Co-creativity in Videogame Level Design

Pedro Miguel Sanches Torres Lucas
Instituto Superior Técnico,
Campus Taguspark,
Av. Prof. Doutor Cavaco, Silva
2744-016 Porto Salvo
Lisboa, Portugal
pedro.lucas@ulisboa.tecnico.pt

Abstract

Motivation for this work comes from the belief that there is some untapped potential in having a computer work as a *colleague* with the videogame level designer as a source of creative stimulus, instead of simply being a tool, in order to achieve overall more creative results than those obtained from solo development. The proposed solution consists of a co-creative level design tool, focused on fostering creativity by allowing human and computer to work together in producing content using the Legend of Grimrock 2 Level Editor, exploring the digital "peer" paradigm. Its interface can be used by the designer to preview generated suggestions and orient its behavior. Suggestions are generated and iteratively evolved by two genetic algorithms and can be guided by the designer on different domains: *innovation*, *objective* and *user map*. Innovation seeks to generate content different from the designer's level, objective seeks to generate content respecting specific layout guidelines and user map seeks to generate content strongly based on the designer's level. Results showed this approach to be promising since it takes into account the smaller nuances of the co-creative interaction as a source of a positive influence. Outlined improvements such as a better way to support designer-specific patterns, selection context or on-demand suggestion generation helped set a direction for future work. We concluded that through an intuitive interface, flexible and adjustable behavior and an agile interaction process, we were able to provide some interesting contributions to the quality of the co-creative level design process.

Introduction

Because most people face computers as facilitating tools, they disregard their potential in providing valuable contributions to creative processes, only associated with human beings. By exploring the paradigm of the digital "peer" in co-creative activities, we seek to enrich the human-computer interaction in such cases, by having the computer work as a *colleague* with the video game level designer. Work written by Lubart (Lubart 2005) states several advantages of this ambitious computer model approach.

There is a respectable amount of work and research on the topic of computational creativity. Among all these, we to draw our attention to the work of Edward DeBono on the Lateral Thinking theory (De Bono 1977), Adam Vile's work on Diagrammatic Reasoning (Vile and Polovina 1998) and the experiments of Yannakakis *et al.* with Mixed-Initiative Co-Creativity (Yannakakis and Alexopoulos 2014) programs, whose behavior is based on a theory which merges both lateral thinking and diagrammatic reasoning.

In the context of videogames, Procedural Content Generation is an often used set of techniques to quickly generate generally correct and useful content, within certain constraints. Some videogames support the creation of user content such as the design of levels. We believe level design in such cases is a good example of an activity which would benefit from a computer-assisted co-creative tool. In this sense, we mainly analyse the finding of Liapis *et al.* in their tool, the Sentient Sketchbook (Liapis, Yannakakis, and Togelius 2013) and the findings of Smith *et al.* (Smith, Whitehead, and Mateas 2011) in the tool Tanagra. We take into consideration both their results, conclusions and observations regarding their respective implementations.

We then present our solution and the chosen approach in the light of the insight acquired from related works, describing how and why this has been implemented. In order to validate the usability and utility of this approach we conducted an evaluation with professional level and game designers and non-professional participants and present its results.

Background

Creativity

We start by addressing the importance of creativity to this work, how can we describe it, what do we know of it and what has been accomplished in the field of artificial intelligence and the theory behind it.

Perhaps one of the most influential individuals in the area of creativity is Margaret Boden. Her work (Boden 2003) introduced and explained the several existing forms of creativity: *combinational*, *exploratory* and *transformational* and the distinction between *psychological* and *historical* creativity in regards to an individual. Additionally, she presents how these forms of creativity influence *conceptual spaces*, structured styles of thought, by combining ideas within them, exploring new ones or even the need to transform themselves to reach otherwise impossible ideas.

On another stage, Boden addresses the contributions of computers as a source of creativity though the use of artificial intelligence and how they fit the previously described theories. Existing computer models of creativity are capable performing combinational, exploratory and transformational creative processes and she mentions a couple of interesting implementations on all levels.

The work of Edward DeBono (De Bono 1977) on describing several creative processes which is also relevant in the context of our work. We draw our attention towards the definition of lateral thinking and how he distinguishes it from the more common process of vertical thinking.

Procedural Content Generation

Videogames, being an area transversal to many others, is a field which has much to gain from such computational models, whether it is for the player or the game developer. Currently, Procedural Content Generation offers mechanisms to facilitate the creation of various types of content for videogames. From audio to visual content, PCG is a widely used and relatively inexpensive way to quickly enrich the player experience in games. The use of such techniques is plausible and has already been subject of some investigation regarding its use as creativity-enhancing computer models, in cases such as the creation of videogame levels.

We refer to the taxonomy used by Togelius *et al.* in (Togelius et al. 2011), where PCG techniques are evaluated according to different aspects:

- Online vs Offline
- Necessary vs Optional Content
- Random seed vs Parameter vectors
- Stochastic vs Deterministic Generation
- Constructive vs Generate and Test

This alternative taxonomy allows us to classify and place different PCG techniques in a continuum, rather than separating them by behaviors.

Creativity-enhancing computer models

As a way to foster the creative process, Lubart identifies several categories (Lubart 2005) on which the computer can help an individual in this way:

1. The management of creative work
2. Communication between individuals collaborating on creative projects
3. The use of creativity enhancement techniques
4. The creative act through integrated human-computer cooperation during idea production

From the mentioned categories, Lubart also presents how they can be translated into four lines of thought about the role of computers in creativity enhancement: computer as a *nanny*, computer as *pen-pal*, computer as a *coach* and computer as a *colleague*. This last one being the most ambitious and perhaps the most interesting to explore, considering how Liapis *et al.* and Smith *et al.* focus on this paradigm in the Sentient Sketchbook (Liapis, Yannakakis, and Togelius 2013) and the Tanagra (Smith, Whitehead, and Mateas 2011) tool, respectively. The main purpose of these tools is to provide a rich mixed-initiative co-creative experience in order to foster the level designer's creativity.

Search for creativity

Both the Sentient Sketchbook and Tanagra tools have a couple of aspects in commons, but at a higher level, both attempt to provoke the creativity of their user by trying to generate novel and valuable content. The way they achieve this is different and, thus, we better identify with the Sentient Sketchbook approach. Remembering the observations made by Margaret Boden(Boden 2003), the problem of creativity can be translated at the *exploratory level* into a *constrained optimization search* conducted on the conceptual space.

Currently, there are several type of algorithms in the field of artificial intelligent which can, and are, used to solve problems through search methods. Interestingly, the Sentient Sketchbook makes use of constrained optimization search algorithms in its implementation to realize its transformational creativity. In this sense, we have found Evolutionary Algorithms to be an interesting and promising approach.

Evaluating creativity

We need, however, to be able to evaluate the human-computer interaction and, not only that, the outcome of this interaction. In the particular case of Mixed-Initiative Co-Creativity (Yannakakis and Alexopoulos 2014), is not so straightforward if we consider the fact there is an ongoing interaction between human and computer which is hard to represent in the final outcome. Yannakakis better explains this as the difficulty of capturing the impact of pro-activeness of the AI on human creativity and vice-versa. Therefore, Yannakakis considers two types of evaluation when a computational creator is involved: the evaluation of the final (or intermediate) outcome and the evaluation of the co-creative process of outcomes, solutions or items. Evaluating the final outcome is easy through the use of heuristics, but the evaluation of the entire co-creative process is far from trivial as the hard part lies in enumerating all the human creativity sub-processes used.

Solution

The presented solution, named Editor Buddy, consists of a GUI based application to be used alongside the Legend of Grimrock 2 in-game Editor. Its primary goal is to foster creativity during a level design activity by presenting visual hints and suggestions iteratively, alongside the designer's work. Its interface serves two purposes: it provides the level designer with visual information as a way to stimulate his creativity and a way to guide the application's behavior towards generating more or less specific content.

The Editor Buddy behavior is defined by two genetic algorithms as well as input from the designers current work. Generated suggestions are then displayed to the designer in the Editor Buddy interface using a 2D preview. This behavior can be configured by the designer using the available interface controls. The designer can also easily export the displayed suggestion, or parts of it, to his current level.

Suggestions are evolved and presented iteratively each time the designer performs any change to his level. This quick response on the Editor Buddy's part allows the designer to consider and evaluate intermediate steps for the Editor Buddy's work, which can hold potentially useful creative value.

Computer colleague paradigm

There is a particular paradigm we are trying to portray with our solution and it is crucial to emphasize it so we are able to understand both the relation between content it generates and respective interface configuration and its iterative level design cycle. To better impersonate a digital "peer", or at least try to produce similar behaviors, our solution acts according to three different domains: *Innovation*, *Objective* and *User Map*. Because more than one domain may be taken into account at a given time, we are bound to find discrepancies between the suggestions it presents and user expectations. These may sometimes be perceived as unwanted, however, we must keep in mind the ultimate purpose of this work is to explore the ability to foster creativity, in this case by allowing multiple, potentially useful suggestions to emerge.

In this sense, the designer is still able to tweak the Editor Buddy's configuration, much like asking a real person their opinion on a more particular topic. For each action the designer performs on the level editor, the Editor Buddy takes that into consideration and generates the following suggestion, ultimately dependant on its configuration, displaying it back to the designer.

In the end, its still up to the designer to choose whether a particular suggestion in worth exporting or even consider. If not, disregarding the displayed suggestion is always a possibility. Figure 1 represents an example interaction between a level editor, where a designer performs any creative actions, and our solution. In theory, this concept is generally applicable to other level editors, other than the one we chose.

Legend of Grimrock 2 Editor

The game Legend of Grimrock 2 (Almost Human Ltd. 2014) is a dungeon crawler genre computer game which includes an in-game Level Editor. The LoG2 Editor, figure 2a, serves as the primary level editing tool for the designer, whereas the Editor Buddy proposes suggestions over content created inside the LoG2 Editor. In the Editor the designer is given a 2D canvas, as well as three main tools to edit the layout of his level as well as its content. The first and second tools, the Pointer and Map Element tools are used to complement the 2D tile based layout after this has been complete. Such steps were disregarded since they fell out of the scope of our work. In this sense, designers exclusively worked with the third tool, the Tile-editing tool, in order to assemble their level's layout.

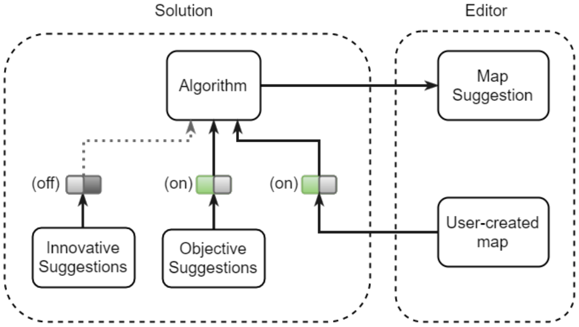


Figure 1: Interaction between the proposed solution and level editor

Editor Buddy

User interface The Editor Buddy User Interface, figure 2b, consists of these main components:

- Controls for defining the applications behavior. These include a normal and expert modes which can be toggled by the designer
- A 2D canvas where the program's generated content is displayed
- The canvas can also be used to perform a selection of the suggestion the designer wishes to export
- A set of buttons which can be used to interact with the generated suggestion
- Highlight setting which help in identifying the delta between the suggestion and the designers level

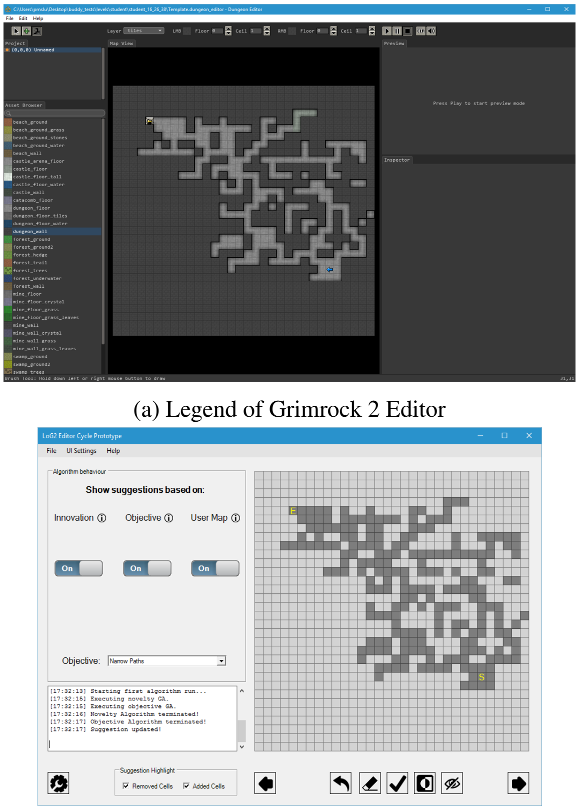
Controls are shown in the form of toggle switches as the standard mode. Alternatively, there is an expert mode where switches are replaced with sliders, to provide a way to fine-tune the Editor Buddy behavior. Specific to the Objective domain, the designer can choose a particular behavior using the appropriate combo-box element named "Objectives".

Suggestions are displayed in the 2D preview canvas and initially remain hidden to avoid over-influencing the designer. To expose the hidden part within an area, the designer can draw a selection over it in the preview canvas to reveal it.

By using a set of buttons, the designer can interact with the performed selection. This selection can also be cleared, inverted or exported to the designer's level in the LoG2 Editor using the respective buttons. Additionally, the designer can peek at the hidden part of the suggestion by pressing the visibility button.

Behavior The Editor Buddy behavior is mainly guided using two genetic algorithms and the designer's level. The mentioned domains, innovation, objective and user map, from which we can receive suggestions can be controlled using the user interface controls and relate to the genetic algorithms in the following way.

The first algorithm, related with the innovation domain, generates content that is novel, that is, suggestions produced by this algorithm are evolved to be topologically distinct from what the designer is doing, the more distinct the better. The second algorithm, related with the objective domain, generates content which adheres to certain level constraints,



(a) Legend of Grimrock 2 Editor

Innovation ⓘ On Objective ⓘ On User Map ⓘ On

Objective: Narrow Paths

```
[17:32:13] Starting first algorithm run...
[17:32:15] Executing novelty GA.
[17:32:15] Executing objective GA.
[17:32:16] Novelty Algorithm terminated!
[17:32:17] Objective Algorithm terminated!
[17:32:17] Suggestion updated!
```

☑ Removed Cells ☑ Added Cells

(b) Editor Buddy

Figure 2: LoG2 Editor and Editor Buddy user interfaces

generating characteristic level layouts, which the level designer can choose in the Editor Buddy user interface. These suggestions, however, in spite of presenting eventual alternatives to the layout of the designer's level, do not actively seek to go against what he has created. The third element which influences the Editor Buddy behavior is the designer's level itself. By generating suggestions heavily based on the layout of the designer's level, we are able to provide more contextual suggestions, in case the designer wished to further explore more refined versions of his level.

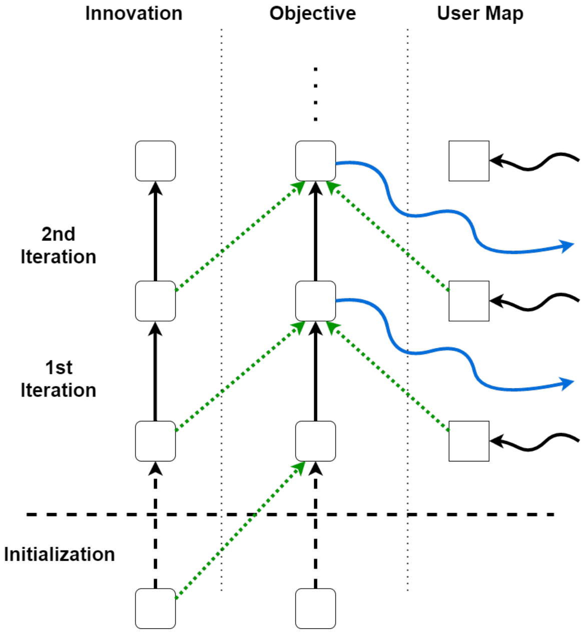
Editor integration The developed application is not a standalone program but, instead, works closely with the LoG2 level editor. All level creation and editing made by the designer is done in the level editor side. The only exception being when and if the designer wishes to replace his work, or a part of it, with the Editor Buddy's suggestion, using the available interface tools. The Editor Buddy handles the automatic saving of the level, so when there is anything new on the designer's end, the Editor Buddy's algorithms begin a new iteration, taking into account the current configuration. When exporting changes to the designers level, the procedure is somewhat the inverse. We take the selected suggestions 2D layout and write it in the appropriate format so the Editor is able to load it. The Editor Buddy also handles the automatic reload of the updated map file, so the level editor view reflects the performed changes.

Execution cycle The Editor Buddy execution cycle comprehends two distinct moments: an initialization phase, where algorithm populations are first generated and a post-initialization phase, or suggestion phase, where the previous algorithm populations are iteratively evolved and from which provide the suggestions presented to the designer.

After the initialization phase, on the suggestion phase, each interaction the designer performs on the LoG2 Editor, effectively resulting on a modification his level, creative or destructive, triggers a new evolutionary run in the Editor Buddy algorithms. After the algorithm run terminates the suggestion is updated and shown to the designer who can then choose to select and export it into his level.

Before executing a new algorithm run, however, the Editor Buddy first evaluates the state of its innovation, objective and user map controls, ensuring the generated content is guided by the interface configuration.

Figure 3 represents how the Editor Buddy algorithm execution unfolds each time the designer performs a change in their level's layout.



(a) Cycle diagram

Element	Description
□	Individual chromosome
▢	Population of inviduals
- - - →	Initialization run
⟶	Algorithm run
········→	Chromosome transfer
←∼	Designer input
∼→	Displayed suggestion

(b) Element caption

Figure 3: Editor Buddy execution cycle diagram

Evaluation

The main goal of this study was the evaluation of the Editor Buddy utility as well as its efficiency. We define utility, in this case, as the ability to contribute, direct or indirectly, with useful content. It remains, however, at the designer's discretion the definition of usefulness. This brings us to the second topic of evaluation, the application's ability to be configured in such a way as to produce coherent and useful content, according to the needs or desires of the designer at a given time.

Evaluation tasks

For this evaluation, we asked participants to create two distinct levels using the LoG2 Editor and the Editor Buddy to support them. They were given a template layout in the beginning of the first task as illustrated by figure 4.

The first task goals were:

- The resulting level was required to have **at least one valid path** from the starting point to the end point. Without this, the level would not be playable
- The level layout must contain **narrow halls**, like a maze, since monsters with a charge attack will be placed in the level, later during development

When changing to the second task, participants were asked to cease their current work on the first task, near the end, and start working on their second task. This allowed us to understand in which case the Editor Buddy behaved better.

The second task goals were:

- The resulting level was required to have **at least one valid path** from the starting point to the end point. Without this, the level would not be playable
- Their level layout needed to be altered to have **rooms** connected by halls, because new monsters would be placed on the level and they were too big to fit narrow halls and would, instead, reside inside these rooms

A time limit was not enforced for these task, ans such was not needed in the end. Still, we considered 20 minutes to be a reasonable amount of time for each task, as a time limit.

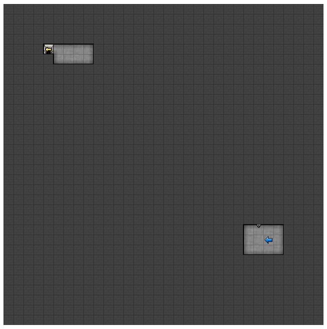


Figure 4: Evaluation task template

Participant description

Two types of participants were involved in this study: professionals and non-professionals. Sampling method was mainly a purposeful sampling, where on one side we had professionals or expert individuals, game or level designers, and on the other side we had non-professionals or inexperienced individuals, computer science students from a games course. This allowed for identification of prominently lacking or redundant or obsolete features.

Data collection methods

In this study, data was collected using a **questionnaire**, **participant observation** and a **structured interview**.

For quantitative data collection, we used a linear scale questionnaire in order to roughly draw a measure of the level of usability for the Legend of Grimrock 2 Editor and the Editor Buddy interfaces, as well as the amount of user expectation versus generated content discrepancy.

Observation was conducted in the form of participant observation and screen recording during the entire test. We chose this method because we were interested in the participant's actions during level design activities including their interaction with both the in-game Editor and the Editor Buddy instances as well as any type input in-between.

After participants were finished with the evaluation tasks, a structured interview was conducted. The goal for these interviews was to draw out patterns from common concepts and insights regarding the personal experience of each participant with both the interface interaction and the generated suggestions.

Results

Thanks to the feedback of participants, our study allowed us to identify several important aspects and limitations in the developed software.

Result presentation

From our evaluation, we were able to collect the following data using questionnaires, observation and interviews.

Questionnaire Regarding the usability of the Editor and the Editor Buddy, the majority of participants found them easy to use and their integration was intuitive and satisfactory. Table 1 shows results from the usability section of the questionnaire, questions 4 through 7, where:

- Question 4: It was easy to edit the layout of my level using the Legend of Grimrock 2 Editor
- Question 5: It was easy to configure the Buddy behavior using the available interface controls
- Question 6: It was easy to make and edit a selection of the map suggestion made by the Buddy
- Question 7: There were no communication issues between the Editor and the Buddy

Regarding the behavior of the Editor Buddy, the following charts, figures 5, 6 and 7, compare the usefulness of content

LoG2 Editor and Editor Buddy usability				
Scale	Question 4	Question 5	Question 6	Question 7
Totally disagree	0	0	0	0
Somewhat disagree	0	0	0	0
Neither agree nor disagree	0	0	1	1
Somewhat agree	2	2	1	2
Totally agree	4	4	4	3

Table 1: Questionnaire usability results

generated by each behavior switch or slider to the results expected by the designer from those individual controls. These results pertain the algorithm behavior section from the questionnaire.

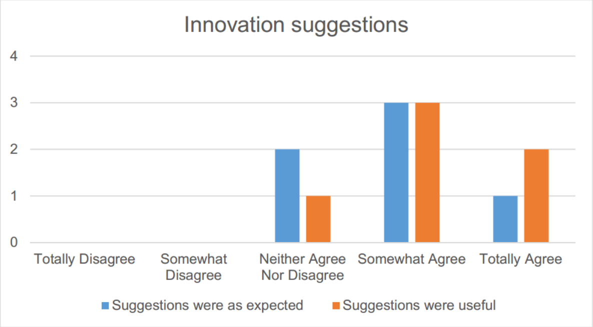


Figure 5: Innovation control expectation vs usefulness

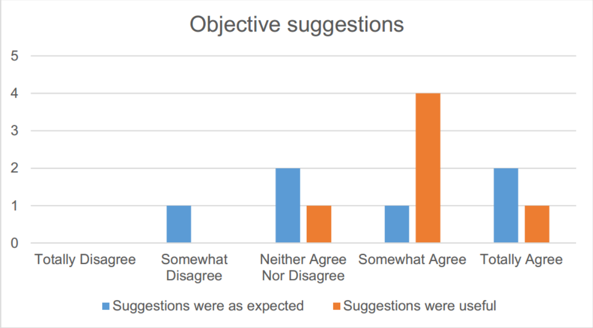


Figure 6: Objective control expectation vs usefulness

Observation From observations, we present the most pertinent results, including key-events and key-issues during participant interaction as well as important processes which occured over time.

Key-events are, in this case, not necessarily ordered by time of occurrence, but ordered by relevance. That being said, we were able to observe:

- A general increase of interaction with the behavior sliders, immediately after the expert mode was introduced
- There was an increase in interaction with the Buddy canvas and suggestion selection in all cases after expert mode introduction

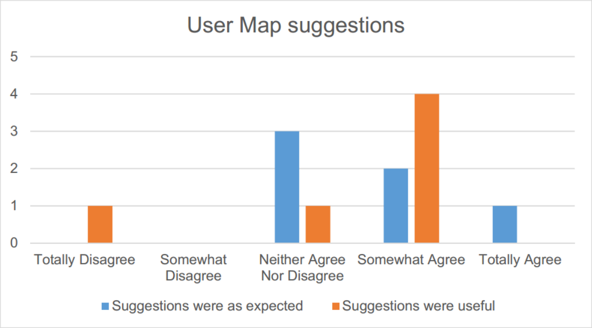


Figure 7: User Map control expectation vs usefulness

- Exporting suggestion selections almost always meant a consequent refinement of the that content in the designer's level
- When asked to work towards a different objective with the Buddy, the vast majority immediately changed the Editor Buddy objective using the appropriate UI element

Key-issues generally detected in the observations were:

- When facing participants with poor suggestions from the Buddy that rarely translated into an attempt to reconfigure its behavior using the standard mode
- When trying to preview the entire suggestion, the majority of participants would perform a selection of the whole canvas instead of using the dedicated visibility button
- Participants almost never interacted with the historic buttons
- After being introduced to the expert mode (sliders), participants didn't change back to the standard mode (switches), despite being told its use was not mandatory

Important processes such as decision making or control interaction were different for each participant. Essentially, we can describe a few of these patterns among the participants.

- The participants who started interacting with the buddy early on, effectively performed selections, exported and improved these changes frequently until they were finished
- Participants which had a predetermined idea for their level, the majority being professionals, wouldn't interact with the Buddy until after they were finished with a first version of it
- Generally, the participant's interaction with the behavior switches and/or sliders increased over time
- Participants would perform selections over the canvas regularly, even if they did not export anything
- Participants would not export a selection very often, however none of the participants used the revert button to undo this step
- Even though it there was no time limit, duration of each task was very even amongst participant, which averaged about 10 to 15 minutes

Interview Results collected from interviews were more subjective and, thus, harder to convey in its entirety. We present a couple of interesting patterns regarding questions on the usability and utility of the proposed solution as well as suggested improvements.

Generally, non-professionals preferred to use the Editor Buddy during the first task where they were asked to create content from a minimal template. Professionals, however, generally preferred to use the Editor Buddy during the second task where they were asked to modify their work towards a different objective instead of creating content from zero, with one exception being a level designer who was interested in receiving much more different alternatives.

On the recommended configurations, Innovation and User Map switches and sliders were the controls which had most disparity concerning their usage and preference but both were related to which task and which stage of design participants were focusing on at the time. There where some who recommended the use of Innovation or User Map controls in early stages in order to generate a large quantity of content quickly and there were others who recommended their use in later stages of the level design, where the exploration of more detailed changes preferred.

Not recommended configurations also depended on the stage of the designer's level and the task at hand, some participants did not recommend the use of Innovation when they wanted to slightly modify their work while others did not recommend the use of User Map when they wanted more different alternatives.

In both cases, recommended and non-recommended configurations, there were exceptions where some participants preferred the opposite of others.

Suggested improvements to the interface included:

- A change in the color scheme of the suggestion highlight
- A way to evidentiate viable paths from start to finish
- A more detailed description of the behavior switches/sliders tooltips
- A way to display real-time suggestion highlight, where more recent suggestions would display more vibrant colors and older suggestions a more faded color
- An initial configuration procedure to help set up the optimal interface and behavior according to the type of user
- Grouping interface buttons to optimize intuitiveness, such as historic buttons and selection-editing buttons

Suggestions towards improving the Editor Buddy behavior included:

- A button to enable the start of a new Buddy iteration in order to receive the next suggestion
- A way to improve the quality of the displayed suggestion, in exchange for a more late feedback, which is to say, adjust the trade-off between suggestion complexity and response time
- A way for the Buddy to identify patterns in the designer sketch and to enable the generation of those patterns such as rooms and branching to complement the main design of the level
- A new slider to control map difficulty, similar to the complexity slider, related to the size of the map
- Take into account the context of the performed selection, and not just the whole map area. Additionally a new control to adjust the granularity of suggestion generated, whether it is on the context of the whole map or the selection

Result analysis

Key-findings The first of our key-findings is directly related with the distinct types of users we invited for these test sessions, which is the fact that the mindset of the user and level of experience ultimately influences the usage and utility of the Editor Buddy. Also, for users who already have a clear idea of the layout they wish to create, the need for innovation or alternatives is greatly diminished. Inversely, designers who carried no preconceptions regarding their level's layout could enjoy a more broad array of options from the Editor Buddy and, possibly, a richer creative experience. However, such tools are still dependant on much more than just professional experience. Personality traits and life experience mean as much as, if not more, to the creative process and, ultimately, to the potential of such tools.

Besides the aforementioned distinction, there was a different kind of need in professional participants. The fact some expected a more contextualized feedback from the Editor Buddy, in the sense that it should be able to pick out patterns from the designer's level and generated novelty in a more coherent way, made us realise the main focus for those types of participants was the context of their own work. This helped us better understand the individual needs of professional level designers what improvements could be made in that direction.

We wish to underline the fact that an interaction-focused co-creative activity seemed to work well in most of the cases. However, particular cases where the less refined suggestions proposed by the Buddy seemed a bit too rough tell us that perhaps it could benefit from having a way to control this parameter. This would mean a more pronounced trade-off between quality of content and response time, where the amount of time dispended would not linearly translate into better results. Still, the option to do so would be present.

In the vast majority of test sessions there seemed to be a common misconception amongst participants, which its that the Editor Buddy would take their selection into account when generating its next suggestion. The fact participants regarded the Buddy's canvas beyond its most simple use and envisioned a way to work more intimately with the Editor Buddy surprised us. This made us reconsider the importance of such actions and would like to point this out as one of the most interesting improvements for future work.

Logical steps Following steps would include the implementation of the more straightforward suggestions, which showed to be a consensual improvement over the current implementation, for example the interface improvements.

For the remaining suggestions, those which required a more careful approach, we would need to perform more in-depth preliminary evaluations to assess implications and costs as well as clearly define which areas are worth being improved and how.

Implications In the end, this study did not dismiss the main purpose of this tool, if anything it helped us validate it was a step in the right direction and discover some aspects of our particular implementation which could benefit from the insight gained from evaluation sessions. These improvements would serve to better adapt to the need of the more particular user which is level designer and, if possible, appeal to a larger universe of more casual users.

Conclusions

We concluded that the overall the performance of the developed solution was positive and it proved to be flexible enough to adapt to several types of users, not taking away any credit to those who were able to make the best out of what they were given, of course. This, in fact, proved to be one of the most important observation we made.

Seeing that, in our particular case, suggestions which were not immediately playable sometimes motivated participants to effectively export that suggestion, or a part of it, and improve on that idea using their own view, which was something we liked to see happen and does not happen in existing solution. Nonetheless, these suggestions would keep on being evolved by the Editor Buddy in order to optimize their contribution for the task at hand. In this sense, we focused more on providing a way for users to interact more precisely with suggestions they were given, as well as watch the Buddy suggestion evolve side-by-side with theirs. Thus, allowing participants to explore every potentially creativity-enhancing aspect of the co-creative level design activity, starting with a blank canvas all the way to a playable solution.

The fact we received so many design suggestions taught us there really are no perfect solutions and finding a middle term, where we attempt to please both experienced and inexperienced users, proved to be both a challenge and a good way to determine the difference in needs and behavior. Still, there are multiple factors we cannot control which end up influencing the outcome of such evaluations or real-life applications, even more so when we are talking about something as volatile as creativity or human beings themselves.

In the end, we believe we were able to portray this paradigm of a digital "peer" and we hope it serves as an interesting contribution to the field of human-computer co-creative science.

Future work

We found several pertinent views on how to improve both the functionality and visual intuitiveness and utility on the Editor Buddy. From all the improvement suggestions we were given, we present the ones we found more pertinent for future work.

Design patterns The implementation of detection and generation of user design patterns could prove as a very interesting starting point when working towards a more user-centric approach. Related with this observation, we can also look with greater detail into the work of Liapis *et al.* in their Sentient Sketchbook paper (Liapis, Yannakakis, and Togelius 2013), more specifically, their work on accommodating the designer's style into the suggestion of the Editor Buddy.

Selection context Another interesting change to the Editor Buddy behavior would be the creation of contextual content depending on the selection made in the Buddy canvas. For example, creating a narrow selection would translate into generation of halls or more straight segments. We found this interesting and cannot help but wonder there is unexplored potential behind contextual generation of content and it might be worth considering for future content.

Evaluating Buddy contributions Regarding future evaluation sessions, besides evaluating the human-computer interaction, we would like to compare the quality of levels created with and without the help of the Editor Buddy. In other words, after integrating a couple of new features and polishing existing ones, we could perform tasks with two different groups, one using the LoG2 Editor and the Editor Buddy to create content and another one using solely the LoG2 Editor. At the end of those tasks, a third group would evaluate the creative value of both levels and could eventually play them to find out which results work best in the player's point of view. Another evaluation we could perform would be providing participants with the Editor Buddy to perform a task and ask them to perform another task afterwards without the help of the Editor Buddy and see if they missed it. Inversely, we could ask participants to perform a task using only the LoG2 Editor and and another task afterwards where we introduced the Editor Buddy and see if they found it useful in stimulating their creativity.

Acknowledgements

The author would like to thank, first and foremost, Prof. Dr. Carlos Martinho, teacher at IST-Taguspark, supervisor of this work. Thank you to all participants of our evaluation phases, including preliminary evaluations. Thanks to the authors of related work, who contributed with the knowledge of their findings on the area of computer co-creativity.

References

Boden, M. a. 2003. *The creative mind: myths and mechanisms.* Routledge.

De Bono, E. 1977. *Lateral thinking: a textbook of creativity.* Pelican books. Penguin Books.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2013. Sentient sketchbook: Computer-aided game level authoring. *Proceedings of the 8th International Conference on the Foundations of Digital Games (FDG 2013)* 213–220.

Lubart, T. 2005. How can computers be partners in the creative process: Classification and commentary on the Special

Issue. *International Journal of Human Computer Studies* 63(4-5 SPEC. ISS.):365–369.

Smith, G.; Whitehead, J.; and Mateas, M. 2011. Tanagra: Reactive planning and constraint solving for mixed-initiative level design. *IEEE Transactions on Computational Intelligence and AI in Games* 3(3):201–215.

Togelius, J.; Yannakakis, G. N.; Stanley, K. O.; and Browne, C. 2011. Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games* 3(3):172–186.

Vile, A., and Polovina, S. 1998. Thinking of or thinking through diagrams? The case of conceptual graphs. *Thinking with Diagrams Conference, The University of Wales, Aberystwyth* 22—-23.

Yannakakis, G. N., and Alexopoulos, C. 2014. Mixed-Initiative Co-Creativity. *Procedings of the Foundation of Digital Games 2014*.