# ChatGPT and Other Large Language Models as Evolutionary Engines for Online Interactive Collaborative Game Design

Pier Luca Lanzi
pierluca.lanzi@polimi.it
Politecnico di Milano
Milano, Italy

Daniele Loiacono
daniele.loiacono@polimi.it
Politecnico di Milano
Milano, Italy

## ABSTRACT

Large language models (LLMs) have taken the scientific world by storm, changing the landscape of natural language processing and human-computer interaction. These powerful tools can answer complex questions and, surprisingly, perform challenging creative tasks (e.g., generate code and applications to solve problems, write stories, pieces of music, etc.). In this paper, we present a collaborative game design framework that combines interactive evolution and large language models to simulate the typical human design process. We use the former to exploit users' feedback for selecting the most promising ideas and large language models for a very complex creative task—the recombination and variation of ideas. In our framework, the process starts with a brief and a set of candidate designs, either generated using a language model or proposed by the users. Next, users collaborate on the design process by providing feedback to an interactive genetic algorithm that selects, recombines, and mutates the most promising designs. We evaluated our framework on three game design tasks with human designers who collaborated remotely.

## CCS CONCEPTS

• **Computing methodologies → Genetic algorithms**; • **Theory of computation → Interactive computation**; • **Human-centered computing → Systems and tools for interaction design**.

## KEYWORDS

Collaborative Design, Large Language Models, Interactive Evolution

**ACM Reference Format:**
Pier Luca Lanzi and Daniele Loiacono. 2023. ChatGPT and Other Large Language Models as Evolutionary Engines for Online Interactive Collaborative Game Design. In *Genetic and Evolutionary Computation Conference (GECCO '23), July 15–19, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3583131.3590351


*GECCO '23, July 15–19, 2023, Lisbon, Portugal*

ACM ISBN 979-8-4007-0119-1/23/07…$15.00
https://doi.org/10.1145/3583131.3590351

## 1 INTRODUCTION

Design is a collaborative problem-solving activity that begins with a design brief providing an approximate description of a target objective. In the opening (divergent) phase, several initial ideas of possible solutions are proposed. These are the seeds for the next exploration (emergent) phase in which ideas are modified and combined to create new solutions over a series of iterations. Finally, in the closing (convergent) phase, developed ideas are critically examined, assessed, and the process result is synthesized [15, 34]. Interactive evolution [45] can model collaborative design processes effectively, and it has been widely applied to design and creative activities like fashion [22, 23, 31, 44], music composition [3, 17, 18, 52], video games [16, 40] and art in general [24, 25]. However, its application has been limited to domains where admissible design solutions could be encoded, randomly generated, mutated, and recombined. For example, it has never been applied to evolve free-form texts that would need advanced operators to maintain semantic coherence in generating the results.

Large Language Models (LLMs) have changed the landscape of natural language processing and human-computer interaction. These powerful tools can answer complex questions and perform challenging creative tasks such as writing screenplays and theater scripts [30], solving complex math problems [9], theorem proofing [38], generating code to solve specific tasks [8], generate descriptions from images [7], etc. More interestingly, they can also generate random texts from high-level instructions, combine two or more texts, and create variations of existing texts—all while maintaining semantic coherence in the generated texts. These functionalities can easily map to the genetic operators at the core of most evolutionary algorithms. Thus, we can use LLMs to implement evolutionary operators that can manage free-form text individuals while also maintaining semantic coherence—something that would be difficult, if not impossible, to achieve with traditional representations.

In this paper, we present an online collaborative design framework that combines interactive evolution with large language models for evolving game design ideas (*game concepts*) represented as free-form texts. The online interactive evolutionary algorithm shows the ideas to the human designers and collects their feedback; the large language model implements all the operators for the random generation, the recombination (crossover), and the variation (mutation) of ideas. Our framework is very general in that it can support any design process based on textual information (and could be easily extended to integrate visuals integrating tools like Dall-E 2 [36]); it can use any viable LLM and therefore, it is intrinsically multilingual; it can be accessed using any medium that can visualize texts like a webpage [5, 6], a 2D-grid displayed on screen [1, 51], or any viable messaging platform that supports user polls or other ways to collect users' feedback.

We performed a preliminary evaluation of our framework on three creative tasks, the design of a board game, the design of a video game, and the generation of game ideas during the 2023 Global Game Jam. The evaluation involved around 80 users (junior and senior game designers). The goal of the two game design tasks was to simulate a typical idea-generation process; accordingly, we set a fixed time limit of four working days for each task. The interactive evolutionary algorithm communicated with the human designers using the Telegram messaging platform [28] while the genetic operators were implemented using ChatGPT [35]. Each task was initiated with a design brief [15] that was sent to the designers as the first message on a Telegram group. The brief was followed by a series of initial design ideas (some randomly generated using the LLM, some collected from existing designs) — the opening (divergent) phase [15]. Then, over the next four days, users evaluated the design ideas they received using the Telegram poll system; the evolutionary algorithm employed the users' feedback to select, recombine and mutate the existing ideas into new ones presented next. Instead, the evaluation performed during the 2023 Global Game Jam was conceived as an open-ended brainstorming activity that lasted less than 24 hours to fit the tight schedule of the event.

There are few examples of applications of LLMs to idea generation and creativity in games. For example, Zhu and Luo [54] explored a generative approach for design ideation combining a language model and a knowledge based of existing patents to generate new ideas. van Stegeren and Myundefinedliwiec [46] investigated the application of GPT-2 for generating dialogues in role-playing games for non-player characters while Värtinen et al. [48] applied GPT-2 [39] and GPT-3 [4] to generate description of quests in the same type of games. Frans [12] presented a framework for building *Language Model Games* whose main mechanic involved players manipulating a language model into behaving in a desired manner with a demo game called AI Charades. However, to the best of our knowledge, our framework is the first example of interactive evolutionary algorithm that apply large language models to implement genetic operator and combines them with interactive evolution to mimic a full-fledged design session from start to end.

## 2 RELATED WORK

Interactive evolution [45] dates back to Richard Dawkins's *The Blind Watchmaker* and *Biomorphs* [10]. Over the years, this approach has been applied to several creative tasks, for which human evaluation might provide a better (if not the only) alternative to the definition of an objective (or fitness) function to evaluate candidate solution, such as fashion [22, 23, 31, 44], ergonomics [2, 49], yacht design [21], visual effects [11, 20], synthetic images [42], music [3, 17, 18, 52], video games [16, 40] and art in general [24, 25].

Interactive evolutionary frameworks collect human evaluations either implicitly or explicitly. In the former case, the systems track human actions, sometimes building a model of users' behavior and deriving an assessment of their preferences [19, 53]. Galactic Arms Race (GAR) [16] is an example of implicit evaluation. It was the first video game using interactive evolution for content generation; it included a weapon system that automatically evolved new weapons based on players' behavior. Petalz [40] was a Facebook game about growing a garden of procedurally generated flowers; players would select what flowers they liked most and wanted to breed; human evaluation used by the interactive evolutionary algorithm was implicitly implemented within the farming mechanics.

Most often, interactive evolutionary frameworks involve an explicit human-based evaluation of candidates and, similarly to Dawkins' *Biomorphs* [10], have a grid-like interface that allows users to evaluate candidate designs iteratively. Hoover et al. [17, 18] integrated interactive evolution with functional scaffolding for generating drum tracks [17] and accompaniments for existing musical pieces [18] using a compositional pattern-producing network. At first, users select a musical piece (in MIDI format) and which parts should be elaborated by the evolutionary algorithm (e.g., the piano, guitar, or bass guitar). Next, users customize and refine the computer-generated accompaniment through an interactive selection and mutation of compositional networks. Noticeably, the user interface in [18] is similar to other music composition tools instead of the grid-based structure used in most frameworks. Cardamone et al. [5, 6] developed a tool that combined an interactive genetic algorithm with procedural content generation to evolve tracks for two open-source racing games. The tool had a web frontend that remained active for several years and currently appears to be discontinued, although still online. Ølsted et al. [55] developed a tool for the interactive evolution of first-person-shooter maps inspired by the popular multiplayer game Counter-Strike. While the tool of [5, 6] is external to the actual racing games, the tool of Ølsted et al. [55] is integrated within the game (FPSEvolver) and operated *live*, collecting players' feedback during the usual map voting and selection process—a typical phase in multiplayer first-person-shooter games. Pirovano et al. [37] developed a tool for creating 3D sword models that combined procedural content generation, interactive evolution, and a grid-based interface to let users explore such a peculiar design space. Generative Adversarial Networks (GANs) [14] represented a milestone in procedural content generation and have been applied to various creative tasks. They learn the probability distribution that generates a collection of training examples and can generate more samples from randomly generated input vectors of Gaussian noise, which form the latent space. Several authors developed methods to explore the latent space to guide the generation of examples [13, 47], including interactive evolution [1, 41]. Bontrager et al. [1] combined generative adversarial networks for image generation with interactive evolution to explore the latent space to improve the quality of sampled images for specific targets. Schrum et al. [41] applied interactive evolution for exploring the latent space for generating levels for the video games *Super Mario Bros.* and *The Legend of Zelda*. Both [1] and [41] require an explicit human-based evaluation using a grid-like interface similar to the ones used in [5, 6, 10, 37].

## 3 THE FRAMEWORK ARCHITECTURE

Our framework combines interactive evolution and large language models to simulate the typical human game design process. It comprises three components: (i) the online database of game design ideas; (ii) the evolutionary algorithm; and (iii) the agent responsible

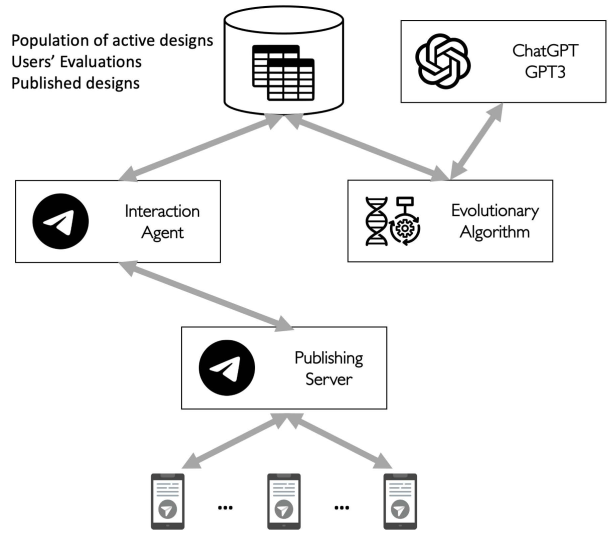



**Figure 1: The framework architecture when collaboration is based on the Telegram messaging platform.**

for publishing the designs and collecting users' feedback. Figure 1 shows the structure of our framework; in this case, the interaction with the human designers uses the Telegram messaging system, which is the one we employed in the experiments discussed in this study.

**The database** contains (i) a list of game design ideas that have been published and are currently under evaluation; (ii) the list of submitted evaluations that are used to compute the fitness; (iii) the current population of active design ideas; this includes both the designs that been already published and under evaluation by the users and the ones that have been already generated but not published yet.

**The evolutionary engine** is implemented as a steady-state genetic algorithm that, at each iteration, (i) selects promising game design ideas from the current population; (ii) applies recombination and mutation (implemented using a large language model like ChatGPT or another LLM like DaVinci GPT-3 [4]); (iii) inserts the newly generated individuals in the population; finally, (iv) deletes individuals to keep the population size constant.

**The interaction agent** manages the communication between the framework and the publishing server; it publishes the design concepts in the population and collects the users' feedback. In Figure 1, we show the architecture used in the experiments discussed in this paper with an interaction agent implemented as a Telegram bot that publishes the design concepts on a Telegram group through the Telegram servers. However, our framework also supports publishing using web servers (as done in [6, 42]).

Note that, when using a publishing-subscribe platform like Telegram, the publication of design concepts is timed to limit the number of concepts (and message notifications) the users receive at any given time (see Section 5). Accordingly, we maintain two separate tables, one for the population of active designs (Figure 1) and one for the published designs currently being evaluated. In contrast, when interacting through a web server, the agent will publish the population all at once; users will autonomously decide how many concepts they want to view and evaluate using the online navigation interface (similarly to [6, 42]).

**User Evaluation.** Most of the interactive evolutionary algorithms we examined (Section 2) ask for qualitative evaluations based, for example, on basic voting systems [41, 55], scales with a limited number of values [5, 6], or let users actively select what individual should go to the next generation [1, 51]. In our framework, we adopted a simple three-value scale (positive, neutral, and negative) that let users express how they felt about a given design idea. Note that, the evaluation is anonymous and users cannot see how other people evaluated an idea before they submitted their evaluation. When using the Telegram poll-based interface, only the summary statistics about the submitted evaluations will appear on the group chat; when using the web-based interface, users can see only their evaluation.

## 3.1 The Workflow

The process starts with a short *design brief*, describing the activity goals [50]. The evolutionary algorithm uses the brief to generate the initial population either (i) by asking ChatGPT or another language model to generate random solutions (similarly to the traditional population initialization in evolutionary algorithms); (ii) by introducing human-designed concepts originating from the brief (e.g., by asking participants to propose solutions or taking existing well-known solutions); or (iii) by combining the previous options (similarly to the half and half initialization in Genetic Programming [26]). The individuals in the initial population are published either at once (when collaboration is web-based) or timely (when using a publishing-subscribe platform like Telegram). Next, the framework waits for feedback from the users, who can score each published concept. When sufficient evaluations have been submitted for all the published concepts, the evolutionary algorithm starts its select-recombine-mutate cycle: it selects two individuals from the population using tournament selection of size two; with a given probability, it applies recombination to generate one offspring that is then mutated; finally, the offspring is added to the population while the worst individual is deleted to keep the population size constant. The new individual becomes eligible for publication; depending on the platform used, the individual is published immediately or in a given time frame. The next iteration of the evolutionary algorithm will start when a sufficient number of evaluations have been submitted for the newly published individuals.

# 4 IMPLEMENTING GENETIC OPERATORS WITH LARGE LANGUAGE MODELS

LLMs are queried using questions (i.e., *prompts*) expressed in natural language. The quality of their answers heavily relies on how well the prompts can specify the objective for the target LLM. In our framework, individuals represent game design ideas expressed as free-form text; all the operators manipulating them are implemented by querying an LLM. Accordingly, we initially studied how to design prompts so that the LLM answers could mimic the behavior of the three operators we needed to implement the evolutionary algorithm: random initialization of individuals, recombination or crossover, and variation or mutation.

### 4.1 Random Initialization of Individual

In our framework, the population can be initialized (i) with random individuals, (ii) with existing designs, or (iii) with a combination of the previous approaches. Random individuals (game ideas) ignite the design process and need to provide an approximate indication of the target objective. In our initial analysis of feasible prompts to implement evolutionary operators we noted that it is convenient to ask the system to structure the answer. Accordingly, the prompts used to generate the initial random game ideas consist of (i) a design brief defining the objective and its constraints in an organized structure; and (ii) a request to *act as a game designer* and generate a novel design based on the brief. Table 1a shows a prompt to generate a random idea for a generic board game. The bold text shows the actual request while the remaining part specifies how the answer should be structured. Note that, the prompt also specifies the length of each section to avoid very long game ideas that could be more difficult to evaluate by the designers. When the request is less generic, the prompt begins with a text that serves as inspiration or as an example of the objective. Table 1b shows a prompt to generate a video game that uses a minimal amount of resources. This example is inspired to the study of minimalism in video game design [32, 33]. The first sentence of the prompt is a statement that serves as an inspiration and sets the basis of what is going to be asked. The second sentence (in bold) shows the actual request (as in the previous case). The remaining text specifies the answer structure. Table 1c shows an example of prompt that was used to generate random individuals for an experiment run during the 2023 Global Game Jam[1] in which the goal was to create a game based on the theme *roots*. Also in this case, the first sentence was used as a sort of inspiration. However, since there could be many interpretations of *roots* (it could refer to cultural heritage, botanic, etc.), the sentence contained the marker $\langle INTERPRETATION \rangle$ which was replaced with sentences suggesting different interpretations (Table 1d). The process generated a great variety of prompts for the same theme.

### 4.2 Recombination (Crossover)

The prompt to implement the recombination of two individuals has a general structure (Table 1e). It starts with the same brief used to generate random individuals. For example, when recombining two video games using minimal resources, the marker $\langle BRIEF \rangle$ would be replaced with the sentence "A white pixel is the minimum amount [...]" from Table 1b. The brief is followed by two examples of games in which the markers $\langle INDIVIDUAL \rangle$ are replaced with the text description of the two parents selected for recombination. Next, the actual request (highlighted in bold) and two sentences specifying a series of constraints to avoid answers that are blunt rephrasing of the parents. Note that, the recombination prompt generates one single offspring and to generate a second offspring (as in traditional crossover) another prompt using two different parents should be used. We tried several prompts to generate two offspring at once but we noted that such approach would lead with very similar offspring so we opted for a recombination prompt that would generate only one offspring.

### 4.3 Variation (Mutation)

The prompt for variation also starts with the brief used to generate random individuals (marker $\langle BRIEF \rangle$ in Table 1f) followed by (i) an existing game idea to be mutated (marker $\langle INDIVIDUAL \rangle$), (ii) the request to the LLM (highlighted in bold) specifying the focus of the mutation (marker $\langle MUTATION \rangle$), and (iv) two sentences specifying a series of constraints to avoid answers that would be simple rephrasing of the parent. Marker $\langle MUTATION \rangle$ specifies the focus of the mutation and it is randomly replaced with a gameplay aspects such as, *the goal of the game*, *the game's resources*, *the level design*, *the player's input*, etc. We experimented with mutation operators that would work at a lower level of detail (e.g., by substituting some keywords of the game idea with synonyms or antonyms generated using word embeddings); however, the approach proved less effective as it did not introduce enough variation.

## 5 EXPERIMENTAL EVALUATION

We performed a preliminary evaluation of our framework with human subjects (junior and senior game designers) on three design tasks. One focused on designing a video game with the least amount of resources possible while having interesting mechanics—a minimalistic video game [32, 33]. One concerned the creation of a generic board game using only one board of any layout and a set of pieces. Finally, we evaluated our framework with people participating in the 2023 Global Game Jam as a tool for the initial brainstorming phase of the jam. All the experiments were run using the Telegram-based version of our framework (Figure 1).

### 5.1 Board Game and Minimalistic Video Game Design

These two experiments were aimed at simulating the typical idea-generation process in a controlled environment (e.g., a design studio) that usually is well-structured [15], has clear goals, and has a fixed duration. We set up two Telegram groups, one for each task. We sent a call for volunteers to game designer communities on Facebook and mailing lists, providing a link for joining each Telegram group. In total, 44 individuals joined one or both groups, with 35 actively engaging in the experiments. Among these participants, 23 were involved in both experiments, while 8 participated exclusively in the video game task, and 4 in the board game task. Not all participants disclosed their identities, limiting our ability to provide comprehensive demographic data. In response to concerns regarding the availability for continuous feedback throughout the day, we scheduled the publication of design ideas during three distinct periods: morning, noon, and late afternoon. The experiments began with a brief that we posted on each Telegram group with information about the timing of the postings so that people could plan their participation. The participants were not informed about the evolutionary algorithm that was running in the background and about the use of large language models. Instead, we informed them that AI will be used to support the design process based on their feedback. The experiments lasted four days, Tuesday to Friday; we allowed a grace period of a few hours, until Saturday morning, to

[1]https://globalgamejam.org

**Act as a board game designer and create a unique board game with the following restrictions:** the game can involve only a board (of any layot) and a set of pieces (of any number, and type). Organize your response as follows: 1) Name of the Game; 2) Number of players and cooperative/competitive game; 3) Game Board: describe the layout of the game board, the number and shape of its tiles (max 200 characters); 4) Game Pieces: describe the number of pieces, their type, and their distribution among players (max 300 characters); 5) Rules: provide a detailed explanation of how the game is played, including how to move pieces and take turns (max 800 characters); 6) Objective: provide a clear description of the win and lose conditions of the game (max 200 characters). Keep the game description as simple as possible, without including visual details, theme and story.

(a)

A white pixel is the minimum amount of information we can show on-screen, and pressing a key (or a button) is the least interaction we can ask players. **Act as a game designer and design a minimalist game.** Organize your response as follows: 1) the name of the game; 2) game concept: describe the idea, the core mechanics, the goal of the game, and player controls (max 1000 chars); 3) game resources: describe the graphical elements involved by the game, which type of player input is required, and any additional resources required by the game (e.g., sound); 4) level design: describe how to do the level design for this game (max 300 chars); 5) game instructions: provide the game instructions for the player (max 300 chars). Do not add additional information.

(b)

**Act as a game designer and describe (max 1000 chars), a concept for a video game about the theme** "roots: $\langle INTERPRETATION \rangle$". Both the game mechanics and the interaction of the players with the game should involve elements directly connected to the theme of the game.
Organize your response as follows: 1) name of the game; 2) game concept: describe the idea, the core mechanics, the goal of the game, and player controls (max 800 chars); 3) level design: describe how to do the level design for this game (max 400 chars); 4) game instructions: provide the game instructions for the player (max 300 chars). Do not add additional information.

(c)

- "the part of a plant that is below the ground and that absorbs water and minerals from the soil"
- "the part of a tooth within the socket"
- "one or more progenitors of a group of descendants"
- "the essential core"
- ...

(d)

$\langle BRIEF \rangle$. Given these two examples of minimalist game.
EXAMPLE1: $\langle INDIVIDUAL \rangle$
EXAMPLE2: $\langle INDIVIDUAL \rangle$
**Act as a game designer and recombine these two games to create a novel game.** In your response do not include any introduction or final comment. In the response, please avoid references to the two games recombined.

(e)

$\langle BRIEF \rangle$. Given the following example of minimalist game.
EXAMPLE: $\langle INDIVIDUAL \rangle$
**Act as a game designer and design a novel game as a variation of the given example, by changing only** $\langle MUTATION \rangle$. In your response do not include any introduction or final comment. In the response, please avoid explicit reference to the original game.

(f)

**Table 1: Example of prompts for generating (a) a generic board game; (b) a minimalistic video game; (c) a game based on the theme *roots* in which the marker $\langle INTERPRETATION \rangle$ can be replaced with sentences suggesting different theme interpretation, like the ones listed in (d). Prompts to implement (e) recombination (crossover) and (f) variation (mutation); $\langle BRIEF \rangle$ identify the brief used to generate the random individuals, $\langle INDIVIDUAL \rangle$ an individual selected from the population, and $\langle MUTATION \rangle$ a high-level characteristic of gameplay.**

let participants submit their final evaluations. We wanted to focus the design process and limit the number of ideas that the participants had to evaluate initially (Section 3). Accordingly, we chose a small population size (10 individuals); selection, recombination, and variation were implemented using ChatGPT [35]; they were triggered as soon as 25 new evaluations were submitted; selection was implemented using a tournament selection of size 2; recombination probability was set to 0.7; variation was always applied.

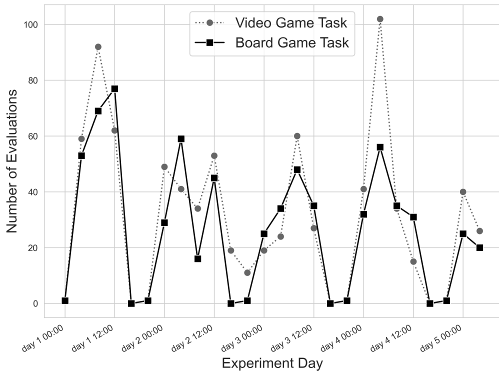



**Figure 2: Number of evaluations submitted during the experiments.**

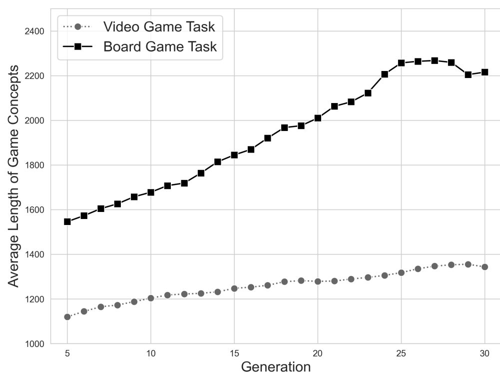
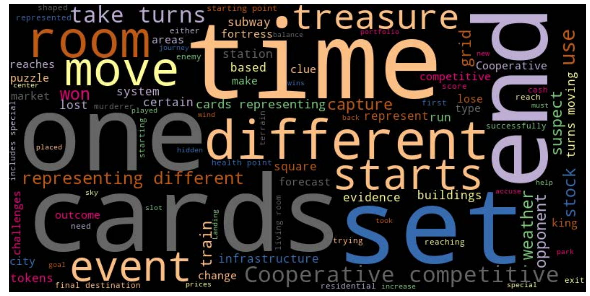


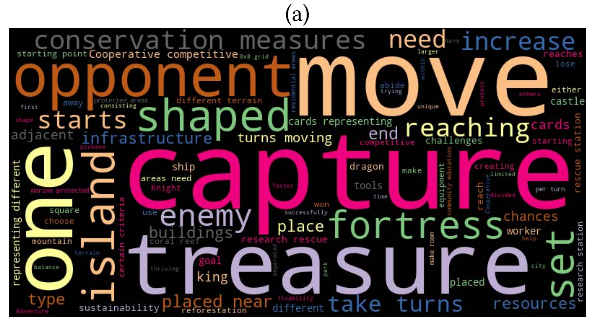

**Figure 3: Average length (as number of characters) of the game concepts, as number of characters in the population computed as the moving average over the last five activation of the evolutionary algorithm.**

Over the four days, the evolutionary algorithm performed 30 iterations and a total of 40 game ideas were evaluated for each task; board game ideas received a total of 799 evaluations; video game ideas received 1025 evaluations overall. Figure 2 shows the number of evaluations submitted over the four and half days. The plots follow the schedule with peaks around the publication slots and no evaluations during the night time. Figure 3 shows the average length of the concepts (as the number of characters) in the population computed as the moving average over the last 5 activations of the evolutionary algorithm. As can be noted, board game concepts tend to be longer on average (Figure 3) and therefore require more effort to evaluate; as a consequence, the board game concepts received fewer evaluations than the ones of minimalistic video games as Figure 2 shows.

Figure 4 compares the word cloud generated using all the board game ideas in the original population (4a) with the one generated from the final population (4b). The clouds share some general terms (e.g., moves, starts), but the game concepts in the final population focus on completely different concepts like the terms related to ecosystem maintenance (e.g., island, conservation, resources). Due to lack of space, we cannot report the word clouds for the video




(a)




(b)

**Figure 4: Word cloud for (a) the initial population and (b) the final population of the board game design task.**

game task as well as examples of game concepts generated (we refer the reader to [27] for additional details).

Finally, we asked participants to complete a form with four open questions. The questions and a summary of the participants' opinions, based on 10 responses collected for the board game task and 13 for the video game task, are presented below.

**(i) Which game concepts did they like most, and why?** Many games in the final population shared similar elements and mechanics. Although participants did not prefer a single game over the 40 evaluated, there was a common preference for concepts that involved one specific mechanic for each task. For the board game design task, most participants liked a cooperative mechanic that asked players to maintain the equilibrium of an ecosystem by placing tiles on a board. Interestingly, no elements of this mechanic were present in the initial population but the mechanic emerged later in the experiment in games with different narratives (e.g., one set in an island environment, one in an urban environment). In the video game task, most participants liked a game mechanic involving *reflection* in its different interpretations (e.g., reflection of light in some games, reflection of bouncing bullets in other ones). In this case, the initial population contained one game description using reflection of bullets however the light reflection mechanics appeared much later in the population.

**(ii) What were the positive features of the generated concepts?** Participants mainly liked the emergence of novelty in mechanics, player inputs, and unexpected element combinations. For instance, in the video game task, a novel mechanic involving color manipulation combined with shadow and light mechanics positively impressed some participants. Additionally, the emergence of different narratives (e.g., environmental protection, urban planning) and scenarios (e.g., a jungle, an island, etc.) in the board game task was positively received.

**(iii) What were the negative features of the generated concepts?** The two main issues reported were the degree of similarity and incoherence in some generated concepts. The similarity issue resulted from the small population size (10 individuals) and a limited number of steady-state iterations (30), which increased the likelihood of presenting several vations of the same concept. However, this design choice was necessary to reduce the evaluation burden. The incoherence issue stemmed from the current limitations of LLMs [29, 43]. Despite we might expect future improvements in this respect, the lack of coherence is a common problem in these models and it is not easy to solve.

**(iv) How could the collaborative design experience be improved?** Interestingly, most participants did not suggest increasing their agency but rather improving concept presentation. Some participants recommended presenting concepts more visually (e.g., using images or videos), while others suggested highlighting similarities and differences among the concepts to facilitate evaluation.

## 5.2 2023 Global Game Jam

We performed a final experiment during the 2023 Global Game Jam that ran February 3-5, 2023, from Friday afternoon to Sunday evening. The event always starts at 5 pm on Friday with the publication of the theme (this year was *roots*) and ends Sunday evening with the presentation of the games. The jam is a global event run with a local timing in that it starts at 5 pm and ends Sunday around 6 pm *in every time zone*; jam sites in the Asia continent are the first to start, those in the Hawaiian time zone are the last ones. We contacted organizers of the Global Game Jam sites in our country, sent them a link to a dedicated Telegram group, and asked them to advertise the tool's availability to the participants of their jam sites; around 35 people joined the group, most of them anonymously. In the previous experiments, at the beginning, participants received a brief with a list of constraints and were instructed to cooperate toward a common objective. In contrast, in this case, participants had a theme (*roots*) that could lead to several different interpretations; they did not have constraints and did not have to collaborate but they brainstormed for ideas to develop their own game. In addition, the time window was more limited since the game must be implemented by Sunday afternoon, so teams need to come up with a game idea by Saturday around noon or early afternoon at the latest. When the theme was revealed on February 3 at 5 pm, we published the first ten game ideas randomly generated using the procedure discussed in Section 4; the framework remained online until Saturday night. The evolutionary operators were implemented using DaVinci GPT-3 [4]; the selection was triggered every 5 new evaluations; recombination and mutation probabilities were set as in the previous experiments. The evolutionary algorithm ran for 15 iterations with a number of evaluations much lower than what was recorded in the previous experiments. This is probably due to the short span that jammers can dedicate to the brainstorming process and the in-person nature of the event. Global Game Jam is a physical jam that encourages meeting, working, and sharing ideas with new people. Thus, it is no surprise that an online framework like ours did not fit well in this in-person scenario.

# 6 CONCLUSIONS

We presented an online collaborative design framework combining interactive evolution and large language models for helping users evolve design ideas represented as free-form text. Our framework can support collaboration through traditional web-based grid interfaces [6, 42] as well as messaging systems, like Telegram, which in our opinion might fit free-text content better. We evaluated our framework in three scenarios involving around 80 participants. Two scenarios mimicked the typical collaborative design process with objectives specified at the start and a fixed duration. The third one was conceived as an open-ended brainstorming activity that took place during the 2023 Global Game Jam. The first two used an evolutionary engine based on ChatGPT; the third one employed DaVinci GPT-3 for the same purpose. All the experiments were run using the Telegram-based interface of our framework. Overall, we received positive feedback from the participants. In our experience, we did not notice any difference between the two large language models that are, in our opinion, both viable options. We focused our evaluation on game design since we had access to several junior and senior game designers employed in the industry. However, the framework can be applied to any design task which can be described in free-text form. And if needed, it could be easily extended with text-to-image platforms (like Dall·E 2[36] or Midjourney[2]) to enrich design ideas with captivating visuals.

# REFERENCES


[1] Philip Bontrager, Wending Lin, Julian Togelius, and Sebastian Risi. 2018. Deep Interactive Evolution. In *Computational Intelligence in Music, Sound, Art and Design - 7th International Conference, EvoMUSART 2018, Parma, Italy, April 4-6, 2018, Proceedings (Lecture Notes in Computer Science, Vol. 10783)*, Antonios Liapis, Juan Jesús Romero Cardalda, and Anikó Ekárt (Eds.). Springer, 267–282. https://doi.org/10.1007/978-3-319-77583-8_18

[2] A.M. Brintrup, J. Ramsden, H. Takagi, and A. Tiwari. 2008. Ergonomic Chair Design by Fusing Qualitative and Quantitative Criteria Using Interactive Genetic Algorithms. *Evolutionary Computation, IEEE Transactions on* 12, 3 (June 2008), 343 –354. https://doi.org/10.1109/TEVC.2007.904343

[3] Andrew R. Brown, Matthew Horrigan, Arne Eigenfeldt, Toby Gifford, Daniel Field, and Jon McCormack. 2018. Interacting with Musebots. In *18th International Conference on New Interfaces for Musical Expression, NIME 2018, Blacksburg, VA, USA, June 3-6, 2018*. nime.org, 19–24. http://www.nime.org/proceedings/2018/nime2018_paper0004.pdf

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, and al. 2020. Language Models are Few-Shot Learners. https://doi.org/10.48550/ARXIV.2005.14165

[5] Luigi Cardamone, Pier Luca Lanzi, and Daniele Loiacono. 2015. TrackGen: An interactive track generator for TORCS and Speed-Dreams. *Appl. Soft Comput.* 28 (2015), 550–558. https://doi.org/10.1016/j.asoc.2014.11.010

[6] Luigi Cardamone, Daniele Loiacono, and Pier Luca Lanzi. 2011. Interactive evolution for the procedural generation of tracks in a high-end racing game. In *13th Annual Genetic and Evolutionary Computation Conference, GECCO 2011, Proceedings, Dublin, Ireland, July 12-16, 2011*, Natalio Krasnogor and Pier Luca Lanzi (Eds.). ACM, 395–402. https://doi.org/10.1145/2001576.2001631

[7] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. 2020. Generative Pretraining From Pixels. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 1691–1703. http://proceedings.mlr.press/v119/chen20s.html

[8] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, and al. 2021. Evaluating Large Language Models Trained on Code. https://doi.org/10.48550/ARXIV.2107.03374

[9] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training Verifiers to Solve Math Word Problems. https://doi.org/10.48550/ARXIV.2110.14168

[10] Richard Dawkins. 1986. *The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe without Design.* Norton & Company, Inc.


[2]https://www.midjourney.com

[11] Marc Ebner, Markus Reinhardt, and Jürgen Albert. 2005. Evolution of Vertex and Pixel Shaders. In *Genetic Programming, 8th European Conference, EuroGP2005, Lausanne, Switzerland, March 30 - April 1, 2005, Proceedings (Lecture Notes in Computer Science, Vol. 3447)*, Maarten Keijzer, Andrea Tettamanzi, Pierre Collet, Jano I. van Hemert, and Marco Tomassini (Eds.). Springer, 261–270. https://doi.org/10.1007/978-3-540-31989-4_23

[12] Kevin Frans. 2021. AI Charades: Language Models as Interactive Game Environments. In *2021 IEEE Conference on Games (CoG)*. 1–2. https://doi.org/10.1109/CoG52621.2021.9619126

[13] Edoardo Giacomello, Pier Luca Lanzi, and Daniele Loiacono. 2019. Searching the Latent Space of a Generative Adversarial Network to Generate DOOM Levels. In *IEEE Conference on Games, CoG 2019, London, United Kingdom, August 20-23, 2019*. IEEE, 1–8. https://doi.org/10.1109/CIG.2019.8848011

[14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144. https://doi.org/10.1145/3422622

[15] Dave Gray and Sunni Brown. 2010. *Gamestorming: A Playbook for Innovators, Rulebreakers, and Changemakers.* O'Reilly.

[16] Erin J. Hastings, Ratan K. Guha, , and Kenneth O. Stanley. 2009. Automatic Content Generation in the Galactic Arms Race Video Game. *IEEE Transactions on Computational Intelligence and AI in Games* 4, 1 (2009), 245–263.

[17] Amy K. Hoover and Kenneth O. Stanley. 2009. Exploiting functional relationships in musical composition. *Connect. Sci.* 21, 2&3 (2009), 227–251. https://doi.org/10.1080/09540090902733871

[18] Amy K. Hoover, Paul A. Szerlip, and Kenneth O. Stanley. 2011. Interactively evolving harmonies through functional scaffolding. In *13th Annual Genetic and Evolutionary Computation Conference, GECCO 2011, Proceedings, Dublin, Ireland, July 12-16, 2011*, Natalio Krasnogor and Pier Luca Lanzi (Eds.). ACM, 387–394. https://doi.org/10.1145/2001576.2001630

[19] Gregory S. Hornby and Josh C. Bongard. 2012. Accelerating human-computer collaborative search through learning comparative and predictive user models. In *Genetic and Evolutionary Computation Conference, GECCO '12, Philadelphia, PA, USA, July 7-11, 2012*, Terence Soule and Jason H. Moore (Eds.). ACM, 225–232. https://doi.org/10.1145/2330163.2330196

[20] Andrew Howlett, Simon Colton, and Cameron Browne. 2010. Evolving pixel shaders for the prototype video game Subversion. In *Proceedings of the 3rd International Symposium on AI and Games - A Symposium at the AISB 2010 Convention (Proceedings of the 3rd International Symposium on AI and Games - A Symposium at the AISB 2010 Convention)*. 41–46. AISB Symposium on AI and Games 2010 ; Conference date: 29-03-2010 Through 01-04-2010.

[21] Shahroz Khan, Erkan Gunpinar, and Bekir Sener. 2019. GenYacht: An interactive generative design system for computer-aided yacht hull design. *Ocean Engineering* 191 (2019), 106462. https://doi.org/10.1016/j.oceaneng.2019.106462

[22] Hee-Su Kim and Sung-Bae Cho. 2000. Application of interactive genetic algorithm to fashion design. *Engineering Applications of Artificial Intelligence* 13, 6 (2000), 635 – 644. https://doi.org/DOI:10.1016/S0952-1976(00)00045-2

[23] Hee-Su Kim and Sung-Bae Cho. 2005. *Fashion Design Using Interactive Genetic Algorithm with Knowledge-based Encoding.* Springer Berlin Heidelberg, Berlin, Heidelberg, 411–434. https://doi.org/10.1007/978-3-540-44511-1_19

[24] Taras Kowaliw, Alan Dorin, and Jon McCormack. 2012. Promoting Creative Design in Interactive Evolutionary Computation. *IEEE Trans. Evol. Comput.* 16, 4 (2012), 523–536. https://doi.org/10.1109/TEVC.2011.2166764

[25] Taras Kowaliw, Jon McCormack, and Alan Dorin. 2011. An interactive electronic art system based on artificial ecosystemics. In *2011 IEEE Symposium on Artificial Life, ALIFE 2011, Paris, France, April 13-15, 2011*. IEEE, 162–169. https://doi.org/10.1109/ALIFE.2011.5954645

[26] John R. Koza. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* The MIT Press.

[27] Pier Luca Lanzi and Daniele Loiacono. 2023. ChatGPT and Other Large Language Models as Evolutionary Engines for Online Interactive Collaborative Game Design. arXiv:2303.02155 [cs.AI]

[28] Telegram FZ LLC and Telegram Messenger Inc. 2023. Telegram. https://telegram.org

[29] Kyle Mahowald, Anna A. Ivanova, Idan A. Blank, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. 2023. Dissociating language and thought in large language models: a cognitive perspective. arXiv:2301.06627 [cs.CL]

[30] Piotr Mirowski, Kory W. Mathewson, Jaylen Pittman, and Richard Evans. 2022. Co-Writing Screenplays and Theatre Scripts with Language Models: An Evaluation by Industry Professionals. https://doi.org/10.48550/ARXIV.2209.14958

[31] P.Y. Mok, Jie Xu, X.X. Wang, J.T. Fan, Y.L. Kwok, and John H. Xin. 2013. An IGA-based design support system for realistic and practical fashion designs. *Computer-Aided Design* 45, 11 (2013), 1442–1458. https://doi.org/10.1016/j.cad.2013.06.014

[32] David Myers. 2009. In search of a minimalist game. In *Proceedings of the 2009 DiGRA International Conference: Breaking New Ground: Innovation in Games, Play, Practice and Theory, DiGRA 2009, London, UK, September 1-4, 2009*, Tanya Krzywinska, Helen W. Kennedy, and Barry Atkins (Eds.). Digital Games Research Association. http://www.digra.org/digital-library/publications/in-search-of-a-minimalist-game/

[33] Andrew Nealen, Adam Saltsman, and Eddy Boxerman. 2011. Towards minimalist game design. In *Foundations of Digital Games, FDG'11, Bordeaux, France, June 28 - July 1, 2011*, Marc Cavazza, Katherine Isbister, and Charles Rich (Eds.). ACM, 38–45. https://doi.org/10.1145/2159365.2159371

[34] Don Norman. 2013. *The Design Of Everyday Things.* Basic Books. Revised edition.

[35] OpenAI. 2021. ChatGPT: a Generative Pre-training Transformer Model for Conversational AI. https://openai.com/models/gpt-3-citation/.

[36] OpenAI. 2022. DALL·E 2. https://openai.com/dall-e-2/. Retrieved 2 February 2023.

[37] Michele Pirovano, Renato Mainetti, and Daniele Loiacono. 2015. Volcano: An interactive sword generator. In *2015 IEEE Games Entertainment Media Conference, GEM 2015, Toronto, ON, Canada, October 14-16, 2015*, Elena G. Bertozzi, Bill Kapralos, Nahum D. Gershon, and Jim R. Parker (Eds.). IEEE, 1–8. https://doi.org/10.1109/GEM.2015.7377226

[38] Stanislas Polu and Ilya Sutskever. 2020. Generative Language Modeling for Automated Theorem Proving. https://doi.org/10.48550/ARXIV.2009.03393

[39] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019). https://openai.com/blog/better-language-models/

[40] Sebastian Risi, Joel Lehman, David B. D'Ambrosio, Ryan Hall, and Kenneth O. Stanley. 2012. Combining Search-Based Procedural Content Generation and Social Gaming in the Petalz Video Game. In *AIIDE*, Mark Riedl and Gita Sukthankar (Eds.). The AAAI Press.

[41] Jacob Schrum, Jake Gutierrez, Vanessa Volz, Jialin Liu, Simon Lucas, and Sebastian Risi. 2020. Interactive Evolution and Exploration within Latent Level-Design Space of Generative Adversarial Networks. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference* (Cancún, Mexico) *(GECCO '20)*. Association for Computing Machinery, New York, NY, USA, 148–156. https://doi.org/10.1145/3377930.3389821

[42] Jimmy Secretan and Nicholas Beato. 2008. Picbreeder: evolving pictures collaboratively online. In *CHI*. 1759–1768.

[43] Murray Shanahan. 2022. Talking About Large Language Models. arXiv:2212.03551 [cs.CL]

[44] Nazanin Alsadat Tabatabaei Anaraki. 2017. Fashion Design Aid System with Application of Interactive Genetic Algorithms. In *Computational Intelligence in Music, Sound, Art and Design*, João Correia, Vic Ciesielski, and Antonios Liapis (Eds.). Springer International Publishing, Cham, 289–303.

[45] H. Takagi. 2001. Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. *Proc. IEEE* 89, 9 (2001), 1275–1296. https://doi.org/10.1109/5.949485

[46] Judith van Stegeren and Jakub Myundefinedliwiec. 2021. Fine-Tuning GPT-2 on Annotated RPG Quests for NPC Dialogue Generation. In *Proceedings of the 16th International Conference on the Foundations of Digital Games* (Montreal, QC, Canada) *(FDG '21)*. Association for Computing Machinery, New York, NY, USA, Article 2, 8 pages. https://doi.org/10.1145/3472538.3472595

[47] Vanessa Volz, Jacob Schrum, Jialin Liu, Simon M. Lucas, Adam M. Smith, and Sebastian Risi. 2018. Evolving mario levels in the latent space of a deep convolutional generative adversarial network. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2018, Kyoto, Japan, July 15-19, 2018*, Hernán E. Aguirre and Keiki Takadama (Eds.). ACM, 221–228. https://doi.org/10.1145/3205455.3205517

[48] Susanna Värtinen, Perttu Hämäläinen, and Christian Guckelsberger. 2022. Generating Role-Playing Game Quests With GPT Language Models. *IEEE Transactions on Games* (2022), 1–12. https://doi.org/10.1109/TG.2022.3228480

[49] Tianxiong Wang and Meiyu Zhou. 2020. A method for product form design of integrating interactive genetic algorithm with the interval hesitation time and user satisfaction. *International Journal of Industrial Ergonomics* 76 (2020), 102901. https://doi.org/10.1016/j.ergon.2019.102901

[50] Wikipedia. 2022. Design Brief (Wikipedia, The Free Encyclopedia). https://en.wikipedia.org/wiki/Design_brief

[51] Brian G. Woolley and Kenneth O. Stanley. 2014. A Novel Human-Computer Collaboration: Combining Novelty Search with Interactive Evolution. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation* (Vancouver, BC, Canada) *(GECCO '14)*. Association for Computing Machinery, New York, NY, USA, 233–240. https://doi.org/10.1145/2576768.2598353

[52] Bin Xu, Shangfei Wang, and Xian Li. 2010. An emotional harmony generation system. In *Evolutionary Computation (CEC), 2010 IEEE Congress on.* 1 –7. https://doi.org/10.1109/CEC.2010.5586210

[53] Georgios N. Yannakakis and Julian Togelius. 2011. Experience-Driven Procedural Content Generation. *IEEE Transactions on Affective Computing* 2, 3 (2011), 147–161. https://doi.org/10.1109/T-AFFC.2011.6

[54] Qihao Zhu and Jianxi Luo. 2022. Generative Design Ideation: A Natural Language Generation Approach. https://doi.org/10.48550/ARXIV.2204.09658

[55] Peter Thorup Ølsted, Benjamin Ma, and Sebastian Risi. 2015. Interactive evolution of levels for a competitive multiplayer FPS. In *2015 IEEE Congress on Evolutionary Computation (CEC)*. 1527–1534. https://doi.org/10.1109/CEC.2015.7257069