AI platform for supporting believable combat in role-playing games

Matteo Bernacchia[1,a)] Hoshino Jun'ichi[1,b)]

Abstract: Character believability is a fundamental component of role-playing games. In video games, characters not controlled by the player are managed by the game itself, and while most of the desired behaviors can be scripted in advance, the combat phase requires the presence of an AI to interact with the player. As classical AI implementations adopted in role-playing games fall short of representing believable characters, we propose a new AI architecture which can produce combat behaviors while considering believability requirements for the characters.

1. Introduction

Role-playing games (RPG) are a game genre where the player controls one or more characters living in a fictional world [1]. Role-playing (RP) means behaving accordingly to a character's role and enacting it by using speech and actions within the game rules. In a more formal way, RP is the creation and interaction of diegeses (a fictional world or the truth about what exists in a fictional world) by the players [2]. The characters not controlled by the player are called non-playing characters (NPCs), and are acted out by a game master, who also specifies the game setting. Although there are many forms of RPGs, in this paper we will focus on offline role-playing video games, also called computer RPG (CRPG). In these games, since there is no game master, NPCs behavior must be generated by the game itself, and the component responsible for this is generally referred to as game AI [3].

There are many CRPGs and they can be structured very differently, but in all of them we can identify a combat phase where the characters fight. Sometimes the transition is not visible (characters simply start fighting where they are), while sometimes a specific combat mode is enabled, usually bringing up its own dedicated interface to the screen. In either case, there is a clear difference between the required behaviors inside and outside the combat phase. Outside combat, characters interact with each other and with the player mainly with dialogues, quests, or by doing normal daily activities. These behaviors most of the times can be statically scripted in the game beforehand with satisfactory results [4]. For example, game designers know what a NPC can talk about at a given point in the story, what he can sell, where he will be on the map, and finally in what way he is related to a quest. As a result, the NPC behavior will stay true to his role exactly as intended, effectively simulating role-playing as if the game designer were acting for him as a player. The combat phase is a completely different matter. Often it is not known in advance which character will join the battle, what will be the state of each character, and what choices will the player make. For this reason the NPCs behavior cannot be prepared in advance, but must be generated on the fly by an appropriate AI. In our research we will focus on the problems related to the game AI during this specific phase of RPGs.

In most CRPGs the AI of the NPCs is implemented with behavior trees, or finite state machines where the state of the NPC changes in response to some designed event inside the game [5]. This is a very common practice, and not only it is quick and easy to implement but also permits to finely control the resulting behavior. The drawback is that such a static algorithm cannot express the complex dynamics of role-playing [6], producing behaviors that don't follow the character's role and consequently disrupting the character believability [7]. In [8], a survey about the expectation for better NPC behavior in RPGs was conducted on several gaming communities. From the results it is clear that there is still much room for improvement about features like personality, emotions, pro-activeness, individual memory and relationships among characters.

In this paper we aim at building an architecture to support the development of a believable character AI for the combat phase of CRPGs. Believable characters contribute to an overall better player experience [9], [10], and having the role-playing component not only outside but also during the combat phase will improve the consistency of NPCs behavior and of game-play in general. We propose a multi-agent architecture, where each agent's internal behavior is based on the BDI model [11]. On top of this we will build a new frame-

[1] Department of Intelligent Interaction Technologies, Graduate School of Systems and Information Engineering, University of Tsukuba
[a)] kikijikispaccaspecchi@gmail.com
[b)] jhoshino@entcomp.esys.tsukuba.ac.jp

work dedicated to the development of a character AI able to role-play in a believable way. A decentralized system where each agent is autonomous naturally matches our research scenario, and the BDI model is one of the best approaches for developing rational agents in a believable way [12], [13]. We also aim to implement this system so that it can be effectively used in real games, meaning that ease of embedding in game engines, ease of content production and computational requirements will also be prioritized. For testing purposes we also developed a test game, a turn-based tactical RPG, by means of which we will evaluate the NPC believability and ability to emulate role-playing. The evaluation will be based on the gamers' feedback in predefined battle scenarios, by comparing a classical AI implementation to ours.

The paper is organized as follows. First we introduce related works inherent to RPGs, believability, and to the technologies we will use. Then, in section 3 we analyze role-playing and how to model it. In section 4 we explain how we intend to evaluate the system, and finally in section 5 we draw our general conclusions.

2. Related works

2.1 Role-playing

Classical RPG formats, like tabletop RPG (TRPG) and live action RPG (LARP), have always been a source of inspiration for creating CRPGs. In [14], the differences between the two game forms, their respective formats and limitations are investigated. It can be seen that due to the very different medium of expression there are strengths and weaknesses in both of them, and while CRPGs help visualizing the game by their virtual representation of the world, the same game engine also poses limits on the freedom of the player and on the flexibility of the story, and these issues must be solved through the game AI. In [15] an analysis about the storytelling process of non-digital RPG is conducted. It is shown how many components are required prior to being able to play, especially the need of establishing a fictional contract between players so that they can immerse themselves in the game and interact with other characters and the game world. It is also observed how RPGs can be seen as information systems, where different entities have different information needs, rights of access and ways of affecting the information flow, and where an information feedback cycle (reaction, processing, decision, action) rules the system. This helps us define the basis for a role-playing substitute for a real player.

2.2 Believability

Believability is a very complex notion, and there is no generally accurate definition of it. Yet, by limiting our domain to computer games we can delineate its meaning in a way pertinent to our case. In virtual reality we can find the concepts of immersion and presence. Immersion is objective, depending on both hardware and software, and is obtained by substituting real world sensations with virtual ones [16] (in a video game we can think for example about graphics and control devices). Presence on the other hand is subjective, the psychological perception of being in the virtual environment where one is immersed [17], and it is mainly related to the environment's content. In RPGs the characters are the main focus [18], and as a consequence enhancing their believability greatly affects the presence too [6]. In [7] it is explained how believability does not concern honesty or reliability, and how instead a believable character is the one that allows the audience's suspension of disbelief. In other words, a believable character is the one that gives the illusion of life [19]. For video games there have been several proposals about the requirements of a believable character [18], [20], [21], [22], also depending on whether we require them to give the illusion of being alive or to give the illusion of being controlled by another player [23]. As we are targeting single-player offline games, the former is considered together with a set of requirements thought for achieving believability during the RPGs combat phase.

2.3 Multi-agent systems

Multi-agent systems (MAS) are based on the concept of agents, and while there is a wide range of definitions [24], we can describe them as reactive, autonomous, internally-motivated entities embedded in changing, uncertain worlds which they perceive and in which they act [25]. Embodied agents are agents manifested in a body, comporting not only a physical representation (in our case, digital), but also meaning that their ability to interact with the environment is limited and must happen by means of their body capabilities. In [26] demonstrates how strongly the adoption of embodied agents and the presence of emotion influences the users' opinion about the agent in a positive way. We can adopt this concept by letting each game character be the embodiment of an agent, allowing us to have a more realistic AI model. While there are already several agent-based frameworks available, for example Mason [27] or JADE [28], we decided to roll our own implementation for many reasons. Many frameworks run on Java or other virtual environments that can't be embedded in a game, and they are also big and complex to set up and to use. In our implementation we aim at building a lightweight and self-contained cross-platform library that can be easily interfaced with an existing game engine and that can provide the necessary functionalities without using too much resources.

2.4 Belief-desire-intention model

The belief-desire-intention (BDI) software model is a model built on Bratman's theory of practical reasoning [11], which is about resolving, through reflection, the question of what one is to do [29]. Practical reasoning comes from folk psychology, which is the natural capacity to predict human behavior, attribute mental states to humans and finally explain the behavior of humans in terms of their possessing mental states [30]. The concepts at the core of the BDI model can be naturally applied when creating human-like agents, and for this reason it has been the model of choice for rational agents. In [9] it is shown how characters which possess unobservable mental-states like goals or who engage in social interactions cannot be built with traditional game AI programming, but how instead the BDI-based approaches are suitable for this purpose. Although still lacking several aspects of human reasoning [31], many BDI-based agent programming languages and frameworks have been developed [32], like Jadex [33], Jason [34] or JACK [35]. Looking at the game industry though, with the notable exception of Black & White [36], there are no commercial games using BDI agents. With this in mind, we decided not use preexisting frameworks but implement our own BDI model suited for games' adoption.

3. Believable combat AI

In many TPRGs like Dungeons & Dragons [37] combat is clearly separated from the normal gameplay. Before it starts, players can act freely without a predefined temporal order, and most of the actions are direct speech (for example, a player says "Guard, open the door please."), or descriptions (for example, "We go back to the tavern", or "I stand up and look him in the eyes."). Once combat triggers though, when can players act, what actions can they do, what are the results of these actions, all must follow the combat rules. To accommodate the fact that applying rules and rolling dices takes time, and to let players think about what action to execute, the normal time flow is interrupted and organized into turns. Video games are very similar, but since they can do all the necessary calculations for the player, depending on the game genre combat can also be real-time (action RPGs).

3.1 Requirements

As stated in the introduction, a major difference with TRPG is that while players can keep on role-playing during the fight, the combat AI usually performs very poorly from that point of view. The first step to solve this problem is defining the requirements needed to achieve believable characters. As cited in the related works, these have been outlined in many papers, but they are all about interactive drama, ignoring the specific requirements needed during the combat phase. Especially in turn-based games, where combat is very complex and tactical, we think that the requirements specific to this phase are different and must be reconsidered.

The combat phase usually involves its own set of possible actions, like skills, spells, items and so on. Movement is limited, as are resources like action points or magic points. Time also is very important in both real-time and turn-based games, as it impacts the ability to act and speak freely. Being a tactical phase the complexity too is higher. Combat may feature complex terrain, a great number of units, many possible skills, status effects and character parameters. Although outside combat there is a certain degree of freedom, while fighting all characters must follow strict rules about what they can do and how they can do it. Finally, a big difference is danger. Most of the times combat means that the characters are fighting for their own lives, and this influences how they behave and their priorities. As there is always the presence of an enemy, interacting with them requires very different behaviors from friends and neutral characters. All these factors put constraints on the characters' believability, but at the same time also offer a new way to present believable behaviors to the eyes of the player. While keeping these differences in mind, the set of requirements for believability laid out by the Oz group [20] is adopted as starting point.

(R1) Personality One of the most important requirements, it is reflected on every action, and if rich enough makes characters unique. Personality is not about the general behaviors but is in the details, specific expressions of given character.

(R2) Emotion Believable characters should have emotional reactions, and express them according to their own personality.

(R3) Self-motivation Characters should not only react to events, but also have their own internal drives and desires. They should also be able to pursue multiple goals in parallel, or in an interleaved way.

(R4) Change As time passes characters have experiences, change and grow.

(R5) Social relationships Believable characters should be able to interact with other characters, consistently with the relationship they belong to.

(R6) Resource-bound Characters should have limits about how much they can think and do.

(R7) Situated Characters must appear to be situated in an environment, and dynamically act in response to the unfolding situation.

(R8) Individuality A character is an independent entity, with its own private knowledge and way of thinking.

4. System

Units in a RPG can be naturally thought of as part of a multi-agent system, where the environment is the current game map. They all need to have their own mental state, knowledge and goals, while keeping the interactions with the world and the other units limited to their local physical interfaces (no blackboard systems, which are an abstraction and can also be considered cheating by the player). The individual agent mental model is implemented following the BDI model, as it allows us to represent such internal states in a natural way.

4.1 Components

4.1.1 Beliefs

Beliefs represent the character's knowledge, but are called this way because are not guaranteed to be true or consistent. They can change through time, but can also be saved and restored, working as the character's long term memory. The belief-base is where all the beliefs of the agent are stored, and contrary to many BDI implementations there is no logic programming involved. While this does not allow inference of new information or belief revision, it allows for grater freedom about what can be stored (arbitrary data) and its memory representation.

Beliefs can be of three types: internal, external and dynamic. Internal beliefs are information only present inside the AI. External beliefs point to data located outside, typically in the game engine. Dynamic beliefs are evaluated when accessed and are implemented as functions (they can be useful to obtain data continuously changing). All belief modifications raise events that can be intercepted by one or more components interested in them (like waiting for a belief to assume a certain value). Albeit very simple, beliefs implemented this give a lot of flexibility, and can be extended to support more complex algorithms like prolog-like unification.

4.1.2 Goals

A goal is a desired state which the agent is trying to achieve. The goal-base is where the goal schemas are stored, indexed by name. We can then instance new goals by just passing the desired name and eventually the parameters to be bound. For example to create the goal to be in a certain place we send "be in location" and a vector containing the coordinates as a parameter. Goals can have dynamic conditions to check if they succeeded or failed, and can also react to events. This allows them to be customized and implement specialized types of goals, for example:

Achieve The classical goal. Can be implemented by adding a success condition to be checked. If the retry flag is on different plans can be also tried in case of failure. Initial conditions and context conditions check the validity of the goal.

Perform The simplest type of goal, succeeds when the plan assigned to it in turn succeeds.

Maintain This not only seeks to achieve a desired state, but also its retention.

Query This is a special type of achieve goal that aims at obtaining information. The success condition checks if we have this information in the belief base.

4.1.3 Plans

A plan is a course of action designed to achieve a certain goal, and is the building block of agent behavior. It contains a specification about which goals it can accomplish, and like goals can have additional conditions (success, failure, context). The plan body can be arbitrary code, has access to the agent interface and is executed in an interleaved fashion by making use of a yield function which suspends the execution until the next cycle. This can be used also to put the plan in an idle state while waiting for an event to occur or a sub-goal to be achieved. Similarly to the goal-base, the plan-base holds the plan schemas for the agent to be instanced.

While normally plans are instanced only when selected to achieve a goal, they can also have creation options which consent them to be activated in a reactive manner, in response to particular events.

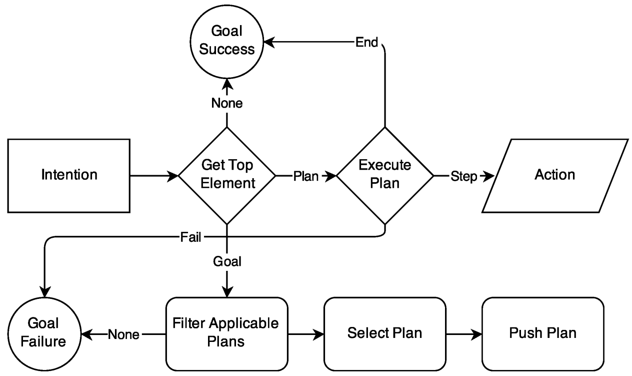

Fig. 1 Stepping into an intention.

4.1.4 Intentions

Intentions represent actively pursued goals, and multiple intentions can be executed in parallel or in an interleaved way. An intention is implemented as a stack where goals and plans are pushed one on top of the other. When an intention is executed, the top element is updated consequently:

Goal If the goal is new then must be managed by finding an appropriate plan, instancing it and pushing it on top of the stack. If it is not new then it means that the relative plan has finished, and in case of success and we can pop the goal and resume the plan below. Conversely if the plan had failed then the goal fails too, with the exception of goals with the retry flag enabled which can try different plans before giving up.

Plan The next step of the plan body is executed. If waiting for a sub-goal to complete it can check the success status and the attached return data. The plan can handle eventual failures in its body, or fail in turn. If it was the last step, the plan is popped, while success status and returned data are passed to the parent goal.

Intentions continuously monitors all plans and goals context conditions (from bottom to top) to assure their validity, and in case of success or failure of an intermediate element of the stack pops everything above it since it is not needed anymore.

4.1.5 Devices

Devices handle all input and output with the environment. An agent can have many devices, each with their sensors and actuators.

Sensor When something happens in the game world, an event is sent to the agent, but it is not directly accessible. Instead, the agent sensors can try to intercept the external event (for example, a sensor to hear noises can detect only sound events within its range), and if case of success dispatch the relative internal event wrapping the elaborated data.

Actuator Actuators are the only means of the agent to interact with the environment. The plan that sends the action request can wait for the relative completion event (synchronous) or continue its execution (asynchronous). A third option is to do an asynchronous call and push a plan on a separate intention to handle the results.

4.2 BDI interpreter

During the game when the turn of a character begins his associated agent is updated, repeating its step function until he has no more thinking to do for that turn. On each iteration events accumulated in the event queue are processed, then an intention is selected and executed. All other computations are triggered by events:

System events Events generated by the AI or by the game engine itself (for example, "begin turn").

External events These are the events generated by the agent sensors when they transpose events coming from the environment.

Internal events Events generated by belief updates, goal/plan state changes and events relative to the devices.

Another crucial component of the BDI interpreter are the selection functions. These are used by the BDI interpreter to control the flow of execution, and can also be overridden to have even more variety of behaviors. The functions in question are:

Select event Selects the next event in the event queue to process.

Schedule Selects the next intention to execute.

Process goal When given a goal the agent must find a way to achieve it. To do so, first filters out all plans that are not applicable (achieve a different goal, already failed, false initial condition), then this function selects a plan to use. The function can also be customized to use parameters like goal priority. If present, a special type of plan called metaplan can help the choice of the best plan.

Having both the classical BDI flow and reactive triggers for plans and goals allows to have great flexibility and the ability to cover many corner cases that may present when developing a concrete game AI.

4.3 BDI and believability

The BDI model allows us to cover many of the believability requirements through its main components. Beliefs store data about the character itself, the other characters and the environment, and each agent has its own belief set (**R8**). Information is not only related to the current fight, but also includes events and experiences from outside the combat phase. This allows the agent to change and learn, adapting its behavior as time passes (**R4**). Goals promote pro-activeness and long term objectives (**R3**). Their priority can be influenced by relationships and emotions (**R2**, **R5**). Plans are probably the most influential component of our system. The set of plans known by an agent (**R8**) can depend on the character parameters (race, class, level), on its personal history (past battle experience, quests) and not only shape its combat tactics (**R7**) but also its personality (**R1**). The set is dynamic, as new plans can be learned from battle experience or for example by gaining new levels (**R4**). When there are multiple plans to achieve the same goal a meta plan is used if available (a plan to select a plan), and

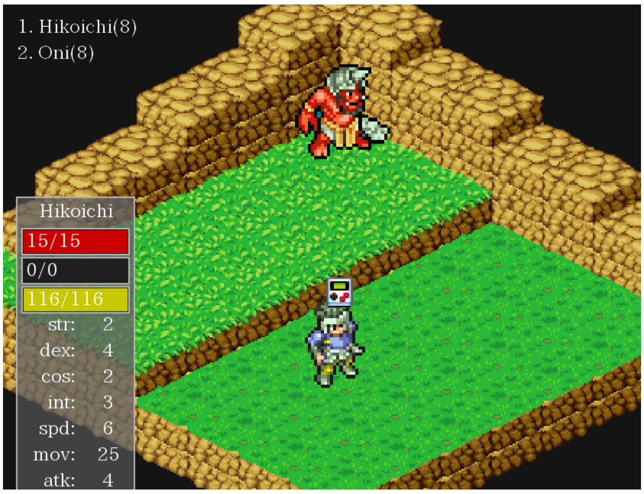

Fig. 2 Test game

this can be exploited to express a peculiar way of thinking (**R1**, **R2**). Sensing and acting is done through sensors and actuators, which are the only interface of the agent to the physical environment (**R7**). These have limitations, for example a sensor could perceive events only within a specific range and not further (**R6**). If available, communication sensors and actuators, paired with a communication protocol specified by plans, can be used to let the agents exchange data. This can allow additional behaviors, for example conversing during combat (**R5**), and by exchanging goals and plans agents can coordinate themselves by giving and receiving orders and requests (**R5**). Lastly, intentions allows the parallel or interleaved execution of plans, allowing the agent to do multiple things at the same time (**R3**), and the dynamic stack of subgoals and subplans allows the agent to behave correctly in the present situation.

5. Evaluation

5.1 Game

To evaluate the system we will use a turn-based role-playing game (SRPG). Since most of these games are not open-source, a sample game will be developed for testing purposes. The map consists of an isometric grid (Fig.2) of square tiles where characters move on (discrete coordinates). Time is organized in battle turns where each character acts in a predefined order (usually from faster to slower). Since characters can only act during their turn, the AI will be executed one instance at a time, storing any event happening during the game in the event list for later processing.

5.2 Survey

Evaluating character believability is not easy to do, as it is part of the player experience. In [38] different strategies to assessing believability are described. Subjective assessment is done by asking the player about their experience directly. This can be done by means of a survey, but when to ask (at the end or during the game), how to ask (boolean answer choice, ranking, comparing two or more versions), and limitations like experimental noise or memory limitations must be considered to efficiently gather data.

We plan to test character believability by providing multiple pre-made battle scenarios. Each of them recreates some of the most common situations that may arise in these game genre, and is also associated to the believability requirements (R1-R8) which are most relevant in that specific case. Some sample scenarios are:

- In the first phase a character faces an enemy without having any prior knowledge about him. In the second phase, the same character faces the same enemy again. Comparing the behavior with and without knowledge allows to evaluate requirements **R4** and **R7**.
- During two phases, two different characters (different personality, personal goals, experience, relationships) that have identical parameters (class, level, equipment) are presented with the same battle situation. Seeing how the behavior is not only dependent on battle-related parameters allows us to evaluate requirements **R1**, **R2**, **R5**, **R8**.
- A group of characters cooperates to defeat an enemy. Information sharing (beliefs, plans), orders and requests (goals) have place through verbal communication visible to the player. This is related to requirements **R3**-**R7**.

The player will have to answer to a set of questions regarding character behavior for each scenario, where each question is directly related to a believability requirement.

6. Conclusions

In this paper we proposed an AI architecture for supporting believability during the combat phase in RPGs. A believable AI can offer an improved game experience, and be of great relevance especially for the next generation of games. We plan on improving our definition and implementation of believability, while conducting thorough tests about its influence on the player.

References

[1] Bartle, R. A.: *Designing virtual worlds*, New Riders (2004).

[2] Montola, M.: Role-playing as interactive construction of subjective diegeses, *As Larp Grows Up*, pp. 82–89 (2003).

[3] Millington, I. and Funge, J.: *Artificial intelligence for games*, CRC Press (2009).

[4] Yildirim, S. and Stene, S. B.: A survey on the need and use of AI in game agents, *Proceedings of the 2008 Spring simulation multiconference*, Society for Computer Simulation International, pp. 124–131 (2008).

[5] Charles, D.: Enhancing gameplay: Challenges for artificial intelligence in digital games, *Proceedings of the 1st World Conference on Digital Games*, pp. 4–6 (2003).

[6] Tencé, F., Buche, C., De Loor, P. and Marc, O.: The challenge of believability in video games: Definitions, agents models and imitation learning, *arXiv preprint arXiv:1009.0451* (2010).

[7] Bates, J.: The role of emotion in believable agents, *Communications of the ACM*, Vol. 37, No. 7, pp. 122–125 (1994).

[8] Afonso, N. and Prada, R.: Agents that relate: Improving the social believability of non-player characters in role-playing games, *Entertainment Computing-ICEC 2008*, Springer, pp. 34–45 (2009).

[9] Sindlar, M. P., Dastani, M. M. and Meyer, J.-J. C.: BDI-based development of virtual characters with a theory of mind, *Intelligent Virtual Agents*, Springer, pp. 34–41 (2009).

[10] Reilly, W. S.: Believable Social and Emotional Agents., Technical report, DTIC Document (1996).

[11] Bratman, M.: Intention, plans, and practical reason, *Harvard University* (1987).

[12] Rao, A. S. and Georgeff, M. P.: Modeling rational agents within a BDI-architecture, *KR*, Vol. 91, pp. 473–484 (1991).

[13] Norling, E. and Sonenberg, L.: Creating interactive characters with BDI agents, *Proceedings of the Australian Workshop on Interactive Entertainment (IE04)*, pp. 69–76 (2004).

[14] Tychsen, A.: Role playing games: comparative analysis across two media platforms, *Proceedings of the 3rd Australasian conference on Interactive entertainment*, Murdoch University, pp. 75–82 (2006).

[15] Tychsen, A.: Tales for the Many: Process and Authorial Control in Multi-player Role-Playing Games, *Interactive Storytelling*, Springer, pp. 309–320 (2008).

[16] Slater, M., Usoh, M. and Steed, A.: Taking steps: the influence of a walking technique on presence in virtual reality, *ACM Transactions on Computer-Human Interaction (TOCHI)*, Vol. 2, No. 3, pp. 201–219 (1995).

[17] Steuer, J.: Defining virtual reality: Dimensions determining telepresence, *Journal of communication*, Vol. 42, No. 4, pp. 73–93 (1992).

[18] Lankoski, P.: Goals, affects, and empathy in games, *Philosophy of Computer Games* (2007).

[19] Thomas, F., Johnston, O. and Thomas, F.: *The illusion of life: Disney animation*, Hyperion New York (1995).

[20] Loyall, A. B.: Believable agents: building interactive personalities, PhD Thesis, Stanford University (1997).

[21] Mateas, M.: *An Oz-centric review of interactive drama and believable agents*, Springer (1999).

[22] Grey, J. and Bryson, J. J.: Procedural quests: A focus for agent interaction in role-playing-games, *Proceedings of the AISB 2011 Symposium: AI & Games*, University of Bath, pp. 3–10 (2011).

[23] Livingstone, D.: Turing's test and believable AI in games, *Computers in Entertainment (CIE)*, Vol. 4, No. 1, p. 6 (2006).

[24] Franklin, S. and Graesser, A.: Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, *Intelligent agents III agent theories, architectures, and languages*, Springer, pp. 21–35 (1997).

[25] Kinny, D. and Georgeff, M.: Modelling and design of multi-agent systems, *Intelligent Agents III Agent Theories, Architectures, and Languages*, Springer, pp. 1–20 (1997).

[26] Brave, S., Nass, C. and Hutchinson, K.: Computers that care: investigating the effects of orientation of emotion exhibited by an embodied computer agent, *International journal of human-computer studies*, Vol. 62, No. 2, pp. 161–178 (2005).

[27] Luke, S., Cioffi-Revilla, C., Panait, L. and Sullivan, K.: Mason: A new multi-agent simulation toolkit, *Proceedings of the 2004 SwarmFest Workshop*, Vol. 8 (2004).

[28] Caire, F. B. G., Poggi, A. and Rimassa, G.: JADE. A white paper (2003).

[29] Wallace, R. J.: Practical Reason, *The Stanford Encyclopedia of Philosophy* (Zalta, E. N., ed.), summer 2014 edition (2014).

[30] Ravenscroft, I.: Folk psychology as a theory (2010).

[31] Norling, E.: Folk psychology for human modelling: Extending the BDI paradigm, *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*, IEEE Computer Society, pp. 202–209 (2004).

[32] Bordini, R. H., Dastani, M., Dix, J. and Seghrouchni, A. E. F.: *Multi-Agent Programming*, Springer (2005).

[33] Braubach, L., Lamersdorf, W. and Pokahr, A.: Jadex: Implementing a bdi-infrastructure for jade agents (2003).

[34] Bordini, R. H., Hübner, J. F. and Wooldridge, M.: *Programming multi-agent systems in AgentSpeak using Jason*, Vol. 8, John Wiley & Sons (2007).

[35] Howden, N., Rönnquist, R., Hodgson, A. and Lucas, A.: JACK intelligent agents-summary of an agent infrastructure, *5th International conference on autonomous agents* (2001).

[36] Studios, L.: *Black & White*, Electronic arts (2001).

[37] Gygax, G., Arneson, D. and Hobbies, T.: *Dungeons & Dragons*, Vol. 19, Tactical Studies Rules Lake Geneva, WI (1974).

[38] Togelius, J., Yannakakis, G. N., Karakovskiy, S. and Shaker, N.: Assessing believability, *Believable Bots*, Springer, pp. 215–230 (2012).