Co-Designing the Lighting of a Game Level

João Miguel Freitas Gonçalves de Oliveira
joao.miguel.oliveira@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

October 2019

Abstract

The inspiration for this work comes from the belief that computers can do more than perform computational tasks, but rather tackle a concept that most people think is impossible for a computer to achieve, due to being unique to the human being: Creativity. To demonstrate this, the proposed solution consists of a level design tool with co-creativity at its core, this is achieved by presenting the user with suggestions of possible modifications to their current level. There are two types of suggestions: level layout and lighting. Level layout consists on which tiles a player can walk through or not. And the latter, lighting is how each part of the level is illuminated to provide the player with the best experience possible. These suggestions are generated by genetic algorithms with parameters that can be adjusted through an user-intuitive interface, either it being more evolved towards level layout or the lighting setup.

Keywords: Level-design, computer co-creativity, procedural content generation, genetic algorithms.

1. Introduction

Creativity is an aspect of the human mind which is yet to be fully comprehended. It is a unique quality that every human being possesses. However not every person thinks the same way and some people are naturally more creative and others need a stimulus to awaken their creativity.

This is the inspiration for our work, to create a tool that works as a colleague providing suggestions and alternatives in order to spark the creativity within a person. More specifically to this work, we believe that our tool will help level designers to achieve a higher satisfaction with the levels they have produced.

Although some tools, have been designed with this concept in mind, none or only a few have the ability to co-creatively build the lighting of a level with its user, which is one of the most important aspects in level design. The illumination of a level has the ability of delivering different feelings and experiences to a player. Our work will be focused on giving a level designer the experience of co-developing the lighting of a level.

2. Problem

Traditionally computers are not coded to be creative, so how can we take this concept and provide a colleague that can absorb a level designer's idea and present meaningful suggestions to it. Can this Artificial Intelligence deliver an experience as good or even better than what a user could achieve with another user's interaction? How can a tool stimulate the user to explore different paths?

As mentioned, similar tools already exist. The Sentient Sketchbook by Antonios Liapis [5] offers a simple yet effective approach of displaying the user with multiple alternatives to his/hers design. However, despite being simple to use, the tool lacks interaction and customization to its parameters, which in the end affects the user's experience. And Tanagra by G. Smith, J. Whitehead, and M. Mateas [8] is comparable to the Sentient Sketchbook but has a deeper focus on platform games and still lacks options for customization, functioning more individually when generating content.

So how can we create a tool that works closely with the user and gives a valuable contribution to the level's design? This is what we aim to accomplish, to create a tool that co-creatively develops a level with its user by generating innovative, correct layouts and interesting lighting.

Two former students from Instituto Superior Técnico have each created their own solutions to this computer as colleague paradigm in level design. Pedro Lucas's work focused on building a tool that co-created a level layout with a user. And Gonçalo Delgado developed a tool that receives a fixed level layout and, along side the level designer, generates puzzles that a player must fulfill in order to successfully complete the level. However creating a co-creative experience is not a simple task. Both Lucas and Delgado followed the same approach, they created an interactable user interface with three sliders. Each slider controls the parameters of the algorithms responsible for generating suggestions and each correspond to one of the three core components of co-creative experience (innovation, guidelines and convergence). By providing the users with the capability to adjust the algorithms' parameters, it guides the application's creativity into generating more valuable suggestions.

Our solution resides in taking the concepts Lucas has constructed and developing our own algorithms that provide suggestions on level illumination.

3. Creativity

Creativity is a quality every human being possesses, some more than others. It is an innate skill of the human mind, that is not fully comprehended and therefore continues to be studied. However with those studies, definitions of creativity become clearer and Margaret Boden [1] perceptions of how creativity can be described are relevant to our work. Margaret Boden categorizes creativity into two terms, P-Creativity and H-Creativity.

Psychological-Creativity or P-Creativity in short, is described as a valuable idea that is completely new to the person who came up with it, even if such idea as previously occurred to someone else. In contrast, Historical-Creativity or H-Creativity is described in the same way as P-Creativity, however it has to be completely new, meaning that no one in history has ever thought about it before.

Margaret Boden work also categorizes creativity into three different approaches:

- **Combinational creativity:** "Produces unfamiliar combinations of familiar ideas, and it works by making associations between ideas that were previously only indirectly linked." [2]
- **Exploratory creativity:** Consists on exploring the preconceptions in a conceptual space to generate new ideas.
- **Transformational creativity:** Is the alteration of the conceptual, leading to the generation of unthinkable ideas.

The idea of generating lighting within a predetermine level layout makes use of Exploratory creativity, as the conceptual space is already known and our interest is in exploring the preconceptions in it.

Lateral vs Vertical thinking

The term Lateral thinking was first promulgated by Edward de Bono. As he describes, Lateral thinking is the process of generating as many alternative ideas as one possible can. Lateral thinking can be seen as producing new ideas by constantly clashing with existing ones. Whereas Vertical thinking is the most typically used by the human mind, where one individual selects the best possible approach to a certain problem and tries to refine it.

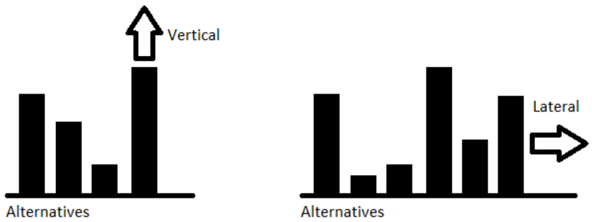

Figure 1: Comparison between Vertical and Lateral Thinking.

4. Procedural Content Generation

Procedural Content Generation (PCG) is a method of generating content through random or pseudo-random algorithms. The main reason for building PCG into the core of a game, is the level of replay value it offers to the user, as the game randomly generates its content, offering its users with constantly diverse experiences between levels. PCG has become somewhat popular nowadays, with a wide range of games using it as a core mechanic, such as the widely popular The Biding of Isaac[1] or more recently the game Dead Cells[2], however the idea of PCG is quite old, dating back to 1980 in the game Rogue [3].

5. Computers as co-creative partners

As briefly described before our project is to build a tool that works along side the user, but how can a computer work as partner? Lubart [6] addresses this concept and categorizes it into four distinct approaches:

Computer as nanny: The computer takes a more passive role, it pressures the user to work in order to complete its deadlines while detecting periods of procrastination.

Computer as pen-pal: It requires multiple individuals and the computers job is to promote the communication between them. It allows the sharing of ideas, to improve the creative process.

Computer as coach: The computer is used as jump-start to the creative process by providing information in a different approach that humans would not necessarily think of.

Computer as colleague: The most ambitious vision, it involves in having the computer working as peer in the creative process.

[1] Edmund McMillen, 2011
[2] Motion Twin, 2018
[3] M. Toy and G. Wichman, 1980

6. Genetic Algorithms

Inspired by Charles Darwin's theory of natural evolution [3], where the fittest individuals are selected and used for reproduction, in order to produce the offspring of the next generation, Genetic Algorithms (GA) (a subclass of Evolutionary Algorithms) are used to generate high-quality solutions to search and optimization problems.

The natural selection process starts by selecting from the population the fittest individuals. They reproduce and generate multiple offsprings that inherits their characteristics, which will be later on passed to future generations. By constantly matching the most fitted individuals, it ensures that their offspring will be parents with higher chances of survival. This process is iterated until a solution to the problem is found, this solution will consist of the fittest individuals.

7. Legend of Grimrock 2

Legend of Grimrock 2 (LoG2) is a tile-based real-time dungeon crawler video game that was developed and published by Almost Human, where a player can control four unique characters. Each character has its own traits and abilities, with each one having both strengths and weaknesses. The goal of the game is to advance successfully through the different maze-like dungeons, which are filled with monsters and bosses that a player must defeat. There are also puzzles contained within the level which the player must decipher in order to advance to new sections of the dungeon.

However, for our work the most relevant feature of the game, is its built in level editor, called Dungeon Editor (Figure 2), which features an intuitive layout with a 32 by 32 square grid that represents the tiles in which a designer can place walkable terrain, walls, puzzle items or monsters. The Dungeon Editor, on the left side of application, provides a complete list of items the designer can place on the level's grid. On the right side of the tool there is one of the most interesting features, a window that allows the user to preview the level without the need of opening the game. Under the preview window, there is a window for LUA scripting that can be used to change the assets parameters.

8. Lighting in level design

The art of level illumination can go by unnoticed in the process of creating a level. However, it is one of the most important elements to give character to a level and it is not all about its placement. Details such as the intensity, color, movement and shadows, all affect the player's experience. The correct use of lighting can transmit different emotions to a player, such as fear or wonder, for example darker levels usually transmit the sense of anxiety and insecurity. However, lighting a level is a difficult task

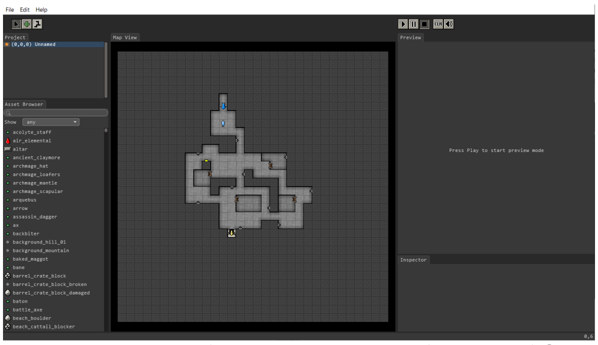

Figure 2: Example of the Dungeon Editor of Legend of Grimrock 2, as can be seen the designer as 32 by 32 grid where he can place any item from list on the left. On the right side there is a preview window of the current level..

to do right, adding or removing one light source can have a tremendous impact on the overall level experience.

John Feil and Marc Scattergood divide lights into two groups: static lights and dynamic lights:

- **Static lights** are pre-rendered into the level, meaning that their are actually a part of the environment. The use of static lights is very common, since they are computationally inexpensive and usually the best way to illuminate most of the level's game objects. However, they come at the cost of realism.
- **Dynamic lights** are generated in real time, meaning the game engine has to render every light, reflection and shadows in every frame. At the cost of performance, dynamic lights are much more natural looking than static lights, since the shadows and light effects are generated in real time and evolve with the movement of the scene.

Level designers usually try to strike a balance in use of the two, so that the level's performance is acceptable while also maintaining visual fidelity.

To our work, as we will focus on the creation of dungeon levels, we will mainly be using the torches as our light sources. Torches in Legend of Grimrock 2, emit light in 360 degree field and can reach a maximum of 5 tiles. They are Dynamic lights, meaning that the illumination they provide is generated in real-time as a player is advancing through the level.

9. Previous works

Pedro Lucas and Gonçalo Delgado have both developed works in the theme of Level Content Cocreation, which will be the foundation for our work.

Pedro Lucas created a tool called Editor Buddy [7], which allows for the creation of a level layout along-side a human using the Legend of Grimrock 2 video game. The Editor Buddy takes as an input the level the user has created and displays suggestions. These suggestions are possible modifications to the user's current level design and are generated based on three algorithms - Innovation, Objective and User Map. Each algorithms' influence can be controlled by adjusting the corresponding sliders in the tool's interface.

Gonçalo Delgado's work was inspired by Lucas's Editor Buddy and shares most of the principles found on his tool. However, Delgado's tool, called Editor Buddy Puzzle Mode [4], instead of co-creating the level's layout, focuses on puzzle generation. The generated suggestions represent changes to the puzzles found on the user's level, such as pressure plate and gate locations, as well as their connections. However, as it was not the focus of his work, Delgado did not use the level layout generation that Lucas developed, instead he decided to use a fixed level layout and construct the puzzle generation around it.

10. Methodology

As our solution's foundation is the Editor Buddy developed by Pedro Lucas, we decided to name our solution Editor Buddy Lighting Mode (EBLM) as its main focus is the illumination of levels. It consists in a GUI based application that works along side the LoG2 Dungeon Editor to simulate creativity. The tool, in conjunction with the level designer, participate in a co-creative process when creating a level. The tool's creativity is represented by the generation of suggestions, which essentially are possible modifications to the user's current level design. The Figure 3 represents the Editor Buddy Lighting Mode final version. The user interface contains several sliders as well as a map canvas, which is used to show the generated suggestions. The mentioned sliders are responsible for defining the EBLM behavior. Its behavior is constituted by a total of four algorithms, two algorithms created by Pedro Lucas and Dr. Carlos Martinho that beside user created map, generate level layout suggestions and two algorithms developed by us, that also take the user created map as an input and generate lighting suggestions.

11. User Interface

The Editor Buddy Lighting Mode interface consists of these main components:

- Controls in the form of sliders that can be used to adjust the application's behavior.
- A 2D canvas where the program generated suggestions are displayed. Which can also be used to draw out paths in the map that are used to influence the generated suggestions.

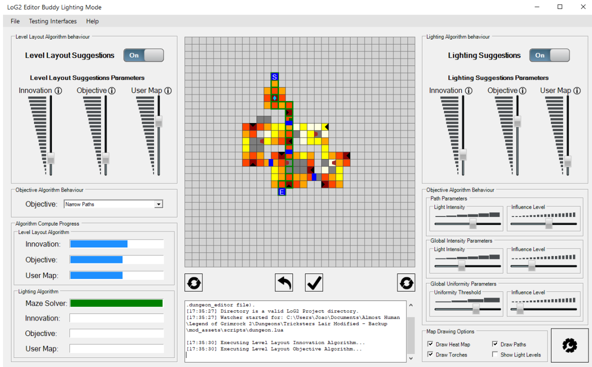

Figure 3: Editor Buddy Lighting Mode interface.

- Two buttons to interact with the suggestion. Revert and Export.
- Two buttons to generate a new suggestion. One for level layout and the other for illumination.
- A set of progress bars that display each algorithm's computational progress.

12. Behavior

The Editor Buddy Lighting Mode concept is to act as a colleague in a co-creative level design process, which means that the algorithms must generate computational creativity. To accomplish this, we followed the same concept as Pedro Lucas, the creation of two Genetic Algorithms. These two algorithms serve different purposes, one is for calculating the Innovation component of the suggestions and the other is responsible for generating suggestions according to the required objectives. The User Map algorithm functions differently from these two. A certain percentage of individuals, measured by the User Map slider, is inserted to the initial population of each of the algorithms.

The algorithms uniqueness, is ensured by the fundamental concepts of Genetic Algorithms, where individuals are selected to breed amongst them in order to spawn a stronger offspring. However the algorithms need to ensure that the population is evolving towards a solution fitting of the user's expectations. This assurance is kept by the fitness functions, which associate a score to each individual, the higher score the better the user's expectations will be matched.

The suggestions generated by the algorithms are a direct result to designer's interaction with the controls in the EB Lighting Mode and the current level he/she has created in the LoG2 Dungeon Editor. Every time the designer makes an alteration in their level in the LoG2 Dungeon Editor, the EB Lighting Mode will automatically start generating new suggestions. However, the process does not reset, both the new level and the previous population from the generated suggestion, are used in the next algorithmic iteration. With this approach the algorithms' initial population will consist of an already evolved population, meaning that instead of restarting the entire growth of the population, the algorithms continue developing the previous one into an even better population, thus making the suggestions better with every algorithm cycle. Figure 4 illustrates the behavior of the Editor Buddy Lighting Mode.

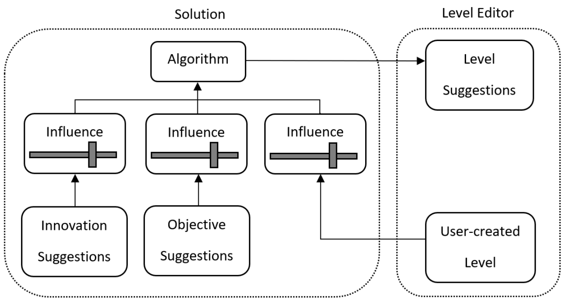

Figure 4: Editor Buddy Lighting Mode behavior.

13. Lighting

After creating the parser to read the current level and displaying it in our interface, we faced our first obstacle as we needed to develop a way for the level designer to see the lighting in our interface's 2D canvas. Our solution was to develop an algorithm that calculated a lighting heatmap of the level.

The first step was to analyze the Legend of Grimrock 2 game lighting behavior. This was accomplished by performing a series of tests with the Dungeon Editor and the placement of several lighting objects. These are the most relevant findings:

- There are multiple objects for illuminating the level, however they all operate by the same rules.
- Torches, the most common form of light objects in dungeons, illuminate a radius of 5 tiles, but can be seen from 12 tiles away.
- There is no reflection when light bounces of walls, meaning that light does not travel through corners.
- Having multiple light sources does not the increase the light intensity of a tile. Meaning that having numerous torches close to each other illuminating a certain tile, does not increase how well illuminated the tile is.

After gathering this information, there were some challenges we had to figure out before developing our heatmap algorithm. The first design decision we made was we would only consider torches has light sources, as it is the most common form of illumination in dungeons. With this decision taken into account, our light levels for a certain tile could only be within 0 and 5, the first being that the tile is completely dark and the latter that the tile has at least one torch, meaning it is fully illuminated. This scale is based on the distance a tile is from its closest light source, for example if the tile is three tiles away from the closest torch and its path is unobstructed, the light level of the tile is 3.

When developing our algorithm three main constraints had to be taken into account:

- The algorithm must know where the closest torch for each given tile is and provide the correct light level considering the distance between them.
- It must always check if there is any wall obstruction between a torch and the tile in question.
- The algorithm must always verify all torches within reach (5 tiles) of a tile before committing to a light level.

After building the algorithm with these constraints in mind, we had to optimize it as much as possible since this heatmap will also be used by the fitness functions of our genetic algorithms.

Figure 5 represents how the heatmap algorithm's outcome is displayed in our interface. With the black triangles representing the torches and the colors red to white each tile's light level, red being the highest and white the lowest. We performed a brief test with a small group of users and its results demonstrated that this color scheme was the best approach.

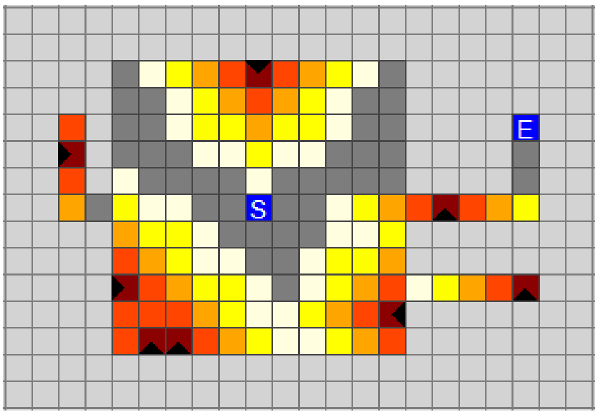

Figure 5: Example of the Editor Buddy Lighting Mode 2D canvas when displaying the generated heatmap.

14. Editor Buddy Lighting Mode execution flow

Over the next subsection, we detail the Editor Buddy Lighting Mode's life cycle, which is illustrated by figure 6. We address when and how it starts, as well as providing relevant information regarding algorithm interaction and how it chooses which suggestion to display.

The Editor Buddy Lighting Mode algorithm execution is comprised of two separate stages: An Initialization phase, where the algorithm first generates the initial populations and a post-initialization phase responsible for generating the suggestion, which takes the previous algorithm generated populations and evolves them to stronger populations until a suggestion is formulated. Each time the level designer makes modifications, creative or destructive, to the level, in terms of both level layout and placement of light objects, it triggers a new evolutionary run in the Editor Buddy Lighting Mode.

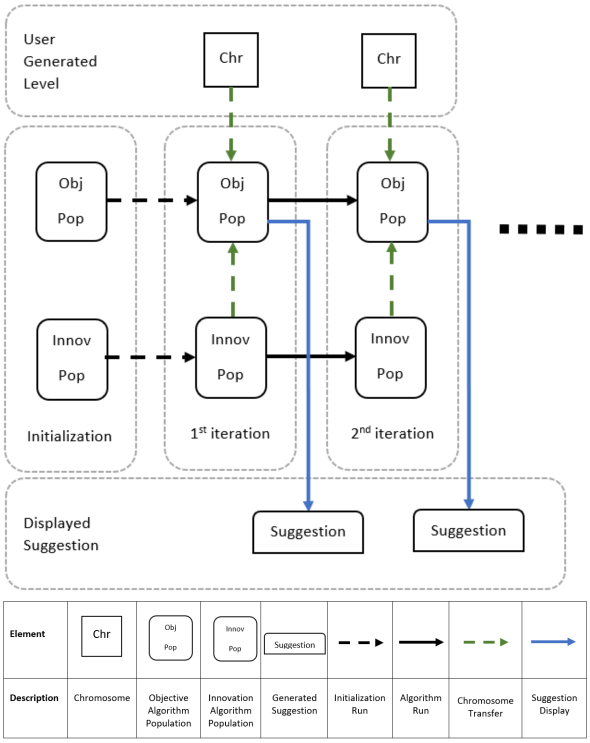

Figure 6: Editor Buddy Lighting Mode algorithm execution flow diagram and its element captions.

Initialization Phase: It consists in the generation of the initial population for both the Innovation and Objective algorithms. This phase only occurs when one of two scenarios happen:

- When the EB Lighting Mode is first launched.
- When the level designer changes any of the sliders in the interface.

User Input: Identified as User Generated Level in the execution flow diagram, it is independent from the Innovation and Objective algorithms, due to its development being solely performed by the level designer. The EB Lighting Mode job is to read it and use it in the generation of suggestions. The only two times the EB Lighting Mode interacts with the user generated map, is when the level designers wish to export the generated suggestion to their design and when revert the button is pressed in the interface.

Each time level designers make a modification to their level, the EB Lighting Mode automatically detects it and proceeds to use current level as an input in the generation of a new suggestion, thus provoking a new evolutionary run of the algorithms.

Configuration and Population / Chromosome transfer: Each action from the level designer triggers a new algorithm run, in which the first step the EB Lighting Mode performs is a verification of the current state of the interface's sliders. Depending on how they are configured, the pool of potential suggestions will be composed of different types and/or amounts of individuals. The Objective algorithm is the algorithm responsible for the selection of which suggestion to display. This is due to the fact that, according to the value of the sliders, a portion of both the level designer's level and the Innovation algorithm's population are inserted into the initial population of the Objective population. For example, if the Innovation slider is set to 25%, then a quarter of the top population generated by innovation population is selected to be a part of the Objective algorithm's initial population for the next iteration.

Algorithm execution: After the Objective algorithm receives a percentage of individuals from the Innovation algorithm's population and the designer's level, a new algorithm run starts, where each algorithm runs independently. The Innovation algorithm generates a solution as distinct as possible from the current level. Whereas, the Objective algorithm generates a suggestion that best fits the parameters set by the user in the interface.

Displayed suggestion: When both algorithms' execution is completed, the EB Lighting Mode must decide which individual is the best suggestion to be displayed to the level designer. This selection process always adheres to the current state of the sliders of the interface. For instance, if all the sliders are set at a third but the Global Intensity Objective slider is set at 100% with its light intensity at 0, then the suggestion chosen is the one who has the least amount of global illumination possible.

Algorithm re-initialization: The re-initialization of the Innovation and Objective algorithms has to occur every time the level designer switches the application's behavior, which means every time the sliders in the interface are altered. Consequentially, the process of initializing populations also has to take place. The next scenario is a perfect example of why this process must happen: imagine a designer has been using the EB Lighting Mode for while without changing the sliders, only asking the application to generate new suggestions. If now, he/she were to change both the Innovation and User Map sliders to 0 and if there was no reinitialization, the next suggestion might still display elements of both, due to the fact that there were still individuals in the Objective algorithm's population.

15. Results & discussion

The objective of this evaluation is to test the utility and the efficiency of the Editor Buddy Lighting Mode. Meaning that we want to test the usefulness of our solution in the generation of lighting suggestions in a co-creative process, by observing its contribution to a designer's when creating a level. Another goal of this evaluation is to test the designer's interaction with the Editor Buddy Lighting Mode's user interface. In other words, evaluate if the designers interaction with the UI is fluid and if they adjust the algorithm's parameters to achieve suggestions that they find valuable.

16. Study Design

The study was performed by a small group of participants in the facilities of IST-Taguspark campus. All the participants play videogames regularly and have a background in level design, except for one, who is an avid gamer but lacks knowledge in level design. All the participants were given a scenario, where they were working for videogame design company and they were responsible for all illumination on the current project. For each of the levels they were only given one guideline about the level's timeline in the overall videogame plot. The reason behind it, was to evaluate how participants change their lighting process considering the difficulty of the level. It is also worth mentioned, that we strongly advised the participants to use the Dungeon Editor's preview window as a way to validate their development.

First the participants were given a document with information about the test, as well as brief guide on how to operate the LoG2 Dungeon Editor. The participants were also given about two minutes so that they could get familiar with the level editor. Afterwards, we presented the first task, which consisted in asking the participants light up an already built level. This task did not have a time-limit but ideally it should not take more than 10 minutes to complete, although the expected time was around 7 minutes. The second task was similar to the first one, however this time we introduced the Editor Buddy Lighting Mode to the participants. It worth mentioning that we first explained every aspect of the EBLM and clarified all doubts. The third and final task, was similar to the second one, however the level was different, as well as the participants' guidelines.

Level guidelines:

- **Task 1 & 2:** This level is one of the first level's of the game, use lighting accordingly.
- **Task 3:** It is the last level of the game and it includes hard puzzles and a difficult boss fight.

17. Participant description

The evaluation process was conducted with the final version of the Editor Buddy Lighting Mode and since the application's target audience is novice level designer's, the tests were performed by people that either have acquired knowledge in level game design and might have previously built levels, or by people that game on a daily basis and have limited knowledge in level design. We decided not to focus our attention on more experienced level designers that work in the field, as they already have a specific mindset when building a level, thus diminishing the usefulness of the EB Lighting Mode.

18. Data Collection Methods

We used three types of data collection, **participant observation**, **questionnaire** and **screen recording**.

Participant observation was conducted throughout the entire evaluation phase, including when the tasks were being performed. The data obtained was collected by observing the participants and taking notes.

The collection of data by screen recording the participants when performing the tasks was conducted so that we could retrieve relevant information after the tests.

Upon concluding the tasks, participants were asked to fill out a questionnaire, regarding their experience of building a level with and without the Editor Buddy Lighting Mode.

19. Results

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

Questionnaires

The beginning of the questionnaire there were three questions with the purpose of defining the demographic of the participants. Question 3 of the questionnaire, was the most important of the three as it allowed us to classify the participants' knowledge of level design. Our small group of individuals contains at least one participant for every level of knowledge in level design, individuals that have no knowledge of level design, individuals that have basic knowledge and individuals with practical experience in building levels.

Regarding the EB Lighting Mode's behavior, the figure 7 corresponds to the responses of the participants on the questions 8 through 13 of the questionnaire. They represent the participants' opinion on the results generate by the algorithms when adjusting the UI's sliders.

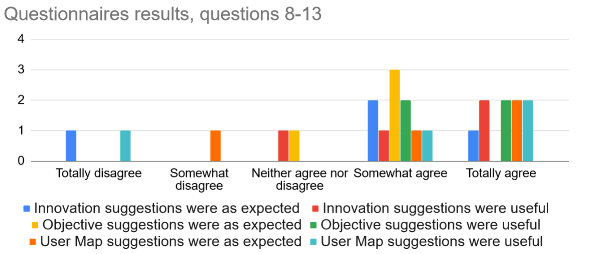

Figure 7: Innovation algorithm generated suggestions - expectation vs usefulness

Screen Recordings

When evaluating the screen recordings we were searching for both significant patterns when the participants were constructing their level as well as some performance metrics.

It worth mentioning that one of the participants took more time adjusting the parameters and generating suggestions in task 3 than the others. The table 1 represents some of the metrics and their respective values obtained from the results of the tasks 2 and 3 without the participant who took longer.

EBLM task 2 and 3 interaction results		
	Task 2	Task 3 without a user
# of interactions	1.7	1
# of generated suggestions	4.25	2.33
# of times sliders were adjusted	2.5	1
# of paths created	0.75	0.33
# of exported suggestions	1.75	1
Total time spent on EBLM	1min 28secs	57secs

Table 1: All results represented are averages. Screen recordings task 2 and 3 results.

Upon analyzing the values of the metrics without the participant that took longer than the others and through our observation of the recordings, there are some interesting findings that can be extracted:

- The participants when familiarized with the EB Lighting Mode take significantly less time adjusting the parameters in the UI to generate a suggestion they are satisfied with.
- In task 2 and 3, all the participants started by using the EB Lighting Mode before interacting with the LoG2 Dungeon Editor.
- In task 3 none of the participants felt the need to adjust the sliders and generate new suggestions, after the first exported suggestion.
- The participants did not seem to have much interest in the Path Objective component of the EB Lighting.
- All the participants showed more interest in the Objective algorithm, more noticeably the Global Intensity sub-goal.
- Two of the participants in both tasks, removed or placed torches after exporting a suggestion.
- All of the participants' level in task 2 had considerably more torches than of the participants' level in task 1.
- In task 3, all participants created a similar lighting to the original LoG2 level "Tricksters Lair" with an average discrepancy of 2.5 torches.

Participant Observation

The participant observation proved to be a valuable component of the evaluation, as we managed to identify some of the problems of the EB Lighting Mode. The most noticeable problem we encountered in our interaction with the participants was a difficulty in understanding the purpose of the algorithms. Some participants required a more detailed explanation than the others in the function of some the algorithms, specially the three sub-objectives of the Objective algorithm. Another problem we have observed was that some participants required a second explanation about the influence sliders of the Objective algorithm.

One key-finding we observed, was the evolution in comfort of the participants between task 2 and task 3. It is natural given that they already had gained experience when experimented with the application in task 2, however when observing the participants in the beginning of task 3, all but one quickly adjusted the parameters and generated a suggestion with the goal they had in mind.

Another curious aspect was how the participants clearly created different lighting for the task 3. As previously mentioned, the only difference, besides the level layout, between tasks 1 and 2 and the last task, was the guidelines. The guideline for the tasks 1 and 2 was that the level they were illuminating was one the first level's in the game and all the participants interestingly created a well lit level, where none of the sections in the level had a low light intensity. Whilst on the last task, all of the participants illuminated the level with a low light intensity, all of the levels generated had at least one dark section and very few areas had more than one torch.

Interpretation

In this subsection, we present our interpretation of the results as well as address some of the problems and the reasons behind them.

The first of the major problems of the EBLM we have identified through the evaluation, was that the participants rarely used the application in a co-creative process, they mostly operated it as a level lighting generator. This is proved by the low number of interactions the participants had with the EBLM, as can be observed in the table 1. In the best case scenario, which happened in the first task the EBLM was introduced, the participants averaged a number of 1.7 interactions with it. This is further supported by the fact that all of the participants always started the requested tasks, by exporting a suggestion and then proceeding to test it, with only two of the participants actually editing and refining the exported level. Furthermore, in task 3, none of the participants asked the EBLM to generate new suggestions after exporting one, thus making us believe that they saw the application as level lighting generator instead of colleague. Despite the concept of co-creation not being apparent to the participants, there is one positive note that we can extract from the low number of interactions with the EBLM, which is that the participants when exported a level, they believed that the generated lighting by the application was at least acceptable and appropriate to level they were illuminating. Thus making the EBLM a useful tool when creating a level's lighting. This is also supported by the fact that only two users felt the need to modify the exported level and that the changes made were minor.

Another major problem we observed was the difficulty the participants had in understanding all of the components of the EBLM user interface. The EBLM has a learning curve due to requiring the participants to learn the purpose of every single one of the nine sliders present in the user interface. Despite its learning curve, we believe that the core concepts of each of the sliders, were simple to comprehend after the participants had time to test them by themselves. This assumption is backed up by the retrieved numbers from the screen recordings. If we compare the metric values of task 2 and 3 without one of the participants, it is evident that the participants felt much more comfortable with the EBLM and its sliders after having time to experiment their effects.

When analyzing the results of the questionnaires, it is obvious that the Objective algorithm was the participants favourite. It was the only one who did not receive a negative evaluation, where both of the Innovation and User Map algorithms received mixed evaluations. We suppose that the shortage in use of the Innovation slider was directly linked to the chosen levels for the evaluation. Given that both levels used in the tasks, in their original state did not have any light sources and the goal of the Innovation algorithm is to generate suggestions as distinct as possible from the current level, it meant that suggestions generated by the EBLM were mostly filled by torches. This translated into the participants being overwhelmed by the amount of torches in the suggestion. This may represent a possible problem with the EBLM, where some of the generated suggestions have too many light sources.

The User Map algorithm suffered from the same problem as the Innovation algorithm, however the effect was the opposite. Since the participants always started the tasks using the EBLM, when they attributed a high value to the User Map slider, the generated suggestions would represent levels with hardly any torches. Which is the expected behavior for the User Map algorithm, but its usefulness might have been dismissed due to the chosen levels for the tests. Another fact that might have influenced the participants dismiss the value of the User Map algorithm, was that they always started the tasks by generating suggestions and rarely went back to the EBLM after exporting a level, they never actually got to discover the true value of the User Map algorithm. The User Map algorithm becomes much more relevant once a level is already fully or partly illuminated and the level designer wants to generate similar lighting to the existing but with a few modifications.

One issue we did not foresee, was the absence of path creation. The results obtained from the question 6 of the questionnaire, reveal that the process of creating a path is simple, so its lack of use must be due to its usefulness. Only one of the participants decided to create a path in task 3. When questioned about it at the end of the test, one participants responded he did not feel the need to create new paths as the level was too small to make a difference, while the other two responded that the provided path was as they wanted. This leads us to conclude that the participants did see value or usefulness of the path sub-objective, however they believed it was more useful to control the lighting intensity on the path that leads towards the level's end, instead of creating new different paths.

The participant who took longer than his colleagues in task 3, expressed that he was having trouble in adjusting the Uniformity objective, as well as seeing any significant changes to the generated suggestions. This observation confirms one suspicion we had, that the uniformity fitness function might need a rework, since its results are not clear in the generated suggestions.

20. Conclusions

We classify the Editor Buddy Lighting Mode impact on the participants as a success. Despite the issues pointed out in the Study Conclusions section, the overall impact it had on the participants experience of illuminating a level was positive. The participants interacted with the application's sliders, generated levels and exported several suggestions, even though we clearly specified that the use of the EB Lighting Mode was optional, all the participants decided to use it. Another reason for us to consider it a success, is the fact that all the participants shared the same opinion that the EB Lighting Mode is useful tool for inexperience level designers to create a level's illumination.

As we mentioned, one area we wish we could have tested more thoroughly was the computer as colleague paradigm. The participants always interacted with the EB Lighting Mode to generate the lighting of a level, but rarely returned to it for to ask the application for different suggestions. Although we believe this might be a problem of how the tests were performed, we still consider that the EBLM can be used and would do quite well in a computer as colleague environment, which could be proven with further testing.

In the end, we hope to have created a useful tool that helps novice designers in the process of illuminating a level.

21. Future work

With the help of the evaluation we realized that several aspects of the EB Lighting Mode could be improved or that it would benefit from the addition of certain features. These are the most relevant aspects we consider to be worthy of a future work:

Making changes to the user interface, in order to make clear the purpose of each of the sliders. Also the sub-objectives of the Objective algorithm should be better defined as being part of it. We noticed that some participants were at first confused by this element of the UI.

Merge the Uniformity sub-objective with the Global Intensity sub-objective. This makes for a cleaner and simpler UI and would prevent some of the doubts the users might have. A user can always make parts of the level darker or brighter by selecting paths.

Adding an option to create zones in the level. This can be achieved in the current state of the EBLM by creating multiple paths, however having the addition of simply selecting an entire zone would make more sense as user's would waste less time. This could be performed by holding a mouse click and dragging over the desired area.

In terms of overall algorithm improvements for a future work we recommend the implementation of some sort of restriction on the number of the light sources that can be added per suggestions. This would prevent levels filled with torches. Another alternative is to check how much does one torch contribute to the heatmap. For example if two torches are placed side-by-side, the overall difference on the heatmap would be insignificant, therefore one of the torches could be removed with almost no impact to the level's lighting.

One aspect we did not have time to pursue but it would be an interesting addition to this work, is the incorporation the EBLM with Delgado's Editor Buddy Puzzle Mode. With this addition, level designers would have a tool that worked as colleague in the process of creating a level layout, illuminating the level and that generated puzzles, for a more complete experience.

To conclude, a more ambitious goal is to incorporate the EBLM with other videogames and their level editors, this is arduous task as the EBLM is deeply integrated into the LoG2 environment.

References

[1] M. A. Boden. *The creative mind: Myths and mechanisms*. Routledge, 2004.

[2] M. A. Boden. Computer models of creativity. *AI Magazine*, 30(3):23–23, 2009.

[3] C. Darwin, A. R. Wallace, et al. Evolution by natural selection. *Evolution by natural selection.*, 1958.

[4] G. Delgado and C. Martinho. Co-creativity in videogame puzzle creation.

[5] A. Liapis, G. N. Yannakakis, and J. Togelius. Sentient sketchbook: Computer-aided game level authoring. In *FDG*, pages 213–220, 2013.

[6] T. Lubart. 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):365–369, 2005.

[7] P. Lucas and C. Martinho. Co-creativity in videogame level design.

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