

Intrinsically Motivated General Companion NPCs via Coupled Empowerment Maximisation

Christian Guckelsberger
Computational Creativity Group
Goldsmiths, University of London
London, United Kingdom
Email: c.guckelsberger@gold.ac.uk

Christoph Salge
Adaptive Systems Research Group
University of Hertfordshire
Hatfield, United Kingdom
Email: c.salge@herts.ac.uk

Simon Colton
The MetaMakers Institute
Falmouth University
Falmouth, United Kingdom
Email: s.colton@gold.ac.uk

Abstract—Non-player characters (NPCs) in games are traditionally hard-coded or dependent on pre-specified goals, and consequently struggle to behave sensibly in ever-changing and possibly unpredictable game worlds. To make them fit for new developments in procedural content generation, we introduce the principle of *Coupled Empowerment Maximisation* as an intrinsic motivation for game NPCs. We focus on the development of a general game companion, designed to support the player in achieving their goals. We evaluate our approach against three intuitive and abstract companion duties. We develop dedicated scenarios for each duty in a dungeon-crawler game testbed, and provide qualitative evidence that the emergent NPC behaviour fulfils these duties. We argue that this generic approach can speed up NPC AI development, improve automatic game evolution and introduce NPCs to full game-generation systems.

I. INTRODUCTION

Dogmeat from *Fallout* or *Ellie* from *The Last of Us* or the pet from *Nethack* – memorable companions are an important part of our gaming experience. But companions can also be a great source of annoyance, especially when their behaviour fails miserably [1]. The vast majority of companions are hard-coded by means of e.g. finite state machines or behaviour trees, and consequently struggle to produce believable or even plausible behaviour in unforeseen contexts ([2], [3]). More advanced companions can adapt their behaviour by means of planning, or by learning a policy via neural networks or traditional reinforcement learning. Nevertheless, they require intense training or pre-specified rewards, which again renders them inflexible especially in sandbox games where players with a large choice of options can change a dynamic world. In the future, the demands on non-player character (NPC) AI in general are likely to increase further [4]. This is particularly emphasised by progress in procedural content generation, which not only focusses on game elements such as levels and game mechanics ([4], [5]), but ultimately aims at generating entire games [6]. How can NPC AI deal with these ever-changing and potentially unpredictable game worlds?

One answer [2] is to drive the NPCs’ behaviour by means of intrinsic motivation, such as artificial curiosity [7] or learning progress [8]. Instead of relying on pre-defined goals or behaviours which might become meaningless when the game changes, intrinsically motivated agents perform “an activity for its inherent satisfactions rather than for some separable consequence” [9]. Models of intrinsic motivation ground behaviour

in the agent’s sensorimotor relationship with the world [8], so changes to the world or the agent’s embodiment are reflected in potentially new behaviour. A curious mouse and a curious bird would consequently behave differently, moderated by their embodiment and environment. In this paper, we will work with the intrinsic motivation of *empowerment* [10], a measure of how much an agent is in control of the world it can perceive. We have previously argued [11] that empowerment reflects an agent’s drive to maintain its own precarious existence, and allows them to adapt to changes in their embodiment and environment. But while empowerment might be very useful to produce an intrinsically motivated *general NPC*, we have to look specifically into how to turn it into a good *companion*.

Players seem to expect a companion to behave differently than a general NPC. For instance, in a qualitative study on companion behaviour, a player said “I dislike that [the companion] prioritises getting to the exit herself over helping [me] first” [1], stressing the delicate balance between support and independence. McGee and Abraham [12] argue that the NPC must account for the player’s goals as part of coordinated decision-making, possibly incorporating uncertainty. To guide our approach, we identify the following three companion duties, which should generalise across a range of game genres:

- 1) *Player Integrity*: Ensure that the player can continue playing the game. Act against any limiting force.
- 2) *Support*: Support and do not hinder the player in achieving their goals. Maintain *operational proximity*, i.e. act towards states where you can support the player.
- 3) *Companion Integrity*: Secure your own existence and ability to act in order to support the player long term.

We did not define any explicit, goal-specific companion duties which could constrain the NPC’s adaptivity. Instead, their goal directedness will arise from the interaction with the player.

We design an intrinsically motivated, general companion NPC based on the *Coupled Empowerment Maximisation* (CEM) principle. CEM establishes a general frame for support by relating an agent’s action selection policy not only to their own, but also to the empowerment of other agents. We provide an intuition and formalisation of the principle. We evaluate our approach qualitatively in a dungeon-crawler testbed, by means of observing whether the NPC fulfils its companion duties. We finish with a discussion, conclusion and future work.

II. BACKGROUND

A wide body of research exists on companion AI and related notions. While the notion of “companions”, “sidekicks” and “assistants” usually refer to a unidirectional, supportive relationship towards the player (cf. [1]), research on “partners” and “team-mates” puts more emphasis on bidirectional collaboration and shared goals (cf. [12]). We are interested in support; Nevertheless, only few projects from the first category are relevant to us, as the majority specialises in specific game genres. We in contrast propose a notion for general companion-like behaviour in an arbitrary game.

We equip NPCs with the skill to coordinate their actions with the player in a supportive way. Most related work addresses this challenge by modelling the player’s goals explicitly. Fern and Tadepalli [13] represent the player as a noisy utility maximisation agent which is more likely to select actions that have a high utility of completing a given task. The player’s intentions are modelled by means of Markov-Decision-Processes (MDPs) which can capture uncertainty in human action-selection and in the environment. Nguyen et al. [14] extend this approach in their *Collaborative Action Planner with Intention Recognition (CAPIR)*, combining pre-computed MDP action policies and online Bayesian belief updates. They improve the performance of Fern’s and Tadepalli’s framework by decomposing tasks into subtasks. Their approach is evaluated in a maze game where a companion has to help the player kill ghosts. Macindoe, Kaelbling and Lozano-Pérez [15] build on *CAPIR* by steering NPC decision-making by the information it can gain about the player’s intentions. They evaluate their framework in a cooperative pursuit game. Our formalism is also concerned with planning, i.e. the simulation of experience, and is capable of accounting for uncertainty in the player’s behaviour and the environment. We do not model the player’s goals explicitly, but introduce them implicitly into the policy. While the utility function in these projects must be pre-defined, our approach employs intrinsic motivation to overcome this limitation.

If we constrain our scope to intrinsically motivated agents, the body of research becomes much smaller. The work by Merrick and Maher ([2], [16]) is most closely related, looking at how intrinsically motivated reinforcement learning [17] can support NPCs in learning complex tasks in a dynamic game world. They propose two models of motivation as reward signals for Q-learning: an agent’s interest in a new situation, given past experiences, and its competence based on the error in learning policy updates. Their qualitative studies in *Second Life* and a quantitative analysis of behavioural variety and complexity in dedicated RPG testbeds confirm that intrinsic motivation allows agents to adapt their behaviour in an unexpectedly changing environment. In contrast to our study, their NPCs act in solitude, and not in favour of other agents such as the player. In a related study, Forgette and Katchabaw [3] focus on the development of more believable NPC behaviour, by letting them choose actions in order to maintain basic desires such as hunger or social contact. Similar to Merrick

and Maher, their focus is on learning complex sequences of actions to satisfy their motives, while we are interested in establishing supportive, immediate reactions to the player’s actions. Forgette and Katchabaw’s desires can only loosely be considered intrinsic, as they rely on sensor semantics (cf. [8]). Unfortunately, this dependency makes the approach unsuitable in PCG or dynamically changing game worlds.

Empowerment as intrinsic motivation has so far only been employed for general game-playing, but not to steer the behaviour of companion or enemy NPCs. Anthony, Polani and Nehaniv [18] analysed empowerment maximisation to drive player behaviour in Sokoban and Pac-Man, and in the same course proposed several optimisation methods. Mohamed and Rezende [19] focus primarily on optimisation, with a likely application in general game playing.

III. FORMAL MODEL

We propose the principle of coupled empowerment maximisation (CEM) as an abstract notion for companion-like behaviour in highly flexible and adaptive NPCs. We recently introduced CEM in a co-creativity context [20], and will expand it here to fit the requirements of companion NPCs. We first provide an intuition and definition of the underlying empowerment formalism and its variations. Equipped with this, we show how the different notions come together in CEM to drive the decision-making of NPCs.

A. Empowerment

Empowerment [10] is defined between an agent’s actuators and sensors. In a deterministic environment, it quantifies the options available to an agent in terms of availability and visibility. In a stochastic setting, this generalises to its potential, perceived influence on the environment. Empowerment is measured in bits; It is zero when the agent has no control over its sensors, i.e. when all actions lead to the same or a random outcome, and it increases when different actions lead to separate perceivable outcomes. For simplicity, we focus on interactions which are discrete in time and space, but continuous implementations exist. An extensive survey of motivations, intuitions and past research can be found in [21].

At the centre of the empowerment definition is the interpretation of an agent’s embodiment as an information-theoretic communication channel between its actuator A and sensor S , where in \mathcal{A} and \mathcal{S} represent the possible actions and sensor states. The agent’s interaction with the world is usually described as a perception-action loop [22] as in Fig. 1. Modelled by means of a causal Bayesian network, the figure illustrates the turn-wise interaction of the player and a companion NPC, unrolled in time. Each agent is represented by their sensor and actuator, and the black, solid arrows imply causation between these random variables. The companion chooses an action based on its sensor input at time t , influencing the rest of the environment R and the player’s sensor in the next time step. The player’s sensor informs the player’s actions, which influence the environment and the companion’s sensor at $t + 2$. The latter in turn affects the companion’s actions,

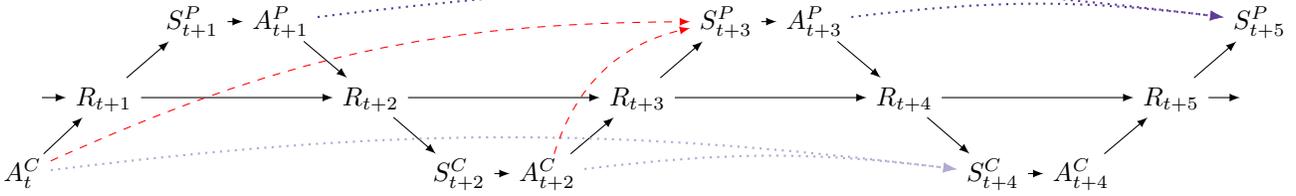


Fig. 1: Perception-action loop for player (S^P, A^P) and companion (S^C, A^C), illustrating three different types of 2-step empowerment. The agents interact in turnwise order, and are represented by their sensors and actuators. Purple, dotted lines (top) represent player empowerment. Violet, dotted lines (bottom) stand for companion empowerment, and red dashed lines for companion-player transfer empowerment.

and the cycle repeats itself. The environment is affected both by its preceding state and the character’s actions. The causal probability distribution $p(S_{t+2}^C | S_t^C, A_t^C)$ then represents the (potentially noisy) communication channel between the companion’s actions *before*-, and its future sensor states *after* the player has performed an action.

Empowerment \mathfrak{E}_{s_t} is then defined as the maximum potential information flow that could possibly be induced by a suitable choice of actions, in a particular state s_t . This can be formalised as the channel capacity:

$$\begin{aligned} \mathfrak{E}_{s_t} &= \max_{p(a_t)} I(S_{t+2}; A_t) \\ &= \max_{p(a_t)} \sum_{A, S} p(s_{t+2} | s_t, a_t) p(a_t) \log \frac{p(s_{t+2} | s_t, a_t)}{\sum_A p(s_{t+2} | s_t, a_t) p(a_t)} \end{aligned}$$

Here, $I(S_{t+2}; A_t)$ represents the mutual information between sensors and actuators. Empowerment as defined here is the maximum amount of information that the active agent can inject into the environment with its actions at t , and perceive again at $t + 2$, after the other agent performed. *N-step empowerment* is a generalisation of this principle where not a set of single actions A , but a set of action sequences $A_t^s = (A_t; A_{t+2}; \dots; A_{t+2(n-1)})$ and their impact on a future sensor state S_{t+2n} are evaluated. The parameter n specifies the agent’s lookahead. For a detailed introduction to the general information-theoretic notions, consult [23].

Fig. 1 illustrates the three variants of 2-step empowerment used in this paper: *Companion-* (\mathfrak{E}^C , \dots , bottom) and *player empowerment* (\mathfrak{E}^P , \dots , top) correspond to the NPC’s and player’s perceived influence on their own future sensor state. *Companion-player transfer empowerment* (\mathfrak{E}^T , $-\cdot-$) maps the companion’s actions to the player’s future sensor, quantifying the companion’s influence on the player’s perception.

Empowerment is *local*, i.e. the agent’s knowledge of the dynamics $p(S_{t+2} | S_t, A_t)$ is sufficient to calculate it. The information-theoretic grounding makes it domain-independent and universal, i.e. it can be applied to every agent-world interaction that can be modelled as a perception-action loop. It can thus be computed for arbitrary setups of what a NPC can see and do, and can cope with changes to a game’s mechanics, affecting how the NPC can interact with the world and player.

B. Coupled Empowerment Maximisation

Empowerment does not measure an agent’s actual, but rather their potential influence on the environment. The em-

powerment maximisation hypothesis [21] suggests that an agent should, in the absence of any explicit goals, choose actions which likely lead to states with a higher influence on the environment, i.e. potentially more options. *Coupled empowerment maximisation* (CEM) is an extension of this principle to the multi-agent case, and we use it to formalise companion-like behaviour in a very general and flexible way.

Fig. 2 illustrates the turn-wise interaction of player P , companion C and enemy E . Each interaction cycle is initiated by the player performing an action, which the companion will react to, followed by enemies. Each agent can affect the others either explicitly, or implicitly through their impact on the shared game world, which can be quantified by empowerment. Given the previous intuition, we suggest that increasing the empowerment of a goal-directed agent can be considered as supporting them in performing and achieving their tasks. We consequently hypothesise that equipping an NPC with an action selection policy which not only maximises their own- but also the player’s empowerment leads to the emergence of companion-like behaviour. We specifically suggest the policy

$$\pi(s_t) = \arg \max_{a_t} \left(\alpha_C \cdot E[\mathfrak{E}_{s_{t+4}}^C]_{a_t} + \alpha_T \cdot E[\mathfrak{E}_{s_{t+4}}^T]_{a_t} + \alpha_P \cdot E[\mathfrak{E}_{s_{t+3}}^P]_{a_t} \right)$$

with parameters $\alpha_C, \alpha_P, \alpha_T$ representing the influence of each empowerment type in the overall coupling. The CEM principle establishes a dependency between the NPC’s decision-making and the player. Given a certain state, the policy returns the action which maximises the NPC’s expected coupled empowerment, i.e. the combination of its own expected empowerment $E[\mathfrak{E}^C]$, the expected empowerment of the player $E[\mathfrak{E}^P]$, and the NPC’s influence on the player’s sensor state, i.e. expected companion-player transfer empowerment $E[\mathfrak{E}^T]$. The latter allows the companion to maintain *operational proximity*, i.e. to put the NPC in a position where it could potentially affect and thereby maximise the player’s empowerment. We look at *expected* empowerment, because the companion must consider all possible ways the player could behave.

The calculation of coupled empowerment therefore involves several estimation steps, which are illustrated in Fig. 2. To select an action in $t + 1$, the NPC has to calculate the expected coupled empowerment for each of its actions a_{t+1} separately. As a first step, we thus have to estimate which potential player states at $t + 3$ each of the companion’s actions and the enemies’ reactions might lead to. From there on, we have to anticipate how the player might react, resulting in potential companion states at $t + 4$. This estimation stage is denoted

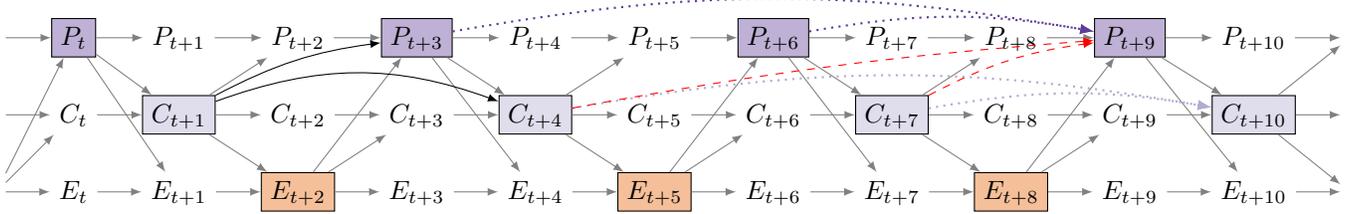


Fig. 2: Causal Bayesian network illustrating the turn-wise interaction of player P , companion C and enemies E . The solid grey lines denote mutual influence. The solid black lines denote the companion’s future state estimation, preceding the calculation of 3 types of 2-step empowerment: player- (purple dotted, top), companion- (violet dotted, middle) and companion-player transfer empowerment (red dashed).

by black, solid arrows in Fig. 2. We then calculate player empowerment (····, top) in $t + 3$, as well as companion- (····, middle) and companion-player transfer empowerment (---) in $t + 4$. This involves another n rounds of estimations, depending on the lookahead n . Expected coupled empowerment is finally calculated by weighting the individual empowerment values with the probability of the states the NPC’s actions trigger.

Note that empowerment maximisation *per se* is not goal-oriented. Nevertheless, coupled empowerment introduces the player’s empowerment into the companion’s policy, and thus guides the companion by means of the player’s goals.

IV. EVALUATION: GENERAL COMPANION-LIKE BEHAVIOUR

We claim that maximising coupled empowerment realises companion-like behaviour. We evaluate this claim qualitatively, by designing several dedicated scenarios to probe each of the companion duties outlined earlier. We describe the emergent behaviour, and highlight the contributions of the individual empowerment types coupled in the agent’s policy, and how they blend together. Our goal is to create highly flexible and adaptive companions. Since explicit constraints decrease this flexibility, we start with the plain formalism, highlight drawbacks in the various scenarios, and propose modifications which maintain this flexibility as needed. Importantly, we do not tailor the formalism to each scenario; instead, we extend it successively to eventually generalise across all of them.

A. Testbed

The following experiments were set in a minimal *dungeon-crawler* game in which the player, supported by a companion NPC, has to navigate through rooms connected by corridors and defeat enemies in order to reach a goal state. Agents interact in turn-wise order, starting with the player, and followed by the companion and enemies. All agents have health points, and can either move one step in each direction, shoot, or idle. They can only hit other agents within a certain range in their view direction, which changes with movement. While the player’s actions cannot be exactly predicted by the companion, the enemies act, for the sake of simplicity, deterministically: They always shoot at, or chase, the closest non-enemy.

We chose this game type for various reasons: It traditionally relies on procedural content generation and elements of chance, and therefore poses interesting challenges to a general NPC policy. Classic examples such as *Nethack*, but also more

recent variants such as *Hashtag Dungeon* illustrate how our minimal testbed can be extended to introduce new challenges to the formalism. Dungeon crawlers are traditionally discrete in time and space and thus simplify the computation and analysis of agent behaviour. The core mechanics are grounded in the behaviour of living beings, and thus connect with the biological origins of empowerment [11].

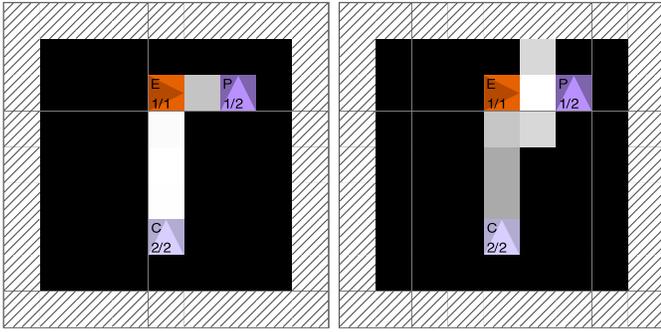
The player and companion sensors are local, non-overlapping and asymmetric. We model locality by only accounting for entities such as other characters in a maximum distance of two units around the agent. They are represented by an id and their relative position. Sensors do not overlap, i.e. they only comprise the agent’s own absolute position, rotation, and health. They are asymmetric, in that the player sensor also comprises the game status (running, lost, won).

The non-overlapping sensor would make our NPC strictly egocentric, if it only maximised its own empowerment. In our simulations though, we weight the player’s empowerment the most by $\alpha_P = 0.5$, the companion’s own empowerment by $\alpha_C = 0.2$, and the companion-to-player transfer empowerment by $\alpha_T = 0.3$, in order to reflect the companion duty hierarchy. These values were determined by experimentation, and work across all scenarios. They can be varied to some extent, and we will provide details on their limits in each scenario. By default, we assume a lookahead of $n = 2$.

B. Duty 1: Ensure Player Integrity

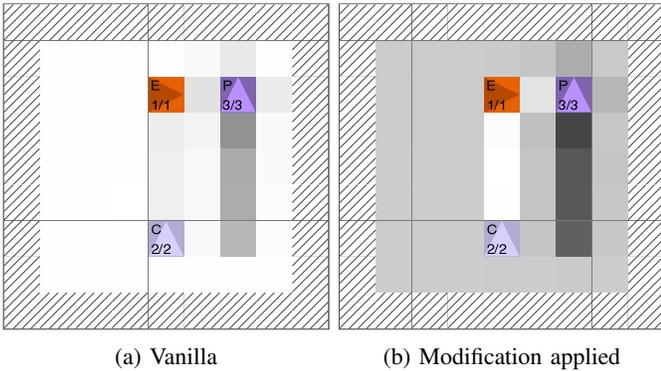
A companion must protect the player, and prevent its death. Fig. 3 show a scenario in which the player is directly threatened: An enemy faces the player, ready to shoot. The companion in turn faces the enemy, and could therefore rescue the player. The figures illustrate the empowerment values relevant to the policy, by means of mapping them as greyscale values to different positions in the scene. Brighter hues indicate higher empowerment. The player, companion and enemies are represented by purple, violet and orange squares with letters “P”, “C” and “E”, respectively. The numbers on the bottom specify their current and maximum health.

In Fig. 3a, the player position was fixed and the companion moved around. The value at a particular location corresponds to the player’s empowerment, if the companion was in that position and chose to shoot. According to the core formalism (cf. sec. III-A), empowerment would drop to zero if the player was killed. Consequently, it is highest if the companion either faces the enemy in close range, expressing the potential to



(a) Player empowerment, given that the companion chose to shoot in a certain position, $n = 2$. (b) Companion-player transfer empowerment for non-local sensor, $n = 2$.

Fig. 3: 1st scenario. The player is threatened to be killed by an enemy. The companion faces the enemy, and could rescue the player.



(a) Vanilla

(b) Modification applied

Fig. 4: 1st scenario. Player empowerment, given the companion shoots in a certain position, and player health $> n, n=2$. Health-performance consistency (right) indicates clearly where to shoot.

shoot, or steps between the two to take the bullet. It is less preferable for the player to be faced by the companion, because the latter could turn its weapon against the ally. Fig. 3b illustrates the companion’s role as bodyguard by means of companion-to-player transfer empowerment, i.e. its influence on the player’s future sensor state. It shows that the companion could still save the player by stepping in from the side.

CEM makes the companion kill the enemy for any value of α_C and α_T , as long as $\alpha_P > 0$. If $\alpha_P = 0$, the companion does not have access to the player’s sensor state. As their sensors do not overlap, the companion would not “care” about the player. Importantly, the companion would defend the player even if the enemies did not pose a threat to itself.

Unfortunately, the companion would only protect the player as long as its lookahead n is larger than the player’s health. The reason can be found in an actual inconsistency present in many (video) games: In nature, a living being’s health not only indicates its closeness to death, but also corresponds to a decline in the ability to interact successfully with the world [11]. In games, health or similar labels for fitness often only represent a mere warning, and affect the agent’s performance irregularly or only when dropping to zero. A companion can thus only foresee the tragic consequences of the enemy’s actions if it evaluates the environment dynamics far enough

ahead. This can be expensive to compute, but could also result in overcautious overall behaviour. We therefore suggest to make the relationship between a character’s health and actual performance more consistent, by introducing noise into the agent’s state transition probabilities:

$$p(S_{t+3}|s_t, a_t)^* = \begin{bmatrix} p(s_{t+3}^1|s_t, a_t) \\ p(s_{t+3}^2|s_t, a_t) \\ \vdots \\ p(s_{t+3}^D|s_t, a_t) \end{bmatrix} \odot \begin{bmatrix} \gamma \\ \gamma \\ \vdots \\ 1 - \gamma \end{bmatrix}, \gamma = \frac{h_t}{h_{max}}$$

Here, h_t and h_{max} stands for the agent’s current and maximum health, as representative for some arbitrary fitness label. The state s_{t+3}^D resembles the agent’s default follow-up state, e.g. the state resulting from idling. The more an agent’s health decreases, the more likely it becomes that its actions will lead to the default state. Applied to all available actions, the modification will lead to a consistent, gradual decrease of an agent’s empowerment with their health.

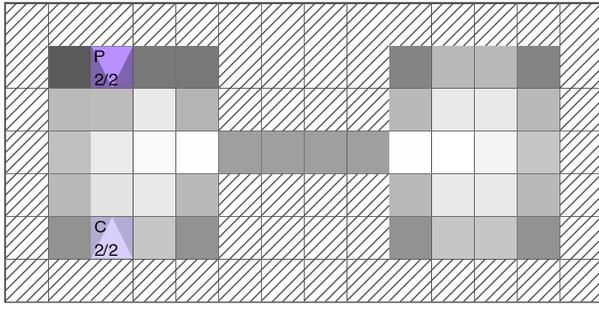
Fig. 4 illustrates the effect of this modification in the previous scenario. Here, *health-performance consistency* allows the companion to clearly differentiate between actions that contribute to the player’s empowerment, despite a short lookahead $n < h_t$. The companion not only acts when the player faces death, but also protects the latter from being harmed. We will assume this modification by default in the following scenarios. The behaviour can be watched online¹.

C. Duty 2: Support the Player

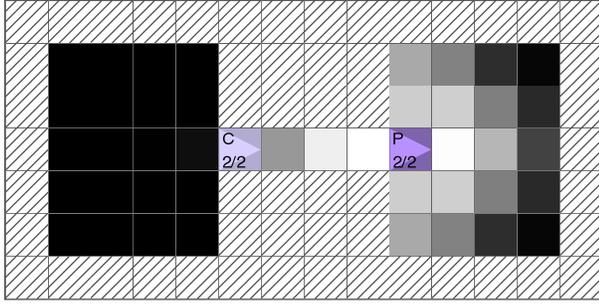
In order to support the player in achieving their goals, the companion should strive for states in which it can affect the player and its perceivable environment best. Depending on the NPC’s action set, this *operational proximity* can be different from *spatial proximity*: We can imagine an NPC which can operate terminals, but has no capacity for melee attacks. Such a companion might support the player most by staying remote, where it could e.g. unlock doors or trap the player’s enemies. In our second scenario in Fig. 5, spatial proximity is key and we expect the companion to stay close to the player and follow them from one room to the other.

However, this is not self-evident: The companion’s empowerment (Fig. 5a) is particularly low at the room edges and corners, but also in the corridor. Here, the NPC’s sequences of navigational actions collapse into very few follow-up states, as the agent can neither move north nor south. If the NPC’s policy was only about maximising its own empowerment, it would move to the centre of the current room, and avoid the corridors. Nevertheless, adding companion-player transfer empowerment to the equation renders all states but the ones in which the player can be directly influenced less attractive (Fig. 5b). When coupled with the other empowerment types, it compensates for the barrier induced by the companion’s empowerment (Fig. 5c). For the default setup and $\alpha_T \geq 0.3$, the companion consequently follows the player through the corridor and maintains spatial proximity. Note that this scenario

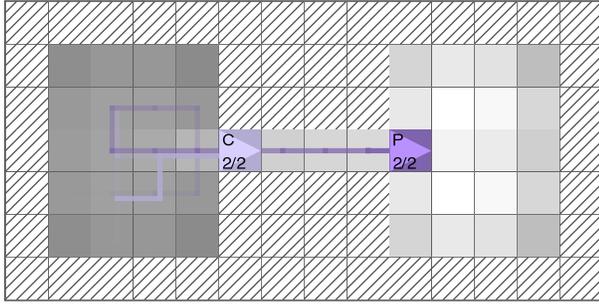
¹Video on duty 1 (Ensure Player Integrity): http://y2u.be/uh3J_ENh11M



(a) Companion empowerment, $n=2$.



(b) Companion-player transfer empowerment, $n=2$.



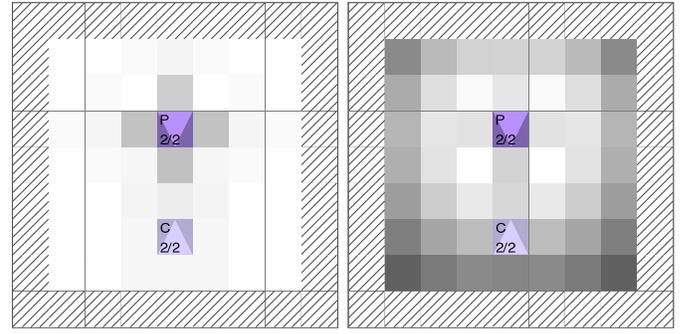
(c) Coupled empowerment with movement trace, $n=2$.

Fig. 5: 2nd scenario. The transfer empowerment in the coupling allows the companion to maintain operational proximity, and thus to follow the player through bottlenecks such as a narrow corridor.

only represents one example of an empowerment bottleneck, and that our formalism should generalise to other situations.

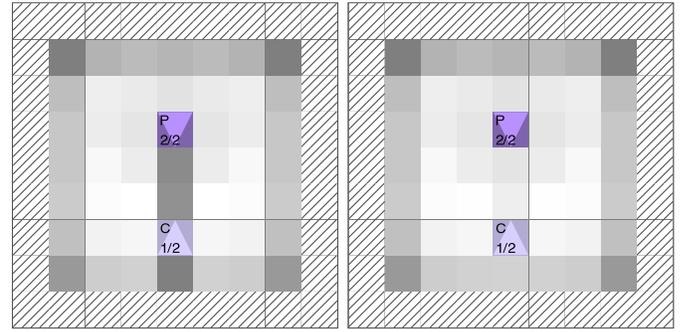
The notion of support implies not hindering the player from reaching their goals. In our third scenario, we check that the companion does not block the player's movement at any time, while maintaining spatial proximity. Fig. 6a shows the player's empowerment for different companion positions. The values are low around the player, because the companion would constrain its movement. The same applies to the companion's periphery in respect to the player. Left alone, they would add up and lead to generally repellent behaviour; Nevertheless, the transfer empowerment in the coupling (Fig. 6b) makes the companion maintain spatial proximity, while not blocking the player whenever possible. It will consequently prefer the corners around the player and either go ahead or follow.

Fig. 7a, showing companion empowerment for different companion positions, highlights a shortcoming in the initial formalism: Here, low values between companion and player illustrate that the player can occasionally be perceived as



(a) Player empowerment, given (b) Companion-player transfer companion positions, $n=2$.

Fig. 6: 3rd scenario. The companion is reflected as an impediment to movement in the player's empowerment. Once coupled, it makes the companion seek proximity off the player's axes.



(a) Health-Performance Consistency (b) Health-Performance Consistency and Minimal Synchronicity

Fig. 7: 3rd scenario. Without applying the trust function, the player will be reflected as thread in the companion's empowerment.

a threat. This is the case because the companion considers scenarios where the player's actions would harm it as feasible as any other, and weights them equally in the calculation of expected empowerment. As a consequence, the companion might e.g. "flee" from the player, preferring not to stay within their shooting range. We suggest that this behaviour is unnatural for the interaction of a supporting agent and one which benefits from this support and its continuity. In general terms, it appears necessary for companion-like behaviour to emerge that player and companion realise *trust*, in terms of not considering actions from their ally which threaten their existence significantly. We consequently applied a correction function to the action transition probabilities *after* the calculation of empowerment, in order to remove actions counteracting such trust from the expected empowerment calculation. Actions are included in the trusted set $A_P^* \subseteq A_P$ as follows:

$$a \in A_P^* \Leftrightarrow \neg (p(s_{t+3}|s_{t+2}, a) > \tau \wedge E^C(s_{t+3}) = 0 \wedge E^C(s_{t+2}) > 0)$$

The function removes player actions which reduce the companion's empowerment to zero with probability $\tau > 0$. The companion therefore still considers player actions which are unlikely to be fatal, but might benefit the player significantly. Not only the player, but also the enemies could decrease the companion's empowerment to zero. In order to not confuse

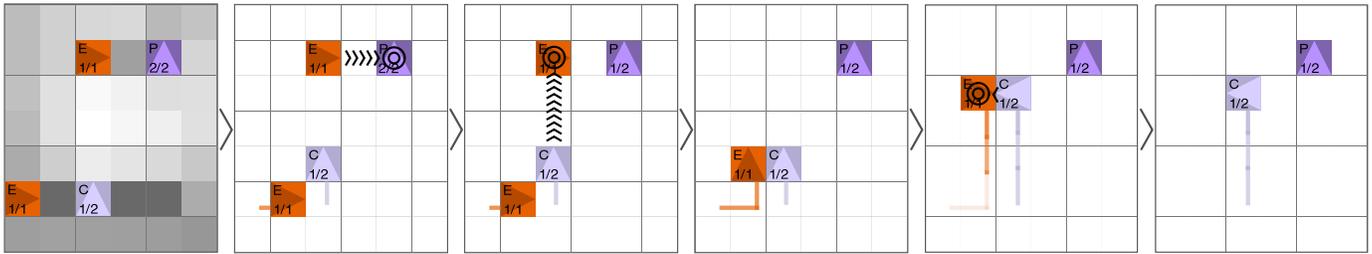


Fig. 8: 4th scenario: Companion and player are threatened simultaneously. Successive moves from left to right: the companion escapes its own death, rescues the player, and finally defends itself. Left: coupled empowerment, for $n=2$. Arrows indicate shooting.

both effects, we calculate and compare companion empowerment before and after the player performed. Applied to the previous scenario (Fig. 7b) the expected empowerment of remaining in the player’s shooting range would increase, making this action more likely to be performed. It also only applies when agents are close to death, i.e. empowerment can actually be decreased to zero in a single step. We will assume the modification to be present in the following scenarios. Our observations were documented in a video available online².

D. Duty 3: Ensure Own Integrity

We demonstrated earlier that the companion will protect the player from threats. Various earlier studies ([18], [24]) showed that empowerment maximisation makes agents death-averse, allowing the NPC to defend itself against threats. In our fifth scenario, we look at a more complex dilemma addressing both the duties to protect the player and itself: What happens if companion and player are threatened at the same time? More specifically, what happens if the player might only be harmed, compared to the situation where it might be killed if the companion did not react immediately?

Fig. 8 illustrates the outcome of the first case by means of a series of movements. The first image in the series shows the coupled empowerment in the initial situation. Here, the dark area between the companion and the enemy on the left renders the latter as a threat, while the white area towards the other enemy highlights the companion’s potential to save the player from harm. The following images illustrate that the companion will first escape from the enemy on the left, while accepting that the player will be harmed. It will then kill the player’s enemy before the latter kills the player. The remaining enemy and companion follow each other, until the companion eventually kills the enemy to save its own existence.

In the second case, the initial situation is the same, but the player’s health is set to one. For the default parameter configuration and $\alpha_P > 0.5$, the companion will sacrifice itself to rescue the player from death, thus fulfilling the hierarchy of duties outlined earlier. A video with both scenarios is online³.

V. DISCUSSION

Our experiments provide evidence that CEM enables the emergence of companion-like behaviour. We proposed to extend the formalism with a consistent relationship between

an agent’s fitness indicator, e.g. health, and its performance. This modification represents a heuristic when the NPC does not have a sufficiently large lookahead n . It thus reduces computational complexity, but is not necessary for supportive behaviour to emerge. Our *trust* function represents a minimal case of coordination in anticipation, which complements the coordination in the agent’s policy established by the CEM principle. In contrast to health-performance consistency, we suggest that coordination in anticipation is strictly necessary for supportive behaviour, and that generalising *trust* to both negative and positive effects on player and companion would result in more fine-tuned, supportive behaviour. We therefore propose to weight the likelihood of actions gradually according to $p(a) \propto \Delta \mathcal{E}_s$ in future implementations. Coordination must be extended into the calculation of empowerment itself, in order to create more reliable biases for the policy. Learning the actual distribution of actions as part of player modelling would also contribute significantly to coordination, and speed up calculation by pruning the search tree.

We fixed the parameters of $\alpha_P, \alpha_C, \alpha_T$ to allow for sensible behaviour across all scenarios. This exact configuration might not work in an arbitrary game; We therefore suggest two alternative solutions: Looking closely at the companion duties, we claim that ensuring its own and the player’s integrity emerges from general player support. We could translate this hierarchy into the policy, and consequently choose actions primarily to maximise player empowerment, followed by transfer and companion empowerment. Alternatively or in addition, we propose to evolve the NPC’s parameters by means of automated play-throughs with a general game-playing agent.

Our goal was to investigate the richness of behaviour induced by the CEM principle, and we consequently abstained from any unnecessary constraints. Employing this principle in an actual game might nevertheless require explicit constraints to meet two industry requirements: predictability and performance. Empowerment as an intrinsic motivation allows for maximum adaptivity and flexibility in NPCs, but as a consequence might trigger surprising behaviour. If predictability is a strict requirement, we can fix the behaviour emerging from CEM before deployment, or use the formalism as a mere intuition pump to assist designers. Alternatively, we can define illegal behaviours as constraints on top of the policy, if the designer favours surprisingness over adaptivity. Such explicit constraints can also help in decreasing computational complexity, by pruning the search tree. Empowerment can be

²Video on duty 2 (Support the Player): <http://y2u.be/6g9qoa5BdwU>

³Video on duty 3 (Ensure Own Integrity): <http://y2u.be/z3gZ0iGE0wg>

approximated and serve as tie-breaker to increase behavioural variety. Optimisation methods allow for larger lookaheads, and thus more behavioural complexity. Existing optimisations use Monte-Carlo sampling [21], the *information-bottleneck* method [18] and deep neural networks [19].

VI. CONCLUSION & FUTURE WORK

We formalised and evaluated the principle of coupled empowerment maximisation (CEM) to design an intrinsically motivated NPC capable of companion-like behaviour. We started with the raw formalism, and successively added modifications which generalise across the scenarios. Our experiments show that CEM establishes a sufficiently general frame for companion-like behaviour by inducing the player's goal into the companion's policy, making it unnecessary to specify the NPC behaviour explicitly at design time. Given our experimental evidence and the universality of the formalism, we hypothesise that the principle generalises to a wide range of game scenarios and genres.

If this proves correct, the flexibility and adaptivity of CEM could make NPCs fit for the most recent challenges in the games industry and academic research. It could allow industry to save efforts and reduce costs of manually authoring NPC behaviour, especially in games with a strong focus on procedurally generated content. Even if a game relies strongly on scripting, our formalism can help in establishing a default mode of interaction with the player and other agents. In automatic game evolution and rapid prototyping, NPCs driven by CEM will allow us to stretch the parameter space and search regions where pre-defined NPCs would break. It could enhance research in computational game creativity, by increasing novelty and surprisingness in NPC behaviour, and serve as cornerstone in the automatic generation of complete games, which presently does not incorporate NPCs. Given these developments, we hope to see NPCs soon in general game playing competitions.

We plan to expand this research in three directions. First, we want to investigate how well the principle generalises by changing the mechanics and increasing the complexity of our testbed. We particularly plan to look at shared resources such as health packs or ammunition, additional actions which might complement with the player's skills, and more sophisticated mechanics such as traps or door openers. A second potential branch of future research concerns human player experience. We are interested in evaluating qualitatively how enjoyable it is to play with our companions, especially in respect to the player's perception of agency. We suggest that empowerment relates closely to agency [11], and that varying the policy parameters might affect the player's locus of control, and the perceived "character" of the NPCs. Finally, we want to investigate how well this principle can be reversed to generate enemy NPCs, demonstrating non-obvious ways to be antagonistic. All branches will require more work in reducing computational complexity and improving coordination.

ACKNOWLEDGMENTS

The authors thank Jeremy Gow for helpful comments on an earlier draft. CG is funded by EPSRC grant [EP/L015846/1] (IGGI). CS received funding from the H2020-641321 socSMCs FET Proactive project. SC is funded by ERA Chair Project GRO (621403).

REFERENCES

- [1] M. Cerny, "Sarah and Sally : Creating a Likeable and Competent AI Sidekick for a Videogame," in *Proc. AIIDE Workshop EXAG*, 2015.
- [2] K. Merrick, "Modeling Motivation for Adaptive Nonplayer Characters in Dynamic Computer Game Worlds," *CiE*, vol. 5, no. 4, p. 1, 2008.
- [3] J. Forgette and M. Katchabaw, "Enabling Motivated Believable Agents With Reinforcement Learning," *Games Media Entert.*, pp. 1–8, 2014.
- [4] G. Smith, "The Future of Procedural Content Generation in Games," in *Proc. AIIDE Workshop EXAG*, 2014.
- [5] A. Liapis, G. N. Yannakakis, and J. Togelius, "Computational Game Creativity," in *Proc. 5th Int. Conf. Computational Creativity*, 2014.
- [6] M. Cook, S. Colton, and J. Gow, "The ANGELINA Videogame Design System, Part I," *IEEE TCIAIG*, no. c, pp. 1–12, 2016.
- [7] J. Schmidhuber, "Developmental Robotics, Optimal Artificial Curiosity, Creativity, Music, and the Fine Arts," *Connection Science*, vol. 18, no. 2, pp. 173–187, 2006.
- [8] P.-Y. Oudeyer and F. Kaplan, "How Can We Define Intrinsic Motivation?" in *Proc. 8th Conf. Epigenetic Robotics*, 2008, pp. 93–101.
- [9] R. Ryan and E. Deci, "Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions," *Contemporary Educational Psychology*, vol. 25, no. 1, pp. 54–67, 2000.
- [10] A. S. Klyubin, D. Polani, and C. L. Nehaniv, "Keep Your Options Open: An Information-Based Driving Principle for Sensorimotor Systems," *PLoS one*, vol. 3, no. 12, pp. 1–14, 2008.
- [11] C. Guckelsberger and C. Salge, "Does Empowerment Maximisation Allow for Enactive Artificial Agents?" in *Proc. 15th Conf. ALIFE (to appear)*, 2016.
- [12] K. McGee and A. T. Abraham, "Real-Time Team-Mate AI in Games: a Definition, Survey, & Critique," in *Proc. 5th Conf. FDG*, 2010, pp. 124–131.
- [13] A. Fern and P. Tadepalli, "A Computational Decision Theory for Interactive Assistants," in *Proc. 23rd Conf. NIPS*, 2010, pp. 577–585.
- [14] T.-H. D. Nguyen, D. Hsu, W.-S. Lee, T.-Y. Leong, L. P. Kaelbling, T. Lozano-Pérez, and A. H. Grant, "CAPIR: Collaborative Action Planning with Intention Recognition," in *Proc. 7th Conf. AIIDE*, 2011.
- [15] O. Macindoe, L. P. Kaelbling, and T. Lozano-Pérez, "POMCoP: Belief Space Planning for Sidekicks in Cooperative Games," in *Proc. 8th Conf. AIIDE*, 2012.
- [16] K. E. Merrick and M. L. Maher, *Motivated Reinforcement Learning. Curious Characters for Multiuser Games*. Springer, 2009.
- [17] S. Singh, A. Barto, and N. Chentanez, "Intrinsically Motivated Reinforcement Learning," *Proc. 18th Conf. NIPS*, vol. 17, pp. 1281–1288, 2004.
- [18] T. Anthony, D. Polani, and C. L. Nehaniv, "General Self-Motivation and Strategy Identification: Case Studies based on Sokoban and Pac-Man," *IEEE TCIAIG*, vol. 6, no. 1, pp. 1–17, 2014.
- [19] S. Mohamed and D. Rezende, "Stochastic Variational Information Maximisation," *Proc. 29th Conf. NIPS*, pp. 1–9, 2015.
- [20] C. Guckelsberger, C. Salge, R. Saunders, and S. Colton, "Supportive and Antagonistic Behaviour in Distributed Computational Creativity via Coupled Empowerment Maximisation," in *Proc. 7th Int. Conf. Computational Creativity*, 2016.
- [21] C. Salge, C. Glackin, and D. Polani, "Empowerment – an Introduction," *Guided Self-Organization: Inception*, pp. 67–114, 2014.
- [22] H. Touchette and S. Lloyd, "Information-Theoretic Approach to the Study of Control Systems," *Physica A*, vol. 331, 2004.
- [23] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Wiley-Interscience, 2006.
- [24] C. Guckelsberger and D. Polani, "Effects of Anticipation in Individually Motivated Behaviour on Survival and Control in a Multi-Agent Scenario with Resource Constraints," *Entropy*, vol. 16, no. 6, pp. 3357–3378, 2014.
- [25] D. Jallof, S. Risi, and J. Togelius, "EvoCommander: A Novel Game Based on Evolving and Switching Between Artificial Brains," *IEEE TCIAIG*, 2016.