- Combo and Delay were best at improving accuracy for the weaker player, with Delay improving the hit ratio of players and Combo improving the number of kills;
- Combo was the most perceptible approach, although mean ratings for perceptibility of all techniques were low;
- Combo was also most perceptible in terms of the stronger player sensing that their opponent received assistance, but only for expert-novice and expert-intermediate pairings;
- Combo and Delay were rated as most fair by the weaker player;
• Combo produced the highest perceived competence in weaker players, with Bullet and Delay following; there were no differences for stronger players;
• Combo and Delay were the most enjoyable approaches for both the stronger and weaker players.

**DISCUSSION**
In the following sections we consider explanations for our results, limitations to our work, the generalizability of our findings to other real-world FPS games, and topics for future research.

**Explanations for results**
Our main result was that all of the balancing schemes except Area worked, and some (Combo and Delay) worked extremely well. There are three underlying reasons for this success that can be valuable for designers. First, our work shows that player balancing is not just about the method of adjusting a single game mechanic – the way in which the adjustment gets applied is also important. Using the 30-second-delay application method made a substantial difference compared to the Bullet scheme alone. This method allowed us to deal with the problem of the weaker player getting near but never reaching the stronger player.

Second, the Combo technique demonstrates the value of combining multiple mechanics in one balancing scheme, which agrees with research using driving games [9]. Multiple techniques appear to be particularly important in FPS games, where there are several aspects to a player’s expertise [24]. The addition of several game balancing mechanics in addition to aim assistance allowed the Combo scheme to perform best in balancing – although this also may have led to it being the most perceptible scheme.

Third, one of our basic techniques (Bullet) performed better than suggested by previous work [24] – this technique did well especially for the large skill differences of novice-expert pairs. The improvement can be explained by the fact that we recognized the magnitude of difference between weak and strong players, and designed the application method (i.e., Levels) to provide a wider range of assistance than in prior work. Again, this shows that the method of applying the assist is as important as the assist itself.

Finally, it was interesting that although Bullet performed well, Area Cursor was not as successful. One reason is that the 3D implementation of Area was a direct transfer of the 2D implementation to work in 3D. This means that at greater distances, the area-cursor manipulation is less effective (i.e., the bullet is not as large as it seems when leaving the gun). This was not encountered in the previous studies that used the area cursor technique [24] due to opponents being in close proximity to the player. In our study, the large map meant more long-range engagements.

Our implementation of player balancing has the interesting quality that it scales with the expertise of the player. As players improve, the amount of assistance naturally declines, keeping games fair and balanced. Players can learn mastery of the skills needed to succeed in FPS play while the balancing scheme naturally adapts to their skill.

**Both individual and shared experience contribute meaningfully to experienced enjoyment**
As expected, weaker players showed differences in aiming accuracy (hit ratio) with the various balancing schemes, whereas stronger players did not. This was also reflected in the rating of perceived competence for the players, which suggests that weaker players attributed the boost in their performance to their improved abilities. On the other hand, because the stronger players did not vary in their accuracy, their ratings of competence also did not change, suggesting that this measure reflects their own abilities, not their ability in relation to another player.

However, the ratings for enjoyment seem to reflect the experience of the dyad, rather than the experience of each individual within the dyad – for both weaker and stronger players, the ratings for enjoyment reflect the score differential between the two players rather than the individual abilities of the player providing the rating.

This distinction is interesting, because Self-Determination Theory specifies that the perceived competence of the player should translate into experienced intrinsic motivation (as measured by the interest-enjoyment subscale) [22]. For weaker players, this is true – the aiming assistance helps the weaker player feel more competent, which translates into increased enjoyment. However, for stronger players, perceived competence does not vary with balancing scheme, as they were not provided with help and their accuracy did not improve. Although competence does not change for stronger players, their ratings for enjoyment do change, which is also not explained by changes in the satisfaction of autonomy or relatedness. In particular, the ratings of enjoyment for stronger players reflect the convergence in score that results from the effective balancing schemes.

Our results reveal how in our multiplayer FPS game, competence ratings reflect the individual’s performance (mirroring accuracy results), rather than the player’s performance relative to their opponent; however, in contrast, enjoyment ratings reflect the level of balance in the dyad (mirroring the score convergence), rather than the performance of the individual player. Therefore the increased competition and increased uncertainty of the outcome of games in which the weaker player is assisted are experienced as more enjoyable for stronger players too.

**Uncertainty in Game Outcome is Enjoyable**
In a study of internet chess players, Abuhamdeh and Csikszentmihalyi [1] found that enjoyment peaked when players held a small performance advantage over their opponent (slightly smaller than the value of a pawn). Although unexplained by self-determination theory, the authors hypothesized that the differences could be explained by the suspense of an uncertain outcome.
To investigate the role of suspense, they conducted further experiments with single-player games in which the player thought that they were competing against a computer (but the opponent was actually a confederate researcher) [2]. The authors showed that peak enjoyment occurred when players were slightly beating the opponent, that peak suspense occurred when players were slightly trailing the opponent, and that perceived competence increased with the difference in score. In addition, they showed that when given the choice to play one more round of a game in which they had either won by a wide margin or a slim margin, 69% of participants chose to play the game that they had won by the slim margin, reflecting higher intrinsic motivation for games with uncertainty in the outcome [2].

Our results mirror these effects in the case of multiplayer FPS scenarios. Our participants experienced greatest enjoyment with the balancing scheme that made the games closest, i.e., where the outcome was most uncertain. In addition, our stronger participants sensed that their weaker opponent was assisted, yet still felt the most enjoyment when outcome was uncertain.

**Practical Significance, Limitations, and Future Work**

One of our main goals in this research was to carry out balancing in realistic game scenarios. We succeeded – we used a real map and real weapons in a real game engine (UDK), and a type of competition that is used in realistic FPS play (deathmatch). Players were allowed to use whatever strategy they wanted, and we did not constrain player movement, evasive tactics, or other hallmarks of expert behavior. This degree of external validity suggests that our player-balancing techniques can be taken up by game developers, and can successfully improve the range of players who can enjoy the game.

However, there were also some limitations in our study setup that require further consideration:

*More than two players.* Both of our application methods use score differential in the two-player game to determine the onset and strength of balancing. In FPS scenarios with more than two players, using different game metrics may be necessary. For example, a metric such as kill-to-death ratio can indicate each player’s relative expertise, regardless of the size of the group.

*Larger maps with more resources.* Our UDK map was small, in order to provide more opportunity for conflict in the study sessions. In addition, our study map contained few spawning resources (e.g., health packs or powerups). In larger maps, expert behaviors such as the ability to memorize the map, or the ability to remember and utilize periodic resources could have a larger effect on score and outcome than what we saw in our study.

*Strategy vs game mechanic.* There are also aspects of FPS games that clearly differ between experts and novices, but that would be difficult to add to a balancing scheme. For example, higher-level strategies such as deployment of multiple players or selecting the best weapon for a particular situation clearly arise through experience, and the decision processes that go into these skills are not easily amenable to balancing. In addition, some strategies that could be part of a balancing scheme might be better left out. For example, experts have the ability to always know when to reload – and although novice players could feasibly be given an auto-reload assist, it may be that this would make the game feel too much like the computer is controlling all aspects of the gameplay.

*Longer-term effects.* Our sessions were short, and players did not build any long-term experiences with balancing. It is possible that our schemes will perform differently over a longer term. For example, stronger players who take great pride in their abilities may begin to dislike the fact that novices get an advantage; or, it may be that experts find exploits that allow them to outwit or negate the balancing strategies. Our participants appeared to like the fact that there was strong competition, however, and so we do not expect either of these situations to occur for most players – but further work over longer time periods is clearly needed. Further work is also needed to examine how aiming assistance may affect skill development.

In addition, future work should consider our results for other game genres, such as fighting or strategy games. Finally, applying player balancing could have impact in non-game playful applications (e.g., collaborative planning or decision-making efforts) or in productivity applications (e.g., meeting systems that aim to balance the contributions from different speakers).

**CONCLUSION**

Player balancing techniques attempt to assist weaker players in order to increase the level of competition in a game. These techniques have been shown to work in some games, but have never been used successfully in FPS. We developed new player balancing schemes based on three ideas: extending the range of assistance, changing the way that a manipulation is applied, and adding more game mechanics to the scheme. We tested these techniques in a controlled study using one-on-one deathmatches in a realistic FPS game developed with UDK. Our results showed that the new balancing schemes were extremely effective – they were able to balance even amongst players with large skill differences. Surprisingly, the techniques that were most effective at balancing were also rated as most enjoyable by both players – even though these schemes were the most noticeable. Our study is the first to show that player balancing can work well in realistic FPS games, providing developers with a way to increase the audience for this popular and competitive genre. In addition, our results demonstrate the general principle that successful balancing is as much about the way the technique is applied as it is about the specific manipulation.
ACKNOWLEDGMENTS
We thank the GRAND NCE, SURFNET, and NSERC for funding, the members of the Interaction Lab at the UofS for their support, and our participants for their time.

REFERENCES