Beyond Computational Intelligence to Computational Creativity in Games

Dan Ventura
Computer Science Department
Brigham Young University
Provo, UT 84602
Email: ventura@cs.byu.edu

Abstract—**This paper argues that computational creativity is the logical next step in the evolution of game design; briefly overviews what is meant by computational creativity and suggests some ways in which it could augment contemporary games; explores some initial ideas for its incorporation into the future of gaming and game design; and argues for increased cross-pollination and collaboration between the computational intelligence and games research community and the computational creativity research community.**

I. Introduction

Computational intelligence has become the (near) future of game design and development, and it is interesting to ask then what might be the next logical step for the longer-term. Here, it is argued that step is *computational creativity* (CC).

The computational intelligence and games (CiG) community has facilitated the wide-spread adoption of procedural content generation (PCG) and artificial intelligence (AI) into game development and experience. Procedural generation of content for games is increasingly allowing designers to focus on higher-level concerns, while automatic generators produce lower-level content such as textures, landscapes, buildings, layouts, music, simple dialogues, etc. [1], [2]. Artificial intelligence is being used most commonly in the form of non-player characters and other types of agents, but it has been used in many other ways as well, including dynamic difficulty balancing [3], [4], player experience modeling [5], [6], and datamining of user behavior [7], [8].

Recently, it has been proposed that the games domain may be the "killer app" for the nascent field of computational creativity [9], and the arguments supporting this position are rather compelling, enough so to suggest a reciprocal kind of relationship—*if games are the "killer app" for computational creativity, then perhaps computational creativity is the future of games*. The beginnings of this idea have, in fact, been suggested elsewhere [10], and the purpose here is to argue for this evolution in the extreme. As the games community has embraced computational intelligence as an integrated augmentation device for the designer's intelligence, this suggests that the next possible step is to do the same thing for the designer's creativity. By embracing computational creativity, game developers can build computational collaborators that take real creative responsibility as a member of a team, and, in the extreme, may build fully autonomous content- or game-creation systems that are legitimate designers themselves.

The integration of CC philosophies and techniques into the game design process is a natural next step in the computational intelligence and games evolution, allowing for greater personalization of the gaming experience, novel and adaptable system behaviors, and the transitioning of intelligence into higher-level components of game creation. It can produce a more immersive game-play experience (e.g. more robust and believable NPC intelligence, personalized and adaptive content), allow for the automatic creation of additional types of game content (and eventually even complete complex games), and make game creation more scalable by relieving the human-creator bottleneck.

II. What is computational creativity?

Because the concept of creativity is very difficult, if not impossible, to formalize and is likely, in fact, an *essentially contested concept* [11], the field as a whole has agreed not to argue (any longer) about what exactly it is but instead focuses on building computationally creative systems. While this may seem disingenuous, it should be noted that the broader field of AI has dealt with a similar crisis of agreement cf. the meaning of intelligence, and yet has still made remarkable discoveries and inventions that everyone agrees are useful progress and likely even constitute "intelligence" in some sense. It is in this same hopeful spirit that the CC field has, in the last few years adopted a circular description suggested by Colton and Wiggins [12] as its stand-in for a proper definition:

> [Computational creativity is] *the philosophy, science and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviours that unbiased observers would deem to be creative.*

So, the goal is to build systems that do things that are creative. This is pretty wide open, as almost any domain of endeavor can be argued to require creativity to meet at least some of its challenges, from the artistic to the scientific to the mundane. Indeed, creativity is certainly an aspect of intelligence and is likely at least part of the reason that human intelligence is so robust. While, to date, computationally intelligent systems have demonstrated and continue to demonstrate some remarkable abilities, they are, at the same time, distinctly fragile, in the sense that their domain of expertise is almost vanishingly narrow. In contrast, humans demonstrate an impressively broad general intelligence, and at least one thing that differentiates human and computational intelligence seems to be creativity.

Perhaps the defining difference between computational intelligence and computational creativity is in the types of problems to which they are applied; or possibly the difference can be captured in the their approaches to solving these problems. In the case of computational intelligence, problems are usually framed as some type of optimization—the system is trying to attain the highest accuracy, or maximize area under the curve, or minimize losses, or maximize discounted future reward, or win. Actually, perhaps the concept of winning could be argued to be not quite optimization, in the sense that there may be more than one winning strategy, and in fact, given the case of an opponent strategy, the winning strategy may be dynamic, but fixing opponent strategy, it is still largely a type of optimization, or, possibly it is a form of satisficing, which is simply a weaker form of optimization—it is just "optimizing" for "good enough". In contrast, the kinds of problems that computational creativity addresses are of an entirely different class. There is no such thing as a best song, or best theorem or best design. One cannot maximize a piece of visual art or a recipe or a poem. There are many interesting songs, theorems, designs, paintings, recipes and poems, and the goal is to find one or more of these. What constitutes a "solution" for these types of "problems" might also be dynamic based on environment, but even if such environmental factors can theoretically be held constant, this is still nothing like an optimization problem, at least in the traditional sense.

Instead, success is qualified (and sometimes possibly quantified) with notions of *novelty* and *utility*. Whatever the domain in question, a computationally creative system should produce artifacts[1] that are novel and useful. Here, these qualities will be defined as follows:

novelty: the quality of being new, original or unusual; this is relative to the population of artifacts in the domain in question and can apply in the personal or historical sense.

value: the importance, worth or usefulness of something; this would typically be ascribed by practitioners of the domain in question.

Note that these qualities are most commonly addressed with respect to product; however, creativity often refers at least as much to process as it does to product, and, in particular, it has been argued that the (perception of the) process by which a computational system produces artifacts is likely at least as important as the artifacts produced [13].

To summarize, computational creativity deals with constructing artificial agents that produce artifacts that are judged novel and useful by those that understand the domain in which the agent works.

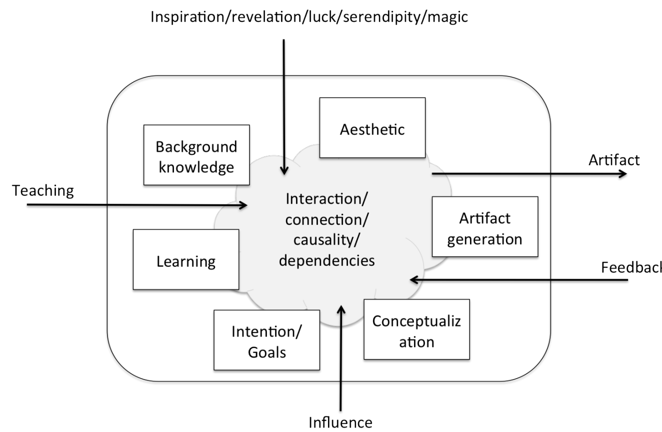

Fig. 1. *An abstraction of a creative agent*. The component internal mechanisms are meant to be likely necessary though possibly not sufficient. No attempt here is made to accurately visualize the dependencies and communication between these mechanisms. The agent communicates with the environment in several ways, represented by labeled arrows entering or leaving the agent (borrowed from [14]).

A. An Abstraction of the Creative Agent

Figure 1 offers a gross visualization of an abstract, archetypal CC agent. Such an agent is composed of multiple internal mechanisms/processes, some of which include *background knowledge*, an ability to *learn*, *intentionality*, an ability to *conceptualize*, a sense of *aesthetic* and some method of *generating* artifacts.

1) Background knowledge: can be encoded in a variety of forms, including rules, associations, semantic networks, iconic representations, prototypical artifacts, a model or set of models (e.g. built from training examples), a database, statistical information, etc.

2) Learning: typically happens by some appropriate machine learning technique such as training (deep or recurrent) neural networks, building a (variable order) Markov model or other form of graphical model, inducing decision trees, forests or other forms of rules, nearest neighbor methods, etc.

3) Intentionality: can be effected by giving the system goals such as the communicating of a concept, innovation (e.g., with respect to its background knowledge), utility (e.g., with respect to some objective function), the accomplishment of some task or the realization of some state, etc. More advanced systems will typically have fewer, more abstract goals and will generate their own concrete subgoals.

4) An aesthetic: can be encoded explicitly (e.g. as a fitness function or a probability distribution over possible outcomes) or implicitly (e.g. in the generative mechanism, etc.), should likely be correlated with both background knowledge and what is learned and should be dynamic in the sense of changing over time in response to changes in either the internal or external environment.

[1]Note that the use of artifact here is abstract, and that, in particular, the artifact produced might, in fact, itself be a process.

5) Conceptualization: includes the ability to represent, manipulate and invent concepts in the domain. What makes a reasonable representation is certainly domain specific, with examples including things like rule sets, statistical models, vectors and mixed modalities. Conceptual representations should be grounded in the domain, and can be quite specific, with a simple mapping to an element of the domain (e.g., like the relationship between the genotype and phenotype in evolutionary computation), or they can be quite abstract, mapping to large subsets of the domain.

6) Generation: of artifacts can be accomplished with a random process, an evolutionary mechanism, a grammar, a generative model, etc. It is important to note that the use of the term *artifact* here is meant in a very abstract sense, so that, in particular, an artifact could be a concrete product, such as a game level, a weapon, an NPC monster, etc.; a more complex product such as a puzzle, a game mechanic, a narrative, etc.; an abstract product like a strategy to be used by an NPC monster, a quest to be assigned to a player, a level theme, etc., or even a process, such as a method for generating a game level, an aesthetic by which to evaluate a potential weapon design, or a new goal to drive the system's behavior.

Of course, in reality there are rarely crisp boundaries that clearly differentiate these mechanisms/components, and some of these are already very developed (in a computational sense) in the computational intelligence and games community, but some are still almost exclusively the purview of the human designer. It is in the consideration, development and incorporation of these others by which the games community might take a quantum leap forward.

In addition to these internal mechanisms, because the agent exists in an environment, it interacts with the environment in multiple ways, including being *taught*, *presenting* artifacts, being *inspired*, receiving *feedback* and other *influences*.

7) Teaching: often comes in the form of a supervisory signal derived from either structured or unstructured data resources, many of which are now freely available on the web; additional sources of supervisory signal can be obtained through human-labeled examples, human or computational reaction to queries, etc.

8) Presentation: of created artifacts is typically constrained by domain (e.g., visual images, written or performed music, a set of rules defining a strategy, a theorem and accompanying proof, a recipe, a process, a game); if the output is intended for human consumption, it obviously must be presented in a form appreciable by humans. In addition, the presented artifact can (and often should) be accompanied with some form of framing information (e.g., a title, a backstory, instructions, context, etc.)

9) Inspiration: is an ill-defined concept that encompasses environmental stimuli that directly or indirectly affect the agent's creative process. The most obvious source of inspiration would typically be example artifacts produced by others (e.g., for a musician, another musician's composition); however, inspiration may be found in artifacts from other domains, outside advice, etc.

10) Feedback: may take the form of immediate and direct (positive or negative) reinforcement with respect to a presented artifact, or, it may take less direct forms, such as a collection of survey responses, sale/resale value, citation or adoption data and so on.

11) Other influences: might include things like feedback from the environment about others' work (when publicly available), the behaviors, opinions, preferences and aesthetics of other creators or consumers, current events, etc.

This abstract notion of a creative agent has been instantiated in many different forms, and a variety of systems of varying degrees of sophistication and efficacy have been built by the CC community for creating artifacts in a broad range of domains, including culinary recipes [15], [16], language constructs such as metaphor [17] and neologism [18], visual art [19], [20], poetry [21], [22], humor [23], [24], advertising and slogans [25], [26], narrative and story telling [27], [28], mathematics [29] and music [30], [31].

At least some of these kinds of systems, and possibly any of them, could be incorporated directly in games, or the techniques they use could be repurposed for use in creating games or components of games. However, in the long run, the most useful takeaway for the CiG community is the set of meta-level ideas driving this research independent of a particular domain. Subscribing to these can have a lasting impact in extending the autonomy/responsibility that can be given to game systems.

The meta-level ideas referred to here are those discussed above in the treatment of an abstract creative agent and should be considered necessary, though possibly not sufficient, for a (sophisticated) CC system. Indeed, because creativity is here assumed to be an essentially contested concept, it is likely impossible to establish a sufficiency condition for computational creativity. Instead, analogous to scientific theories, a system likely may only conclusively be discredited as *not* creative, and this list is a starting point for avoiding the obvious ways this might happen (cf. some of the arguments in [13]). The avoidance of such a discrediting may seem like a dubious goal to which to aspire; however, a system that cannot easily be argued to be uncreative is likely a quite impressive one.

B. How is this different than PCG?

It is possible to consider this proposal of computational creativity as just procedural content generation on steroids, and in one sense, that may not be completely incorrect.

However, the idea of content should be expanded beyond the low-level (though certainly important and by no means trivial) components currently being tackled (e.g., maps, levels, skins and other visuals, some music, simple dialogue, etc.) to include much higher-level constructs such as complete quests, complex mechanics, goals and objectives, governing rules, ludic considerations, full narratives and eventually complete games (cf. Smith's position on the future of PCG [32]).

Perhaps the most significant improvement over traditional PCG approaches is the consideration of agent *intentionality*—CC research targets the building of systems that demonstrate deliberation or purposiveness in their creating. Accomplishing this is nontrivial, to be sure, and is a topic of ongoing research and debate. One way in which this might be approached is by building systems that share a perceptual grounding with users. Such shared grounding facilitates successful communication of intention between the system and those with which it interacts by providing a common medium for motivating and interpreting system actions. Games are an ideal domain for this kind of research, because it is possible to forge that shared grounding through the game experience (which may even facilitate types of grounding and types of intentionality not possible in the real world).

Of course, CiG research is already beginning to explore some of these possibilities, and that's why this discussion is timely—the computational creativity perspective can lend a transformative momentum to this trend.

III. Imagining the Possibilities

Several classic and recent games provide good examples of how CC ideas might be incorporated; here, the games are categorized as research or commercial, with the allowance that others may see the dichotomy somewhat differently.

A. Research Games

These games have typically been used as research platforms over many years, are well-understood and are relatively simple, in the sense that one might be able to envision an automated system for inventing something like them in its entirety.

For example, consider the class of platform- or level-based puzzle games. *Super Mario Bros.*[2] is certainly the most studied of these, likely because of both its original popularity and because of its conservative nature—it is easy to generate playable levels. Examples like *Ms. Pac-Man*[3] and *Spelunky*[4] complexify the genre by adding ghosts and bombs, respectively, that make both level design and play more challenging. Is it possible to abstract a complete description of such games in such a way that comparable new, cohesive and interesting games of the genre could be created automatically?

Or consider card-based games such as *Hearthstone*[5] and *Lords of War*[6]. Because these games lack mechanics, it is perhaps even simpler to imagine abstracting the genre and building a system that creates new complete games. The challenge is the invention of diverse card packs and their coherent incorporation into a set of gameplay and ludic rules.

Racing games like *TORCS*[7] provide another relatively simple class of games to consider for abstraction. The mechanics are well-defined and immutable (though, an interesting variation would allow mutation of the mechanics in coherent ways). Vehicle types, tracks, obstacles, race conditions, skins, and soundtracks could all be the subject of CC intervention, and while the basic ludic principle is to finish first, perhaps even this could be tampered with in interesting ways.

First person shooters like the very popular *Unreal Tournament*[8] and strategy games like *Starcraft*[9] and their variants and expansions introduce multi-player, tactical decision making, and real-time considerations as well as scale and additional complexity issues. Perhaps the most obvious focus here is the development of sophisticated and believable AI NPCs, but not using traditional tricks that allow NPCs game-knowledge not available to normal players (cf. the comment below about *The Flame in the Flood*[10]).

Games like *Galactic Arms Race*[11] and *Neuro-Evolving Robotic Operatives (NERO)*[12] are interesting in that they might be described as exploring some of the themes taken up here in a non-cognitive way—by using neuroevolution to procedurally generate content that dynamically responds to the game environment.

Other types of research games exist, of course, and the interactive fiction game *Façade*[13] deserves a special mention as a very different kind of "game", that is in some ways much more complex than those mentioned above and is also in some ways a true pre-cursor to the idea of CC in games. (A more recent version of this idea, *Versu*[14], introduces some groundbreaking work on social modeling and how it affects narrative and in particular the progression of a group conversation.) Imagine the core engine for Façade coupled with the ability to potentially invent any human-drama-based interactive fiction, based on current events, online novels, movies, etc.

B. Commercial Games (AAA and Indie)

Commercial games have a high "cool" factor, but their implementations are usually opaque, and therefore it is difficult to evaluate their process—there is always the possibility of smoke and mirrors, but if it works, it works—their goals are different than a research community's (though hopefully they are informed by such goals). Given that, what follows are some potential CC contributions to such games.

The Flame in the Flood (a river journey) and *No Man's Sky*[15] (a mind-bogglingly large universe) and, to a lesser extent *Secret Habitat*[16] (a series of art galleries), offer impressive PCG of the main game content. However, in the first case, NPC AIs are supported by dynamic markup of generated terrain, which would be considered "cheating" in the CC sense, and in the latter two, there are no NPCs. CC techniques might facilitate the introduction of believable NPCs that inhabit the PCG worlds and thereby provide a cohesive plot (of sorts).

At the other end of the spectrum are games with very strong NPC AIs, with examples including Incognita from *Invisible,*

[2]Nintendo, 1985
[3]Bally/Midway Manufacturing, 1982
[4]Mossmouth, 2008
[5]Blizzard Entertainment, 2014
[6]Black Box Games Publishing, 2012
[7]Eric Espié and Christophe Guionneau, 1997
[8]Epic Games and Digital Extremes, 1999
[9]Blizzard Entertainment, 1998
[10]The Molasses Flood, 2016
[11]Evolutionary Games, 2010
[12]Neural Networks Group, CS Dept., UT Austin, 2005
[13]Michael Mateas and Andrew Stern, 2005
[14]Richard Evans and Emily Short
[15]Hello Games, 2016
[16]Strangethink, 2014

Inc.[17], the Xenomorph from *Alien: Isolation*[18] and Elizabeth from *Bioshock: Infinite*[19]. These AIs might be made even more believable with a focus on intentional novelty and utility.

Adventure games such as *Assassin's Creed*[20], *Sunset Overdrive*[21] and *Dying Light*[22] are hailed for their visuals, their combat systems, their mechanics, etc. while at the same time being criticized for their relatively weak stories. Of course all games do not need to be all things to all people, but here is another example of a subcomponent of the game that might be co-opted by a CC system to positive effect.

Other games of interest here might include *Total War: Rome II*[23] for its impressive incorporation of Monte Carlo tree search (discussed in the next section) and *Cities: Skylines*[24] a building/planning simulation game that might be made more game-like if it could pose interesting problems to be solved.

IV. Computational Creativity in Games

As mentioned above, the field of computational creativity is already beginning to produce working systems that themselves produce artifacts that could be used as components of games. However, the bigger vision is, of course, the building of a system that can contribute significantly to the creation of complete games (or, in the extreme, autonomously create complete games). A good deal of recent work has begun to explore various approaches to this for simple games [33]–[38], with the most explicit example probably being Cook's ANGELINA system [39], [40], so this is of course not a novel idea; however, the widespread adoption of CC ideas will help move this effort forward significantly.

The first necessity is some way to describe a game in the abstract, and many attempts at this exist, including the classic *Zillions of Games* [41], the game description language [42] used in the AAAI general game playing competition, the game description language used in the Ludi system [33], the recently developed puzzlescript [43] and a general video game description language [44].

Now, imagine something like these description languages that admit the description of a space of possible games in an abstract, hierarchical manner, as suggested in Figs 2, 3, and 4. Each level of abstraction consists of a set of design choices, and the hierarchical organization allows the exploration of the space by traversal of a tree, as in Figure 5. Making choices at each node in the tree corresponds to design decisions, both general and specific for creating a particular game, and the leaf nodes in the tree correspond to completely specified games, given the representation/description language. One obvious possibility for exploring such a tree would be some variation on Monte Carlo tree search (MCTS), which has become widely adopted in the CiG community [45]. (In fact, something like this has been done very recently on a limited scale, for the creation of platformer levels [46].)

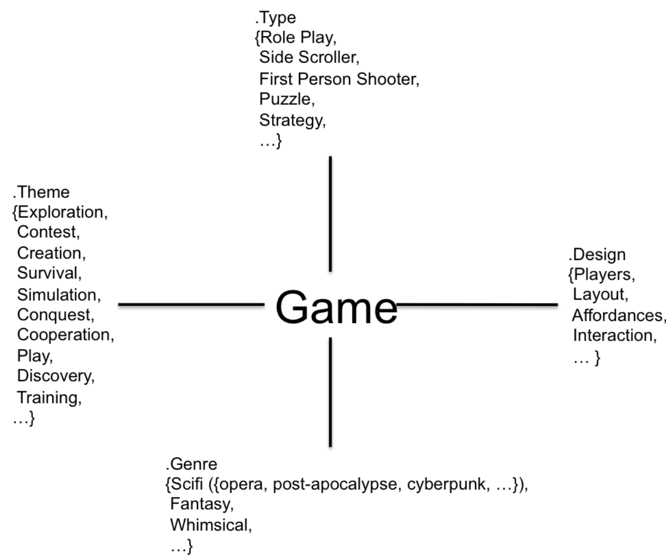

Fig. 2. An abstract (representative, incomplete) model of a *game*. A game has *theme*, *genre*, *type* and *design* elements (among other things), and each of these elements can take different values, each of which can be expanded with further detail at a lower level of the abstraction (see Fig. 3).

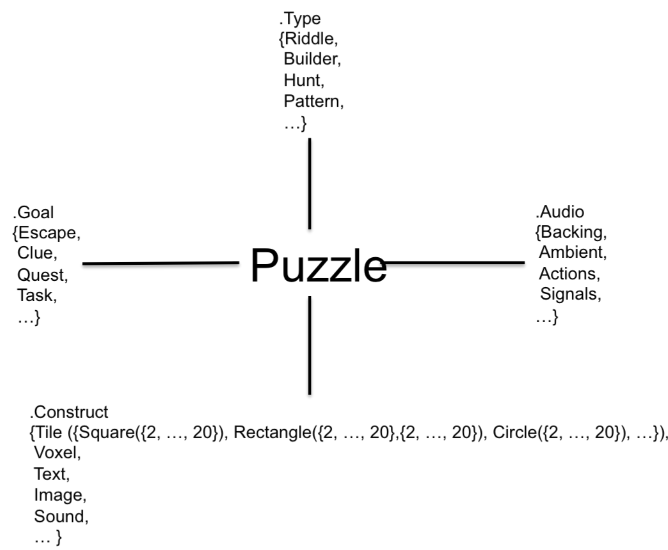

Fig. 3. An abstract (partial) model of a *puzzle*-game. A second level of abstraction specializes the game-type puzzle, which itself has associated elements *goal*, *type*, *audio* and *construct*. Again, each of these elements can be expanded with further detail at a lower level of the abstraction (see Fig. 4).

In traditional MCTS, nodes in the tree represent game states, and leaf nodes represent completed games, with an outcome (win or lose, or possibly draw). A path from a node to a leaf represents a sequence of moves, each resulting in a new game state further down the tree, until the leaf is reached and the game is completed. Following one of these paths is called a *playout*, and the search involves making many playouts and

[17] Klei Entertainment, 2015
[18] Creative Assembly, 2014
[19] Irrational Games, 2013
[20] Ubisoft, 2007
[21] Insomniac Games, 2014
[22] Techland, 2015
[23] Creative Assembly, 2013
[24] Colossal Order, 2015

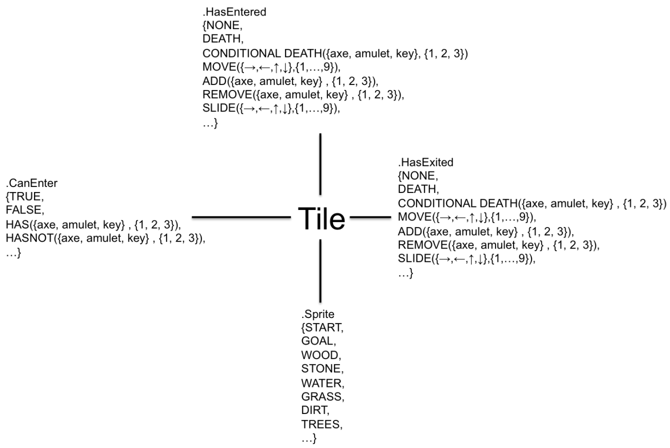

Fig. 4. An abstract model of a *tile* for a tile-based puzzle-game. This third level of the abstraction provides concrete detail that can be implemented—a tile has three parameterized functional characteristics: *.CanEnter*, *.HasEntered* and *.HasExited* and an associated *sprite*, which can take any of several visual values (of course in general, the levels of abstraction could continue).

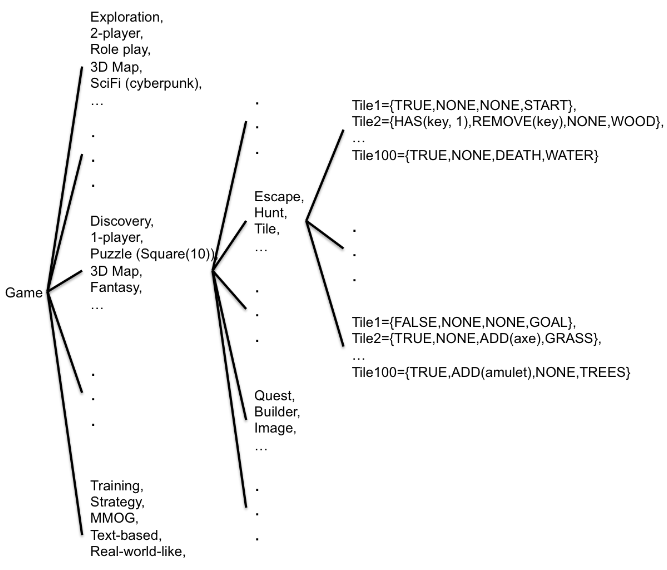

Fig. 5. Search tree for exploring the hierarchical space of possible games (partially) described by the abstractions of Figs 2, 3, and 4. Each node in the tree represents a partial game design, and each branch represents a particular design decision that further specifies the game. Leaf nodes represent fully specified games.

collecting statistics on their results (wins/losses/draws) and backpropagating those statistics up the tree. Eventually, based on these collected statistics, a move is actually made, and the game progresses. In this search, the objective is to win the game, and the search explores different paths down the tree in an effort to find a "good" move to make, which increases the chances of meeting the objective. Each time the search reaches a leaf, the result is known and the relative quality of the moves in the path can be updated.

In searching the proposed game creation tree of Fig. 5, the objective is to create a "good" game, and the search explores different paths down the tree in an effort to find one of these "good" games. Each node in the path is a design decision, with the idea being to make "good" decisions that lead to the creation of a "good" game, in the same way that making good moves leads to winning a game. In other words, the game creation process can be thought of as a meta-game of sorts; however, it is not a competitive game—there is no explicit opponent nor explicit concept of winning—another example of the difference between traditional AI problems and CC problems. Since there is no objective concept of winning, how can the leaf nodes of this tree be evaluated? How can it be determined when one of them represents a "good" game (or piece of a game, in the case of creating a soundtrack or mechanics or visuals, etc.)?

This, it turns out, is a really difficult question, that can be posed in a variety of ways. For example, consider the idea of using an inductive logic program (ILP) to represent a quest and the (pre- and post-) conditions that apply to various stages of that quest. It is not perhaps that difficult to think about how one might do this (though for a complex quest, this may not be trivial, either), and this might, in fact, be an interesting way to approach the problem of realizing a game quest. Having taken that step, it might not be (too) difficult to then take the next step—considering how to build a system that could generate an ILP to represent a quest. However, what determines if the generated quest is a good one? Is there some way to formalize an aesthetic for (ILP-encoded) quests, so that the system has a method for evaluating its own output? This is a difficult question, of course, and the discovery and implementation of useful aesthetics for various domains is one of the key points of study in the field. However, even this is not yet satisfactory. For, given a formalized aesthetic for "good" quests, the system now can (in principle) create such quests, but it will not create other quests that may be considered "good" under some other defendable aesthetic. In other words, what is really wanted is a system that can (also) create an aesthetic, and for that, what is required is an aesthetic for aesthetics (about quests in this particular example).

This meta-level reasoning has its finger prints all over computational creativity, and once recognized, it immediately begs the question, "what about an aesthetic for aesthetics for aesthetics?" Ad infinitum. This is a fair question, but for now a satisfactory accounting for one additional level of meta will constitute serious progress. It is also worth mentioning here that it is not clear that human creators are capable of this type of higher-order meta-evaluation either, so if artificial systems are limited to "only" one meta-level of evaluation, they may be still in good company.

Now, assume the existence of a system for exploring the (tree-structured) space of possible games, that the system knows what constitutes a "good" game, and even that it knows how to "change its mind" about what makes a game "good" in a defensible way. There is *still* work to do, in the sense that the search space has been given to the system in the first place and is immutable from the system's point of view. What if the system could invent new branches for the tree? This could be something as simple as adding an additional item to associate with the *Tile.CanEnter.HAS* property of Fig. 4; or as complex as adding a new *Game.Type* in Fig. 2; or even inventing an additional element, *Game.NewElement*, analogous to the existing elements: *Game.Theme*, *Game.Type*, *Game.Design*, and *Game.Genre*. This is another form of meta-creativity in which the space to be explored is the space of all possible trees that define spaces of game representations. But then, what kind of aesthetic would guide *that* search?

Of course, MCTS may not be the right approach to this general problem of exploring the space of possible games. The search tree envisioned here is possibly relatively shallow, and possibly very broad (even likely infinitely so). Is MCTS the best approach for this shape of tree, let alone this kind of search problem? There are certainly other traditional forms of search that could also be adapted as a possible mechanism for exploring this space, and it is possible that the shape of the tree might be changed significantly by changing the abstraction. Perhaps a tree isn't even the appropriate structure for describing this space. Perhaps it is even possible to search the space of abstractions for good ones for searching for games. Many of these considerations are related to interesting foundational work by Wiggins [47]; however, this is likely beyond the realm of (immediate) interest for the CiG community.

One additional suggestion that deserves further exploration is the idea of building games in which computational creativity is the main feature of the game. It is not yet clear what this might entail, but a similarly intriguing idea has been suggested for games featuring AI [48], and as argued there, it seems likely that taking full advantage of CC as the main event will require re-thinking at least some accepted ideas about games and may open the way for entirely new types of game.

A. Evaluation

As with any creative endeavor, it is not sufficient that the creator believe that the result is novel and useful, though this is certainly necessary; other creators or consumers or other "gatekeepers" of the field must also attribute these values to the result. This sort of external feedback can be measured in any number of ways, and certainly gross measures such as sales ranking, hours played and other adoption/popularity metrics represent something of a bottom line when it comes to games. However, there is another sense of external measurement that is also critical to the advancement of computational creativity as a field and is somewhat more difficult to assess—the "creativity" of the system. This is a difficult question, made the more so by the lack of a concrete definition for creativity. Still, some progress has been made and varying proposals suggest ways of dealing with the problem, including suggesting metrics for quantifying various qualities of system output [49]; an abstract ontology of behaviors potentially demonstrable by a system [50]; a proposal for a standardized evaluation methodology that uses case-specific requirements-based testing [51]; and a spectrum of prototypical abstract landmark algorithms that characterize varying levels of system intentionality in producing novelty and utility [52].

V. Conclusion

This paper argues that the logical next step for computational intelligence and games is the incorporation of computational creativity in games. It gives a necessarily brief overview of the field of computational creativity, imagines some initial uses for it in contemporary games, and explores the beginnings of a few ideas for its incorporation into the next generation of games and beyond. Many more questions have been raised than have been answered, with the goal being to arouse interest in the CiG community reciprocal to that which has begun to grow in the CC field. For those whose interest is piqued, a good resource for all things CC, and in particular an extensive and growing bibliography can be found at `www.computationalcreativity.net`.

Computational creativity is itself basically domain agnostic. However, since it is very difficult to make an effective study in the abstract, it is typical to settle for being agnostic in the statistical sense of "averaging" over many domains. This provides interesting opportunities for collaboration between CC researchers and researchers in a particular domain to which CC may be applied, and that, in turn, can strengthen research agendas in both fields. Here the call by Liapis, et al. for such a collaboration between the CiG and CC communities is reciprocated as being perhaps the most natural of possible collusions. Though the extreme possibility of a fully autonomous system that creates complete games has been considered, in reality most games are now so complex that they are built by (often large) teams of creative individuals, and so the more likely positive outcome would be a system or systems that can participate in that collaborative process as true co-creative members of such a team.

Acknowledgment

Thanks to Jim Whitehead, Michael Mateas, Michael Cook, Antonios Liapis and Julian Togelius for many useful suggestions that helped guide the development of this manuscript.

References

[1] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, "Search-based procedural content generation: A taxonomy and survey," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 3, pp. 172–186, 2011.

[2] M. Hendrikx, S. Meijer, J. V. D. Velden, and A. Iosup, "Procedural content generation for games: A survey," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 9, no. 1, p. 1, 2013.

[3] R. Hunicke and V. Chapman, "AI for dynamic difficulty adjustment in games," in *AAAI Workshop on Challenges in Game Artificial Intelligence*, 2004.

[4] G. Chanel, C. Rebetez, M. Betrancourt, and T. Pun, "Emotion assessment from physiological signals for adaptation of games difficulty," *IEEE Transactions on Systems Man and Cybernetics, Part A*, vol. 41, no. 6, pp. 1052 – 1063, 2011.

[5] A. Drachen, "Crafting user experience via game metrics analysis," presented at NORDICHI 2008, Lund, Sweden, 2008.

[6] G. N. Yannakakis and J. Togelius, "Experience-driven procedural content generation," *IEEE Transactions on Affective Computing*, vol. 2, no. 3, pp. 147–161, 2011.

[7] C. Bauckhage, A. Drachen, and R. Sifa, "Clustering game behavior data," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 7, no. 3, pp. 266–278, 2015.

[8] N. Ducheneaut, N. Yee, E. Nickell, and R. J. Moore, "Alone together?: Exploring the social dynamics of massively multiplayer online games," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2006, pp. 407–416.

[9] A. Liapis, G. N. Yannakakis, and J. Togelius, "Computational game creativity," in *Proceedings of the 5th International Conference on Computational Creativity*, 2014, pp. 46–53.

[10] K. Compton and M. Mateas, "Casual creators," in *Proceedings of the 6th International Conference on Computational Creativity*, 2015, pp. 228–235.

[11] W. B. Gallie, "Essentially contested concepts," *Proceedings of the Aristotelian Society*, vol. 56, pp. 167–198, 1956.

[12] S. Colton and G. Wiggins, "Computational creativity: The final frontier?" in *Proceedings of the 20th European Conference on Artificial Intelligence*, 2012, pp. 21–26.

[13] S. Colton, "Creativity versus the perception of creativity in computational systems," *Creative Intelligent Systems: Papers from the AAAI Spring Symposium*, pp. 14–20, 2008.

[14] D. Ventura, "The computational creativity complex," in *Computational Creativity Research: Towards Creative Machines*, T. R. Besold, M. Schorlemmer, and A. Smaill, Eds. Atlantis Press, 2015, pp. 65–92.

[15] R. Morris, S. Burton, P. Bodily, and D. Ventura, "Soup over bean of pure joy: Culinary ruminations of an artificial chef," in *Proceedings of the 3rd International Conference on Computational Creativity*, 2012, pp. 119–125.

[16] L. Varshney, F. Pinel, K. Varshney, A. Schorgendorfer, and Y.-M. Chee, "Cognition as a part of computational creativity," in *Proceedings of the 12th IEEE International Conference on Cognitive Informatics and Cognitive Computing*, 2013, pp. 36–43.

[17] T. Veale and Y. Hao, "Comprehending and generating apt metaphors: A web-driven, case-based approach to figurative language," *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, vol. 2, 2007.

[18] M. R. Smith, R. S. Hintze, and D. Ventura, "Nehovah: A neologism creator nomen ipsum," in *Proceedings of the 5th International Conference on Computational Creativity*, 2014, pp. 173–181.

[19] D. Heath, D. Norton, and D. Ventura, "Conveying semantics through visual metaphor," *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 2, p. article 31, 2014.

[20] S. Colton, "The Painting Fool: Stories from building an automated painter," in *Computers and Creativity*, J. McCormack and M. d'Inverno, Eds. Springer-Verlag, 2011, pp. 3–38.

[21] J. M. Toivanen, H. Toivonen, A. Valitutti, and O. Gross, "Corpus-based generation of content and form in poetry," in *Proceedings of the 3rd International Conference on Computational Creativity*, 2012, pp. 175–179.

[22] H. G. Oliveira, "PoeTryMe: a versatile platform for poetry generation," in *Proceedings of the ECAI 2012 Workshop on Computational Creativity, Concept Invention, and General Intelligence*, 2012.

[23] O. Stock and C. Strapparava, "HAHAcronym: Humorous agents for humorous acronyms," *Humor - International Journal of Humor Research*, vol. 16, no. 3, pp. 297–314, 2003.

[24] K. Binsted and G. Ritchie, "A symbolic description of punning riddles and its computer implementation," in *Proceedings of the Association for the Advancement of Artificial Intelligence*, 1994, pp. 633–638.

[25] C. Strapparava, A. Valitutti, and O. Stock, "Automatizing two creative functions for advertising," in *Proceedings of 4th International Joint Workshop on Computational Creativity*, 2007, pp. 99–105.

[26] G. Özbal, D. Pighin, and C. Strapparava, "BRAINSUP: Brainstorming support for creative sentence generation," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 2013, pp. 1446–1455.

[27] M. O. Riedl and R. M. Young, "Narrative planning: Balancing plot and character," *Journal of Artificial Intelligence Research*, vol. 39, no. 1, pp. 217–268, 2010.

[28] R. Pérez y Pérez and M. Sharples, "Three computer-based models of storytelling: BRUTUS, MINSTREL and MEXICA," *Knowledge-Based Systems*, vol. 17, no. 1, pp. 15–29, 2004.

[29] S. Colton, A. Bundy, and T. Walsh, "HR: Automatic concept formation in pure mathematics," in *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, 1999, pp. 786–791.

[30] B. Houge, "Cell-based music organization in Tom Clancy's EndWar," in *Proceedings of the 1st International Workshop on Musical Metacreation*, 2012.

[31] F. Pachet and P. Roy, "Non-conformant harmonization: the real book in the style of Take 6," in *Proceedings of the 5th International Conference on Computational Creativity*, 2014, pp. 100–107.

[32] G. Smith, "The future of procedural content generation," in *Experimental Artificial Intelligence in Games: Papers from the AIIDE Workshop*, 2014, pp. 53–57.

[33] C. Browne and F. Maire, "Evolutionary game design," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 1, pp. 1–16, 2010.

[34] C.-U. Lim and D. F. Harrell, "An approach to general videogame evaluation and automatic generation using a description language," in *Proceedings of the IEEE Conference on Computational Intelligence and Games*, 2014, pp. 1–8.

[35] T. S. Nielsen, G. A. B. Barros, J. Togelius, and M. J. Nelson, "Towards generating arcade game rules with VGDL," in *Proceedings of the IEEE Conference on Computational Intelligence and Games*, 2015, pp. 185–192.

[36] A. Khalifa and M. Fayek, "Automatic puzzle level generation: A general approach using a description language," in *Computational Creativity and Games Workshop*, 2015.

[37] D. M. LeBaron, L. A. Mitchell, and D. Ventura, "Intelligent content generation via abstraction, evolution and reinforcement," in *Experimental AI in Games: Papers from the AIIDE 2015 Workshop*, 2015, pp. 36–41.

[38] J. Gow and J. Corneli, "Towards generating novel games using conceptual blending," in *Experimental AI in Games: Papers from the AIIDE 2015 Workshop*, 2015, pp. 15–21.

[39] M. Cook, S. Colton, and J. Gow, "The ANGELINA videogame design system, part I," *IEEE Transactions on Computational Intelligence and AI in Games*, 2016, to appear.

[40] ——, "The ANGELINA videogame design system, part II," *IEEE Transactions on Computational Intelligence and AI in Games*, 2016, to appear.

[41] J. Mallett and M. Lefler, "Zillions of games," 1998. [Online]. Available: http://www.zillions-of-games.com

[42] N. Love, T. Hinrichs, D. Haley, E. Schkufza, and M. Genesereth, "General game playing: Game description language specification," Stanford, LG-2006-01, 2006.

[43] S. Lavelle, "Puzzlescript." [Online]. Available: http://www.puzzlescript.net/

[44] T. Schaul, "A video game description language for model-based or interactive learning," in *Proceedings of the IEEE Conference on Computational Intelligence in Games*, 2013, pp. 1–8.

[45] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of Monte Carlo tree search methods," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 1, pp. 1–43, 2012.

[46] A. Summerville, S. Philip, and M. Mateas, "MCMCTS PCG 4 SMB: Monte Carlo tree search to guide platformer level generation," in *Experimental AI in Games: Papers from the AIIDE 2015 Workshop*, 2015, pp. 68–74.

[47] G. A. Wiggins, "A preliminary framework for description, analysis and comparison of creative systems," *Knowledge-Based Systems*, vol. 19, pp. 449–458, 2006.

[48] I. Horswill, "Game design for classical AI," in *Experimental Artificial Intelligence in Games: Papers from the AIIDE Workshop*, 2014, pp. 28–34.

[49] G. Ritchie, "Some empirical criteria for attributing creativity to a computer program," *Minds and Machines*, vol. 17, pp. 67–99, 2007.

[50] S. Colton, J. Charnley, and A. Pease, "Computational creativity theory: The FACE and IDEA descriptive models," in *Proceedings of the 2nd International Conference on Computational Creativity*, 2011, pp. 90–95.

[51] A. Jordanous, "A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative," *Cognitive Computation*, vol. 4, no. 3, pp. 246–279, 2012.

[52] D. Ventura, "Mere generation: Essential barometer or dated concept?" in *Proceedings of the 7th International Conference on Computational Creativity*, 2016, to appear.