Experimental AI in Games: Papers from the AIIDE 2015 Workshop

Sarah and Sally: Creating a Likeable and Competent AI Sidekick for a Videogame

Martin Černý
Charles University in Prague
Czech Republic
cerny.m@gmail.com

Abstract

Creating reasonable AI for sidekicks in games has proven to be a difficult challenge synthetizing player modelling and cooperative planning, both being problems hard by themselves. In this paper, we experiment with designing around these problems: we propose a cooperative puzzle-platformer game that was designed to look similarly to the mainstream of the genre, but to allow for an easy implementation of a quality sidekick AI, letting us test player reactions to the AI. The game was designed so that it is easy for the AI to find optimal solutions while the problem is relatively hard for a human player. We gathered survey responses from players who played the game online (N=28). While the AI sidekick was reported as likeable and helpful, players still reported greater enjoyment of the game when they were allowed to control the sidekick themselves. These findings indicate that the AI itself is not the only obstacle to truly enjoyable gameplay with an AI sidekick.

1 Introduction

Despite huge advances in both computing power and algorithms, contemporary computer games still struggle to convey interesting and effective AI-controlled sidekicks for human player. In almost all cases, players would strongly prefer even an unskilled human to AI as a sidekick or a teammate, or — if technically possible — to fully control the sidekick themselves. A naturally arising question is to what extent we can attribute this to poor quality of the AI and to what extent it is a design problem — maybe humans simply do not like playing along with computers.

To examine this question, we will work with two definitions of collaborative gameplay:

> 1) In a collaborative game, all the participants work together as a team, sharing the payoffs and outcomes; if the team wins or loses, everyone wins or loses. (Zagal, Rick, and Hsi 2006)
>
> 2) Collaboration is a coordinated, synchronous activity that is the result of a continued attempt to construct and maintain a shared conception of a problem. (Roschelle and Teasley 1995)

The former definition comes from game research and stresses the importance of competence — if the sidekick AI is weak, the player suffers as well. This view accounts for one problem with AI sidekicks: in most dynamic game environments humans are often able to find better solutions than AI and naturally maintain a good model of the other player increasing the effectivity of their decisions.

The latter definition originates in research on cooperative learning and puts more importance to communication. Once again this is an important area where humans are much better than AI: humans can quickly communicate their beliefs and intentions to the other player and easily share the "conception of a problem".

A further AI disadvantage is friendship — humans playing together are able to create and maintain positive relationships involving empathy and trust. We consider competence to be the easiest to match as a huge body of research is focused on improving AI competence in games. The latter two are however more demanding.

The goal of this work is to examine the problems for sidekick AI that will remain even if the AI competence can be made satisfactory. To let us focus on communication, friendship and the way players perceive sidekick AI, we decided to design around the problem of AI competence. We have built "Sarah and Sally", a game which is AI-friendly by design, i.e. it allows for near optimal behavior with well-known AI methods while still being similar to a large group of dynamic games. The game also features only simple communication and portrays the AI-controlled character in a friendly way.

We share the experience we gathered during the implementation of the game, both from game design and AI perspective and report on evaluation of player responses to the game, with focus on player's enjoyment of the game and the sidekick's perceived friendliness.

2 Related Work

Achieving reasonable competence of sidekick AI has been the focus of multiple academic studies. In (Nguyen et al. 2011) and (Macindoe, Kaelbling, and Lozano-Pérez 2012), the problem of finding on optimal action on the side of an AI sidekick in cooperative chasing scenario is represented as a fully or partially observable Markov decision process (MDP). Although their results are promising, the approach is extremely computationally expensive even for very small domains (the domains in the papers are 10×10 grids) and also needs a reasonable model of human behavior. A similar approach (Fern et al. 2007) has been used to guide a sidekick AI in a simple real-time game[1]. MDP-based approaches have also been used to coordinate human actions with robots in simple scenarios (Nikolaidis et al. 2014).

A sidekick may also be used as a measure to shape gameplay experience: In (Tremblay and Verbrugge 2013) a scripted adaptive sidekick for first-person shooter games changes his behavior to maintain the flow of the game.

One of the most interesting sidekicks in AAA games is Elizabeth in BioShock: Infinite[2]. The creators designed Elizabeth to not be a fighter, but rather a support in combat, throwing ammo and medikits at the player. They also equipped Elizabeth with rich dialogue fostering player's friendly emotions towards her. To make Elizabeth effective, they put an extensive amount of annotations into the environment and heavily scripted the character to cover a wide range of situations[3]. Elizabeth was very well received. Nevertheless, Elizabeth's competence as a sidekick is very limited as her actions have only very small impact on the actual gameplay. While limits on Elizabeth's impact make a lot of sense from the design perspective, this approach can be applied only to a small subset of sidekicks in games. In particular, it cannot be used as a replacement for a human player in games designed as cooperative multiplayer. In our approach we want the AI sidekick to have the possibility to directly influence the game and to refrain from pre-scripting the behavior. While there are many games with AI sidekicks, it is very difficult to get technical details of their implementation. Nevertheless, from searching informal sources on the Internet and our own gaming experience we expect that the vast majority of contemporary AI sidekicks in games are fully scripted.

As one of the reviewers of this paper pointed out, our experiment design was completely oblivious to research on the perception of computer personalities. Studies have found that very simple hints are sufficient to make the user act as if the computer had personality (Nass et al. 1995) or to treat computer as a teammate (Nass, Fogg, and Moon 1996). While the former study also suggests that users prefer to interact with a similar personality, this has been disputed (Isbister and Nass 2000). Nevertheless, both studies agree that the perceived personality of the computer has a strong impact on the way the human perceives the interaction with the computer.

3 Game Design

Our primary inspiration was a subset of puzzle-platformer Flash games such as One and One Story[4] or Home Sheep Home[5]. In those 2D games, the player switches control between multiple characters and has to complete a task requiring complex collaboration of all the characters — hence the "puzzle" part. The "platformer" ingredient is the fact that the characters are generally affected by gravity and an important part of solving the puzzles is figuring out how to reach certain platforms in the level.

The most desirable property of those games is that regardless of player skill, the game simply cannot be completed without the characters cooperating. We thus designed a similar game with two characters where one is controlled by the player, and the other one is an AI-controlled sidekick. We chose the protagonists to be two girls with distinct appearance: Sarah (player character) is small while Sally (AI sidekick) is tall. The cooperation of the characters is enforced by complementary abilities. Further, we wanted the sidekick's abilities to be very powerful, to increase the feeling that the sidekick is helping.

To make the AI simple to implement, the game logic operates on a grid. At the same time we try to hide this fact from the player by making the movement of the characters smooth. Moreover, only one character is active at a time and the other character cannot perform any action until the active character ends her turn.

To experimentally evaluate the technique, it has to be possible for the player to control both characters. This will let us compare player experience in both cases, but it enforces further limitations on game mechanics.

Game Mechanics

To make the game accessible and let the players learn the game mechanics quickly we kept the amount of mechanics minimal. Beyond simple "physics" there are only 4 mechanics: target location, keys and barriers, levitation and level rotation[6]. See Fig. 1 for a screenshot of the game.

Characters can move only sideways, there is no jumping. The gravity pulls the characters down, but unlike many similar games, the characters cannot move sideways while falling. The player character is only 1 tile tall, while the AI sidekick is 2 tiles tall, letting the player reach some places the sidekick cannot. In each level, the goal is that both characters reach their target locations portrayed as a small and a large door. Keys and barriers follow a common game trope: when a character enters a tile occupied by a key, the key is automatically picked and some previously non-traversable tiles become clear. A key always has the same color as the barrier it removes.

The levitation mechanic is available only for the sidekick and lets her lift the player up. To make the AI implementation easier, levitation is available regardless of position of either character. To simplify user interface when the player is controlling the sidekick and to make levitation less powerful, player character is always lifted in straight line up, until she hits a ceiling. This way, certain areas in the level remain unreachable for the player character even when levitation and

[1] http://gambit.mit.edu/loadgame/dearth.php

[2] http://www.bioshockinfinite.com/

[3] Interested reader is referred to a video where developers comment a playthrough of the first encounter between the player and Elizabeth: http://www.youtube.com/watch?v=2viudg2jsE8

[4] http://armorgames.com/play/12409/one-and-one-story

[5] http://armorgames.com/play/12700/home-sheep-home-2

[6] The best way to understand game mechanics is through play. We thus encourage the reader to play the game before proceeding. The game can be played/downloaded at http://martin-cerny-ai.itch.io/sarah-sally

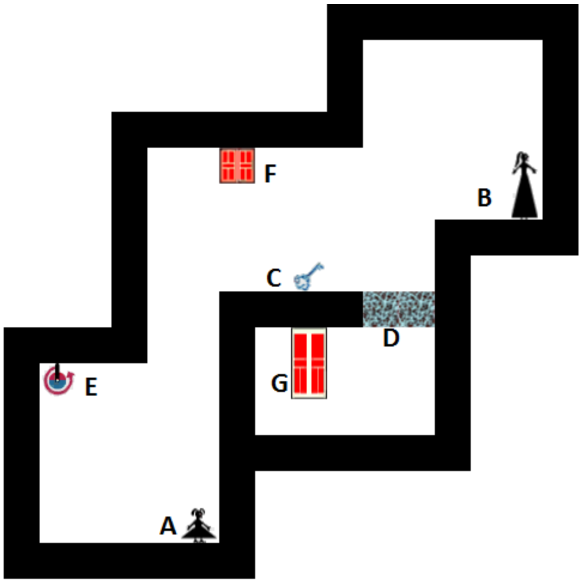

Figure 1: Screenshot of the first non-tutorial level of the game. One of the solutions is that the sidekick (B) moves left and picks up a key (C) which removes the barrier (D). The sidekick then moves to a spot just below its target location (G) and ends its turn. The player character (A) moves below the lever (E) and ends its turn to let the sidekick levitate her. Once she has been lifted up, she can use the lever to rotate the level counterclockwise two times. Now both the player character and the sidekick can easily reach their respective target locations (F,G).

movement is combined. The tile below player's final destination is made non-traversable until the end of player's turn, further simplifying user interface for the sidekick as there is no need to specify which direction should the player character be lifted: the direction is decided by player's movement next turn.

The last mechanic is level rotation. If any of the characters reaches a special lever object, she can use the lever to rotate the level by 90 degrees, which is equivalent to changing the direction of the gravity. Level rotation is the only way the sidekick character can reach places that were above her prior to rotation, requiring the player to help the sidekick in reaching her target location.

4 AI Implementation

As the game logic operates on a grid and the characters do not act simultaneously, the AI can be implemented as a simple heuristic forward search. The game state is fully described by the (immutable) level geometry, positions of both characters, the character that is active, the set of keys that were picked up, the direction of gravity, and the location of the tile that is not traversable due to levitation (if any). There are at most 5 actions available to the active character: move left, move right, change active character, levitate (Sally only), rotate level (if at the same location as a lever). After each action, the effects of gravity are applied.

The AI expects the player to perform optimal actions and thus nodes where the player chooses an action are treated the same as those where the AI decides. We used a simple heuristic that takes into account the number of keys that were not picked-up and the distance from both characters to their target locations. Our level design ensures that the levels are not completable without picking all keys so the heuristic is admissible. To reduce the branching factor, we used a simple pruning that considers the "end turn" action only when ending the turn makes sense, i.e. after rotating the level or using levitation, after the character's vertical position has changed (so the result of rotating the level changes) and after the player character moves below a possible levitation target. This pruning cannot remove any solutions and also cannot increase the minimal number of turns necessary for completing the level.

The search space is relatively modest — the larger levels allow less than 60 valid locations per character, there are 4 possible directions for gravity, up to 3 keys that may and may not be picked up, up to 50 valid locations for the tile not traversable due to levitation, and two options for active character. In total, the search space is definitely smaller than 1.2×10^7. For the levels we use the search space is probably much more constrained — the search algorithm never evaluated more than 13 154 nodes. The solutions to levels in the game vary in length from $\tilde{1}5$ to $\tilde{4}0$ actions.

Assigning costs to actions in the search algorithm turned out to be an interesting problem, as the solution needs to appear reasonable to a human player and the game needs to be fun. The cost of moving a character is simply the number of tiles traversed. Initially, we assigned large cost to ending the turn, so that the AI will optimize the number of switches between characters as we expected them to be the most annoying to the player. This however resulted in the AI performing long and/or unintuitive chains of actions, even though some of the actions could be accomplished by the player as well (e.g., the AI would move a long distance to pick a key close to the player). After experimenting with the turn cost, we achieved best results by removing the turn cost altogether and only incurring cost after a turn where the character performs no action. This no-op cost was useful to minimize the amount of situations where the AI is unhelpful (doesn't do anything although it can, because it found a shorter solution starting with no-op). However, increasing no-op cost too much led the AI to perform redundant actions instead of a no-op, which made the AI look unreasonable. This redundant actions were almost always levitation actions as levitation can almost never reduce the possibilities for player character, and movement without side effects is removed by our pruning procedure. Therefore levitation had to cost more than a no-op.

Further issue with the no-op cost is that there is a balance to be hit from the design point of view: while high cost on inactivity results in redundant actions, AI that puts low cost on player inactivity may be too active and the player may feel redundant and AI that does not incur penalty on its own inactivity may be perceived as lazy or unhelpful.

To let the player better cooperate with the AI, the AI exposes some of its internal state to the player. Note that it is not desirable to expose its complete state, because the AI knows the solution for the level and showing it to the player would render the game uninteresting. The player has no explicit way to communicate with the AI.

The most important moment to communicate is when the AI decides to perform no action at all. The AI can communicate the following reasons for its inactivity: 1) the sidekick is at target location and the player can reach target location without her help, 2) the sidekick can perform no meaningful action without the player doing something first, 3) the optimal solution requires the player to do something first and 4) there is no solution. All those situations can be easily detected by analyzing the optimal action sequence and/or the top levels of the search tree.

In pilot experiments we noticed that players often think that a solution is feasible, when in fact it is not. They would then get frustrated because the AI refuses to help them. This state is very difficult to communicate, as the AI would have to understand what the player is trying to do and explain him why it is not possible.

We settled for a simple approximation of this communication: if no solution is found, the AI tells the bad news to the player but instead of doing nothing, it continues to perform actions that achieve the state with lowest possible heuristic estimate. If no state with lower estimate is reachable, the AI selects actions at random, with a bias towards levitating the player. In our pilot experiments this helped players realize where the mistake was while posing no risk: as the level was already impossible to finish, no action could do any harm.

An even trickier situation arises when the solution is found, but the player arrived at a different (longer or incorrect) solution. Some players than get frustrated as the AI "stays it's ground" and waits for the player to do the optimal action before continuing. We tried to reduce the occurrence of this phenomenon by increasing AI no-op cost, but as already noted, it could not be set too high and those situations still occurred relatively frequently. This situation is both harder to detect and harder to handle: unlike the "no solution" situation we cannot just "do something" as that could harm the team's prospects to finish the level.

When the AI has chosen no-op for two or more consecutive turns and the optimal solution is the same in both turns, the AI concludes that the player has a different solution in mind. In this case, the AI would sometimes (randomly decided) levitate the player, as this is almost always a reversible action and usually it is the action the player expects from the AI.

While the AI "standing it's ground" is possibly frustrating, we felt that it makes the position of the AI and the player more equal and thus decided to keep this behavior in the game. It is also important to note that correctly handling such situations would be very difficult without a two-way communication channel. To further establish that the AI is player's partner, the AI lets the player know, if the player performed a non-optimal action i.e. when the cost of the current solution is notably larger than the cost of the solution found in the previous turn.

One of the most frustrating technical aspects of the implementation was the necessity to implement most mechanics twice — once for continuous game world and once for the discrete simulation used by the AI. The code reuse among the two cases is limited mostly by the fact, that the discrete simulator the AI uses has to be optimized for fast evaluation. This is a problem shared by many applications of symbolic AI in games and we see no way to easily alleviate it.

	After all levels
Q1	I had fun.
Q2	The level was difficult.
	After a baseline level
Q3	I understood Sally's controls and abilities.
Q4	It felt better to control both characters.
	After a coop AI level
Q5	I liked Sally as a character.
Q6	Sally acted intelligently.
Q7	Sally was helpful.
Q8	Sally was lazy.

Table 1: Statements rated in the questionnaire.

5 Experimental Setup

To evaluate the quality of our AI we compare player sentiments when playing the game with the AI-controlled sidekick and when controlling both characters. The players played the game online in an uncontrolled setting. We let all players complete first five levels with the aid of the AI sidekick. The first four form a simple tutorial introducing the game mechanics and the 5th level is the only one that is at least a little challenging (see Figure 1). Once the player completes the 5th level we have a high confidence that he understands the game and is familiar with the way the cooperation with the AI sidekick works. In one of the following three levels, the player is given control of both characters. In further text we refer to this level as the *baseline level* in contrast to coop AI levels where the sidekick is controlled by the AI. The baseline level is chosen randomly for every player session. After each of the three levels, the player is asked to fill in a short questionnaire consisting of several 4-degree Likert statements. The actual questions are given in Table 1. When the user is presented with the first questionnaire we also gather basic demographic data (gender and age). As the game was part of AIJAM online event[7] we expected a large amount of the players to be game developers or AI researchers and added specific questions to check for that. We also measured player retention throughout the game.

6 Results

Initially, there was a bug in recording player retention and only the runs that resulted in filling in a questionnaire were recorded. During this period, 7 people completed level 6 and filled questionnaire before measuring game starts. Out of

[7] http://ai-jam.com/

Level	# Participants	# Baseline
6	28	9
7	17	5
8	17	6
Sum	**62**	**20**

Table 2: Number of study participants — players that completed the given level and filled out the questionnaire. The number of participants that formed the baseline (controlled both characters) is also reported. The "Sum" row reports the total number of questionnaires filled in.

Fun	--	-	+	++
Baseline	0	1 (05%)	14 (70%)	05 (25%)
Coop AI	0	7 (17%)	22 (52%)	13 (31%)
Difficult	--	-	+	++
Baseline	5 (25%)	04 (20%)	10 (50%)	1 (5%)
Coop AI	8 (19%)	21 (50%)	12 (29%)	1 (2%)

Table 3: Summed responses over all levels for the questions on perceived fun (Q1) and difficulty (Q2).

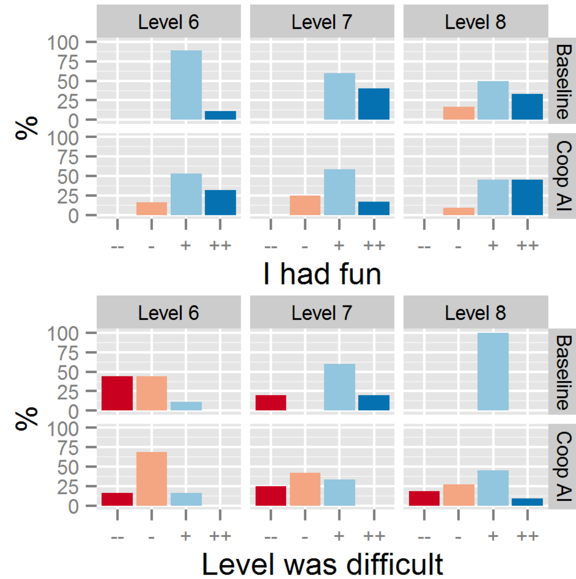

Figure 2: Respones for the questions on perceived fun (Q1) and difficulty (Q2) broken down by game level. Bear in mind that for levels 7 and 8 the results have to be interpreted carefully, as few participants played the baseline variant.

those 4 also completed level 7 and 3 completed level 8. After the bug was fixed, 41 further players started the game. 25 (61%) reached the first measured level (level 6), all of them completed the level Further dropout happened during level 7, only 16 players completed the level, all of which then completed level 8. Not all participants did however complete the questionnaires given. The summed counts of questionnaires we analyze is given in Table 2.

Of the 28 players who filled in at least one questionnaire, 7 were female. Regarding age, majority (18) was between 20-30 years old, one participant reported to be 40-50 years old and the rest (9) was 30-40 years old. Seventeen participants were either game developers or AI researchers.

Quantitative User Feedback

As our game intends to be a casual game, we consider fun to be the most important part of user experience. Participants rated the baseline level as slightly more fun than the coop AI levels. Additionally, the baseline level was rated as more difficult. The differences are however not statistically significant with any relevant test. See Table 3 for the actual numbers and Figure 2 for a breakdown by level[8]. While for the hardest (8th) level the coop AI variant was deemed more fun, there are too few participants that completed the level for the results to be conclusive.

The results for perceived fun are contrasted by the fact that majority of the users reported that they did not prefer controlling both characters (Q4) and that the AI was perceived overwhelmingly as intelligent (Q5), helpful (Q6) and likeable (Q7) and generally also as not lazy (Q8). As almost all players reported that they had no trouble with controlling the sidekick in the baseline level (Q3), we can also rule out that the results were affected by difficulties in controlling the sidekick, see Table 4 for the actual numbers.

Qualitative User Feedback

We also gathered qualitative feedback from users as freeform text. The users mostly reported minor glitches in both design and implementation. Below we present all the qualitative feedback that was not concerned with easily fixable issues:

1. "It is different than other games"
2. "I thought that I do not like to control Sally (the sidekick); but now that I dont it was annoying to wait for what she needs me to do."
3. "She's a bit mean; but I like that."
4. "When Sally told we could not win I thought she was just kidding me and having negative mood in the level; waiting if Ill finish it anyway :D"
5. "I dislike that Sally prioritizes getting to the exit herself over helping first (it's not what I expect of her to do in that situation)"
6. "I would have liked it more if I wasn't told I screwed up immediately after screwing up; just let player figure it out himself (or tell him after a minute or so)"

[8] In all tables and figures "--" and "-" stand for "Strongly disagree" and "Disagree" respectively, "+" and "++" stand for "Agree" and "Strongly Agree" respectively.

Table 4: Responses to questions Q3 — Q8. Questions on sidekick controls (Q3) and preference for the baseline levels (Q4) were asked only after the player completed the baseline level, while questions regarding the perceived properties of the sidekick — intelligence (Q5), helpfulness (Q6), likeability (Q7) and laziness (Q8) were asked only after the player completed a coop AI level.
The total number of responses slightly vary between questions because not all participants answered all questions.

	--	-	+	++
Controls OK	1 (5%)	0	2 (10%)	17 (85%)
Prefer Baseline	3 (15%)	8 (40%)	4 (20%)	5 (25%)
Intelligent	0	5 (12%)	17 (40%)	20 (48%)
Helpful	1 (2%)	2 (5%)	21 (51%)	17 (41%)
Likeable	2 (5%)	5 (13%)	22 (56%)	10 (26%)
Lazy	13 (31%)	16 (38%)	11 (26%)	2 (5%)

The statement 5 is directly related to the problems with tweaking action costs and is probably caused by the high cost of the levitation action. Statement 4 is an interesting illustration of the inherent difficulties in communicating AI state to the player. Finally, statements 2 and 6 hint us at one possible explanation why the players enjoyed the baseline level more: they were in full control and could figure the solution out at their own pace.

Discussion

The most interesting part of the results is that although the players rated the AI very positively in all of the investigated qualities and reported a slight preference for playing with the AI, the players still experienced more fun when controlling both characters themselves. Although the difference is not statistically significant, we clearly failed to show that the gameplay with the AI sidekick is more fun than the traditional approach. Our interpretation is that in creating an AI sidekick, a very delicate balance has to be found: if the sidekick is incompetent, the gameplay is frustrating as the player has to babysit the AI. On the other hand, if the sidekick is too good at helping the player in the game, it diminishes the player's sense of achievement. Our game probably falls in the latter category. This explanation is supported by a few instances of the qualitative feedback and also by the fact that this trend is not visible in the most difficult level, indicating that once the difficulty reaches a certain threshold, the level is challenging even when hints from the AI are taken into account.

There are however other explanations: the higher fun ratings for the baseline level may be attributed to its novelty — the baseline level plays differently than the previous ones which may contribute to increased engagement. A similar argument can however be also used to diminish the fun ratings for the coop AI levels: playing alongside a competent AI is an uncommon experience which may have contributed to the positive ratings. It is not certain whether similar ratings could be achieved if competent sidekick AI was a frequent sight in mainstream games. Further, many of the participants were game developers and/or AI researchers, who are likely to be positively biased to a novel AI-based gameplay. In addition, people who filled in the questionnaires liked the game — otherwise they would not have persisted through the first 5 levels. It is also possible that players are uncomfortable with the fact that the AI itself can be more assertive than the player. A study where the baseline would be formed by another human controlling the sidekick while being limited to the same action space and communication channel could provide better insights.

Other factor that hints that the fun ratings for the baseline level may be underrated is that the game was designed and tweaked to be played with AI. Therefore the player's experience when controlling both characters was not optimized and could probably be easily improved.

7 Conclusion

We have successfully created a puzzle-platformer game with a competent AI sidekick. The game was generally well received and the players perceived the AI sidekick as intelligent, helpful and likeable. Despite our best design efforts, the positive sentiments towards the AI sidekick did not result in increased enjoyment of the game, likely because the AI was too powerful and made the game too easy. We nevertheless believe that our project is an interesting exercise in both game AI and game design and hope that it will serve as an inspiration for sidekick AI implementation in other games. Cooperating with an AI that is a true partner for the player is an intriguing experience that deserves to be explored.

Although our study was relatively small and on a very simple game and there are many limitations, it hints that even if near-optimal sidekick or teammate behavior was achievable for mainstream games, it might not improve gameplay. It all boils down to a fact every game AI practitioner probably already knows all too well: When it comes to character behavior, even the best AI algorithms often do more harm than good unless the AI is perfectly aligned with the overall game design.

The complete sources of the game and complete data gathered during the evaluation are freely accessible at https://bitbucket.org/martin_cerny/coopaigame/.

Acknowledgment

This research is partially supported by the student grant GA UK No. 559813/2013/A-INF/MFF and by SVV project number 260 224.

References

Fern, A.; Natarajan, S.; Judah, K.; and Tadepalli, P. 2007. A decision-theoretic model of assistance. In *International Joint Conference on Artificial Intelligence*, 1879–1884.

Isbister, K., and Nass, C. 2000. Consistency of personality in interactive characters: verbal cues, non-verbal cues, and user characteristics. *International journal of human-computer studies* 53(2):251–267.

Macindoe, O.; Kaelbling, L. P.; and Lozano-Pérez, T. 2012. Pomcop: Belief space planning for sidekicks in cooperative games. In *Proceedings of the Eighth Artificial Intelligence and Interactive Digital Entertainment Conference*, 38–43.

Nass, C.; Moon, Y.; Fogg, B.; Reeves, B.; and Dryer, D. C. 1995. Can computer personalities be human personalities? *International Journal of Human-Computer Studies* 43(2):223–239.

Nass, C.; Fogg, B.; and Moon, Y. 1996. Can computers be teammates? *International Journal of Human-Computer Studies* 45(6):669–678.

Nguyen, T.-H. D.; Hsu, D.; Lee, W.-S.; Leong, T.-Y.; Kaelbling, L. P.; Lozano-Perez, T.; and Grant, A. H. 2011. CAPIR: Collaborative action planning with intention recognition. In *Proceedings of the Seventh Artificial Intelligence and Interactive Digital Entertainment International Conference*, 61–66.

Nikolaidis, S.; Gu, K.; Ramakrishnan, R.; and Shah, J. 2014. Efficient model learning for human-robot collaborative tasks. arXiv preprint arXiv:1405.6341.

Roschelle, J., and Teasley, S. D. 1995. The construction of shared knowledge in collaborative problem solving. In *Computer supported collaborative learning*, 69–97.

Tremblay, J., and Verbrugge, C. 2013. Adaptive companions in fps games. In *Foundations of Digital Games*, 229–236.

Zagal, J. P.; Rick, J.; and Hsi, I. 2006. Collaborative games: Lessons learned from board games. *Simulation & Gaming* 37(1):24–40.