Explainable AI for Designers: A Human-Centered Perspective on Mixed-Initiative Co-Creation

Jichen Zhu
Drexel University
Philadelphia, USA
jichen.zhu@drexel.edu

Antonios Liapis
University of Malta
Msida, Malta
antonios.liapis@um.edu.mt

Sebastian Risi
IT University of Copenhagen
Copenhagen, Denmark
sebr@itu.dk

Rafael Bidarra
Delft University of Technology
Delft, The Netherlands
R.Bidarra@tudelft.nl

G. Michael Youngblood
Palo Alto Research Center
California, USA
Michael.Youngblood@parc.com

***Abstract*—Growing interest in eXplainable Artificial Intelligence (XAI) aims to make AI and machine learning more understandable to human users. However, most existing work focuses on new algorithms, and not on usability, practical interpretability and efficacy on real users. In this vision paper, we propose a new research area of eXplainable AI for Designers (XAID), specifically for game designers. By focusing on a specific user group, their needs and tasks, we propose a human-centered approach for facilitating game designers to co-create with AI/ML techniques through XAID. We illustrate our initial XAID framework through three use cases, which require an understanding both of the innate properties of the AI techniques and users' needs, and we identify key open challenges.**

***Index Terms*—explainable artificial intelligence, mixed-initiative co-creation, human-computer interaction, machine learning, game design**

I. Introduction

With the swift development of artificial intelligence (AI) and machine learning (ML) in recent years, their applications (digital games included) have become more sophisticated. With the rise of algorithmic complexity, however, it is becoming increasingly difficult for humans to understand these algorithms and hence to have trust in them. For instance, while recent development of deep learning techniques produced impressive results, it is notoriously difficult for humans (programmers included) to gain full insights into the system's function.

In this vision paper, we focus on one group of human users. We propose a new research area of *eXplainable AI for Designers (XAID)* and specifically for game designers. The increase in game AI sophistication opens up a new creative design space for potentially new gameplay and/or more efficient production. However, game designers (such as rule designers, level designers and artists) often find these techniques inaccessible and difficult to explore their full creative potentials without a deep understanding of how they function. To the best of our knowledge, there has been a lack of XAI research to address this particular problem.

By focusing on a specific user group, their needs and tasks, we provide the basis of a human-centered XAID approach which facilitates game designers to co-create with AI/ML techniques. XAID can enhance game designers' capabilities to co-create playable experiences with AI, including but not limited to ML, agent control, procedural content generation, and planning. We believe that, although fundamental understandings of the properties of different AI/ML techniques are essential, the goal of XAID includes investigating the actual *usability* of XAI in terms of how it supports game designers in specific design tasks.

Below, Section II presents related work on XAI and mixed-initiative human-AI co-creativity. We present our framework on explainability and the three axes of XAID in Sections III and IV. Through three use cases, we illustrate our initial framework for XAID, requiring understanding both the innate properties of the AI/ML techniques and users needs. Finally, we identify key open challenges for future XAID research.

II. Related Work

Current XAI research can be classified by the types of techniques being illuminated (e.g. black-box techniques, white-box techniques). Given the limited research on XAI for game design, we also provide background on mixed-initiative co-creation systems where AI and human designers work together, an interaction model we envision XAID to extend. Finally, we review current evaluation methods of XAI.

A. Black-Box XAI approaches

Current XAI approaches for black-box systems such as neural networks can be roughly divided into approaches that aim to (a) visualize features, and (b) elucidate the relationship between neurons. Visualizing hidden layers' features [1]–[4] can give insights into what network inputs would cause a certain reaction in an internal neuron or in the network output. In contrast to optimizing the network's weights, as normally done through gradient to train the network, a researcher can use the same process to optimize an image that would maximally activate a certain neuron. These techniques can help to identify what certain neurons in a DNN pay attention to.

While these feature visualization techniques can offer insights into particular neurons, other approaches aim at under-

978-1-5386-4359-4/18/$31.00 ©2018 IEEE

standing how multiple neurons in a network interact to reach a decision. Techniques that aim to explain relationships between neurons are known as attribution and a variety of different approaches exists [2], [5], [6]. One of the simplest attribution examples is the saliency map, which is a heatmap highlighting which areas of the input image are most responsible for reaching a certain output classification.

These approaches—especially when combined—offer some insight into the inner workings of a neural network, making DNNs more of a grey than a black box. While earlier work tried to address this problem by e.g. creating decision rules or a decision tree out of a neural network [7], how these approaches will scale to modern DNNs is an open problem.

In a recent article [8], Olah *et al.* demonstrate how these interpretability building blocks can be combined in a unified interface to gain a deeper understanding of the workings of a neural network. We believe these techniques could help in creating compelling interfaces for designers.

B. White-Box XAI approaches

There is a large body of work in helping human users better understand white-box AI techniques, whose inner workings are transparent (e.g., simple decision trees). Earlier work of XAI can be traced to expert systems and Bayesian networks [9]. In a review, Lacave and Díez [10] categorized existing approaches into three main types: explanation of evidence, explanation of the model, and explanation of the reasoning. Notably, they pointed out a serious limitation in research which focused mainly on theoretical models of explanation without empirically validating these approaches with human users. We argue that the current state of XAI, including both white- and black-box AI, shares a similar limitation.

In the domain of planning, research in *plan explanations* attempts to make the systems' output more understandable through a more understandable representation of plans [11] and by generating explanations using natural language [12]. More recently, the focus has shifted from explaining the plans themselves to explaining how the planner produces its output [13]. There is growing interest in explaining the planner's behavior through verbalization [12], [14].

Specifically for games, limited work exists on explaining the underlying white-box systems. As an example, the graphical representation of behavior trees has made it easier for game designers and artists to understand how the underlying AI functions. Another loosely related work is on explaining utility AI: [15] annotates positions in a 3D shooter game based on their strategic value (e.g. at the right distance to an enemy with coverage from secondary threat). However, there is very little work on explaining white-box AI techniques for the purpose of facilitating design tasks.

C. Mixed-Initiative Co-Creative Systems

We envision XAID as a useful way to facilitate game designers in their work. Interfaces intended to help designers create content and, more broadly, design games have long been challenged to provide appropriate, informative feedback to their end-users. Game engines and their editors offer a variety of intuitive interfaces for simplifying a user's tasks. Through the use of AI, these computer-aided design tools are elevated to *mixed-initiative co-creative systems* [16] where 'both the human and the computer proactively make contributions to the problem solution, although the two initiatives do not need to contribute to the same degree'. Likening the design process to a dialog between colleagues [17], computational initiative can refer to the task initiative (i.e. who initiates the dialog), speaker initiative (i.e. when each actor will speak, and whether actors can interrupt each other), and outcome initiative (i.e. who decides when the dialog is finished or the problem is solved). The dialog analogy clarifies how explainable AI is vital in conveying to the user its reasons in taking any of task, speaker or outcome initiatives mentioned.

Numerous mixed-initiative co-creative tools have been developed over the last decade for game design, although for the most part as academic rather than commercial endeavors. Many of these tools focused on explaining the properties of game design artifacts that the computational designer produced for direct use or for further editing by the human designer [18]–[20]. For example, *Sentient Sketchbook* [18] autonomously creates levels as alternatives to what the designer is currently doing, and there is no explanation regarding such a task initiative. If the designer stops and observes each computational suggestion, the interface displays numerically which functional level properties (e.g. area balance or exploration) improve or decrease compared to the current human sketch. Through fairly simple visual feedback (e.g. plus and minus signs), the tool attempts to explain why this suggestion could be desirable or undesirable to the designer.

Numerous mixed-initiative tools have focused on visualizing such properties of their specific artifacts for each user (and in the case of Danesh [20], properties of a large sample of artifacts). However, there is little research in explaining the creative process (rather than the final artifact) in co-creative tools, and all attempts to date have focused on visualizations rather than on natural language generation of the explanation. On the other hand, there have been several interesting attempts at explaining autonomously creative processes (without a designer involved either as a co-creator or as a consumer of the explanation) both in game generators such as *Angelina* [21] and in broader creative software such as *The Painting Fool* [22]. Many of the positions in this article, especially in Section VI-A, borrow from these white-box generative systems.

D. Measuring Explanations

While there is no established definition of explainability and how to measure it, ultimately explanations serve to build understanding and possibly trust between the AI and the user or beneficiary of the AI. Testing understanding of software has a long history in human computer interaction and education. For complex mechanisms of AI and ML, understanding requires testing model induction: how does the induced mental model a person holds match or differ from the actual model?

The typical route for model testing is through instance testing: given a specific instance, can a human predict or determine how the model will act? Complex instances may be decomposed into sub-instances or competencies and tested in smaller measures to determine the level of model match and understanding. The learning rate, precision, and recall of the process and the induced model are important factors. Explanations that improve these can lead to a qualitative measure for comparison between explanations. Some explanations may require more mental processing to learn, so task loading is also a consideration. Measuring trust in machines is a complex issue and still an active area of research in psychology [23].

III. Explaining Explainability

Developing XAID to facilitate design tasks first requires a thorough understanding of explainability and how it connects to the properties of different AI/ML techniques. Interest has been increasing in better understanding some of the learned AI models, specifically in the field of machine learning. Techniques that are reduced into networked structures of weights with complex topologies and varying transformation functions embedded in them are notoriously difficult to understand by humans. Yet, with the recent advances in deep learning, these models are being used in a broadening and critical set of everyday life applications. Reliance on ML for critical tasks and especially those involving human-life requires trust, which is typically gained through some level of transparency that facilitates a comfortable level of model understanding for the person giving that trust [24].

Explainability is not just needed in opaque, machine-learned models, but in many facets of AI. We define explainability as being clear of obscurity and understandable in all aspects. This means that to truly understand something, we must be able to introspect all of its mechanisms. Some argue that this a white-box view of explainability; we maintain that this is the only true explainability: the ability to answer *why* questions.

Axiom 1: **Explanation without introspection is not explanation.**

An argument can be made that some reactive (black-box) techniques are fully understandable from observation of all potential combinations of input and their related outputs. This black-box view does provide an understanding of behaviors, but it does not address the obscurity of the underlying model, and thus we call this Observable AI, which is valuable and may suffice for proper model induction in humans. Observation, however, is not an explanation: it cannot truly answer *why* it does what it does. It is also important to note that black-box testing of complex models may be intractable, so observable behavior may have a level of uncertainty that matches the inconsistencies and incompleteness of the observations made.

Axiom 2: **Understanding through external probing is observation.**

If we look at the spectrum of AI techniques as shown in Fig. 1, we can reduce them to a dimension of reactive to deliberative—or through Daniel Kahneman's lens [25], fast and slow thinking[1]. Performing thinking as a reflex, which is a stimulus-driven control policy, is something that an agent may have been created with. Such a thought (which came into existence as a black box) can not be inspected and may never be truly explainable.

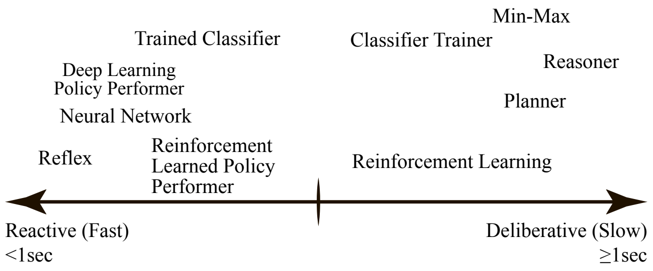

Fig. 1. AI/ML Techniques on a reactive to deliberative Scale.

Axiom 3: **Reactive elements that have always been reactive from inception are not explainable, but may be observable.**

Many reactive elements are created through deliberative training processes as shown in Fig. 2. Deliberative processes have the property of all being inspectable and procedural. This is not to say that all deliberative techniques are explainable, but their obscurity comes from complexity (e.g., processes with many steps, stages, or interoperable rules), not reduction.

Conjecture 1: **Deliberative elements may require explainability due to complexity.**

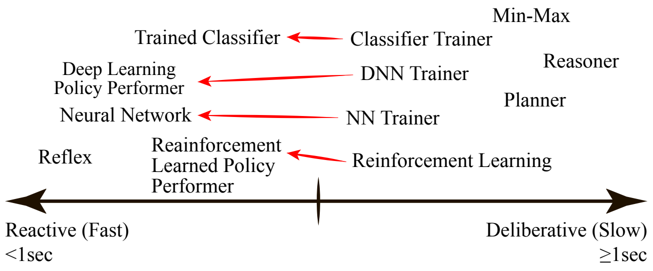

Fig. 2. Mapping deliberative AI/ML techniques to reactive processes.

Obscurity in reactive elements comes from dimensionality reduction of the state space or the data. This reduction is the condensation of training data into a model, by transforming data and state space topology from the deliberative process to the reactive.

Conjecture 2: **Reactive elements may require explainability due to reduction of training into a model through data and topology transforms.**

If we want to understand and inspect a reactive model, the explanation lies in the deliberative portion that created the model. This typically has a temporal aspect to it. The

[1] For a complex agent, such as humans, the line between how much of what an agent uses for decision-making exists in the reactive versus the deliberative scale is an open area of research.

exception may be one-shot learning techniques, which is a topic for another paper.

Axiom 4: **The explanation of a reactive model lies in the deliberative process that created it.**

For reactive processes, the ways in which eXplainable AI (XAI) techniques may manifest themselves is via injection of explainable notes to the reactive process; these notes are created by the deliberative process(es) that led to the reactive process, and provide a surface for post-hoc analysis of the training of the reactive model in the deliberative process(es). Alternatively, the deliberative process(es) may generate a separate explanation in the process of creating the reactive model. These explanations serve to illuminate the reduction process and the impact of what is learned that drives the reactive model behavior. These aim to largely answer the *what* and *why* details of decisions the model will make based on perception or who, what, when, and where influenced the behavioral response. It may also answer the more mechanistic *how*.

For deliberative processes (such as those shown in Fig. 2), the ways in which XAI techniques may manifest themselves is by making the complex tractable for human understanding. It is a reduction, reorganization, or reframing of the complex into something understandable that maintains the transparency and introspection of the model.

IV. eXplainable AI for Designers (XAID)

It is generally agreed upon that the goal of XAI is to increase users' trust, their ability to interact with the systems and with their decisions, and improve the transparency of the system [13]. However, most existing work focuses on new algorithms of XAI rather than on usability, practical interpretability and efficacy on real users [26]–[28]. Although we believe that fundamental understandings of the properties of different AI algorithms are an essential part, XAI techniques should be developed with specific users and their needs in mind if they are to fulfill their promise.

We propose a new area of research in eXplainable AI for Designers (XAID) who create interactive digital products built on AI components. As AI and ML techniques are mature enough to reach commercial products (e.g. computer games, virtual assistants, smart objects), designers need to understand how the AI component works in order to devise desirable ways for the end-users to interact with the systems. Unlike the end-users of an AI system, designers constitute a unique user group because they not only consume the results of AI systems, but also *co-create* with them. To the best of our knowledge, no XAI work focuses on designers and co-creation.

In the rest of this paper, we focus specifically on game designers (such as rule designers, level designers and artists) who do not have a strong technical background in AI. The key purpose for XAID is to (a) provide designers with sufficient understanding of the underlying AI system and its behavior, and hence (b) facilitate their design tasks through co-creation. Our positions are the following:

1) Work in XAID needs to build on understanding of the nature of underlying AI techniques. As argued above, different AI techniques afford explanations with introspection while others afford only observations. Although both can be useful for designers, understanding the option of explanation and/or observation can help shape how XAID can support the co-creative process between designers and AI.
2) Work in XAID needs to center on specific human users (e.g. game designers) and their specific needs and tasks. Compared to more general XAI research, XAID as proposed here has the advantage of a more concrete if narrow scope. Through the three specific use cases in Section VI, we argue that work in XAID should be designed for and evaluated with specific users.

V. Mapping the Space of XAID

In a broad stroke, we describe the XAID space along three main axes, each one spanning its own spectrum.

A. Spectrum of Explainability

We first identify the XAID spectrum of explainability, ranging from *explanations* that provide introspection into the operation of AI techniques to *observations* that offer insights of the input-output pattern. Explanations can provide designers with information such as the chain of actions and why the algorithm takes a specific decision. Observations, for instance, can be used to inform game designers of all the possible actions an AI-controlled non-player character (NPC) will take at a given game state and the likelihood of each action.

Although the decision between offering explanations or observations relate to the properties of the underlying algorithms, as argued above, it also depends on the needs of the designers and their tasks. For example, observations may be the best choice for level designers working with a white-box NPC AI because understanding the exact operation of the AI is not necessary to their design task and may cause information overload. By contrast, a game designer tasked with game balancing in the same project may need to know exactly why the NPC performs certain actions—especially if they are unexpected—in case the game attributes themselves (which may influence these decisions) need to be corrected.

B. Spectrum of Initiative

A crucial aspect of any AI-assisted design system is the degree and type of initiative that it can take in performing its tasks. The spectrum of initiative traces the limits of a system's intervention, and it determines the kind of explanations that it may be expected to provide. We can distinguish three bands in this spectrum, corresponding to the *level of system initiative*, each with its typical kind of explanation.

At the lowest level, the system passively waits for the designer to request assistance, e.g. some on-demand analysis. For this, a typical explanation can include a simple description of the task performed, possibly with a number of meaningful parameters used to yield its output.

At the next level, we can devise AI assistance that requires a higher degree of autonomy. Correspondingly, the kind of explanations involved in their execution has an increasing complexity. Some examples of these (together with a possible explanation) are:

- explore how to proceed (e.g. describe the space of possible alternatives, sampling methods used, evaluation criteria);
- sketch a range of choices (e.g. characterize the extreme points of a range of options, to give insight into what it involves);
- warn a designer regarding some risk ahead (e.g. look ahead for what-if analysis, to identify and describe conflicts or risks)

The explanations mentioned in these examples require a considerable understanding of the processes and goals at hand.

Going even higher on the spectrum of autonomy and initiative, we can devise an AI system working on par with the human designer, taking on activities more as a colleague than as an AI assistant. At this level the tasks, outcomes and explanations are currently only expected from a human co-designer, not an AI assistant. Examples of these could be:

- making informed design choices (e.g. based on the awareness of the goals of the design task);
- intervene to suggest the best way to proceed, e.g. switch focus to another task, attempt some alternative solution or try to avoid an early commitment (justifying such suggestions requires a much deeper understanding of both the design situation, its history and the available options);
- signal and correct a 'mistake' made by the designer (this requires explaining why it is perceived as a design mistake, e.g. violating some previously stated intent, and finding out alternatives with a better outcome);
- propose a sensible task division (explaining such a proposal will likely need a formidable amount of knowledge; in addition to the above, this typically human activity requires meta-knowledge on both the nature of each subproblem, their relation to the ultimate goal sought and the competences of team members, AI or otherwise).

For some more down-to-earth activities, one can imagine them taking place at any level of the spectrum above. For example, the creation of a specific type of content could be either explicitly issued from a procedural content generation (PCG) algorithm by a designer, suggested by the system at some appropriate stage, or autonomously performed as a fitting complement to whatever else the designer is doing. However, explanations on that same activity will likely have to vary according to the level of initiative being taken.

C. Spectrum of Domain Overlap

Another aspect of XAID is the amount of overlap between the tasks performed by the designer and the tasks performed by the AI. To a certain degree, this can be considered the degree of co-creativity that is needed, and can similarly affect how (or how much) each task by the AI needs to be explained. The spectrum of domain overlap ranges from the scenario that AI and a human designer making use of the same tools applied to the same task (*on-task* co-creative activities) to the scenario that a human designer is working on an aspect of the game while the AI handles another that only tangentially be affected by the designer's input (*off-task* co-creativity).

Let us consider how the explanations differ between on-task and off-task AI co-creativity. If an AI and a designer work on the same domain, e.g. changing the same game level using the same tile-based structure as is the case in *Sentient Sketchbook*, the explanations provided by the AI should be fairly specific as the designer (a) is very aware of the terminology and current problems of the work in progress, (b) can directly observe the elements that the AI refers to, and (c) must be able to take immediate decisions regarding the AI suggestions, e.g. to accept or reject them. In an example of an off-task collaborator, we consider a designer who is creating a level which an AI agent playtester attempts to solve, similarly to *Roppossum* [29]. In such a case, the human-made artifact (level) directly affects the AI agent, but the explanations provided by the playtester should focus on level-specific concerns (e.g. "this platform is 90% likely to cause me to overshoot, if the player has poor reflex time"), rather than explanations of its behavior. The level designer may not be knowledgeable of (or interested in) the AI agent's internal decision-making priorities, but instead may be interested why the level is deemed unplayable by the AI agent.

VI. Three Use Cases

In order to illustrate the human-centered perspective on XAID described so far, we discuss three different use cases.

A. Use Case 1: White-Box PCG System

The first use case tackles the problem of a level designer who is using a computer-aided design (CAD) tool to create the perfect overworld map for a free-roaming car-racing game similar to *Mad Max* (Warner bros 2015). Apart from typical CAD functionality such as texture brushes, mesh placement and camera movement, the tool can generate the entire level (or parts of the level) on command. The algorithm for this generative component is based on grammars, which have been inserted into the system not by this level designer, but by a tool programmer who may not be working in the same company.

The above instance is one of *on-task* collaboration (the two designers literally work on the same map), and the level of initiative is *on-demand*. Due to the deliberative steps that a grammar generator takes, the algorithm can narrate its generative process as textual output, which is in the form of *explanation*. Given the fact that the generative grammars follow a fairly transparent process, we could conceive that the explanation generator could be included within the procedural generator with a sentence produced after relevant commands, function calls or choices: in this particular case, grammar expansions. This goes beyond a simple log of steps and decisions taken, e.g. it may also carry information on the context influencing those decisions. Generative grammars have an ideal generative architecture for such an explanation because it is, in many respects, a pipeline: each expansion produces an intermediate output that is passed on as input to a further expansion [30]. Presenting to the designer a compelling and intuitive narrative regarding the choices taken by the PCG system can be done in a variety of ways, including:

- *sequentially* in the order that the system makes decisions. This explanation can follow some form of story structure which simulates e.g. the generative pipeline [22]. In order to enhance (e.g. via natural language processing) how the connections are made between different steps of the generation, we can investigate work on story generation so that the narrative is coherent and causal links are made obvious. This can be achieved, for example, by post-processing the generated sequence of sentences to introduce throwbacks to past generative decisions which affect future outcomes, or to foreshadow how one early decision affects the final outcome.
- *summary of highlights* of the generative process, by filtering out and omitting less interesting points in the generated sentence structure. For this to happen, a number of evaluation mechanisms are needed, defining criteria to assess each sentence on its relevance (will this be interesting to a human user?), clarity (will this be understandable by a human user?), or creativity (will this step be a creative milestone [16] where the design shifts?).
- *non-sequentially*, summarizing the explanation starting from the most important points regardless of when they were performed in the generative process. Indeed, it is possible to start by presenting a description (visual, textual, or otherwise) of the final artifact and backtracking some of its most interesting elements on points in the generative process where those happened. Moreover, tropes such as sports game summaries can be used as inspiration, presenting the main outcomes of the generative process first (as non-sequential highlights) followed by a longer form of the sequential narrative regarding how generation progressed from unformed to fully formed content.

B. Use Case 2: Black-box PCG System

In a black-box PCG system such as a level generator or world builder, the AI assistant needs to (a) share a common language of design with the human designer, (b) communicate its current understanding of this language, and (c) update this understanding in response to designer feedback. Essentially, this is the notion of establishing ‘common ground’ as forwarded and explored by Herb Clark [31]. The designer will provide input into the black-box AI assistant with the goal of receiving a full or partial design from the system. For example, assume that the AI assistant aims to generate the 3D geometry for a city similar to CityEngine [32]. The input will be a size of the land with topology on which to generate the city and a set of parameters that are used in the construction of the road and transit structures, building designs, and placement of buildings in blocks along streets (as this is how humans build cities). There is a lot of information to convey and a nearly infinite way to construct cities; however, cities are created all the time—even virtual ones. The black-box PCG system will need to be clear about how it takes that input and ultimately how that connects to and affects the output. It is the transformation by the AI from input to output that needs to be explained and this is where common ground is leveraged.

A designer first working with an AI assistant could spend a great amount of time probing the system with variations and developing a mapping (or model) of how changes in input impact output (learning by observation). However, that is quite tedious and a more abstract, explained transformation process would be faster to comprehend and work with for a designer. A human may direct an AI assistant to build a city in the ‘American style’; knowing that this means a city laid out in generously-sized square blocks with most streets having simple intersections is an easy and powerful way to produce a desired design. Ultimately, this is a direction for an agreed concept in the transformation process of the technique. There are a lot of details needed to produce that design, which are encapsulated in a specific design concept. Mechanisms that build and update that common ground language and mapped meaning need to be added to the techniques inside the black-box. For reactive techniques, the artifacts of training data and the process of machine learning may need to be included in some form to facilitate explanation of the internal mechanisms.

Imagine a black-box AI assistant using a DNN to recognize ideal topology for road placement, which is then placed by a set of construction rules biased by a provided set of city road layout examples filtered by design language labels. In order to build *common ground*, the AI assistant will need to interactively show the designer how the provided land and topology are perceived and how its prior examples are used to generate roads based on the language provided, as well as how these concepts may roll-up hierarchically in the system. This may involve keeping connections to the training data used in the deliberative creation process of reactive techniques in the system. Thus, the system reveals enough information to allow the induction of a model of the AI in the designer. When a mapping of the designer’s internal model is connected to a correct induction of the AI assistant’s internal model, common ground is established by sharing a language, an understanding, and the ability to update both sides easily. The key challenge is how and what to share to build that model in the designer’s mind without exposing them to the potentially massive amounts of data used to train the network and used by the system for making decisions. Induced models in humans can be tested by predictive capability and accuracy.

On the XAID spectra, black-box PCG AI assistants for designers require the most explainability as they involve learning, recognizing, and extending patterns to create content the subtleties of which a designer will want to understand and work on together with the AI. On initiative, these systems are likely to be *on-task* colleagues or have high-functioning autonomy. On domain overlap, as the example given in Section V-C, creating content with an AI will have high overlap, but *on-demand* and often *turn-based*. The PCG AI assistant and the designer work closely together, refining until the desired content is produced. The designer provides the vision, the AI provides capabilities, and they merge that into the creation.

C. *Use Case 3: Black-box NPC Behavior System*

In a third use case, imagine an enemy NPC behavior controlled by a trained DNN. The goal of the game designer is to see whether the NPC behaves as intended in a new game level (in our case, an infirmary). Since the network encodes complex NPC behaviors in an opaque way, an XAID system should be able to help the designer to better understand the NPC AI. We design this XAID task to be *observable* and *passively* awaiting the designer's request to provide insights on how the NPC will behave.

Two types of information are of particular importance to the designer. First, given the layout of the level, what is the likely distribution of actions the NPC will take? For example, at the entrance of the infirmary, how often will the NPC walk straight inside, turn left to interact with another NPC, or turn right and avoid the level altogether? This distribution can be approximated through sampling, i.e. letting an NPC play a certain level multiple times with slight variations in starting position, etc.. If it is crucial for the player to encounter this NPC and the latter has a high chance of leaving, the designer needs to be informed and be provided with a reason why this happens and how to correct it. Through a mixed-initiative approach, the system could also suggest changes to the environment that would make the desired behavioral outcome more likely.

Second, given a particular NPC action, what are all the possible situations that can lead to this action? If the NPC sometimes has the unexpected behavior of shooting at a window, it would be helpful for the XAID to show all the situations where this will happen. By providing a full list of scenarios that will lead to a particular action, this feature will make the NPC behavior system more predictable and thus may increase designer's trust in the behavior system. Methods such as feature visualization and attribution (Section II-A) can give insights to what stimulus the network will react to. To the best of our knowledge, however, there is currently no approach that can do all of this in an automated way.

Given the large number of possible actions and/or situations, similar to highlights in white-box PCG systems, a good design guideline for XAID is to highlight the unexpected and reduce the visibility of the common ones. A key open challenge to providing both types of information to a human designer is how to design the reward function for the NPC.

VII. Open Challenges

In this section we point out some of the open challenges in providing useful XAID in relation to both white-box and black-box systems, as well as their combination.

A. *White-Box Systems*

An open challenge in providing useful XAID is how to fit the entire process of the white-box system into something that is compact and yet sufficient for designers. Similar to how a black-box ML model can show all of the training data (Section VII-B), a white-box model can explain (i.e. narrate) the sequence of all actions that it takes (including iterations within loops). The challenge is how to cluster or omit activity reports that are less relevant for the designer to know. Some of the actions reported may be too 'esoteric' (i.e. tied to the system's internal method of understanding the world or producing new artifacts) for the user to understand. In order to create 'highlights' as noted in Section VI-A, the challenge of evaluating subjective notions such as *interestingness* or *relevance* may require a computational model of the individual designer [33]. Moreover, such criteria might need to operate beyond the horizon of a single sentence; the whole narrative (sequence of sentences) must be produced before the most interesting points within it are chosen in a post-processing step. In addition, the concrete features of the final artifact (be it game content, NPC behavior, etc.) may also be relevant for this post-processing.

B. *Black-Box Systems*

While different techniques are now emerging that can give some insights into the working of black-box systems such as neural networks (surveyed in Section II-A), they can currently only help to interpret a model but lack full explainability. The more complex these models become, one can wonder if it will ever be possible to fully explain their inner workings.

Meaningful abstraction from base provenance can be difficult. While it is possible to show all the training data or even clusters of training data for explanation, this process may overwhelm a user and fail to induce a model of understanding. Providing proper, meaningful and likely hierarchical abstractions of training data and the transformation into learned models is an open challenge. While the core question that leads to understandable AI is 'why?', the answer should come from introspection, which is an open challenge for many techniques, especially black-box methods.

C. *Combined approaches*

While explaining white-box and black-box systems is difficult in the context of XAID, understanding complex AI systems with many parts and multiple techniques becomes an even more challenging problem; understanding the sum is not the same as understanding the parts. An important future direction will be the development of dialog and concept grounding, building common ground between AI and designers.

VIII. Conclusions

In conclusion, we proposed the new research area of eXplainable AI for designers (XAID), to help game designers better utilize AI and ML in their design tasks through co-creation. Our position is that, in order to make usable and efficient XAID systems, we need to build on understandings of both algorithmic properties of the underlying AI techniques and the needs of human designers. We mapped the space of XAID with three axes—the spectra of explainability, initiative, and domain overlap—and illustrated our approach through three specific use cases. Based on a deeper analysis into use cases, we identified key open challenges.

Acknowledgements

This article extends the work performed during the 2017 Dagstuhl seminar 17471 on "Artificial and Computational Intelligence in Games: AI-Driven Game Design". An initial report on the topic, titled "Explainable AI for designers", can be found in [34]. We would like to thank all of the researchers who contributed their ideas in writing the Dagstuhl report; their insight has led to this publication.

References

[1] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 427–436.

[2] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[3] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski, "Plug & play generative networks: Conditional iterative generation of images in latent space," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3510–3520.

[4] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," University of Montreal, Tech. Rep. 1341, Jun. 2009.

[5] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proceedings of the European conference on computer vision*. Springer, 2014, pp. 818–833.

[6] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," *arXiv preprint arXiv:1704.03296*, 2017.

[7] O. Boz, "Converting a trained neural network to a decision tree dectext - decision tree extractor," Ph.D. dissertation, Lehigh University, 2000.

[8] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev, "The building blocks of interpretability," 2018, [Online; accessed 13 May 2018]. [Online]. Available: https://distill.pub/2018/building-blocks/

[9] O. Biran and C. Cotton, "Explanation and justification in machine learning: A survey," in *Proceedings of the IJCAI Workshop on Explainable Artificial Intelligence*, 2017.

[10] C. Lacave and F. J. Díez, "A review of explanation methods for bayesian networks," *The Knowledge Engineering Review*, vol. 17, no. 2, pp. 107–127, 2002.

[11] B. Seegebarth, F. Müller, B. Schattenberg, and S. Biundo, "Making hybrid plans more clear to human users-a formal approach for generating sound explanations," in *Proceedings of the International Conference on Automated Planning and Scheduling*, 2012.

[12] J. Bidot, S. Biundo, T. Heinroth, W. Minker, F. Nothdurft, and B. Schattenberg, "Verbal plan explanations for hybrid planning," in *Proceedings of MKWI workshop on Planung/Scheduling und Konfigurieren/Entwerfen*, 2010.

[13] M. Fox, D. Long, and D. Magazzeni, "Explainable planning," in *Proceedings of the IJCAI Workshop on Explainable Artificial Intelligence*, 2017.

[14] S. Rosenthal, S. P. Selvaraj, and M. M. Veloso, "Verbalization: Narration of autonomous robot experience," in *Proceedings of the IJCAI conference*, 2016, pp. 862–868.

[15] R. Straatman and A. Beij, "Killzones ai: dynamic procedural combat tactics," in *Game Developers Conference*, 2005.

[16] G. N. Yannakakis, A. Liapis, and C. Alexopoulos, "Mixed-initiative co-creativity," in *Proceedings of the Foundations of Digital Games Conference*, 2014.

[17] D. Novick and S. Sutton, "What is mixed-initiative interaction?" in *Proceedings of the AAAI Spring Symposium on Computational Models for Mixed Initiative Interaction*, 1997.

[18] A. Liapis, G. N. Yannakakis, and J. Togelius, "Sentient sketchbook: Computer-aided game level authoring," in *Proceedings of the Foundations of Digital Games Conference*, 2013.

[19] G. Smith, J. Whitehead, and M. Mateas, "Tanagra: Reactive planning and constraint solving for mixed-initiative level design," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, 2011.

[20] M. Cook, J. Gow, and S. Colton, "Towards the automatic optimisation of procedural content generators," in *Proceedings of the IEEE conference on Computational Intelligence and Games*, 2016.

[21] M. Cook and S. Colton, "Ludus ex machina: Building a 3d game designer that competes alongside humans," in *Proceedings of the International Conference on Computational Creativity*, 2014.

[22] S. Colton, J. Halskov, D. Ventura, I. Gouldstone, M. Cook, and B. Perez-Ferrer, "The Painting Fool sees! New projects with the automated painter," in *Proceedings of the International Conference on Computational Creativity*, 2015.

[23] B. F. Malle, *How the mind explains behavior: Folk explanations, meaning, and social interaction*. Cambridge, MA: MIT Press, 2004.

[24] P. Pu and L. Chen, "Trust building with explanation interfaces," in *Proceedings of the 11th international conference on Intelligent user interfaces*. ACM, 2006, pp. 93–100.

[25] D. Kahneman, *Thinking, fast and slow*. Farrar, Straus and Giroux, 2011.

[26] A. Abdul, J. Vermeulen, D. Wang, B. Y. Lim, and M. Kankanhalli, "Trends and trajectories for explainable, accountable and intelligible systems: An hci research agenda," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2018.

[27] F. Doshi-Velez and B. Kim, "A roadmap for a rigorous science of interpretability," *arXiv preprint arXiv:1702.08608*, 2017.

[28] T. Miller, P. Howe, and L. Sonenberg, "Explainable AI: Beware of inmates running the asylum," in *Proceedings of the IJCAI Workshop on Workshop on Explainable Artificial Intelligence*, 2017.

[29] M. Shaker, N. Shaker, and J. Togelius, "Ropossum: An authoring tool for designing, optimizing and solving cut the rope levels," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2013.

[30] J. Charnley, S. Colton, M. T. Llano, Rodriguez, and J. Corneli, "The FloWr online platform automated programming and computational creativity as a service," in *Proceedings of the International Conference on Computational Creativity*, 2015.

[31] H. H. Clark, R. Schreuder, and S. Buttrick, "Common ground at the understanding of demonstrative reference," *Journal of Verbal Learning and Verbal Behavior*, vol. 22, no. 2, pp. 245 – 258, 1983.

[32] ESRI, "City engine," 2018. [Online]. Available: http://www.esri.com/software/cityengine

[33] A. Liapis, G. N. Yannakakis, and J. Togelius, "Designer modeling for personalized game content creation tools," in *Proceedings of the AIIDE Workshop on Artificial Intelligence & Game Aesthetics*, 2013.

[34] P. Spronck, E. André, M. Cook, and M. Preuß, "Artificial and Computational Intelligence in Games: AI-Driven Game Design (Dagstuhl Seminar 17471)," *Dagstuhl Reports*, vol. 7, no. 11, pp. 86–129, 2018.