Learning from Demonstration to be a Good Team Member in a Role Playing Game

Michael Silva, Silas McCroskey, Jonathan Rubin, G. Michael Youngblood, Ashwin Ram
Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304 USA
{Michael.Silva, Silas.McCroskey, Jonathan.Rubin, Michael.Youngblood, Ashwin.Ram}@parc.com

Abstract

We present an approach that uses learning from demonstration in a computer role playing game to create a controller for a companion team member. We describe a behavior engine that uses case-based reasoning. The behavior engine accepts observation traces of human playing decisions and produces a sequence of actions which can then be carried out by an artificial agent within the gaming environment. Our work focuses on team-based role playing games, where the agents produced by the behavior engine act as team members within a mixed human-agent team. We present the results of a study we conducted, where we assess both the quantitative and qualitative performance difference between human-only teams compared with hybrid human-agent teams. The results of our study show that human-agent teams were more successful at task completion and, for some qualitative dimensions, hybrid teams were perceived more favorably than human-only teams.

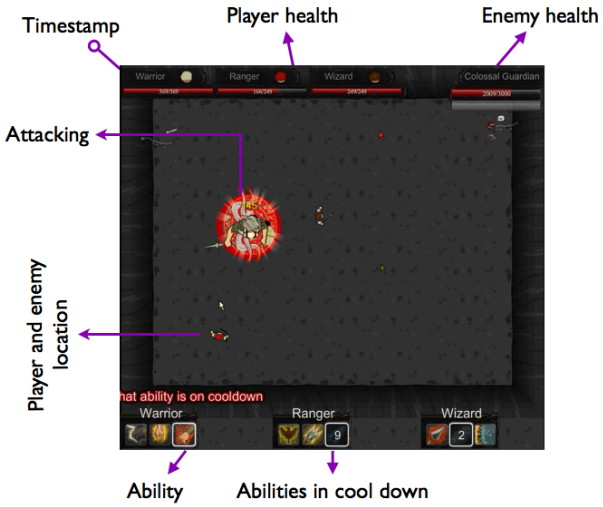

Figure 1: Hands of War 2 gaming environment

Introduction

In this paper, we present an approach to automatically learn behaviors for characters in computer role playing games. Our approach employs learning from demonstration. Observation traces are collected from human players as they interact within a gaming environment. These traces are processed by a behavior engine that employs case-based reasoning (Riesbeck and Schank 1989; Aamodt and Plaza 1994). The end result is fully autonomous behavior control for characters within a computer role-playing game.

The major advantage of learning from demonstration is the ability to derive autonomous behavior by simply observing human play. This reduces the need for programmers to construct large behavior sets by manually encoding individual rules. Instead, a behavior engine that employs case-based reasoning dictates the type of behavior that should be performed based on similarity between the present and previous gaming environments. The resulting recommendations are sent to non-player characters within a virtual environment, which then carry out the required behavior. Overall, the authorial burden of character control is eased. This allows a range of character styles and behaviors to be automatically captured and re-used in gaming and virtual environments.

In this work, we utilize a case-based behavior engine that employs learning from demonstration for team-based play, where an autonomous agent is required to interact and cooperate with a team of human players in order to complete a task within a gaming environment. We present the results of a study that determines the effect of replacing a human team member with an autonomous agent. In particular, our study attempts to capture differences observed between human-only teams and hybrid human-agent teams, in terms of both task performance and teamwork perception (i.e, how well teams were perceived to work together as judged by team members).

Our research takes place in the domain of role playing games. In particular, our study was conducted in a commercially available RPG game called Hands of War 2 (Axis-Games 2012). For this research, we modified the game Hands of War 2 to accommodate team based play within collaborative battles. Teams were made up of three members, each with separate roles to play. The team roles consisted of a warrior, a ranger and a wizard. The warrior could perform

close-ranged attacks, whereas the ranger would attack from a distance. The wizard could attack at a distance, as well as cast healing and shielding spells for the group. In order to win the game, team members were required to work together to defeat an enemy character within a constrained time period of five minutes. Figure 1 depicts a snapshot of the game play. First, teams composed entirely of human players were used to gather game traces. A mixture of human subjects were used to gather data, with varying levels of experience when it came to gaming. The second group of teams involved a mix of human subjects and one automated agent. To reduce the effect of bias in the study, human team members were not made aware that one of their team mates was an AI agent. This was made possible by having a member of the research team sit in on the experiment and pretend to control one of the characters in the game. In reality, this character was actually entirely controlled by the case-based behavior engine, which was trained with the data traces recorded from the first group of participants. Both a quantitative and qualitative analysis was conducted. We present a comparative evaluation between all-human vs. hybrid human-agent teams. We found that hybrid teams were more successful at task completion compared to all-human teams and hybrid teams were perceived more favorably, for some qualitative dimensions.

First, we present related work in both learning from demonstration and team-based AI. We then provide further details regarding our case-based behavior engine and describe the methodology used within our study. Finally, we present results of the study, followed by discussion.

Related Work

Our work is related to other research that employs case-based reasoning and learning from demonstration. Floyd, Davoust, and Esfandiari (2008) detail an approach for constructing an autonomous, spatially aware agent that uses case-based reasoning within the domain of RoboCup soccer. They employ learning from demonstration by observing the playing decisions of other RoboCup soccer players. Their work is similar to our own, as a team of agents are required to coordinate and reason within a spatial dimension. However, our own work requires coordination among a team of players, where team members are a *mix* of both artificial agents and human players.

The work we describe in this paper focuses on role playing game environments (RPG). The related environment of real time strategy games (RTS) has often been employed as a test-bed for learning from demonstration research (Ontañón et al. 2007; Palma et al. 2011; Weber, Mateas, and Jhala 2012). In particular, (Ontañón et al. 2007) employed learning from human demonstration, together with real-time case-based planning to play the RTS game of Wargus and (Weber, Mateas, and Jhala 2012) who investigated learning from demonstration in the domain of StarCraft. While the strategies these works produced may be used to challenge human opposition, the goal was not to augment human teams with artificial agent members. This has been the goal of research presented in (Abraham and McGee 2010) and (Merritt and McGee 2012). Abraham and McGee (2010) describe considerations involved in developing an AI "buddy" agent that can dynamically adapt to the needs of a human player while cooperating against an adapting AI opponent. The authors describe a simple game called *Capture the Gunner* where cooperation is required to succeed. The basic game structure involves a team composed of a human-player and an AI agent, both of whom must attempt to reach an opponent, the *gunner*, before they are shot. Success only occurs when both teammates have reached the gunner. The game is designed such that the difficulty of the opponent increases with each level and the AI teammate's skill adapts to the needs of the human-player and the opponent's skill. Our work differs from that of (Abraham and McGee 2010; Merritt and McGee 2012) as we focus on the more complex domain of computer role-playing games. Moreover, the agents we produce are required to interact with multiple human team members, not just a single teammate.

Case-Based Behavior Engine

We use a case-based approach in order to generate behavior for artificial agents within the gaming environment. There are four main components involved with generating behaviors. They are:

1. Trace generation and capture
2. Preprocessing of traces
3. Similarity assessment
4. Solution generation

We provide further details for each of these components.

Trace Generation and Capture

Traces are generated by having a team of human players play the game Hands of War 2. Figure 1 depicts a snapshot of the game play. Figure 1 also lists the features captured from the environment and stored within a game trace. As we are dealing with a real-time environment, in which the order of actions matter, all game traces record time-stamp information. For every 100 milliseconds of game play, a trace is captured and stored. Each trace records information regarding all players' current locations, as well as the enemy's location. Location information is encoded as (x,y) coordinates in screen space within the trace. Traces also capture players' current health values, as well as the enemy's current health. The ability being performed by each player is recorded, as are the abilities that are currently ready and able to be performed by the player. Finally, we also record whether a player is currently attacking, as well as the player currently being attacked by the enemy.

Preprocessing of Traces

Before generating agent behavior by utilizing the captured traces within our behavior engine, a preprocessing step occurs. The only preprocessing performed involves mapping (x,y) location values into a grid system. A grid based encoding was chosen for similarity assessment and case retrieval, rather than relying on raw pixel information captured in the

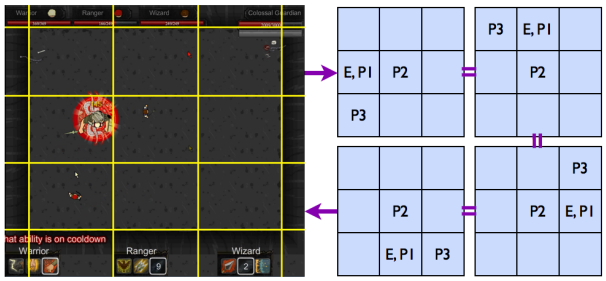

Figure 2: Location information is encoded using a rotation/reflection agnostic grid based system

initial traces. This was done in order to generalize entity position information. Figure 2 shows how an entity's (x,y) position is mapped into a 3x3 grid. In Figure 2, E stands for the enemy and P1, P2 and P3 are the players' positions. We use an encoding that ignores reflections and rotations of the grid. Figure 2 shows an example of our rotation/reflection agnostic grid based encoding. The right hand side depicts a series of grid rotations that are all treated as similar during case retrieval. This allows a larger number of similar cases to be retrieved by our behavior engine at runtime.

Similarity Assessment

Case retrieval involves determining similarity between the current environment and traces captured from previous human demonstrations. The result of each human demonstration is a separate case-base composed of time-stamped actions and behavior. As each demonstration results in a separate case-base, a prioritized list of case-bases is kept, which dictates the order in which distinct case-bases are searched. Case-bases were manually prioritized by the authors based on the perceived quality of demonstration. As such, case-bases where team members were perceived to have worked well together to complete the task were queried first by the behavior engine.

We use a two stage process for case retrieval. In stage one, we encode the current spatial environment using our grid-based encoding, which ignores differences between rotations and reflections. A prioritized search of case-bases takes place, where each case-base is searched for matching grid encodings. Higher priority case-bases are searched before lower priority ones. If no spatial matches are found within the current case-base, the search continues on to the next case-base. If the current case-base does contain spatial matches, a smaller subset of matching cases is extracted from this case-base.

The second stage of the retrieval process involves a deeper similarity assessment, performed on the subset of cases retrieved from stage one. Here, similarity assessment between weighted feature vectors takes place, where feature vectors capture the game state information from Figure 1. Larger weights were assigned to features that captured player and enemy heath information, with lesser importance placed on a player's ready ability and attack status. The case which is most similar to the current environment is selected and passed to the solution generator.

Solution Generation

Once we have determined the most similar case, the final requirement of the behavior engine is to generate a solution. A solution determines the destination a character should proceed to, as well as a sequence of actions it should perform within the environment. In order to determine a series of actions, a look ahead is performed through the time stamped cases that directly follow the most similar case in the case-base. Beginning with the most similar case, the actions that follow are extracted. The amount of look ahead to perform is specified by look-ahead parameter, Δt. In our behavior engine, we used $\Delta t = 25$, which corresponds to roughly two and a half seconds of look ahead. The final case after Δt dictates the cell location a player should proceed to. A final reverse spatial mapping is performed to determine the actual location the agent will move to, in the current environment, given the rotation/reflection agnostic grid encoding.

Methodology

One of the motivations of this work was to evaluate any observed differences between all-human groups compared with hybrid human-agent groups. We sought to evaluate differences in how well teams performed their task, as well as how team members judged their team's performance. We conducted a study that comparatively evaluated the performance of two groups. The first group was made up of three human players only. We refer to this team as the *all-human* team. The second group was made up of two human players and one agent. The agent was controlled by our case-based behavior engine. We refer to this team as the *hybrid* team or the *human-agent* team.

First, we collected game traces from the all-human teams. Teams were asked to play together to defeat an enemy in the game. Teams were first given instructions on the abilities of each character type, as well as the game controls. Each team was allowed one practice round, followed by one or two actual rounds where game traces and statistical results were recorded. Three rounds were played at most to limit the amount of time required by subjects to participate in the study. Players had at most five minutes per round to work together to defeat the enemy in the game. Teams were free to communicate with each other regarding tactics and strategies to use. Following game play, subjects were asked to complete a short questionnaire about how they felt

Table 1: Quantitative results recorded between groups

	Human-Only	Human-Agent	Control
Average Successes (%)	66.67	**100.00**	75.00
Average Time	221.89	205.34	**156.36**
Average Time (Success only)	168.51	197.78	108.48
Average Deaths per round: Warrior	**0.17**	**0.17**	0.25
Average Deaths per round: Ranger	**0.00**	0.50	0.00
Average Deaths per round: Wizard	0.67	**0.50**	1.00
Average Deaths per round: Total	**0.83**	1.17	1.25

about their team's performance. The questionnaire was constructed by selecting an appropriate set of questions from the Team Diagnostic Survey (Wageman, Hackman, and Lehman 2005), an instrument intended to assess team efficacy.

The traces collected from the first group of players were then used within our case-based behavior engine to automate the actions of an artificial agent within the gaming environment. The second set of human-agent teams consisted of two human players and one artificial agent. The agents could play any of the available character types by simply adjusting the set of traces it used to generate behavior. As we did not want to introduce any bias into the experimental setup, human subjects were not told that one of the players was an artificial agent. Instead, a member of the research team pretended to control the character in the game, when in fact the behavior witnessed was entirely produced by the behavior engine. All other details of the study were held constant between the two groups.

A third and final control group was also included in the study setup. All details for the control group were the same as for the human-agent group, except for the fact that the agent's behavior was no longer generated by the case-based behavior engine, but instead generated randomly.

In total, there were four *human-only* teams (12 human subjects), six *human-agent* teams (12 human subjects) and four *control* teams (8 human subjects). For each group we conducted a quantitative and a qualitative analysis. For the quantitative analysis we measured 1: task outcome (success or failure), 2: task completion time, 3: relative damage inflicted upon enemy per player, and 4: number of player deaths.

The qualitative analysis consisted of a set of 13 team related questions, listed in Table 2. The questions from Table 2 were selected from the Team Diagnostic Survey (Wageman, Hackman, and Lehman 2005), based on their appropriateness for our experimental setup. A 5 point Likert scale was used to record responses. Values ranged from 1: very inaccurate, 2: somewhat accurate, 3: neither, 4: very accurate to 5: very accurate. Subjects were also given the option of declining to respond.

Experimental Results

Quantitative Results

Table 1 summarizes the quantitative results data obtained from the study. Within each row, the bold values highlight the group that performed the best. First, notice that hybrid

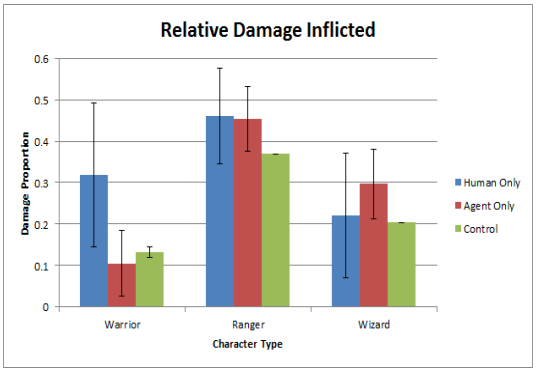

Figure 3: Proportion of damage inflicted by player types between groups

human-agent teams outperform the human-only and control group at task completion. In fact the human-agent groups never fail at completing the task, whereas the human-only group only manages to successfully complete the task two thirds of the time. Next, we observe that the human-only group takes the longest time to complete the task. This makes sense, as more task failures results in a larger average completion time. When we consider only instances where the task was completed successfully, the human-agent group takes the longest time on average. Interestingly, the control group takes the least amount of time to defeat the enemy. Table 1 also captures information about how many times each player type died while performing the task. Overall, members of human-only groups died the least, followed by human-agent team members. The most number of deaths occurred in the control team, where the agent took random actions.

We also measured the amount of damage each player type inflicted upon the enemy during task completion. Figure 3 shows the relative damage inflicted for characters within each group type. The leftmost (blue) bars record the damage inflicted when a human player controlled the particular character type (warrior, ranger, wizard respectively). The middle (red) bars, record the relative damage inflicted when the character was controlled by our case-based behavior engine. The rightmost (green) bars record the damage inflicted by each character type, when they were controlled by the system and given random actions to perform. We can see that both the case-based behavior engine and random play, do

Table 2: Questions selected from the team diagnostic survey

1	Members of this team have their own individual jobs to do, with little need for them to work together.
2	This team's purposes are not especially challenging - achieving them is well within reach.
3	Members of this team are too dissimilar to work well together.
4	This team has too few members for what it has to accomplish.
5	This team is larger than it needs to be.
6	Some members of this team lack the knowledge and skills that they need to do their parts of the team's work.
7	Generating the outcome or product of this team requires a great deal of communication and coordination among members.
8	Members of this team have to depend heavily on one another to get the team's work done.
9	This team is just the right size to accomplish its purpose.
10	Everyone in this team has the special skills that are needed for team work.
11	Members of this team agree about how members are expected to behave.
12	This team has a nearly ideal "mix" of members.
13	Generally speaking, I am very satisfied with this team.

poorly (compared to human players) when they control the warrior. If we consider damage inflicted by the ranger, both humans and agents perform similarly, with random play performing slightly worse. Finally we observe that, when controlled by the behavior engine, wizards inflict more damage on average than when controlled by their human counterparts.

Qualitative Results

A further objective of the study performed was to gauge team member perceptions about their team's dynamics and to determine if any differences exist between human-only teams compared with hybrid human-agent teams. Table 2 lists the questions that were posed to human team members after completion of the task. The question numbers from Table 3 correspond to the questions listed in Table 2. In Table 3, we have grouped questions into a *mostly critical* set (light red background) and a *generally positive* set (light green background). Values closer to one are preferred for *mostly critical* questions, whereas values closer to five are preferred for *generally positive* questions. Also depicted are p values calculated by conducting an unpaired, two-tailed t-test between the human-only and human-agent groups. Once again values in bold highlight the group with the most favorable response and bold-italicized p values indicate statistical significance with 95% confidence.

The results from Table 3 show that for three out of the six *mostly critical* questions, human subjects disagreed the most when one of the team members was an AI agent (i.e. questions two, five and six). For one of these questions (question six), the result was statistically significant, with 95% confidence, when compared with the human-only team.

For the *generally positive* set of questions, subjects from the human-agent team agreed more strongly on average with four out of the seven questions, compared with responses from members of the human-only and control teams. More favorable responses were recorded from the human-agent team for questions eight, nine, eleven and thirteen. For two of these questions (eight and eleven) the difference observed was significant, when compared with the human-only team.

Table 3: Qualitative results recorded between groups

Q.	Human-Only	Human-Agent	p =	Control
1	2.83	2.83	1.00	**2.38**
2	4.00	**3.75**	0.594	3.88
3	1.75	2.00	0.594	**1.63**
4	**1.42**	1.50	0.784	1.88
5	1.92	**1.83**	0.790	2.25
6	3.33	**1.91**	***0.006***	2.25
7	3.17	3.75	0.218	**4.00**
8	3.67	**4.58**	***0.041***	3.88
9	4.50	**4.67**	0.581	4.50
10	3.91	4.50	0.199	**4.88**
11	3.33	**4.36**	***0.029***	3.88
12	4.30	4.09	0.496	**4.38**
13	4.00	**4.50**	0.097	4.00

Responses to the final question from Table 3 (question thirteen) indicate that, overall, team members from the human-agent group appeared slightly more satisfied with their team (4.5 on the Likert scale), compared with the human-only and control teams (4.0 on the Likert scale, for both).

Discussion

We noticed some interesting differences in observed behavior between character types when they were controlled by our behavior engine. The behavior patterns of both the ranger agent and the wizard agent were reasonable and appeared consistent to human play. However, the behavior of the warrior agent was less ideal. In general, the warrior spent considerably more time transitioning between locations than it did actually attacking the enemy. The explanation for this difference between characters has to do with the type of attack characters perform. The warrior used a close ranged attack, whereas the ranger and wizard both use long ranged attacks. In general, the system appeared to struggle with close ranged attacks due to the added requirement that the agent

be within close proximity to the enemy. For ranged attacks this requirement did not exist, making attacks easier to complete. This explains the reduced damage values in Figure 3 for the warrior agent-only and control entries. For long ranged attacks, the ranger agent competed at a level comparable with human play, although it incurred more deaths on average. Finally, the wizard agent inflicted more damage than its human counterparts and also incurred fewer deaths. This is likely due to the behavior initially witnessed during trace capture, where human wizard players typically adopted a strategy which used the wizard's abilities to protect the group, while consistently avoiding the enemy. This behavior was successfully replicated by the behavior engine. The fact that hybrid human-agent teams outperform human-only teams at task completion is likely due to the prioritized case-base search, which ensured higher quality case-bases were queried first.

The authors were not expecting the hybrid human-agent groups to perform any better than the human-only groups in the qualitative evaluation. However, the results from Table 3 showed that for a number of questions the human-agent group members were less critical and more positive of their team compared to the human-only and control teams. In particular, hybrid human-agent groups performed the best on the final question: "*Generally speaking, I am very satisfied with this team*".

All of the subjects involved in the study thought that the member of the research team was making and executing playing decisions for their character, when in fact this character was controlled by the case-based behavior engine. After revealing to participants that they had been playing with an agent rather than a human team mate, a few subjects stated that they were so focused on their own actions they did not have the opportunity to fully observe the actions of the other players. One possible explanation for why human-agent groups were more satisfied with their team is that, given this high *cognitive load* subjects had to deal with, they may have relied more on the eventual outcome of the task to influence their responses to the questionnaire. The human-agent group had a 100% success rate at task completion, compared with lesser success rates for the human-only and control teams. This may have led to an improved qualitative judgment by team members.

Conclusion

This work involved the design and construction of a case-based behavior engine, which learns from demonstration. Traces collected from multiple human demonstrations are used as input into the behavior engine, which then suggests a sequence of actions to perform based on those traces. By applying learning from demonstration, new behaviors can very quickly and easily be incorporated into the system without requiring additional programming effort. We utilized our behavior engine within a study to determine differences in performance between teams composed entirely of human players and teams augmented with artificial agents. We designed our study, such that human team members who played on the hybrid human-agent team, were not aware of this fact. This was done in order to not introduce any bias into the study. We found that hybrid, human-agent teams, were more successful than all-human teams at task completion. We also found that on some qualitative dimensions, hybrid teams were perceived and evaluated better by their human team members, compared to human-only teams.

Acknowledgments

The authors wish to thank Danny Jugan and Axis Games for kindly providing the source code for Hands of War 2.

This material is based upon work supported by the National Science Foundation under Grant No. (IIS-1216253). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Aamodt, A., and Plaza, E. 1994. Case-Based Reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications* 7(1):39–59.

Abraham, A. T., and McGee, K. 2010. AI for dynamic teammate adaptation in games. In *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games, CIG 2010*, 419–426.

Axis-Games. 2012. Hands of war 2. http://armorgames.com/play/10987/hands-of-war-2.

Floyd, M. W.; Davoust, A.; and Esfandiari, B. 2008. Considerations for real-time spatially-aware case-based reasoning: A case study in robotic soccer imitation. In *Advances in Case-Based Reasoning, 9th European Conference, ECCBR 2008*, 195–209.

Merritt, T. R., and McGee, K. 2012. Protecting artificial team-mates: more seems like less. In *CHI Conference on Human Factors in Computing Systems, CHI '12*, 2793–2802.

Ontañón, S.; Mishra, K.; Sugandh, N.; and Ram, A. 2007. Case-based planning and execution for real-time strategy games. In *Case-Based Reasoning Research and Development, 7th International Conference on Case-Based Reasoning, ICCBR 2007*, 164–178.

Palma, R.; Sánchez-Ruiz-Granados, A. A.; Gómez-Martín, M. A.; Gómez-Martín, P. P.; and González-Calero, P. A. 2011. Combining expert knowledge and learning from demonstration in real-time strategy games. In *Case-Based Reasoning Research and Development - 19th International Conference on Case-Based Reasoning, ICCBR 2011*, 181–195.

Riesbeck, C. K., and Schank, R. C. 1989. *Inside Case-Based Reasoning*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc.

Wageman, R.; Hackman, J. R.; and Lehman, E. V. 2005. Team Diagnostic Survey: Development of an instrument. *Journal of Applied Behavioral Science* 41(4):373–398.

Weber, B. G.; Mateas, M.; and Jhala, A. 2012. Learning from demonstration for goal-driven autonomy. In *Proceedings of the Twenty-Sixth Conference on Artificial Intelligence (AAAI-12)*.