Periodica Polytechnica
Electrical Engineering
and Computer Science

61(2), pp. 198-208, 2017
https://doi.org/10.3311/PPee.10313

Applying and Augmenting Deep Reinforcement Learning in Serious Games through Interaction

Aline Dobrovsky[1*], Uwe M. Borghoff[1], Marko Hofmann[1]

RESEARCH ARTICLE

Received 22 November 2016; accepted 08 March 2017

Abstract

Serious games belong to the most important future e-learning trends and are frequently used in recruitment and training. Their development, however, is still a demanding and tedious process, especially when regarding reasonable non-player character behaviour. Serious games can generally profit from diverse, adaptive behaviour to increase learning effectiveness. Deep reinforcement learning has already shown considerable results in automatically generating successful AI behaviour, but its past applications were mainly focused on optimization and short-horizon games. To expand the underlying ideas to serious games, we introduce a new approach of augmenting the application of deep reinforcement learning methods by interactively making use of domain experts' knowledge to guide the learning process. Thereby, we aim to establish a synergistic combination of experts and emergent cognitive systems to create adaptive and more human behaviour. We call this approach interactive deep reinforcement learning and point out important aspects regarding realization within a novel framework.

Keywords

Serious Games, Deep Reinforcement Learning, Interactive Learning, Cognitive Systems, Game Artificial Intelligence

1 Introduction

The term serious game (SG) origins from [1] and refers to games not exclusively developed for mere entertainment, but primarily for creating educational value. SGs are counted among the current e-learning trends and are gaining more and more acceptance and influence [2]. SG development slightly differs from that of entertainment games: SGs are often individual products for a restricted target audience, industrial branch or company. This results in high expenditure and deficient reusability, although the market shows growing interest in cost-efficient and customized applications [2]. One of the most relevant, but also most time-consuming tasks is the creation of reasonable, human-like non-player character (NPC) behaviour [3-5]. Thus, simplifying authoring and adaptation of AI in SGs seems desirable and profitable.

Machine learning, especially reinforcement learning (RL), is occasionally used for automated NPC behaviour generation. Nevertheless, several issues arise when applying RL to create believable and diverse behaviour in complex scenarios. We indicate a way to overcome these issues by combining RL with human guidance, including effective collaboration between a learning system and human experts.

In this paper, we give a short introduction to some challenges of behaviour generation in serious games and to the background of RL, deep reinforcement learning (DRL) and interactive reinforcement learning (iRL). We show related approaches and depict current issues of applying DRL methods to SGs. Furthermore, we introduce SanTrain as a SG providing challenging scenarios for NPC behaviour generation. Finally, we show how our approach of interactive deep reinforcement learning (iDRL), integrated into a flexible framework, could enhance AI development in SGs and exemplarily indicate valuable application opportunities in SanTrain.[1]

[1] Fakultät für Informatik, Universität der Bundeswehr München, 85577 Neubiberg, Germany

* Corresponding author, e-mail: aline.dobrovsky@unibw.de

1 This paper is an extended version of a preliminary conference paper that was presented at CogInfoCom 2016 [6].

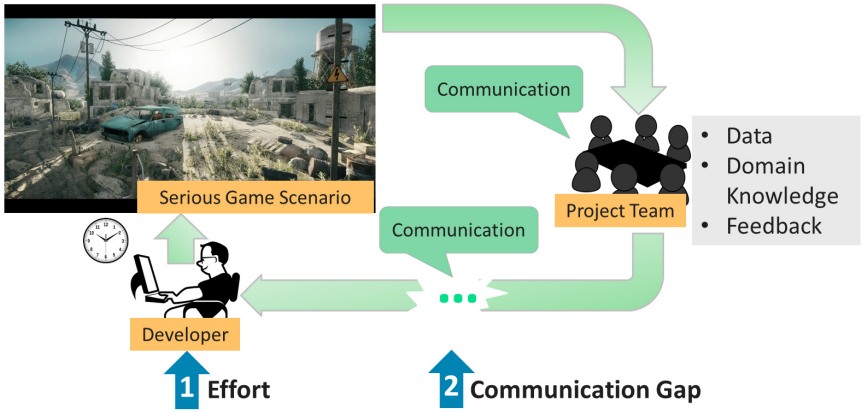

Fig. 1 Simplified, iterative process of serious game AI generation and occuring challenges

1.1 Challenges in Serious Game AI Development

Fig. 1 shows a simplified outline of the serious game AI development process and depicts two prevalent challenges we identified: the effort of AI generation and the communication gap between developers and domain experts.

- **Effort of AI generation** Non-player character behaviour is still mainly produced through writing scripted instructions as hand-coded, rule-based systems. The effort needed to create reasonable behaviour for every scenario of the game is dependent on its complexity, but always requires the ability to thoroughly anticipate reasonable behaviour for all possible game situations.
- **Communication gap** In regular meetings during an repetitive process, the stakeholders (project leaders, domain experts, developers) iteratively refine the AI model. Domain experts give feedback to the current, presented behaviour and provide additional data and domain knowledge. They describe the intended behaviour of the AI model in a qualitative manner, e.g. 'search for cover when under fire'. This behaviour is subsequently translated into a rule-based behaviour system with quantified values, e.g. 'distance of enemy bullet impact to avatar'. Developers and experts communicate about behaviour that is strongly dependent on training, experience and personality. The goal of trying to find a game implementable definition and description of desirable AI behaviour easily leads to misunderstandings and creates a communication gap (cf. [7]).

1.2 Reinforcement Learning (RL)

Reinforcement learning means learning a mapping of situations to actions from interaction with an environment. RL agents try to maximize a numerical reward signal received from their environment. Fig. 2 shows the standard interaction model over a sequence of discrete time steps. At each time step, the agent selects an action according to the current state and receives a reward in the following state. The learned mapping is called a *policy* and describes the probability of selecting action a in state s. In Q-learning, the agent follows a policy that promises to maximize the function $Q(s,a)$, where Q describes the maximum discounted future reward when performing action a in state s and continuing with optimal choices. A comprehensive RL overview can be found in [8].

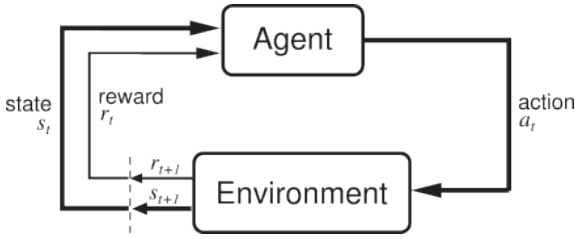

Fig. 2 Agent-environment interaction in reinforcement learning [8]

1.3 Deep Reinforcement Learning (DRL)

DRL means the combination of RL with deep machine learning methods. Deep machine learning is inspired by the research of structure and information processing of the neocortex. It tries to capture spatio-temporal dependencies and involves training on large sets of observations to overcome the curse of dimensionality [9]. Within RL, a deep learning architecture can be used as function approximator in large state-action spaces. In deep Q-learning, the conventional table containing all state-action pairs and the learned rewards is replaced by a deep artificial neural network (ANN). The ANN is more compact, generalizes better on unknown states and captures hierarchical features of the problem. The successful applications of deep Q-Learning to a set of different Atari games [10] and computer Go [11] are commendable examples.

When applying DRL in a serious game, the game provides the environment from which the agent receives a representation of the current game state and a reward value (as illustrated in Fig. 3). A deep ANN can be used as the agent's decision making component. The action to be executed in the game is selected depending on the current state representation input and the learned behaviour.

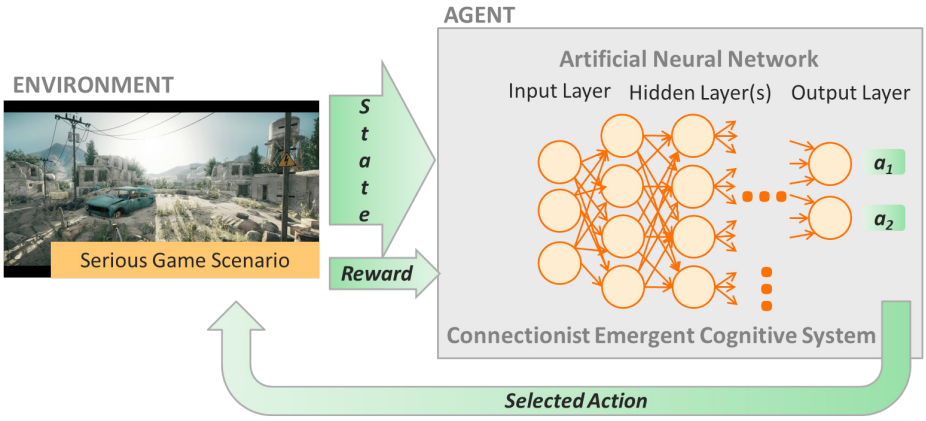

Fig. 3 (Deep) Reinforcement learning with a serious game as environment and an artificial neural network as the agent's action selection component

1.4 Interactive Reinforcement Learning (iRL)

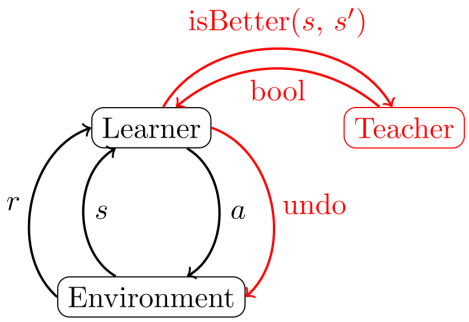

Fig. 4 Example of an iRL framework with interactive feedback and the teacher's ability to undo the last action [12].

In iRL, an agent obtains reward signals not only from its environment, but additionally or exclusively from interactions with a human trainer. Human trainers could be domain experts and maybe non-programmers. They are able to give feedback at any time and not only in goal states. Objectives of using an interactive approach in RL are to improve convergence speed and to create biased and influenceable behaviour in complex tasks. Different interaction methods are possible, e.g. a human trainer can give numeric rewards, advise a concrete action or demonstrate a desired behaviour. Objectives of using interactive approaches are improved convergence speed in learning tasks and biased behaviour in complex scenarios. However, there are still a lot of open issues in combining expert knowledge and automated learning. Deciding for a method and tuning of the form and intensity of human feedback cannot be standardized. It remains an intriguing task and strongly depends on the application scenario.

1.5 Objectives

Our motivation is to help developers in creating believable NPC AI and thereby simplify serious game development. The use of machine learning techniques can partially automate the process but requires machine learning experts, complex software integration, parameter tuning, powerful hardware and long training times. Furthermore, vague objectives of game AI like 'fun' and 'learning' are hard to encode. We aim to overcome some of the current problems by integrating expert knowledge in an efficient learning process and by establishing a synergistic collaboration between experts and cognitive systems. Our concepts are to be implemented within a flexible framework, which shall be easy to use by AI developers and domain experts.

2 Related Work

To the best of our knowledge, we found no similar approach of combining DRL and iRL. Particularly, we found no framework offering interactive DRL to support SG development. Nevertheless, we were inspired by different engaging ideas and current research in the areas of DRL, iRL and General Game Playing (GGP) [13]. A general overview of ML techniques used in SG can be found in the literature overview of [14].

Ongoing research in the area of General Video Game Playing (GVGP) [15] attempts to develop algorithms that play any game without knowing it a priori. GVGP provides relational, object-oriented representations of 2D Atari-like game world features and offers information about game states and a forward model. The planned learning track of the corresponding competition, excluding the forward model, is considered to induce promising new approaches in the areas of unsupervised learning and RL. A related attempt is pursued by the Arcade Learning Environment (ALE) [16]. It offers a platform for evaluating domain-independent AI by providing an interface to a large number of various Atari 2600 game environments. The available games can be used as RL problems, whereby the state and action space is given through pixel-arrays of the 2D game screen and possible joystick controller moves. ALE notices the agent about the current accumulated score and whether the game has ended.

Several approaches use evolutionary techniques in combination with artificial neural networks (ANN) to produce human-like behaviour. The authors of [17] present a system for

automatic evolution and adaptation of ANN based on evolutionary algorithms. Their approach is based on long offline learning sessions with subsequent testing procedures. In [18], neuro-evolution for deep learning is investigated, showing good results in training a feature extractor for use by other ML approaches. The authors of [19] apply neuro-evolution to general Atari game playing in the ALE. They investigate the mutual influence of different state representations and the application of different neuro-evolution algorithms and show that the use of neuro-evolution overcomes previous RL problems with large state spaces and sparse reward gradients. A variant of deep learning is used in [20] for player goal recognition, a player-modeling task. It is used to predict players' goals in an open-ended digital game world by learning from a collection of player interactions.

In 2013, the authors of DeepMind Technologies, later Google DeepMind, stated to present the first successful approach of applying RL for learning control policies directly from high-dimensional sensory input [10, 11]. A convolutional neural network is trained with an algorithm called 'Deep Q-learning with Experience Replay' on several games of the ALE and partially outperforms other approaches and human players. This approach is combined with Monte Carlo Tree Search (MCTS) in [21]. The use of model-based, slow planning agents proves to be a good way of supplying training data for a deep learning architecture used in real-time gameplay. The combination of RL and MCTS is also used by Google DeepMind for their computer Go program 'Alphago', which is the first to ever have defeated a human professional Go player [22]. The authors combine supervised learning of expert moves with RL in self-plays to train policy- and value-networks used by MCTS. Recent research shows that DRL for Atari game-playing can also be trained efficiently on customary hardware using asynchronous gradient descent and parallel actor learners [23]. The method of asynchronous actor critic even succeeds on continuous motor control tasks and on finding rewards in a 3D random maze.

3 Problem Statement

Our examinations focus on the support of SG AI developers in NPC controlling. We intend to cover a wide range and variety of games, e.g. single-player board games like 8puzzle up to complex multiplayer 3D games. Thus, we need a method to cope with different state-action space representations and accesses to game-relevant information. Traditional NPC behaviour generation methods, primarily containing scripting and rule-based systems like finite-state machines, imply laborious hand-crafting of NPC AI. Especially in complex game worlds, self-adapting AI can relieve developers of the need to anticipate reasonable behaviour for every possible game situation [24]. Additionally, the lack of dynamic, adaptive behaviour makes hard-coded approaches easily exploitable [25].

Machine learning can help to ease the authorial burden of creating such game AI. However, there are some prevalent challenges in applying ML to games [26]. ML algorithms often need long time and vast data for training and a meaningful target function. Additionally, game AI is often granted only a specific period of time and limited resources. A high-level symbolic knowledge representation would allow for a straightforward and understandable AI manipulation, but also require expense in obtaining this knowledge representation [27]. Furthermore, explainability of algorithms and their results is an important factor to gain trust in the algorithms' decisions, because learning methods are known to sometimes produce unpredictable outcomes. Moreover, specific problems of RL must be handled in computer games [28]: curse of dimensionality (large state-action spaces), partial observability problem (hidden states), generalization and exploration-exploitation dilemma, credit structuring and temporal credit assignment problem (delayed and sparse rewards).

DRL can resolve some of these issues: it enables efficient training on large datasets and raw data input. The use of Artificial Neural Networks (ANN) improves generalization, even on unknown states. The presented approach [10] also showed that there's no need to adapt a game and that efficient application is possible. Although DRL can handle quick-moving, complex games based on visual input, the game score results of [11] indicate that there remain issues with long-horizon games that offer only sparse rewards (e.g. platformers like 'Amidar' or games with open, complex worlds and differently visualized information like 'Battle Zone').

7	2	4
5		6
8	3	1

Fig. 5 8puzzle game example

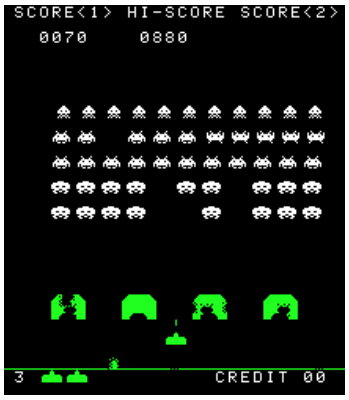

Fig. 6 Atari Space Invaders screenshot [29]

The described challenges grow with size of the state-action space and scenario complexity. Figs. 5, 6 and Figs. 7, 10 illustrate increasing demands on AI for different types of games. In Fig. 5, the state-action space can easily be defined by the positions of all tiles and the player-actions of 'do nothing', 'select tile' and 'move tile'. Time limits for action selection are optional (cf. GGP competition [13]). Fig. 6 shows an exemplary Atari game. States can be defined as is done in ALE, by 2D pixel arrays. The action space consists of 18 possible actions. Games of this type require fast perception and reaction in real-time environments containing non-deterministic elements.

If we move on to modern 3D multiplayer games, we often can't specify a straightforward definition of the state-action space. In these games, AI has to deal with real-time environments containing large amounts of various game objects and many (non-)deterministic events. Players and NPCs can take different positions and stances, possess equipment and perform sets of different actions (e.g. move, shoot) with varying constraints (e.g. stamina, perception). In the following section, we will introduce SanTrain as an example of a modern 3D serious game we use as an application scenario for research.

3.1 SanTrain: A Serious Game for Tactical Combat Casualty Care (TCCC)

SanTrain is a serious game in development in the domain of military first aid. It provides a game-based learning and training platform to train TCCC decision making skills in an ego-shooter perspective.

Fig. 7 Screenshot of SanTrain

TCCC means specialized first aid on the battlefield and is based on a series of simple life saving steps and clear priorities for the first minutes following injuries. The goals are to prioritize and treat life-threatening injuries, to prevent additional injuries and to be able to complete the military mission. The underlying principles origin from US special operation forces' experiences in real combat scenarios. It has been shown that, through prioritized emergency treatment on battlefield, lives of injured with life threatening injuries can be saved [30]. At present, TCCC training is practised in armed forces not only by medical personnel but also by regular servicemen.

SanTrain can provide parts of TCCC training in a cost effective manner through simulating the major aspects in a game, whereas purely traditional TCCC training is limited through time, budget and the number of competent tutoring personnel. As an educational tool, SanTrain has to teach an extremely complex subject matter efficiently. One of the most important aspects for realistic TCCC training is a very precise pathophysiologic model of the human body. It has to represent all visible vital signs of a patient's body, because diagnosis and treatment is based on their perception. Furthermore, it has to model all relevant vital parameters, consequences of injuries and effects of medical treatment and their evolution over time. A realistic simulation and versatile modeling is needed, as the outcome of survival is dependent on timeliness and correctness of a trainee's decisions. Furthermore, as a 3D game, SanTrain has to enhance learning motivation through convincing visualization capabilities, natural interaction between players, demonstration of realistic stories in domain specific scenarios and convincing, supportive AI capabilities.

The development of serious games like SanTrain requires effective cooperation of subject-matter experts, didactic experts, simulation experts and game developers [32]. Furthermore, there are requirements for cost-effective development and simple adaptability, with limited number of potential users and limited availability of trainer capacities in very specific domains. One major challenge is that algorithms modeling learning matter have to be validated, which is often done by showing SME typical courses of action in the game. Therefore, a playable version must be available; a late validation can delay development because thorough validation is critical regarding plausibility and learning effects [32]. SanTrain meets these challenges through the design of a flexible game-architecture with well defined-interfaces, as shown in Fig. 8 and the use of separate development teams; game developers for designing an attractive game and a pedagogical expert team for teaching of medical principles. Various application scenarios for using elaborate AI exist in a SG like SanTrain and the flexible architecture offers simple integration potential. Some examples of integration possibilities of our interactive DRL approach are described in Section 4.4.

3.2 Application Goals and Implications

The described characteristics and development challenges illustrate the implications on the SG AI development process and objectives. Serious game AI has to be supportive, meet the needs of teaching and the learning goal and, when applied on human NPCs, has to exhibit realistic and variable behaviour to enhance learning success. Nonetheless, there is need for a cost-effective development process and less effort in AI development due to limited time and capacities. Furthermore, regarding the need for validation in SGs, AI development must be integrated in the general SG development process, including and involving different subject matter experts. Therefore, AI and its development have to be accessible to, comprehensible for and influenceable by different experts.

As we intend to cover games of differing complexity, we have to deal with the most difficult cases. Furthermore, we can't generally assume to have direct access to game state information or subsequent game states or to get recorded and stored replay data. DRL has successfully proven to generate satisfying behaviour even when only provided with visual input. This supports our assumption that DRL could also generate reasonable behaviour in more complex 3D games. Nonetheless, we will

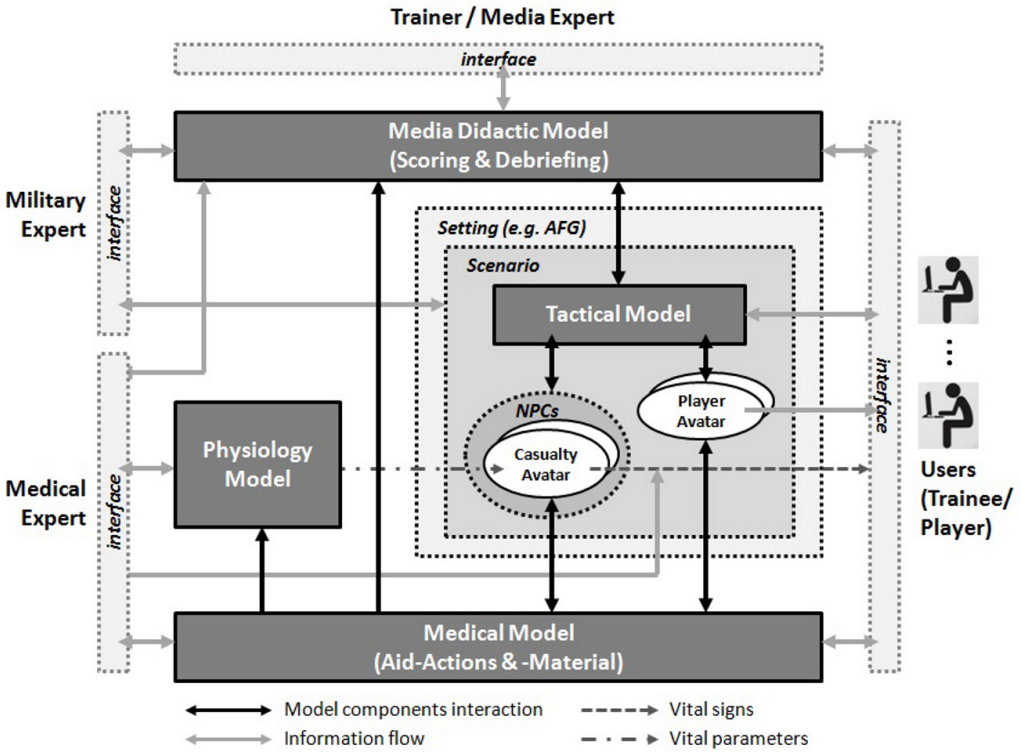

Fig. 8 Generic SanTrain architecture [31]

be confronted with the issues of long training times, slow or deficient convergence and generation of undesired behaviour. In particular, we have to consider that DRL is an optimization algorithm, which means that it will eventually lead to optimal, but not necessarily human, behaviour.

In contrast, human players are able to quickly capture a situation and rhetorical objectives like 'fun' and 'learning'. Human programmers can provide characters with knowledge exciting their own perception and with diverse, interesting actions. To improve DRL with these human capabilities won't only lead to faster convergence but also to more reasonable and varying behaviour, increasing the trust of users and trainers in a SG application.

4 Approach to an Interactive Deep Reinforcement Learning (iDRL) Framework

To overcome the mentioned challenges and support SG developers by reducing complexity and effort in AI generation, we propose to offer relevant functionalities within a new interactive DRL (iDRL) framework. We aim for a system that provides a general solution for applying DRL to SG. The focus is on user- and game-specific adaptability of the framework and its AI components, whereby no profound changes to games should be needed. Furthermore, the communication between experts and the DRL component should besimple and efficient. Our prospect is a modular, reusable and easy to use framework, which provides help in developing believable and variable AI through interactive online-learning.

4.1 Outline of iDRL Framework

The general demands we impose on our framework are modularity, efficiency and scalability (particularly on customary hardware) and parametrized control of structures and processes. The basic architecture will offer default components and functionality (outlined in Fig. 9). The flexible and generic approach will allow for the possibility to compare different learning and expert knowledge integration methods. Multiple learning instances will facilitate efficient use and quick convergence (cf. [23]). Establishing a connection between game instances and the learning component by a game interface has to be simple. DRL will be the default learning process, which is controlled by the framework. The learner's input and output will be composed of the current game-screen and possible actions, encoded as keyboard inputs. This will lead to a human-like perception and avatar control. The only necessary game changes are to allow execution control and to provide reward values. The flexible architecture will offer configuration options, exchangeable components and extensions, exemplarily shown in blue colour. Nevertheless, the developer will get support through default parameters and automated configuration options. Important configuration possibilities are specification of input quality (game screen resolution, abstraction), number of game instances, possible actions and definition of a numeric reward function. Several interactive aspects are shown in red colour. Domain experts can play multiple games, offer rewards and get visualizations of the learner's state and decisions. One of the most important parts regarding usability is

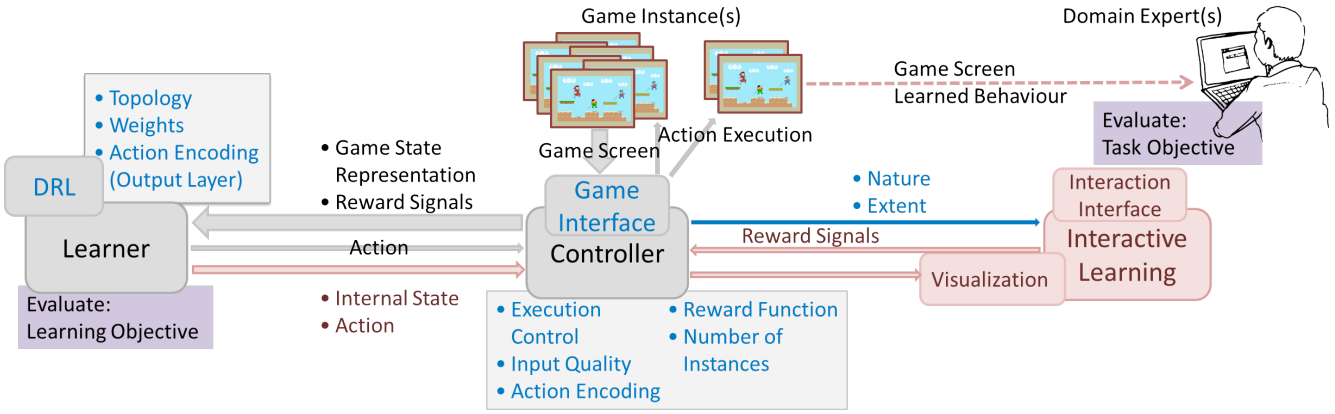

Fig. 9 Outline of essential components and interactions of the iDRL Framework

offering an adaptable interface for interaction of experts with the framework's learner, which must be easy and natural to use by non-programmers. More specific aspects of our DRL and iRL design objectives are discussed in the next subsections.

4.2 Application of Deep Reinforcement Learning

As DRL is a key aspect of our framework, we show which conditions for its application in SGs arise from the issues described in Section 3 and why the existing approach of [10] is an appropriate basis for use in our framework. We assume to have no explicit state information, no forward model, different possible actions (depending on the game) and no game log or replay data for training of ML algorithms. If only visual information of the game screen is used, the state space is composed of all possible screen representations (e.g. 160*210 coloured pixels in ALE Atari games). Thus, the described issues with RL (cf. Section 3) in large state spaces obligatorily lead to the application of model-free learning methods and approximation functions. Model-free RL methods learn a value function directly instead of a state-transition model and don't presume the functionality of predicting next states and consequences of actions. Furthermore, the use of approximation functions allows handling of large and continuous state-spaces. They scale well and are able to generalize over unknown states whereby retaining a compact representation.

At first, we will pursue a similar approach to Google's Atari player [10] including some further developments and extensions (e.g. [23]). A state is given by visual input signal from several consecutive frames. A convolutional neural network as function approximator implicitly models spatial information. The use of several hidden layers corresponds to abstraction layers of hierarchical state representation. Actions are encoded straightforward as single nodes in the output layer of the neural network. We will include and compare the Temporal Difference (TD)-learning based model-free approaches of Q-learning, Sarsa and Actor-Critic (cf. [8]). Our focus lies primarily on flexibility; we want to offer applicability and comparability of different network topologies, connection weights and different output action encodings. In the long term, we intend to include different approaches, like different forms of action encoding, allowing multiple output actions or output of parameters for easier human-understandable methods.

4.3 Essential Aspects of Interactive Reinforcement Learning

As mentioned above, DRL is an effective method, but also entails slow convergence on complex tasks. Furthermore, the goal of RL is to optimize an expected reward. Thus, a fixed reward function implies telling the agent what to achieve but not how, which doesn't necessarily lead to believable behaviour. To address these drawbacks, we present the interactive learning component as important part of our framework. By combining machine learning and human intelligence, a trainer could guide an agent in a goal-oriented way, biasing it's behaviour towards a desired outcome. The trainers to be included should be able to give feedback to specify behaviour and share their task-knowledge as domain experts, regardless of their programming skills or machine learning knowledge. This combination will decrease problem complexity on both sides - reducing the need for laborious hand-crafted behaviour and support DRL in finding solutions in complex environments. Nevertheless, the nature and extent of human trainer integration have to be determined beforehand. We have to consider two different angles: How should interactive learning be included in DRL and how does a human trainer want to interact with a learning system? Both imply investigating the communication between a human expert and a learning system.

In the area of cognitive infocommunications, this problem falls within the categories of inter-cognitive, sensor-sharing and sensor-bridging communication (cf. [33] and [34]). The DRL component, containing an ANN, is a connectionist emergent cognitive system that is able to adapt and act effectively through interaction with its environment (cf. [35]). Thus, the efficient use of iRL requires efficient infocommunication between the DRL system as cognitive thing and human experts as cognitive beings. In the following, we will summarize different possibilities of

human-RL interaction, state our selected approach and indicate remaining issues. Thereafter, we will show some implications from DRL properties and human evaluation characteristics for our interactive learning component.

Integrating human domain experts as trainers in the traditional machine learning workflow often means an iterative process, whereby nature and extent of involvement are mediated by practitioners. Experts only provide data, answer domain questions, and give feedback about the learned model. This procedure creates communication gaps between experts, programmers and agent behaviour and is seen as inefficient involvement of experts [7]. The approach of *interactive shaping*, whereby an agent receives exclusively human reward in the form of positive and negative values [36], requires intensive participation and effort on the part of domain experts. *Learning from human demonstration* or *inverse reinforcement learning* means the automated reconstruction of a reward function from a sample of policies provided by human players, which requires sufficient training data. Examples are TD-learning in backgammon by offline policy learning from experts plays [37] and learning policies for first person shooter games to generate human-like behaviour [38]. In *active learning*, an agent queries a trainer for getting labelled samples at specific learning states [39]. The role of a trainer as a pure question-answering oracle without control on the interaction process is often perceived as frustrating [7]. In *heuristically-accelerated multiagent reinforcement learning* (HAMRL) [40], hand-crafted heuristic functions are used to accelerate RL by suggesting the selection of particular actions over others. This approach implies laborious finetuning of heuristics and machine learning skills of experts.

In our framework, we will focus on an approach of combining capabilities of DRL and human experts by applying a combination of pure RL and interactive shaping; thus letting an agent learn from both environment and trainer reward. It has been shown that, in complex settings, interactive learning improves learning speed and quality compared to non-interactive learning and that interaction seems to make learning more robust [12]. An advantage of giving reward over demonstration is that an agent learns relative values of actions instead of merely when to choose a specific action [41]. Further advantages are relatively simple realization through an interface and simplicity of use [42]. In general, we have to differentiate between the *task objective* as the objective of an expert (what he gives reward for) and the *learning objective* of the agent (maximize expected reward) [42]. Several concrete combination techniques of human and environment reward, applicable in action-value functions, are described and compared in [41]. Initially, we have to take into account some more general issues that occur when applying human reward in RL (see [42]). These include reward positivity of trainers' reward values, which means the tendency to give more positive than negative reward. Especially episodic tasks seem vulnerable to positive reward circles, whereas continuing learning sessions are less sensitive. Furthermore, the temporal discounting of human reward has to be considered. A discount rate of 0 accords to a form of supervised learning, which is easier to handle but leads to laborious micromanagement. Experiments showed that non-myopic rewards proved successful in continuing tasks by pursuing higher-level goals and leading to more robust learning regarding unknown states.

We briefly summarize some demands on and desirable characteristics of our interaction component that arise from the selected methods and mentioned challenges. At first, the most promising time period of expert involvement is at the beginning of learning; early training seems more effective in terms of mean reward [41]. Furthermore, an agent should show steady behaviour by default, but nevertheless apply reward immediately and appropriately [41, 7]. Additionally, domain experts should be provided with a variety of complex feedback and control mechanisms [43], because users favour transparency and are willing to learn how a system works to give nuanced feedback [44]. Even though, the user should be supported with an appropriate level of guidance and offered summaries and explanations of system behaviour, which should preferably be lightweight but also scalable [43]. For example, a visualization of higher level features of convolutional layers in deep networks [39], as was used in [21]. In the long term, we want to offer a comprehensive repository of visualized information, resulting in system transparency and understandable NPC behaviour.

4.4 iDRL Application Scenarios in SanTrain

This section offers a short outline of how a SG like SanTrain can profit from iDRL techniques. We found several possible application scenarios we think inspiring to support SG development and offering some additional value over classical approaches. The following areas are not meant as strictly separable but rather conceptual application categories.

- **Single NPC Control** This is the most obvious possible use of iDRL and has been the main focus of our previous descriptions. In SanTrain, using iDRL for decision making of NPC enemies offers less effort for individual programming and leads to heterogeneous and adaptive NPC AI. Multiple NPCs can be controlled by iDRL within a single scenario as well.
- **Multiplayer Human Replacement** Single or multiple NPC control can be used as replacement for human players in multiplayer games. This reduces the need for additional persons in multiplayer scenarios and iDRL is assumed to lead to more varying decisions than scripted NPCs and therefore showing more realistic and entertaining behaviour.
- **Scenario Control** Multiple instances of NPCs in a scenario can be controlled in order to manage an entire

scenario development. The control objectives or learning objectives of a scenario can be defined according to different criteria, e.g. in a multiplayer scenario in SanTrain, 'the trainees' team should have one injured person on average'.

- **Game Adaptation** Scenario control and NPC control can be used for adapting the game to individual player capabilities, depending on an appropriate definition of objective function in form of rewards for DRL.
- **Game Testing** NPCs controlled by iDRL can also be used for gameplay-testing purposes during development. The created behaviour may be used to determine flaws and inconsistencies in gameplay, level-design and agent behaviour.

Fig. 10 Demonstration of enemy position possibilities for scenario control and adaptation in SanTrain

As mentioned in Section 3.1, algorithms that are important for learning matter should be validated as soon as possible, ideally already in earlier phases during development. This means that behaviour generation with iDRL should also be integrated in this process from the beginning and thereby get influence and bias from experts. The learning and calibration process before the real game is finished can be partially realized by using the concepts of fishtanks and sandboxes (cf. [45]). Fishtank means a simplified version of a game with limited gameplay complexity. Sandbox describes a game version with similar gameplay but more positive outcomes. Both concepts are often used, possibly combined, as a tutorial or first levels in computer games.

We are aware that an autonomous learning algorithm should be used with caution. Not all NPCs in a SG should be controlled by adaptive AI, as sometimes a strict and scripted scenario is more important for learning. Additionally, a large proportion of successful application is also depending on finding and defining an appropriate reward function. In all cases, the combination of DRL and interactivity enables relief of already limited trainer availability and capacity due to the offline learning capability of DRL. The integration of expert knowledge in learning is expected to lead to faster convergence, counteract unpredictable outcomes and increasing acceptance of resulting behaviour. This process can be supported by our proposed framework; by including domain experts and users more directly without the need of translating a desired behaviour into a technical layer and thus lessen the communication gap.

5 Conclusion

Combining deep reinforcement learning methods with interactive human guidance may be a promising solution to reduce effort and complexity issues in NPC behaviour generation while at the same time creating believable and diverse behaviour. Since SGs often require convincing human-like NPC behaviour, their development often includes laboriously hand-crafted AI design. We showed that although the application of machine learning techniques like DRL has already been successful in different games, there are still enormous issues with more complex scenarios, resulting from incomprehensibly large state-action spaces. However, we also mentioned that previous studies have shown that interactive RL methods can improve learning speed and quality, particularly in complex tasks. We therefore proposed a flexible and easy to use framework, providing the possibility to apply a combination of their genuine qualities during SG development. Furthermore, we showed some concrete implications for the application of DRL and interactive learning, especially considering effective collaboration between a learning system and human experts. We presented SanTrain for soldier TCCC training as practical example for a recent serious game development and introduced some of the challenges it provides for AI behaviour generation. Regarding some exemplary but easily transferable problem settings from SanTrain, we described possible application scenarios and expected benefits from applying our iDRL approach in this SG. In the long term, we hope to overcome the current issues and communication gaps between developers and experts during SG development in general, thus transforming the generation of reasonable NPC AI into an efficient, collaborative process with reduced complexity.

References

[1] Abt, C. C. "*Serious games*." University Press of America. 1987.

[2] Doujak, G. "*Serious Games und Digital Game Based Learning. Spielebasierte E-Learning Trends der Zukunft.*" GRIN Verlag, 2015.

[3] Brisson, A., Pereira, G., R. Prada, R., Paiva, A., Louchart, S., Suttie, N., Lim, T., Lopes, R., Bidarra, R., Bellottiet, F., Kravcik, M., Oliveira, M. "Artificial intelligence and personalization opportunities for serious games." In: Human Computation and Serious Games: Papers from the 2012 AIIDE Joint Workshop. AAAI TechnicalReport WS-12-17. pp. 51–57. 2012.

[4] Vik, E. "State of the Art Report on Serious games: Blurring the lines between recreation and reality." In: *INF358 Seminar on Visualization*. (Violaand, I., Hauser, K. (Eds.)). The EurographicsAssociation. 2008.

[5] Lara-Cabrera, R., Nogueira-Collazo, M., Cotta, C., Fernández-Leiva, A. J. "Game artificial intelligence: Challenges for the scientific community." In: Proceedings 2st Congreso de la Sociedad Española para las Ciencias del Videojuego. (Camacho, D., Gómez Martin, M. A., González Calero, P. A. (eds.)). Barcelona, Spain, Jun. 24, 2015. pp. 1-12

[6] Dobrovsky, A., Borghoff, U. M., Hofmann, M. "An approach to interactive deep reinforcement learning for serious games." In: Proceedings of 7th IEEE International Conference on Cognitive Infocommunications. (CogInfoCom 2016), Wroclaw, Poland, Oct. 16-18. 2016. pp. 85-90. https://doi.org/10.1109/CogInfoCom.2016.7804530

[7] Amershi, S., Cakmak, M., Knox, W. B., Kulesza, T. "Power to the people: The role of humans in interactive machine learning." *AI Magazine.* 35(4), pp. 105–120. 2014.

[8] Sutton, R. S., Barto, G. A. "Reinforcement learning: An introduction." MIT press, 1998.

[9] Arel, I., Rose, D. C., Karnowski, T. P. "Research frontiers: Deep machine learning - a new frontier in artificial intelligence research." *IEEE Computational Intelligence Magazine*. 5(5), pp. 13–18. 2010. https://doi.org/10.1109/MCI.2010.938364

[10] Mnih, V., Kavukcuoglu, K., Silver, K., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M. "Playing Atari with deep reinforcement learning." Deep LearningWorkshop NIPS. 2013.

[11] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D. "Human-level Control through Deep Reinforcement Learning." *Nature*. 518(7540), pp. 529–533. 2015. https://doi.org/10.1038/nature14236

[12] Stahlhut, C., Navarro-Guerrero, N., Weber, C., Wermter, S. "Interaction is more beneficial in complex reinforcement learning problems than in simple ones." In: Interdisziplinärer Workshop Kognitive Systeme: Mensch, Teams, Systeme und Automaten, Bielefeld, Germany, March, 2015. pp. 142-150.

[13] Genesereth, M., Love, N., Pell, B. "General game playing: Overview of the AAAI competition." *AI magazine*. 26(2), pp. 63-65. 2005.

[14] Frutos-Pascual, M., Zapirain, B. G. "Review of the use of AI techniques in serious games: Decision making and machine learning." *IEEE Transactions on Computational Intelligence and AI in Games*. PP(99), p. 1. 2016. https://doi.org/10.1109/TCIAIG.2015.2512592

[15] Perez-Liebana, D., Samothrakis, S., Togelius, J., Lucas, S. M., Schaul, T. "General video game AI: Competition challenges and opportunities." In: Thirtieth AAAI Conference on Artificial Intelligence. (AAAI-16). Phoenix, Arizona, USA, Febr. 12-17, 2016, pp. 4335-4337.

[16] Bellemare, M. G., Naddaf, Y., Veness, J., Bowling, M. "The arcade learning environment: An evaluation platform for general agents." *Journal of Artificial Intelligence Research*. 47, pp. 253–279. 2013. https://doi.org/10.1613/jair.3912

[17] Asensio, J. M. L., Donate, J. P., Cortez, P. "Evolving artificial neural networks applied to generate virtual characters." In: IEEE Conference on Computational Intelligence and Games (CIG), Dortmund, Aug. 26-29, 2014. pp. 1–5. https://doi.org/10.1109/CIG.2014.6932862

[18] Verbancsics, P., Harguess, J. "Generative neuroevolution for deep learning."

[19] Hausknecht, M., Lehman, J., Miikkulainen, R., Stone, P. "A Neuroevolution Approach to General Atari Game Playing." *IEEE Transactions on Computational Intelligence and AI in Games*. 6(4), pp. 355–366. 2014. https://doi.org/10.1109/TCIAIG.2013.2294713

[20] Min, W., Ha, E. Y., Rowe, J., Mott, B., Lester, J. "Deep learning-based goal recognition in open-ended digital games." In: AIIDE'14 Proceedings of the Tenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. Raleigh, NC, USA, Oct. 3-7, 2014, pp. 37-43.

[21] Guo, X., Singh, S., Lee, H., Lewis, R. L. Wang, X. "Deep learning for real-time atari game play using offline Monte-Carlo tree search planning." In: Advances in Neural Information Processing Systems 2014. pp. 3338–3346. 2014.

[22] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctotet, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., Hassabis, D. "Mastering the game of go with deep neural networks and tree search." *Nature*. 529(7587), pp. 484–489. 2016. https://doi.org/10.1038/nature16961

[23] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., Kavukcuoglu, K. "Asynchronous methods for deep reinforcement learning." arXivpreprintarXiv:1602.01783, 2016.

[24] Mehta, M., Ontanón, S., Ram, A. "Adaptive computer games: Easing the authorial burden." In: *AI Game Programming Wisdom 4*. (Rabin, S. (ed.)) pp. 617–632. 2008.

[25] Ontañón, S., Synnaeve, G., Uriarte, A., Richoux, F., Churchill, D., Preuss, M. "A survey of real-timestrategy game AI research and competition in star craft." *IEEE Transactions on Computational Intelligence and AI in Games*. 5(4), pp. 293–311. 2013. https://doi.org/10.1109/TCIAIG.2013.2286295

[26] Muñoz-Avila, H., Bauckhage, C., Bida, M., Congdon, C. B., Kendall, G. "Learning and game AI." In: *Dagstuhl Follow-Ups*, 6. (Lucas, S. M., Mateas, M., Preuss, M., Spronck, P., Togelius, J. (eds.)). pp. 33-43. Schloss Dagstuhl, Leibniz-Zentrum fuer Informatik. 2013.

[27] Galway, L., Charles, D., Black, M. "Machine learning in digital games: a survey." *Artificial Intelligence Review*. 29(2), pp. 123–161. 2008. https://doi.org/10.1007/s10462-009-9112-y

[28] Mahajan, S. "Reinforcement learning in complex real world domains: A review." *Indian Journal of Computer Science and Engineering (IJCSE)*. 5(2), pp. 32-40. 2014.

[29] Taito "Atari space invaders screenshot." https://web.archive.org/web/20040902162808/http://archive.gamespy.com/legacy/halloffame/spaceinvaders.shtm. [Accessed: 21st June 2016].

[30] Butler, F. K., Jr. Butler, F. K., Jr. Holcomb, J. B., Giebner, S. D., McSwain, N. E., Bagian, J. "Tactical combat casualty care 2007: evolving concepts and battle field experience." *Military Medicine*. 172(11), pp. 1-19. 2007.

[31] Lehmann, A., Hofmann, M., Palii, J., Karakasidis, A., Ruckdeschel, P. "SanTrain: A Serious Game Architecture as Platform for Multiple First Aid and Emergency Medical Trainings." In: AsiaSim 2013. Communications in Computer and Information Science. 402. pp. 361–366. Springer, Berlin, Heidelberg Berlin, Heidelberg. 2013.

[32] Feron, H., Hofmann, M. "Tactical combat casualty care: Strategic issues of a serious simulation game development." In: Proceedings of the 2012Winter SimulationConference (WSC). Dec. 9-12, 2012. pp. 1–12. https://doi.org/10.1109/WSC.2012.6465005

[33] Baranyi, P., Csapo, A. "Definition and synergies of cognitive infocommunications." *Acta Polytechnica Hungarica*. 9(1), pp. 67–83. 2012.

[34] Baranyi, P., Csapo, A., Sallai, G. "*Cognitive Infocommunications (CogInfoCom).*" Springer International, 2015. https://doi.org/10.1007/978-3-319-19608-4

[35] Vernon, D., Metta, G., Sandini, G. "A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents." *IEEE Transactions on Evolutionary Computation*. 11(2), pp. 151–180. 2007. https://doi.org/10.1109/TEVC.2006.890274

[36] Knox, W. B., Stone, P. "Combining manual feedback with subsequent MDP preward signals for reinforcement learning." In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems. (AAMAS 2010), 1. pp. 5–12. 2010.

[37] Tesauro, G. "Neurogammon: A neural-network back gammon program." In: 1990 IJCNN International Joint Conference on Neural Networks, San Diego, CA, USA, June 17-21, 1990, pp. 33-39. https://doi.org/10.1109/IJCNN.1990.137821

[38] Tastan, B., Sukthankar, G. R. "Learning policies for first person shooter games using inverse reinforcement learning." In: Proceedings of the Seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. AIIDE'11. Stanford, California, USA, Oct. 10-14, 2011. pp. 85–90.

[39] Erhan, D., Bengio, Y., Courville, A., Vincent, P. "Visualizing higher-layer features of a deep network." University of Montreal. Technical Report. Number: 1341. 2009.

[40] Bianchi, R. A. C., Martins, M. F., H. Ribeiro, C. H. C., Costa, A. H. "Heuristically-accelerated multiagent reinforcement learning." *IEEE Transactions on Cybernetics*. 44(2), pp. 252–265. 2014. https://doi.org/10.1109/TCYB.2013.2253094

[41] Knox, W. B., Stone, P. "Reinforcement learning from simultaneous human and mdp reward." In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems. 1, pp. 475-482. 2012.

[42] Knox, W. B., Stone, P. "Framing reinforcement learning from human reward: Reward positivity, temporal discounting, episodicity, and performance." *Artificial Intelligence*. 225, pp. 24–50. 2015. https://doi.org/10.1016/j.artint.2015.03.009

[43] Amershi, S. "Designing for effective end-user interaction with machine learning." PhD dissertation, University of Washington. 2012.

[44] Kulesza, T., Burnett, M., Wong, W-K., Stumpf, S. "Principles of explanatory debugging to personalize interactive machine learning." In: Proceedings of the 20th International Conference on Intelligent User Interfaces. Atlanta, Georgia, USA, March 29 – Apr. 01. 2015. pp. 126–137.

[45] Gee, J. P. "Learning by design: Good video games as learning machines."*E-Learning and Digital Media*. 2(1), pp. 5-16. 2005.