AI for Dynamic Team-mate Adaptation in Games

Aswin Thomas Abraham and Kevin McGee

***Abstract*—There is a long tradition of developing games in which the difficulty level is dynamically adapted to the performance of human players. However, there has been less work on the creation of game systems that perform dynamic team-mate adaption – and even less on developing team-mate NPCs (Non Player Characters) that adaptively support players in the face of opponents that adaptively increase the difficulty for the player. This paper is based on preliminary research to identify the key elements involved in developing "buddy" NPC team-mates that dynamically adapt to the needs and behaviors of human players while cooperating to compete against adaptive AI opponents. We discuss the computational and design challenges involved in developing such agents in the context of a simple test game called *Capture the Gunner (CTG)*. The main contributions of the paper include: a proposed vocabulary and framework for understanding/modeling team-mate systems with adaptive difficulty, a particular technique for adaptive team-mate cooperation in the face of an adaptive opponent, and the identification of several significant new issues that arise in the process of developing computer games that involve adaptive NPC team-mates that cooperate with the player in the face of adaptive opponents.**

I. Introduction

An interesting challenge in AI is to develop characters that adapt to the player. Systems with adaptive opponents make use of "escalating tension" or "Dynamic Game Difficulty Balancing" (DGDB)[1] which involves making automatic, real-time changes to aspects of a game based on player performance. A more recent trend is to develop systems with adaptive team-mates that play alongside or under the command of the human player. Before describing our own work, we review work that has been done to develop adaptive NPC team-mates for team-based computer games – including team-based games that have adaptive opponents.

II. Related work

There has been work to date on adaptive difficulty ("opponents") in non-team games – and some work on adaptive AI for "ally" NPC team-mates in team-based games. Below is a brief summary; for a more extensive survey and critique, see [1].

Adaptive Difficulty There are numerous examples of adaptive difficulty in games. A few among them include the algorithmic increase of speed in *Tetris* (and many other classic arcade games), increase of opponent aggression in the game of *Halo*, adaptive opponent attack in *Max Payne*, removal of the radar in *Metal Gear Solid* with level up, improved combo attack in *Street Fighter*, and adaptive zombie spawning in *Left4Dead*. Research has explored different metrics of player performance [2], [3] or means of increasing the difficulty for the player [4].

Adaptive "Ally" Team-mates Examples of adaptive team-mates in games include the creation of allies for team sports games (NPCs that avoid blocking the player and taking passes only intended for them in *FIFA 10*); action games (allies that heal the player in *Left4Dead* or allies that drag the player to safety in *Killzone 2*); RPGs (allies with immediate and delayed responses to player action in *Neverwinter Nights 2*); God games (units which follow player commands in *Black & White*); FPSs (allies that shoot objects indicated by the player in *Call of Duty*); platformer games (providing a mask to get past levels in *Crash Bandicoot 3*); and even driving simulation games (driving assistance in *Need for Speed: Shift*). Research on adaptive team-mates tends to be very genre-specific, focusing on the appropriate team-mate decisions necessary for team-based sports games [5], [6], [7], [8], action games [9], [10], [11], [12], [13], [14], [15], RPGs [6], [16], and RTS games [17], [18].

Adaptive "Ally" Team-mates Against Adaptive Opponents Finally, there exist a few games in various genres where team-mates adapt to the player while the opponent adapts the difficulty of the game. In *Left4Dead*, the player allies provide cover from all directions and heal the player when in low health status; the spawning location of the opponents (or zombies) vary, their number increases and more stronger members are added on to increase the difficulty. In *Warcraft III*, the allies when commanded, attack the opponents in a coordinated and aggressive manner; the opponent construction strength increases, enabling them to build stronger towers and buildings, making the game harder. In *Baldur's Gate*, there are team-mates that cast spells (e.g. for healing), intelligently select short and long range weapons, perform coordinated attacks and defenses; opponents with higher power levels are introduced as the player skill level goes up. In the game of *Madden NFL*, if the player's team is being defeated, team-mates adaptively improve their team-work; the opponent AI on the other hand, quickly adjust to the player team winning strategies and begin stopping the player's favourite plays. In terms of research, there are contributions in which the team-mate cooperates while the game complexity increases [19] or varies [20].

III. Research Problem

Although there has been work to automatically adjust game difficulty, as well as some work on AI systems that adaptively cooperate with human team-mates, there has been very little

Aswin Thomas Abraham and Kevin McGee are with the National University of Singapore, Singapore (email: {aswinthomas,mckevin}@nus.edu.sg).

[1]Also known as "Dynamic Difficulty Adjustment" (DDA) or "Dynamic Game Balancing" (DGB).

work that formalizes the development of adaptive autonomous agents that cooperate with human players against one or more adaptive opponents. Indeed, although the section on "related work" points to a number of examples, it is not much of an exaggeration to say that the state of the art on team-mate AI is still much the way Jansen described it in 2007: "In most current computer games however, artificial intelligences do not really cooperate with their companion. They co-exist beside their partner, doing their own tasks, without attacking the partner player" [18].

The purpose of the current paper is to report on our recent work to develop an autonomous "buddy" team-mate that cooperates with the player in the face of an opponent that adapts the difficulty of the game. To this end, we have developed a game – *Capture the Gunner* (*CTG*). The game is trivial in its simplicity; nonetheless, it allows us to explore and address a number of problems fundamental to the creation of autonomous, "buddy" team-mates that support a player competing against an adaptive opponent.

The main contributions of work reported here are:

- Basic components in an agent that behaves adaptively to both a team-mate and its adaptive opponent are formulated.
- The modules identified for the team-mate and adaptive difficulty adjustment for the opponent are analyzed in the context of developing a working game.
- Challenges and design considerations in developing adaptive team-mates are also discussed.

IV. System description

There does not seem to be any existing consensus about how to describe the components of a game system that makes use of adaptive difficulty – nor does there seem to be any published work that has identified the main components of adaptive AI team-mates. Therefore, before describing the system, we introduce some vocabulary and concepts relevant to analyzing and modeling NPC team-mates.

Adaptive opponents typically consist of:

- a *performance-evaluator (PE)* which measures player performance in terms of overcoming a particular type of challenge. The *PE* often involves simple parameters which are related to the survival of the player (e.g. player lives left, health remaining), damage estimators or even the likelihood of player death or involve the rate of success in completing the challenge at hand (e.g. time lapsed, numbers of won and lost, density of blocks etc.).
- an *opponent adjustment mechanism (OAM)* that changes some aspect(s) of the game (or the AI opponent's behavior) in order to increase/decrease the difficulty. *OAMs* typically involve simple parameters or sophisticated adjustment goals which are a set of configurations and intervention strategies employing a combination of parameters. Simple adjustment (common in classic games) involves an increase (or decrease) of parameter values for such things as speed or size. More complex parameters include random speed levels, challenges of increasing complexity (e.g. increased block complexity), reducing the functionality (e.g. perception or weapons) available to the player via the interface/tools, or providing a mix of challenges which allow players to master a few basic techniques and then later require players to use them in combination.

In light of the above, the main elements of an adaptive team-mate system can be described as follows:

- a *team-mate and opponent evaluator (TOE)* which measures human performance (from the perspective of the NPC team-mate) and AI opponent(s) performance. A *TOE*, in general, involves learning about, modeling, predicting, and reasoning about player and opponent actions. The *TOE* is therefore similar to *PE*. However, while *PE* may focus on an individual (player) or the player team, the *TOE* always evaluates on an individual basis (player, opponent).
- a *team-mate adjustment mechanism (TAM)* that supports the player and hence promotes team success. Depending on inferences from *TOE*, the *TAM* coordinates the team-mate positioning, movement, provides supportive actions, complements player preferences and even prioritizes the player.

To study and understand some of the issues about adaptive challenges, we have developed the *Capture the Gunner* (*CTG*) game (shown in Figure 1).

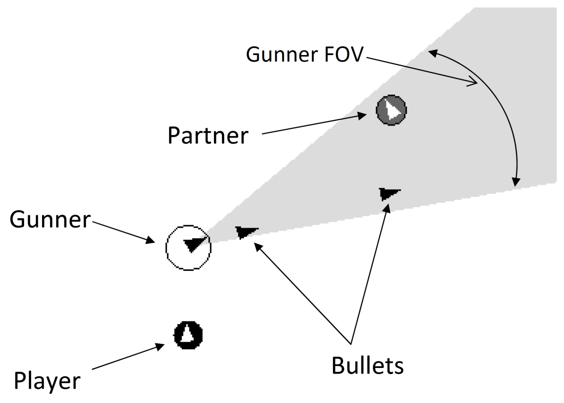

Fig. 1. Snapshot of the game Capture-the-gunner (CTG)

The game consists of an autonomous NPC gunner, a player-controlled character, and an autonomous NPC team-mate (the player's ally). The gunner adaptively attempts to kill the player or player's team-mate; the NPC team-mate adaptively attempts to both capture the gunner *and* to ensure that the human player also succeeds at capturing the gunner (thus, the player plus team-mate function as a team that attempts to capture (touch) the gunner). The team completes a level when both team members touch the gunner without getting shot – at which point the game is reset and adaptively increases the difficulty for the player's team to capture the gunner. The game ends when either the player or the team-mate are killed.

The gunner sits in the center of the screen and is displayed as a white circle with a black arrow head indicating its direction of focus, the player-controlled character is the black circle with a white arrow-head indicating the character's heading, and the autonomous team-mate is the dark grey circle with a white arrow head indicating the heading of its movement. The gunner is capable of sensing any object within its *visual* field-of-view (displayed as light grey) and has an unlimited supply of bullets (black arrow heads). The gunner also has an accuracy of aim and a firing rate which are both constant – and it tracks the target continuously by rotating at a speed that is constant for the level. The player controls the character via arrow keys and is capable of moving in four directions (*north, south, east* and *west*). The team-mate is capable of three movements: *rotating around the gunner*, *moving towards gunner*, and *moving away from gunner*.

It is assumed that there are at least two individual actions that can be performed by either the player or the autonomous team-mate.

- *Pursue the gunner:* move directly forward to touch the gunner.
- *Draw fire:* enter (or stay within) the gunner's field-of-view in order to distract the gunner away from the other team-mate. While doing so, it attempts to avoid getting shot by the gunner.

Figure 1 shows the team-mate trying to draw away fire by staying in the gunner field-of-view without getting shot. The player, on the other hand, is moving forward to touch the gunner.

In order to capture the gunner, *both* team members must touch the gunner from opposite sides without getting shot. The capture area on the gunner is highlighted once either the team-mate or player comes in contact – making it clear where the other team member must touch the gunner. This is illustrated in Figure 2 where the player has touched the gunner and the capture area of the team-mate is labeled and indicated in dark grey.

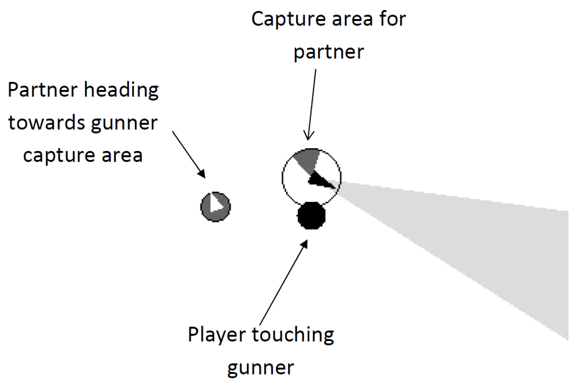

Fig. 2. Capturing the gunner

Note that once a team member comes in contact with the gunner, the gunner does not exploit this information to focus on the opposite side and and simply wait for the partner to shoot it; in this game the gunner continues to track and fire upon the team as before. Also, even if the player touches the gunner, he is not safe from getting shot – after the first team-mate touches the gunner that team-mate needs to keep itself alive until the other team-mate also succeeds in touching the gunner. Once both team-mates touch the gunner the level is complete, the game restarts with the gunner and NPC team-mate performing at a slightly higher level (see below).

A. Gunner Algorithm

This section describes the gunner algorithm: how it chooses to target either the team-mate or the player, how performance is evaluated, and how difficulty is adjusted.

There are three situations in which the gunner has to make a decision on which team-mate to target.

1) **Both opponents visible.** In this situation the gunner always chooses the target closest to it in linear distance.
2) **One opponent visible.** If the position of either the human player or team-mate is unknown, the gunner continues rotating to acquire it. Once both positions are known, the gunner chooses the target closer to it in linear distance. If the player is closer, the gunner spends time $t = \frac{0.1d}{4}$ in total (where d is the distance from the team-mate) for tracking and shooting the player. Thus, the time spent on the player is a function of the team-mate's distance from the gunner. The same procedure applies if the team-mate is closer.
3) **No opponents are visible.** When neither the player nor the team-mate is visible, the gunner searches for them by rotating in the direction it was rotating previously. This search continues until either the team-mate, the player or both of them are visible.

Once the target is chosen, the gunner rotates toward it and fires bullets at it. The rate of rotation and firing are within a range of accuracy that remains constant through-out the game.

1) Performance Evaluator (PE): Performance evaluation is in terms of the *team* completing a level. As discussed earlier, a level in the game is complete when both the player and team-mate cooperate in capturing the gunner. The capture area for the gunner is set $180\,^\circ$ apart from the present touched location on the gunner (as shown in Figure 2).

2) Opponent Adjustment Mechanism (OAM): The difficulty or the opponent strength is improved by increasing the gunner rotation speed. All other parameters remain constant (e.g. bullet firing rate, gunner field-of-view, firing accuracy etc.). The difficulty adjustment curve is a geometric series $a + ar^2 + ar^3 + ...$ where a is the initial gunner rotation speed of $25\,^\circ/sec$ and $r = 0.5$. Thus, the gunner starts rotating at a speed of $25\,^\circ/sec$ (level 1), then increases to $43.24\,^\circ/sec$ (level 10) and beyond. Since the PE is based on level completion in the game, the OAM for the system is executed at the end of every level.

B. Team-mate Algorithm

This section describes the team-mate algorithm: obtaining player and gunner intentions to decide on the task of *drawing away fire*, *pursuing the gunner* and finally *capturing* the gunner. The manner in which the autonomous team-mate provides adaptive support to the player by prioritizing the player and complementing player actions are also discussed.

1) Team-mate and Opponent Evaluator (TOE): The team-mate draws conclusions about the player and opponent beliefs based on which team member the gunner is tracking. There are two conditions under which the team-mate has to make a decision:

- **Gunner tracks the player.** If the player is tracked by the gunner, then the team-mate *pursues* (move towards to touch) the gunner.
- **Gunner tracks the team-mate.** If team-mate is being tracked by the gunner, then the team-mate *draws fire* away from player (by moving in the opposite direction).

Since the gunner tracking could result from the gunner's or the player's intention, these simple rules help the teammate succeed in evaluating them.

2) Team-mate Adjustment Mechanism (TAM): The TAM for the system involves supportive actions through positioning and movement, prioritizing the player and adapting to the adaptive opponent.

- **Supportive positioning and movement.** This consists of staying out of the player's way, drawing fire, and pursuing the gunner.
  The team-mate *stays out of player's way*. The team-mate does so by always rotating in a direction away from the player. This ensures that the team-mate does not block the area between the player and the gunner, hence causing no disturbance to the player in doing his task.
  While *drawing fire* from the gunner, the team-mate stays closer to the gunner than the player. The team-mate does so by moving to a closer position along the same line between itself and the gunner. This ensures that the gunner opts to target the team-mate. When the player moves closer to the gunner, the team-mate also moves in. When the player moves away from the gunner, the team-mate also does so, but making sure that the gunner always targets it. The team-mate here adapts to the player movement. However, if the player has already touched the gunner, the partner abandons drawing fire and moves forward to the gunner capture area – to complete the level. Therefore, in order to capture the gunner, both team members have to be *pursuing* it.
  When *pursuing* the gunner, the team-mate rotates until it is sufficiently opposite the player and then goes in close proximity to the gunner. For this, the team-mate ensures that a minimum separation of at least 150° is maintained between itself and the player. This is to make it easy for the player to hone in to touch the gunner, once the team-mate does so. The team-mate also circles around the gunner, waiting for the player to come close while giving him several opportunities for coordinated capture. Therefore the team-mate waits for the player to abandon his task of drawing away fire and capture the gunner along with the team-mate.
  The team-mate also *avoids being shot* by bullets. When in close proximity to bullets, the team-mate starts rotating around the gunner until no bullets are close by.
- **Prioritizing the player.** The team-mate always assumes a role that is not taken up by the player. So if the player chooses to *pursue* the gunner, the team-mate *draws fire* away from the player and vice versa. Even in the special case of final gunner *capture* where both team members have to *pursue* the gunner (as mentioned above), priority is always given for the player to either abandon the task of drawing fire or go in first to touch the gunner while pursuing it.
- **Adapting to opponent.** After every capture of the gunner, the team-mate improves its speed at a rate proportionate to the amount by which the opponent adapts the difficulty i.e. at the end of every level. This is reflected as an improvement in several team-mate behaviors including staying out of the player's way, drawing fire, pursuing the gunner, and avoiding being shot.

V. Discussion

This section analyzes the proposed adaptive mechanisms such as difficulty adjustment for teams, adaptive team-mate support and justifies the design and implementation choices in the game of *CTG*. Models of team-mates and characteristics of cooperating team-mates are also discussed.

A. Performance Evaluator (PE)

In *CTG*, the team performance evaluation is based on *both* the player and the NPC team-mate succeeding. In this section we discuss alternatives.

The simplest way to evaluate the player/team-mate performance would be to monitor whether either the player or the team-mate touches the gunner; and then allow the player to move to the next level. This approach, whereby the difficulty will increase based on the success of *either* the player or the team-mate, seems like a natural extension of the model used in games such as *Tetris*. However, this approach to difficulty adjustment in team-based games has a significant draw-back: the game may progress based on the autonomous team-mate succeeding at some, most, or even all of the levels. In other words, this approach might work fine if the team-mates were *both* human players, but if we want to ensure engagement for the human player in *CTG*, we need something else.

Said another way, since the focus of team based games such as the *CTG* is cooperation, teams should be evaluated based on how well they cooperate. Therefore, the PE proposed in this paper evaluates the *combined* performance of player/team-mate. There are numerous ways to do this, but one should be cautious. For example, one possible combined measure is the time the team takes to complete a level. However, even though well-coordinated teams could exhibit quick behaviors, teams in combat situations such as ours, often employ time-consuming strategies that allow them to successfully attain the goal by adopting clever planning and minimal risks. Such parameters also make the implementation complex and unreliable.

To further understand the choice, consider the characteristics of a strong collaborative team [21]:

- Individual plans: An individual in the team must have knowledge of the overall task at hand, must be able to perform any sub-task and must have intentions to perform it. As discussed before, the main task of the team is to *capture the gunner* and have two subtasks of *pursuing* the gunner and *drawing fire*. If a team-mate is either not able to or does not have any intention to perform a sub-task undertaken, a change of roles take place. In order for the team to succeed, another team-mate must switch over to undertake the abandoned sub-task, compensating for the weakness in the team. If this behavior is frequent, this prevents the team from succeeding.
- Group plans: Individuals in the group must have a commitment to the overall task and belief in the team-mate's ability to succeed in their actions. For completing the overall task, a team-mate does a subtask after inferring the intentions of the other (as discussed in Section IV). The team-mate also believes that the player can do his job and does not bother him. However, any discrepancy in mutual intention recognition and commitment results in team failure.

From the above, it is clear that simply completing a level involves good individual performance, mutual understanding and coordination. Therefore, the choice for the performance measure is justified.

B. Difficulty Adjustment

In *CTG*, the difficulty adjustment parameter is the speed of rotation – and this adjustment is made as a geometric progression.

There are numerous ways in which the difficulty for the game of *CTG* could have been adjusted: bullet firing rate, bullet speed, gunner field-of-view and gunner shooting accuracy. Each of these was tried and was fairly successful in increasing the game difficulty for first few levels, after which the tension in the game did not seem to increase. The game then appeared to be boring and monotonous.

One major drawback in all of the above, lie in their theoretical upper limits. For example, the maximum gunner field-of-view could only increase to $360\,^{\circ}$. Apart from this there are other disadvantages. For example, an increase in the gunner field-of-view may increase visibility and promotes faster decision making. However, the speed at which the action is performed, remains the same throughout in all levels since the gunner has to turn at the same speed to shoot a target it chose. An abnormally high rate of bullet firing and bullet speed turned out to be an unfair challenge to the player. A gunner with lower accuracy shoots a target with an offset. This makes it evident to the player that the system is deliberately making the game easy for him and is therefore not a convincing or interesting behavior. It may be true that in many games, the difficulty has to be adjusted using several parameters over several levels. However, choice of a single parameter reduces the burden on the developer as he does not have to design each level. Therefore we chose to increase the gunner rotation speed, which was not only observed to be convincing and fair, but also felt that the game was increasingly challenging and interesting. The parameter is also scalable over various levels and therefore is considered to be a reasonable choice for difficulty adjustment.

Once the difficulty adjustment parameter has been selected, there is still the question of how the parameter will be varied. The main aim of a difficulty adjustment profile is to increase the difficulty such that the player experiences an even increase in challenge. One approach would be to increase the speed in a linear manner. However, it turns out that a linear profile does not bring about a satisfying experience. During testing (as discussed in Section VI), it was observed that a linear speed adjustment profile is highly challenging for the player and forces him to stop playing at low levels. An alternative is to start the first level at a very low gunner speed, but this introduces a new problem: players may find that the increase in difficulty takes too long.

CTG therefore makes use of a non-linear function, with gunner rotation initially quite slow, but increasing dramatically at lower levels (but progressively more slowly at higher levels). This pattern of variation in the gunner speed seems to create the right level of challenge.

C. Inference mechanisms

In *CTG*, the intentions of the opponent and the player are inferred by monitoring which team member the gunner is targeting (as discussed in Section IV). Before finalizing such an implementation choice, other parameters were tested.

Among them, one of the obvious parameters is the *time duration* the player stays in the gunner focus. If the player is in the gunner focus for a considerable amount of time, the result could be due to the gunner's intention (i.e. gunner chases player) or the player's intention (i.e. player wants to be chased by gunner) or both. However, it was observed that relying on this parameter resulted in a team-mate behavior that was uncertain. There were few instances where the team-mate did not choose an appropriate task which prevented the coordinated capture of the gunner. Since the agents in *CTG* are reactive, numerous activities occur in a short interval of time. Even when the team-mate did succeed in producing a satisfactory behavior, the team would sometimes spend long periods of time to complete a level in the game.

To make this more robust, we experimented with monitoring the *angular distance* between the player and team-mate. If the angle is small, it signifies that the player is trying to come close and do what the team-mate is doing – which gives the team-mate an indication that it should change the task it has undertaken. However, this made the system complex and did not improve the system as expected.

Another parameter explored was the *linear distance* of the player from gunner. The assumption was that if both the team-mate and player are in the gunner field-of-view, and the player is closer to the gunner, then the player intends to be chased by the gunner; and if the player is not in the gunner field-of-view, then the player could be trying to touch the gunner. Even though this implementation seems to enable the team-mate with the required behavior, the inferences about player intentions were varied and unreliable. This is because the gunner is always in motion and both team members fall in its field of view during rotation – causing the team-mate to change its decisions often.

Therefore, we finally chose to have the team-mate monitor which team member the gunner currently targets. Although there are several reasons why the gunner would target a team member, for any of those reasons it quite practical to immediately undertake the task of *drawing away fire* from the other team member – rather than trying to move away from the gunner field-of-view and continue *pursuing* it. This not only reduces the time involved in team-mate decisions in the rapidly changing environment, but also results in a smooth team-mate movement.

D. Models of Team-mates

There are numerous models that can be applied while designing a team-mate. A few among them which were considered during the development of *CTG* are described below.

Master-Slave model. In this model, the team-mate is a slave and the player commands the team-mate to do tasks. Since the complete decision is made by the player at a "manager" or "general" level, there is no room for conflicts and the behaviors are executed faster. However, this does not lead to an effective cooperation. There is either no or little artificial behavior for cooperation embedded into or generated within the team-mate system. For example, in a RTS game the player demands units to move from one place to the other. The units are capable of movement, but do not require cooperation with the player. Our aim however, is to develop synthetic team-mates that cooperate and play alongside the human player.

Semi-autonomous slave model. A related but slightly different variant is the semi-autonomous slave model. The human player commands the team-mate when he desires. If not, the team-mate behaves autonomously. Such behavior is sometimes termed *adjustable autonomy* [22]. Consider an example where both the player and team-mates are hiding in an FPS game. The player now commands the team-mates to go forward to check the level of danger in the area. The team-mates cooperate and split up while executing low level operations autonomously. Even though there is some amount of coordination involved between team-mates, they do not cooperate with the player. The player in this model is always at the commanding level like the master-slave model.

In order to circumvent problems with the master-slave and the semi-autonomous slave model, we have not provided the player with the ability to provide commands. Even though communication is a typical aspect of team-based games, the problem (for our purposes) is that skilled human players can take control and complete a level in the game without cooperating with others.

Clone model. A different model of cooperation is the team of clones. This represents a team of agents that have equal abilities and roles. One example is that of collaborative robot map building. A set of robots communicate and share their plans to complete the task of mapping out an unknown area. The main motivation in using the model is to use more agents and finish the task in polynomial time. Furthermore, a single agent is capable of completing the overall task. In the actual problem of coordination where a team of agents play against an adversary, time is not the main concern. It is also preferred that team-mates cooperate in completing the overall task – rather than by themselves. Therefore in *CTG*, there are unique sub-tasks, cooperation is made mandatory (as discussed earlier in this section), and both the player and team-mate have different movement abilities (as discussed in Section IV).

"Buddy" model. We designed *CTG* as a "buddy" model in which the team-mate and the player are equally capable – and also have comparable weaknesses. In other words, does not just behave like the player's slave.

The need for an autonomous "buddy" team-mate really comes into focus when we consider the problem of the specific adaptive opponent in *CTG*. One might imagine it would be possible to make use of some of the team-mate techniques used in, say, *RoboCup*. However, in such games, the team adapts as a whole (e.g. by changing formations) by observing and imitating the opponent. But in the case of *CTG*, the opponent has very different skills, weapons, and behaviors. A related limitation is that in *RoboCup* (and in many software implementations of NPC teams), there is no human player. Thus, the team-mate in *CTG* cannot "do what the player does" in any simple sense (or, actually, in anything except a simple sense).

The fact that the team-mate in *CTG* is not just an intelligent slave can be seen in two examples:

- When the team-mate is focusing on rotating in a direction away from the player, there were instances in which it got shot by the gunner. The team-mate cannot always avoid bullets while making sure it does not block the player.
- The team-mate shows difficulty in choosing a direction to rotate when the gunner is exactly in between the team-mate and player. The gunner has an advantage in this situation and shoots down the team-mate if the player does not act.

In both the cases, the player can act to save the team-mate from being shot. This requires understanding the team-mate's limitations and that the player's actions affect the behavior of the team-mate as well. The player and the team-mate in the team are inter-dependant. In fact, there is a quite subtle issue that arises because of the fact that the team-mate improves its speed at the rate at which the opponent adapts. This means that the team-mate indirectly employs the team performance evaluation. This is interesting because the team-mate adapts depending on how the entire team performed – that is, based on the performance of the team of which the team-mate is a part. This co-evolution of the opponent and team-mate provides the player with a challenging and supportive playing experience. Unlike the gunner, the strength of the team is dependant not only on the team-mate which infers player actions, opponent actions and team performance, but also the game learning curve and adaptation of the player to the team-mate.

In this context, it is worth saying a few words about prioritizing player participation [23]. In *CTG*, both the opponent (gunner) and the team-mate are constrained to adapt in ways that *prioritize* the player's enjoyment and participation. This means that the opponent cannot simply try its best to defeat the player at the expense of adaptive challenge – and the team-mate must not overcome the opponent (or help the team overcome the opponent) at the expense of player participation.

In terms of the team-mate, this does not mean that the player must be given total control or must be able to make the team-mate do uninteresting tasks, but rather that the team-mate must give the player priority in making choices. In an FPS this would mean that the player would be the first one to collect ammo or shoot enemies. In the game of *CTG*, we give the player the freedom of choosing any task to do. The team-mate infers the player intention and chooses a different task. Even though player prioritization improves the player experience, it promotes team success under the assumption that both the player and team-mate are both quite good at executing their individual tasks. Consider the scenario in which the interval between the bullet firing is small and the player cannot handle the job of drawing fire from the team-mate well. Since the player does a poor job of distracting the gunner, the gunner moves over to the team-mate and then the team-mate assumes the role of *drawing fire*. If such task switching is frequent, it affects the team performance. Therefore, even though we follow the buddy model in *CTG*, the team-mate gives priority for the player. This is simply to enhance the player experience.

E. Traits of cooperating teams

There are a number of additional issues about cooperative teams that arise in the context of analyzing *CTG*.

Choosing non-conflicting sub-goals. Agents in a team cooperate by choosing appropriate non-conflicting sub-goals that lead to the common goal of the team. Team-mates could choose to perform sequential sub-tasks (i.e, tasks that could be performed in sequence and then summed together). However, when a team is involved, it is efficient if each member can perform several sub-tasks simultaneously. Furthermore, as Grosz [21] points out, collaborative activity is not just sum of individual plans. In the game of *CTG*, the team-mate and player perform independent tasks simultaneously and in parallel. The contribution by each team member is more or less equal. This is in contrast to team-games such as football where the action execution depends on acquiring a token (i.e. ball) and the contribution made by each member in the team is uneven at any point of time.

Evolving plans. Another characteristic of cooperative teams is the ability for plans to evolve. There are many systems in which cooperation and plans are built into the system right from the start. However, these systems fail in dynamic circumstances. In the game of *CTG*, there is an adaptive opponent which causes team-mates to change plans and switch tasks. As the game progresses, the execution of plans and strategies for completing a level must vary for the team to succeed. The game of *CTG* is vaguely analogous to the situation in football in which two players near the goal face the goal-keeper. One of the players in the situation has to shoot the ball and the other has to pass the ball to the team-mate. If the goal-keeper moves closer to one player, he has to pass the ball and the team-mate shoots the ball into the goal, exhibiting good coordination. However, the team will not always succeed if it makes use of the same strategy each time it confronts a similar situation. In *CTG*, there is no ball visible to the gunner, but task switching takes place between the player and team-mate.

VI. Evaluation

As yet, there has been no formal evaluation or rigorous testing of *CTG*. However, roughly 20 different players have played the game and there is some informal feedback.

Capturing the gunner. Although the player and team-mate are required to maintain an angular separation of 120° while capturing the gunner, in the initial version of the game players were confused because it was not always clear where they needed to touch the gunner after their team-mate had done so. The visualization of the capture area was therefore introduced (as discussed in section IV).

Difficulty increase. In early versions of the game the difficulty adjustment for the gunner was linear. This prevented most of the participants from going past level 2 and none of them beyond level 4. The difficulty increase profile was therefore changed to be a geometric series as discussed in section IV (more on this in Section V).

Team-mate's fault. There were cases where the team lost because the NPC team-mate got shot by the gunner. Players were asked whether they felt this was "unfair" or otherwise made the game unappealing; most players found it acceptable to lose the game because of the failure of the NPC team-mate. In fact, some players said that, since the player has so much influence on the team behavior, they felt that losing as a result of NPC team-mate being killed was partly their (player's) fault/responsibility. It is interesting to note that none of the players felt that the team-mate was inadequate to the task.

Waiting for the team-mate. Many players initially maneuvered in the world without considering the team-mate: it was quite common for players to immediately go and touch the gunner by escaping its field of view. Players would then wait for the team-mate to move around and touch the gunner while avoiding bullets. Such players/teams all died at this first level: either the player got shot (when the gunner turned towards them) or the team-mate got shot (due to the number of overwhelming tasks of avoiding bullets, prioritizing player, and moving to the opposite side for gunner capture). While some players adapted their play to be more cooperative, others continued to ignore the partner. The ones that continued to ignore the partner never made it past the first level which suggests that the game does indeed support team cooperation.

VII. Conclusion and Future Work

As we have seen, there are numerous challenges in building team-mates for games equipped with opponents with adaptive difficulty adjustment mechanisms. The challenges in developing such team-mates include issues such as player influence, maintaining flow, modeling beliefs, and online learning – but these are also opportunities for future innovation in the development of AI for dynamic team-mate adaptation in games.

VIII. Acknowledgments

This research is supported by the Singapore-MIT GAMBIT Game Lab research grant "Designing Adaptive Team-mates for Games"

References

[1] K. McGee and A. T. Abraham, "Real-time team-mate AI in games: A definition, survey, & critique," in *FDG 2010*, 2010.

[2] R. Hunicke, "The case for dynamic difficulty adjustment in games," in *Advances in Computer Entertainment Technology*, 2005, pp. 429–433.

[3] M.-V. Aponte, G. Levieux, and S. Natkin, "Scaling the level of difficulty in single player video games," in *Proceedings of the 8th International Conference on Entertainment Computing*, 2009, pp. 24–35.

[4] S. W. Um, T. Y. Kim, and J. S. Choi, "Dynamic difficulty controlling game system," in *IEEE Transactions on Consumer Electronics*, vol. 53, 2007, pp. 812–818.

[5] T. Wellmann, "Building a Sports AI Architecture," in *AI Game Programming Wisdom 2*. Charles River Media, 2004.

[6] J. A. Glasser and L.-K. Soh, "AI in computer games: From the player's goal to AI's role," University of Nebraska, Nebraska Lincoln, USA, Tech. Rep., 2004.

[7] I. Millington, *Artificial intelligence for games*. Morgan Kaufmann, 2006.

[8] S. Coradeschi and L. Karlsson, "A role-based decision-mechanism for teams of reactive and coordinating agents," in *RoboCup-97: Robot Soccer World Cup I*, 1998, pp. 112–122.

[9] J. Reynolds, "Team member AI in an FPS," in *AI Game Programing Wisdom 2*. Charles River Media, 2004.

[10] J. Orkin, "Simple techniques for coordinated behavior," in *AI Game Programing Wisdom 2*. Charles River Media, 2004.

[11] D. Garces, "Achieving coordination with autonomous NPCs," in *Game Programming Gems 6*, M. Dickheiser, Ed. Charles River Media, 2006.

[12] D. Silver, "Cooperative pathfinding," in *AI Game Programing Wisdom 3*. Charles River Media, 2006.

[13] C. T. Tan and H.-L. Cheng, "Personality-based adaptation for teamwork in game agents," in *Proceedings of Artificial Intelligence and Interactive Digital Entertainment*, 2007, pp. 37–42.

[14] F. D. Laramee, "Dead reckoning in sports and strategy games," in *AI Game Programming Wisdom 2*. Rockland, MA, USA: Charles River Media, 2004.

[15] R. Straatman and A. Beij, "Killzone's AI: dynamic procedural combat tactics," in *Game Developers Conference*, 2005.

[16] M. Mateas and A. Stern, "Facade: An experiment in building a fully-realized interactive drama," in *Game Developers Conference*, 2003.

[17] R. Houlette, "Player modeling for adaptive games," in *AI Game Programing Wisdom 2*. Charles River Media, 2003.

[18] T. J. A. Jansen, "Player adaptive cooperative artificial intelligence for RTS games," BSc. Thesis, Universiteit Maastricht, 2007.

[19] C. Tan, , and H. L. Cheng, "IMPLANT: an integrated MDP and POMDP learning AgeNT for adaptive games," in *Artificial Intelligence and Interactive Digital Entertainment Conference*, 1996, pp. 80–87.

[20] D. Doherty and C. O'Riordan, "Evolving team behaviours in environments of varying difficulty," *Artif. Intell. Rev.*, vol. 27, no. 4, pp. 223–244, 2007.

[21] B. J. Grosz, "AAAI-94 presidential address: Collaborative systems," *AI Magazine*, pp. 67–85, 1996.

[22] P. Scerri, D. Pynadath, and M. Tambe, "Adjustable autonomy in real-world multi-agent environments," in *Proceedings of the fifth international conference on Autonomous agents*, 2001, pp. 300–307.

[23] J. Reynolds, "Tactical Team Al Using a Command Hierarchy," in *AI Game Programing Wisdom*. Charles River Media, 2002.