Co-creativity in Videogame Puzzle Creation

Gonçalo Nuno Dias dos Santos Delgado
Instituto Superior Técnico,
Campus Taguspark,
Av. Prof. Doutor Cavaco, Silva
2744-016 Porto Salvo
Lisboa, Portugal
gon.n.delgado@gmail.com

Abstract

This work proposes a solution to improve the cooperation between humans and computer Artificial Intelligence (AI), as a *colleague*, in the creation of puzzles for video game levels. With this interaction we hope to give the designer a source of creative stimulus, in order to achieve overall more creative results than those obtained if said designer was working alone. The proposed solution consists of a co-creative puzzle creation tool, focused on improving 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 three genetic algorithms and can be guided by the designer on different domains: *objective*, *innovation*, *user map*; all then combined in a fourth one that re-evaluates the best suggestions of each previous algorithm, again on the three domains, with different weights based on the users configuration, to choose the best suggestion overall. Results showed a positive influence on the puzzle creation because our approach takes into account the smaller nuances of the co-creative interaction. Outlined improvements such as a better way to support designer-specific interaction patterns, improved algorithm behaviour and integration with past tools set the direction for future work. We concluded that through an intuitive interface, flexible and adjustable behavior, we were able to provide some positive contributions to the quality of the co-creative puzzle creation process.

Introduction

Most people think of computers as facilitating tools, they discard the possibility of their potential in providing valuable contributions to creative processes, only associated with human beings. By exploring the "peer" or "colleague" paradigm in co-creative activities, we seek to improve the human-computer interaction in such cases, by having the computer working alongside a human designer, as a *colleague*, in the creation of puzzles for a video game map. The work written by Lubart (Lubart 2005) states several advantages for this approach.

There is a respectable amount of work and research on the topic of computational creativity. Among all these, we took particular 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), the experiments of Yannakakis *et al.* with Mixed-Initiative Co-Creativity (Yannakakis and Alexopoulos 2014) programs and the work of Lucas (Lucas 2016), the Editor Buddy, whose behavior is based on a theory which merges both lateral thinking and diagrammatic reasoning.

Procedural Content Generation, in the context of video games, is an often used set of techniques to quickly generate generally correct and useful content, within certain constraints. Some video games support the creation of user content such as the design of levels and, more relevant to us, puzzle creation. We believe level design, in particular puzzle creation, is a good example of an activity that would benefit from a computer-assisted co-creative tool. In this sense, we analyze the findings of Liapis *et al.* in their tool, the Sentient Sketchbook (Liapis, Yannakakis, and Togelius 2013), the findings of Smith *et al.* (Smith, Whitehead, and Mateas 2011) in the tool Tanagra, the work of Lucas (Lucas 2016) with the Editor Buddy and the Evolutionary Dungeon Designer of Baldwin *et al.* (Baldwin et al. 2017). We take into consideration 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 inexperienced and amateur users in level design, with the majority having interest in it and present its results.

Background

Level-Design

Level design is one of the key aspects of the game design process, as Totten, C.W says on (Totten 2014) "it is also one of the most exhilarating" and Rollings, A and Adams, E in (Rollings and Adams 2003) "there is no standard way to design level". Totten,C.W. tries to answer this by saying "as you watch people play your level"(Totten 2014) we must keep these, among other reasons, relevant to our study:

- **Do they understand how to play the level?**Meaning, that teaching is an important mission of level design, and if a player doesn't understand a puzzle type that we want

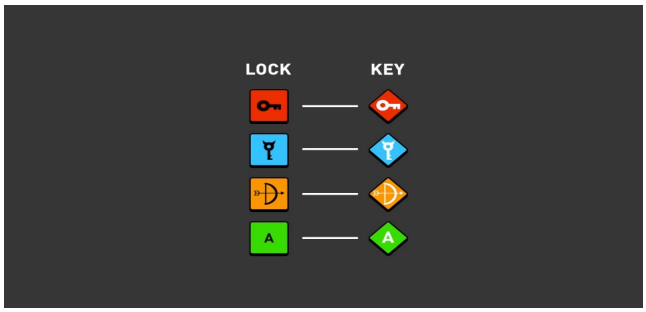
to repeat during the game, the player may need a better or more transparent introduction to the mechanic.

- **Is the level too hard for the player?** It must be avoided sudden increases in difficulty without proper balancing or player preparation. If a player is stuck in a early puzzle of the game, we may need to place in a later time of the game, or build easier puzzles to prepare the player.
- **Embrace happy accidents.** Sometimes players may resolve puzzles in an unique way that the designers didn't expected them to.

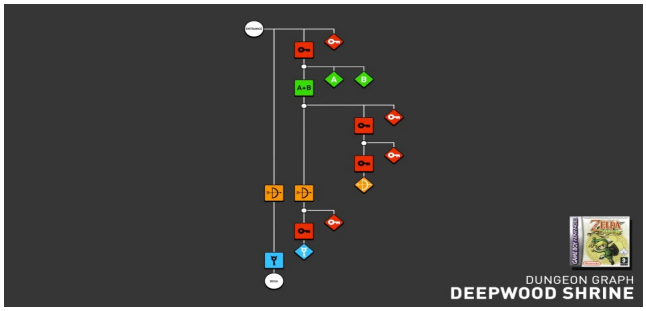
For our project we will keep simple individual puzzle pieces, pressure plates and gates, that the player by looking at it can understand how to interact, but create a more complex puzzle made of simpler smaller ones. The difficulty for a player is a subject of matter, but we want to achieve with the level of puzzle difficulty, an easier one that has overall less steps to complete than a harder one.

In terms of layout, we'll use Legend of Grimrock 2 in game layouts, since it's a dungeon crawler, the predominant gamespace structure is a *Maze*. Totten,C.W. describes mazes(Totten 2014) not as unicursal labyrinths, but as "branching spatial puzzles where occupants and players must find their way through an elaborate structure of walls and pathways with multiple dead ends to find an exit point". He also points that due to their branching nature, mazes are *multicursal*, by having more than one defining path. Due to this nature, it has potential dead ends that implies a risk-reward structure, where the player must choose from different uncertain options and hope to choose the most advantageous one. This type of gamespace structure is a good choice for puzzle creation, because the player must explore the maze to solve the puzzles, and face the uncertainty in order to proceed with the game.

Another interesting idea for our approach of level design, came from an Youtube series, called Boss Keys, from Brown, M. (Brown, M. 2016), a game journalist that wrote for various important gaming news websites. In this video[1], he explains The Legend of Zelda: The Minish Cap's[2] dungeon design and to do so, he converts the dungeons layouts and puzzles into graphs, as shown in figure 1. We can use a similar approach to our algorithm, convert the level layout into a graph, since it will be a fixed layout it will be easier than a random generated one.



(a) Caption of the graph, a square represents a lock and the diamond represents the key/item/switch to open the lock of the same color and image.



(b) The Legend of Zelda: The Minish Cap's Deepwood Shrine's graph made by Mark Brown.

Figure 1: Brown's graph representation of a dungeon from Legend of Zelda: The Minish Cap.

Creativity

We address how important creativity is 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 Boden, M., in her work (Boden 2003) she 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 and how these forms of creativity influence *conceptual spaces*, structured styles of thought, by combining ideas within them, exploring new ones or even the alteration of them, to give a new perspective and lead to new ideas that were unthinkable..

Boden also 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 de Bono, E. (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

[1]Link: https://www.youtube.com/watch?v=KEVJXqV7XMc
[2]Nintendo, 2004

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. We identify our application as being *offline*, a mix of *necessary* and *optional content* (depending on the algorithm), *parameters vectors* , *stochastic* and *constructive*.

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 this categories Lubart (Lubart 2005) translated into four lines of thought a computer's role in creativity enhancement:

- **Colleague paradigm:** or a "peer", where the creative process would be a shared activity between the human and the computer.
- **Nanny paradigm:** in this paradigm, the computer takes a more passive role in the creative process and is used as a manager of the user's productivity by detecting periods of procrastination and productivity breaks.
- **Pen-pal paradigm:** the computer assists the user by promoting communication, mainly in a collaborative creative process with two or more individuals. This communication is a fundamental tool to fulfill the need of idea sharing and synthesis.
- **Coach paradigm:** in this paradigm it's proposed an user's cognitive domain and narrowed thinking style, that can be inconvenient and can hinder his performance for a certain task. For this reason, the computer can assume the responsibility of kick starting a sentence of topic to the creative process, as well as offering information about existing techniques to stimulate creativity, thanks to its knowledge in creativity-relevant techniques.

This last one being the most ambitious and perhaps the most interesting to explore, considering how Liapis *et al.*, Smith *et al.* and Lucas focus on this paradigm in the Sentient Sketchbook (Liapis, Yannakakis, and Togelius 2013), the Tanagra (Smith, Whitehead, and Mateas 2011) and Editor Buddy (Lucas 2016) 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.

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 (MI-CC) (Yannakakis and Alexopoulos 2014), this evaluation it isn't trivial on an ongoing interaction between human and computer because it's hard to represent it in the final outcome. Yannakakis *et al.* explains that is difficult to capture the impact of pro-activeness of the computational activity on the human creativity and vice-versa. Due to this, they consider two types of evaluation when a computational creator is involved: the evaluation of the final(or intermediate) outcome, through the use of a number of heuristics for the task at hand (such as novelty and usefulness) or through crowdsourced estimates of creativity from a human audience; and the evaluation of the co-creative process for the generation of outcomes,solutions or items. This evaluation is less straight forward because the exact human creativity processes are either completely unknown or only partially known. It requires that we identify milestones within the co-creative process through heuristics of novelty, value, surprise or other relevant heuristics. Or, some type of meta-level mechanism of the quality of the process. Or, finally, an evaluation based on a temporal model of the co-creative process. They argue that MI-CC supports and fosters the creative process towards a certain outcome in addition to fostering the creative value of that same outcome.

Solution

The developed solution, a new mode for the previously created GUI based application by Lucas(Lucas 2016), named Puzzle Mode (Editor Buddy Puzzle Mode), where Figure 3b represents its interface. The application's primary goal is to foster creativity during a level puzzle design activity by presenting visual suggestions according to the designers settings. The interface serves two purposes: it provides the level designer with visual information, in the form of the accessibility Graph names (Map) Sections' Graph and the Map representation with optional toggles to show Pressure Plate and Gate connections, and a way to guide the application's behavior towards generating a specific content. The Editor Buddy Puzzle Mode(EB Puzzle Mode) behavior is defined by four genetic algorithms, that work in a three plus one (3+1) manner. Generated suggestions are displayed to the designer in the interface using a 2D preview of the map with the puzzle elements and the corresponding Sections' Graph associated to the map. The designer can also easily export the displayed suggestions or parts of it, to his current level.

Computer colleague paradigm

There's a particular paradigm that we are trying to represent with our solution and it's important to emphasize it so we are able to comprehend the relation between content it generates, it interface configuration and its iterative puzzle design cycle. To better impersonate a digital "peer", or to at least try to have similar behaviors, our solutions works on three different domains:

- *Objective*, seeks solutions towards a particular objective(such as more or less length and/or more or less backtrack;
- *Innovation*, seeks solutions that have the most distinct number of map elements in each room/corridor from the designers puzzle;
- *User Map*, is the opposite of Innovation, that means that it seeks solutions that have more equal number of map elements collocation in each room/corridor from the designers puzzle.

Since multiple domains can be used at the same time, we expect to find discrepancies between suggestions generated and what the user expects to be generated. These suggestions may be perceived as unwanted or wrong, however, the purpose of this work it isn't to give an absolute answer to what the user seeks, but to explore the ability to foster creativity, by allowing potential multiple useful suggestions to appear. Even with these "wrong" suggestions we hope to disrupt potential preconceived ideas or objectives in the user's mind and perhaps trail a more interesting path.

In the end, the designer will still be given all the means to guide or orient the type of content generated by the program (like asking a human partner to give suggestions more focused on a specific topic). For example, if a level designer feels more receptive to a wider array of suggestions with potential opposing traits, he can make these domains guidelines more lax, on the other hand, if he wants to explore a particular type of suggestions that he can be more strict on how he sets up these domain guidelines. He can either accept the suggestion and export it to work on it (or even consider it's a good solution and accept as the final suggestion) or discard and generate another suggestion. Figure 2 represents an example of an interaction between a level editor, where a designer creates a puzzle, and our solution. *Innovation* and *User Map* have into account the puzzle created by the user, hence their connection on the diagram, *Objective* evolves towards the objective defined by the user on the Buddy interface. According to the weight defined by the user for each algorithm, a percentage of the best individuals of each algorithm are blended in the *Combined* algorithm, that re-evaluates in the three domains each solution (again with the same weight, defined by the user, for the final fitness for each fitness function), and the best one is displayed as the puzzle suggestion from the Buddy that can be exported to the LoG2 Editor. In theory, this concept is generally applicable to other level editors that support puzzles (using the *door* and *key* micro-patterns(Dahlskog, Togelius, and Björk 2015)) , other than the one we chose.

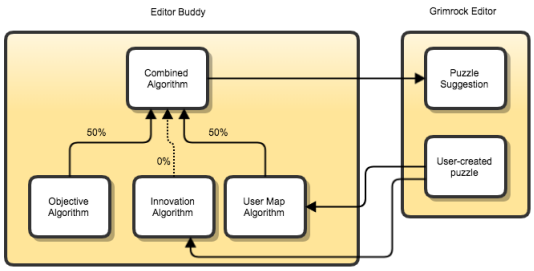


Figure 2: Example interaction of the solution with a level editor

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 3a, 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.

Editor Buddy Puzzle Mode

User interface The Editor Buddy Puzzle Mode User Interface, figure 3b, consists of these main components:

- Controls for defining the applications behavior. The Objective,Innovation and User Map that control its respective algorithms;
- An Objectives' section where the user defines the Objective Algorithm behaviour in terms of searching of a puzzle that creates a long or short puzzle, with more or less backtracking and the preference between either setting;
- 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 run the algorithms, interact with the generated suggestion, analyze a map and re-open the section's graph window;
- Four checkboxes of optional settings;
- A logger that shows extra information from the map to the user;
- Four progress bars to inform the current processing state for each algorithm.

Controls are shown in the form of sliders for the user to fine-tune the EB Puzzle Mode behavior. Suggestions are displayed in the 2D preview canvas. Each map section or corridor has a distinct color related to it, that corresponds to the same color node of the Sections' Graph. The light grey tiles are non-walkable ones. Additionally, the Editor Buddy can detect where the start and exit points of a level are and identify them with the character "S" for the starting location and the character "E" for the exit, and also represents with a "G" for the gates and "P" for the pressure plates of the map. The left and right arrow buttons, navigate through the suggestion historic, allowing the designer to revisit previously displayed suggestions or imported maps from the games editor that were analyzed and, a first and current or last suggestion arrow.

The user can make a selection on the 2D canvas with the left mouse and deselect with the right mouse button to select the area that he want's to export. This selection can also be cleared using the clear selection button or inverted using the invert button.

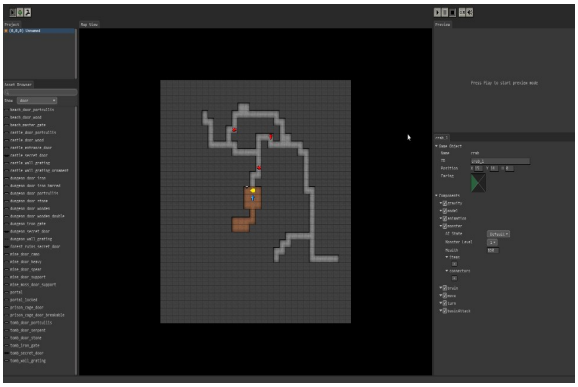
The four checkboxes control optional settings:

- The "Show Corridor" shows in black the corridors of the map that are the relevant zones to place a gate for our algorithm;
- "Show Gate/Key connection" shows with a black line;
- "Show Graph for Optimal Path Only" toggles between the Optimal Path Graph if checked and the Complete Graph of the level;
- "Export only connected Keys" when checked exports only the Pressure Plates(Keys) that are connected to a gate;

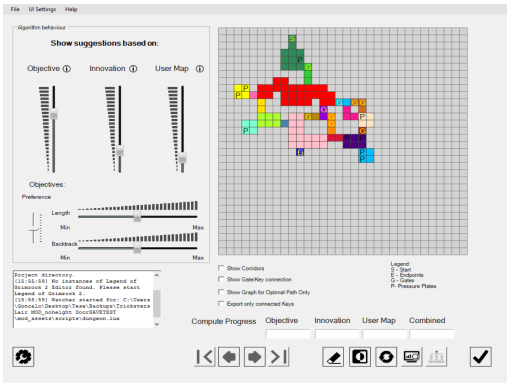
Behavior The Editor Buddy behavior is mainly guided by four genetic algorithms. The mentioned domains, innovation, objective and user map, from which we can receive suggestions can be controlled using the user interface controls, and a fourth that combines the results of the previous three:

The first algorithm, related with the objective domain,generates content closest to a particular objective(such as more or less length and/or more or less backtrack). The second of these three algorithms is Innovation, where the fittest individuals are those that have a most distinct number of map elements in each room/corridor from the designer's puzzle. And the last one is User Map, that is the opposite of Innovation, that means the fittest individuals are the ones that have a more equal number of map elements collocation in each room/corridor from the designer's puzzle. Additionally there's the said extra algorithm, called Combined Algorithm, that blends a percentage of each fittest population and evolves them towards a combination of the previous three algorithms fitness functions, which weight for them is set by the user's settings on each of the sliders for the previous three algorithms.

In all of the four algorithms, only the Objective component (either the algorithm itself or the fitness function on the Combined algorithm) need to be a feasible puzzle, meaning the user must reach the exit from the start, Innovation and User Map component only look to a more distinct or equal number of map elements per section, respectively, independently if the map is or isn't feasible.



(a) Legend of Grimrock 2 Editor

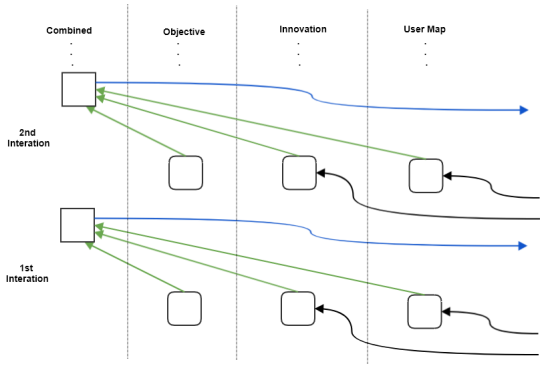


(b) Editor Buddy Interface: On the left side there's the three sliders for each algorithm and under them the sliders for the *Objective* algorithm and the logger, on the right side there's the 2 canvas and under it the different control checkboxes, progress bars and interaction buttons.

Figure 3: LoG2 Editor and Editor Buddy user interfaces

Editor integration The developed application is not a standalone program but, instead, works closely with the LoG2 level editor.All the puzzle elements addition 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 EB Puzzle Mode'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 only shows on the 2D canvas the added puzzle elements, this can be useful to see if a gate is placed in a corridor and the pressure plate in a room, no gate/pressure plates are updated though, the designer must actively analyze the map for the Buddy to create the most efficient connections. When exporting changes to the designers level, the procedure is 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 following diagram, illustrated by figure 4, represents the Editor Buddy Puzzle Mode execution flow. Each algorithm execution is independent from the ones before, the only possible connection is if the user export's the map and the Innovation and User Map take into account that new map generated before. The three algorithms, Objective, Innovation and User Map, run at the same time and, when they finish a percentage of each population, calculated according to the user input, is transferred to the Combined algorithm that re-evaluates and evolves that new population again according to objective,innovation and user map fitness functions with different weights according to the user input. The Combined algorithm calculates the best individual and shows that suggestion to the user, that can either export or ignore and run again the algorithms with the same or new objective and/or algorithms weight configuration. Each run is triggered by the user by clicking on the Run button.



(a) Cycle diagram

Element	Description
	Combined Population of Individuals
	Algorithm Population of Individuals
	Population of individuals transference
	Designer's map input for comparison
	Displayed Suggestion

(b) Element caption

Figure 4: Editor Buddy execution cycle diagram

Evaluation

The main goal of this study was the evaluation of the Editor Buddy Puzzle Mode 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 puzzles using the LoG2 Editor and the Editor Buddy Puzzle Mode to support them. They were given a previously created map layout (based on the Legend of Grimrock 2 map "Trickster's Lair") in the beginning of the first task as illustrated by figure 5.

Both task had shared goals:
They would be using LoG2 editor while interacting with EB Puzzle Mode to support them during the puzzle creation process; the map layout could not be changed; 1 pressure plate only opens 1 gate; a pressure plate could only be placed in rooms(non-black cells) and gates placed in corridors(black cells), for that they could use the "Show Corridor" checkbox to help them, and; the solution should be as interesting as possible for the player.
The first task particular goal was:

It must teach the players that pressure plates can open gates in non-adjacent corridors.

When the participants finished the first task they were shown the second task soon after and started working on it over the map from the first task. This allowed us to understand in which case the Editor Buddy Puzzle Mode behaved better.

The second task particular goal were:

The puzzle must make the player pass multiple times in the same room, where in the future will be placed "lore" related objects and we want the player to find these objects, which room or rooms are at the designer's criteria.

A time limit was not enforced for these task.

Participant description

We used a purposeful sampling method and selected participants that were inexperienced and amateurs in level design. One of the participants works in the game industry, not in a level design position, and all the participants either are or were Computer Science MSc students.

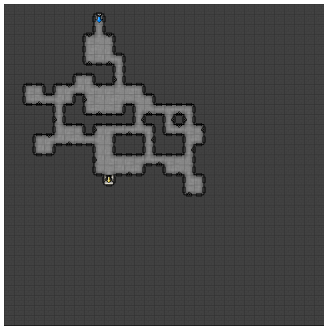


Figure 5: Evaluation task map layout

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 Puzzle Mode 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 Puzzle Mode 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 Puzzle Mode,participants said, in general, interface made easy to configure its behaviour. Table 1 shows results from the usability section of the questionnaire:

- (U1:) It was easy to configure the Buddy behavior using the available interface controls;
- (U2:) It was easy to make and edit a selection of the puzzle suggestion made by the Buddy;
- (U3:) There were no communication issues between the LoG2 level editor and the Buddy;
- (U4:) The section's Graph was useful;
- (U5:) It was easy to make the connection between the section's Graph and the corresponding Map sections.
- (U6:) It was easy to make the connection between the section's Graph and the generated puzzle topology.

There was some minor constraints related to the Editor Buddy Puzzle Mode that may have affected negatively these results, that we'll discuss in the analysis.
Regarding the behavior of the Editor Buddy, the following charts, Figure 6 shows how EB Puzzle mode's suggestions, when using the *Objective* control slider, were expected and how useful they were perceived by the participants. Figures 7 through 11 do the same for the *Length*,*Backtrack*, *Preference*, *Innovation* and *User Map* control sliders.

EB Puzzle Mode usability						
Scale	U1	U2	U3	U4	U5	U6
Totally disagree	0	0	1	0	0	0
Somewhat disagree	0	1	2	1	0	0
Neither agree nor disagree	1	1	0	1	0	1
Somewhat agree	2	1	1	3	3	3
Totally agree	3	3	2	1	3	2

Table 1: Frequency of answers to usability questionnaire

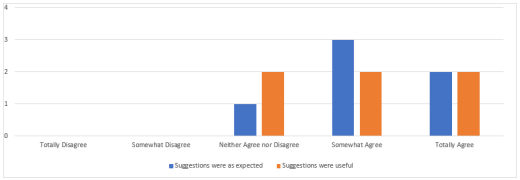


Figure 6: Objective slider expectation vs usefulness.

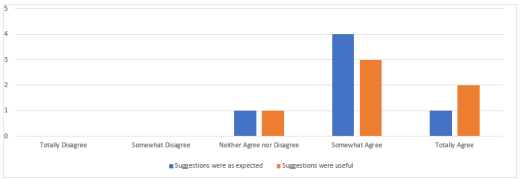


Figure 7: Length slider expectation vs usefulness.

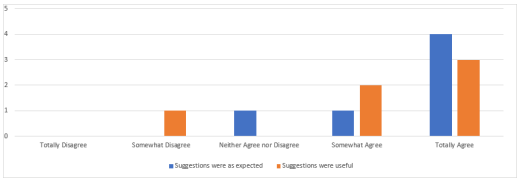


Figure 8: Backtrack slider expectation vs usefulness.

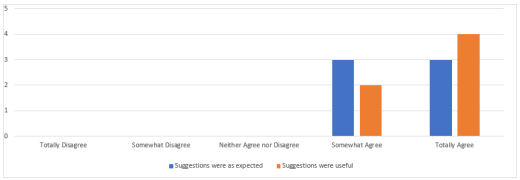


Figure 9: Preference slider expectation vs usefulness.

Observation From the observations we have been able to identify some key-events, key-issues and two polarizing ways to interact with the EB Puzzle Mode and the LoG2 Editor.

The identified *key-events* were:

- In the second task almost everyone changed backtrack,objective to max and some even preference towards backtrack;

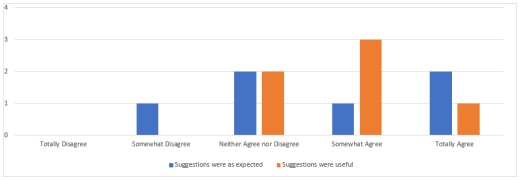


Figure 10: Innovation slider expectation vs usefulness.

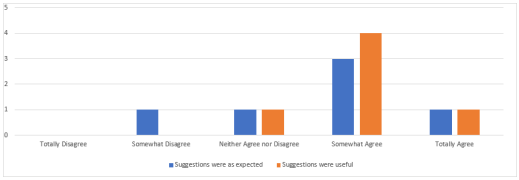


Figure 11: User Map slider expectation vs usefulness.

- One participant used the EB Puzzle Mode history for the second task, to make the map on the editor completely empty again so he could build from scratch.
- Most of the participants used the "Show Corridors" option to check if a gate was in a corridor;
- Every participant used the "Show Gate/Key connection" option.

The identified *key-issues* were:

- The aforementioned expectation of the pressure plate-gate connection to be taken into account for *innovation* and *user map* oriented solutions was not fulfilled;
- Some participants wanted for a certain disposition of the puzzle elements multiple puzzle solutions could be generated (different gate-pressure plate connections);
- In one of the tests the application crashed once for unknown reasons that was never replicated again;
- Some participants ignored the Section's Graph altogether;
- The selection of cells to export partial solution didn't have much use.
- The "Export only connected keys" was rarely used.

Finally there was two types of users, some of them worked their puzzle from scratch and then used the Buddy to give suggestions based on their design, while others run multiple times the Buddy to find a suitable base to work their puzzle from that in the editor. There were three participants that created their puzzle from scratch before using the Buddy to show suggestions, and the other three used the buddy first to generate a suitable first version of the puzzle to work over it and adapt to his vision. With this two behaviors we believe the EB Puzzle Mode can support different ways of interaction and adapt to each user's preferences.

Interview The EB Puzzle Mode was widely accepted as being and overall positive influence more specifically in the second task. The EB Puzzle Mode interface was found to be user-friendly and intuitive in general, *Innovation* and *User Map* for some wasn't all that clear.

Overall "Show Corridor" and "Show Gate-Key connection" was very well received and used by the participant, especially the "Show Gate-Key connection" had the overall consensus. A participant said that "Show Corridor" wasn't useful. We think the participant that expressed this opinion was because he used primarily the EB Puzzle Mode to generate suggestions first and then he would work and adapt them to his vision. In that way the Buddy would always put the puzzle elements on the right section, with small changes within sections or deletion of elements was more difficult to put a puzzle element on the wrong place, and so the "Show Corridor" didn't feel that important. Some participants expressed that the *Logger* window was optional, in their opinion, from the designer's standpoint that extra information wasn't necessary. Overall *Export only connected keys* was deemed unnecessary or situational. The majority wanted the *Innovation* and *User Map* algorithms to take into account the gate-pressure plate connections on the editor. There was a participant who even said that *Objective* was the main slider and *Innovation* and *User Map* could disappear. This participant felt the behavior of the Buddy not taking into account the connections that he made for the *Innovation* and *User Map* were frustrating that he said that with this behavior of these two algorithms they felt less relevant as the *Objective* algorithm, and so this last one was the main one and the others could be discarded.

Suggested improvements to the interface included:

- A reset button to set every slider at 50%;
- Change some button icons to other more representative;
- A dedicated box to show current map Length and Backtrack value instead of being on the Logger;
- Show on the map the number of time that each section was visited;
- Show on the map with another color(or even highlighting one section at the time) to show the path that the algorithm took.

Suggestions towards improving the Editor Buddy behavior included:

- *Innovation* and *User Map* take gate-pressure plate connections into account;
- Without changing the puzzle element's disposition show alternative ways to connect gates and pressure plates;
- A way to lock element's to prevent their deletion in new generations.

Result analysis

From this evaluation we confirmed that the EB Puzzle Mode worked really well to generate objective-oriented content, the participants saw the potential of this tool in the creation of puzzles for game levels. We saw a significant number of comments towards the EB Puzzle Mode taking into account gate-pressure plate connections that users make, unfortunately in our approach it's difficult to map connections into chromosomes and since these connections appeared in later stage of development they were not introduced in the

genetic algorithms, but for future work this is something that has to be taken into account. The other behavior suggestions are easier to integrate in the current tool. Overall in the interface and usability level there's a good opinion from the participants, one or another had more problems with them but we think might be explained by the behavior of the Buddy itself in discarding gate-pressure plate connections (on the question "There were no communication issues between the LoG2 level editor and the Buddy") and the small improvements suggested in the previous section. We'll explain on the future work section why this isn't a trivial problem to solve.

From the interaction of some participants we standard lateral thinking techniques were observed, in the case of participants that generate first multiple suggestions for a certain tool configuration, choose the one that feels better to them, modify them, and then generate again suggestions, if either *Innovation* and/or *User Map* had values other than 0, the modifications would take into account the puzzle elements position for the new suggestion. As said before, Lateral thinking it's all about generating *conflict* and challenging old ideas, a user express that the Buddy was "too disruptive" on the *User Map* algorithm, so the new idea conflict was there, maybe a little too much for some users, again by not taking into account gate-pressure plates connection, may have been perceived as much different, because of the new generated connections, but for the algorithm, by looking to only puzzle elements position it was similar, but by changing one or two elements from different sections made the *objective* algorithm go through another path and the consequent generation of connections was completely different.

Conclusions

We were pleased to observe that an interaction-focused co-creativity seemed to work well in most of the cases, even with two distinct types of user, one more LoG2 focused (that constructed his puzzle from scratch) and another that used the EB Puzzle Mode to generate a suitable first version of the puzzle to work over it, both user type worked back and forth with Buddy and the editor, showing the "peer" or "colleague" paradigm interaction between computer and human, defined by Lubart (Lubart 2005), that we hoped to see. We also noticed that standard lateral thinking techniques were observed during interaction and that the overall performance of the developed solution was positive and was flexible enough to adapt to the aforementioned two types of users that we identified.

We described EB Puzzle Mode simple interface, based on Lucas' work, that allows to fine tune the Buddy in three dimensions: *Objective*, following specific design goals; *Innovation*, wanting to explore new directions, and; *User Map*, focus on the current co-proposal. The design goals are puzzle suggestions in two domains: length and backtrack. The Buddy's 2D canvas allows to intuitively and visually communicate its suggestions to support diagrammatic reasoning, a cognitive process inherently present in level design

The fact we received so many design suggestions, which we'll discuss in the next section, taught us there's still some aspects that need to be improved regarding how the Buddy better utilizes the designer's created puzzles (take into account gate-pressure plate connections) and some interface modifications to adapts for different users.

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 research.

Future work

Throughout the development process up until the evaluation phase, we encountered several alternatives or ideas to improve certain aspects of the Editor Buddy. These were the most pertinent ones we kept for future work reference:

The integration of our solution with Lucas' work was something that we ambitioned from the beginning, but for time-related reasons and to have one less variable in the evaluation phase, we decided to have a fixed map layout based on the map Trickster's Lair from the LoG2 game. This is something that can be done in the future, as mentioned before, all map and graph sections related code are hard-coded for the Trickster's Lair modified map, but all the algorithms are generic for any map.

Another change that could be implemented in the future, was the one expressed by the participants, the EB Puzzle Mode take into account gate-pressure plate connections made by the user in the LoG2 editor for the Innovation and User Map algorithms. Since the connection came into play in later stages of development these algorithms didn't take it into account for the results, but the participants expressed that was better if these algorithms would take the connections made by them. This is a situation that isn't trivial to tackle , participants wanted to keep the connections, but they can become irrelevant, having no value whatsoever with other map elements, affect the most efficient way to finish a map, that's our main criteria to stop the objective algorithm, or even the change of places of the gate or the pressure plate, keeping their connection, the new solution, for those elements and their connection, might not make sense.

Another behavior change that some users said was for the displayed solution, show other ways to connect gates and pressure plates even if it wasn't the most optimal way to finish the map, this already can be supported with small code changes.

A interesting addition for the EB Puzzle Mode is to provide different types of puzzle, it already supports the *Environmental Mechanical Puzzles*, the pressure plates, but could also support *observational challenges*, *logic puzzles*, *riddles* and *hidden treasure* puzzles.

Various interface improvements were given in the evaluation chapter, some examples are a dedicated box for the puzzle metrics, *length* and *backtrack*, show in the map, with another color for example, the sections that the algorithm passed through, amongst other small things detailed on the chapter.

Regarding future evaluation sessions, besides evaluating the human-computer interaction, we would like to compare the quality of the puzzles created with and without the help of the EB Puzzle Mode. 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 EB Puzzle Mode 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 EB Puzzle Mode 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 EB Puzzle Mode and see if they found it useful in stimulating their creativity.

A more ambitious goal would be adapting the EB Puzzle Mode to other existing level editing tools. Apart from the Legend of Grimrock 2, this would be a huge challenge, because the EB Puzzle Mode is so deeply integrated with the LoG2 Editor. Because the LoG2 Editor works on the 2D basis, we would require at least those tools to work in a similar fashion, otherwise it would need to be redesigned to generate content for 3D game levels.

Acknowledgements

The author would like to thank, first and foremost, Prof. Dr. Carlos Martinho, teacher at IST-Taguspark, supervisor of this work. A big thanks goes to Pedro Lucas that created Editor Buddy and that helped with technical insight that helped in the creation of this Puzzle Mode for his tool. 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

Baldwin, A.; Dahlskog, S.; Font, J. M.; and Holmberg, J. 2017. Mixed-initiative procedural generation of dungeons using game design patterns. *2017 IEEE Conference on Computational Intelligence and Games, CIG 2017* 25–32.

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

Brown, M. 2016. Mark brown's youtube channel.

Dahlskog, S.; Togelius, J.; and Björk, S. 2015. Patterns, dungeons and generators. *Proceedings of the 10th Conference on the Foundations of Digital Games* (Fdg).

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.

Lucas, P. 2016. Co-creativity in Videogame Level Design. 1–92.

Rollings, A., and Adams, E. 2003. *Andrew Rollings and Ernest Adams on Game Design.* New Riders.

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.

Totten, C. W. 2014. *An Architectural Approach to Level Design*. A K Peters/CRC Press.

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*.