EvoDesigner: Towards aiding creativity in graphic design *

Daniel Lopes[0000−0003−0984−9775], João Correia[0000−0001−5562−1996], and Penousal Machado[0000−0002−6308−6484]

University of Coimbra, CISUC, DEI
{dfl,jncor,machado}@dei.uc.pt

Abstract. Graphic Design (GD) artefacts aim to attract people's attention before any forward objectives. Thus, one of the goals of GD is frequently finding innovative aesthetics that stand out over competing design artefacts (such as other books covers in a store or other posters on the street). However, as GD is increasingly being democratised and broadly shared through social media, designers tend to adopt trendy solutions, lacking disruptive and catchy visual features. *EvoDesigner* aims to assist the exploration of innovative graphic design solutions by using an automatic evolutionary approach to evolve the design of a number of text, shapes, and image elements inside two-dimensional canvases (pages). To enable the collaboration human-machine, the process has been integrated into *Adobe inDesign*, so human designers and *EvoDesigner* may alternately edit and evolve the same design projects, using the same desktop-publishing software. In this paper, an overview of the proposed system is presented along with the experimental setup and results accomplished so far on an evolutionary engine developed. The results suggest the viability of the development made in this first iteration of the system, which aims to reinterpret existing layouts in an unexpected manner.

Keywords: Automatic · Evolutionary · Graphic Design · Layout · Poster

1 Introduction

The goal of Graphic Design (GD) artefacts may vary according to the context of their applications. Nevertheless, it may be reasonable to classify them into two main separate groups: (i) communication artefacts, which final objective is passing information objectively to a given public, and (ii) artistic artefacts, which might seek only to be aesthetic or pass information in a non-objective way (e.g. presenting hidden messages or ones that are susceptible of personal

* This work is funded by national funds through the FCT - Foundation for Science and Technology, I.P., within the scope of the project CISUC - UID/CEC/00326/2020 and by European Social Fund, through the Regional Operational Program Centro 2020 and is partially supported by *Fundação para a Ciência e Tecnologia*, under the grant SFRH/BD/143553/2019.

interpretation). Regardless of their final aims, often, GD artefacts must first of all menage to attract the attention of the target public and only thereafter the public may read the given information or enjoy the presented aesthetics.

One of the most established and commonly adopted approaches to make designs stand out over others (e.g. other books in a store or other posters on the streets) might be enhancing aesthetics. But as GD is getting increasingly democratised and broadly produced and shared (e.g. through social media, television and even on the streets), many design artefacts tend to converge into trendy solutions which lack disruption and eye-catchy features. Therefore, finding novel graphic design solutions might not be a trivial task or otherwise, creators would come up with innovative and surprising solutions for every work produced. Also, if that was the case, maybe professional designers would not be needed at all to create catchy brands, posters, book covers and others, nor co-creative tools would be continuously researched.

To keep innovating and surprising the public, graphic designers have been constantly evolving their work processes by taking advantage of the technologies of their times. Recently, that may be observed in the exploration of digital techniques to create moving and interactive designs (such as moving/interactive posters), which most times can stand out over static ones. Nonetheless, such approaches might not be a possibility in all contexts, not only because of technicalities but also because the development of such digital artefacts is often more time consuming and expensive. Thereby, the creation of disruptive aesthetics might always be a key necessity in the GD area, either to be applied in static, moving or interactive artefacts. For that reason, we believe that automatic tools for assisting creativity in GD might help designers not only achieve more disruptive aesthetics, but also free them up to explore and innovate regarding other aspects, such as dynamism or interactivity, or even come out with new features that one cannot imagine yet.

In that sense, *EvoDesigner* aims to fasten the exploration and employment of innovative GD aesthetic solutions. To accomplish that, the system employs a conventional Genetic Algorithm (GA) to automatically evolve an undetermined number of text-boxes, shapes and images into two-dimensional canvases (pages). Also, to facilitate the application of the generated ideas, as well as to allow the collaboration human-machine (so both agents may contribute to the work with their valences), the system was integrated into *Adobe inDesign* (a widely used desktop-publishing software for GD) in the form of an installable extension. By doing so, human designers and *EvoDesigner* may alternately edit and evolve designs using the same software, until a satisfactory result is achieved.

In this first iteration of the system, the Mean Squared Error (MSE) between the generated individuals and a given image is calculated to assess fitness. This approach is tested for the generation of unexpected poster layouts, by approximating the page balance of both sketched and camera-ready posters. In further developments, several other modules must be developed, such as ones for estimating how balanced, legible and novel the generated pages might be.

Primarily in this paper, an overview of *EvoDesigner* is made. Then, follows the description of the developments made so far, consisting of the aforementioned evolutionary engine for evolving a number of given items (text-boxes, shapes and images) within *inDesign* pages, and towards the page layouts of a given image (poster). Lastly, the experiments for technically validating the system are presented. The preliminary results suggest the viability of the developed system for evolving GD artefacts in the form of *inDesign* pages, as well as the feasibility of manually editing and automatically evolving pages, alternately.

2 Related Work

This first iteration on *EvoDesigner* aims to contribute by presenting (i) an evolutionary tool for aiding the creation of two-dimensional graphic design artefacts (e.g. posters or book covers), (ii) which can be easily integrated in the workflow of professional designers and (iii) that takes advantage of the editing capabilities of existent desktop-publishing software. Thus, *EvoDesigner* relates mostly to page layout, including the style and geometry of the displayed items. In this section, related work is presented while identifying the respective pros and cons.

So far, more than evolutionary techniques, generative approaches have become increasingly common in the development of GD applications. In GD, these systems usually take advantage of stochastic parameters to define visual features such as the colour, size or position of certain elements [3]. However, in many cases (if not most), the aesthetics of the generated artefacts may be relatively predictable. For that reason, generative systems are many times purposely developed for specific design projects, for example, for generating variations within a defined style (graphic identity), such as creating variations of book covers within a given layout [1, 9, 15, 14] or varying visual features on logos according to the context of their application [23, 48].

Besides, generative projects for more broad applications can also be pinpointed. For example, projects for aiding the generation of typography [10, 47, 39] or the generation of two-dimensional GD layouts. This latest, better relating to our purposes. And although some of the work on the GD layouts topic may have limited capabilities, e.g. not permitting the declaration of concept-wise preferences [16, 24], the work of Ferreira (2019) [17] or Cleveland (2010) [6] stand out by allowing the users to fix indented parameters and then let the system vary the remaining, ensuring the maintenance of an intended style (e.g. useful for creating layouts for given graphic identities). A similar approach must be adopted in future developments of *EvoDesigner*. Lastly, we highlight the work of Rebelo et al. (2020) [40] on the creation of layouts for websites based on the semantic analysing of its textual content. Also, similar approaches must be implemented in further developments of *EvoDesigner*, for the generated designs to visually represent given concepts.

Furthermore, there has been research endorsing more intelligent approaches [27], such as training Machine Learning (ML) models using existing work and exploring the latent space to return interpolations of these. In GD, that has been endorsed, for example, for the generation of logos [35], typography [30, 19, 28, 5] or editing images [4, 8]. Because of its closer relation to our project, we highlight the work of Zheng et al. (2019) [53] on the creation of content-aware layouts. The shortcoming on the aforementioned ML approaches is these may often lead to pastiche results (imitations of existing styles) [50], so many times these lack in capabilities for exploring more disruptive solutions.

For that reason, we argue that Evolutionary Computation (EC) approaches may have greater potential for the exploration of innovative GD solutions, due to their similarity to the work processes of human designers [49], i.e. both humans and EC can explore the space of possibilities towards a given target (important as GD projects usually have a briefing to respond to), yet EC systems have the advantage of allowing a higher number of experiments per time, comparing to humans. Even so, humans still being crucial for the ultimate judgement of the results, fine-tuning and several other tasks.

Many (if not most) EC systems for GD applications endorse interactive approaches, i.e. in which the user must drive the generation process. For example, there is work on the generation of figures [22], icons [11, 13, 12], logos [45, 18], typography [38, 52, 44, 51], websites [46, 36] or posters [25, 26]. From the reviewed work on this topic, the most robust might be the work of Önduygu (2010) [37], by being able to evolve typographic fonts, lines, shapes, colours, images and visual filters. *EvoDesigner* seeks to expand this range of abilities even further for trying to put the system on a pair with human designers, as much as possible.

Regarding automatic EC creative systems, less work seems to be published. This might be due to the difficulty of objectifying aesthetics to create appropriate fitness functions. Thus, even though some frameworks have been presented so far [2, 21, 43, 32], none of these might fully solve the aesthetics evaluation problem. Nonetheless, one may identify some successful automatic EC systems for GD applications, such as the work of Rebelo et al. (2017) for evolving moving posters according to the actions of the spectators [41]. Furthermore, we highlight the work of Rebelo et al. (2018) [42] by allowing both automatic and interactive evaluation, so the system and human designers may collaborate in the evolutionary process.

Another relevant hybrid approach might be applying ML techniques for assessing fitness in EC creative systems. In the computational art scope, a considerable number of works might be found [20, 31, 7]. Nevertheless, regarding GD, there might not be many references. Besides, one may identify relevant work, such as the one of Martins et al. (2016) for evolving typefaces out of given modules [33].

Lastly, we refer to existing work that integrates EC systems in existing desktop-publishing software. As long as we could assess, such integrated systems might not be highly common. However, it is possible to identify a few examples already. One of these is *Microsoft PowerPoint's Design Ideas* [34] which may be a good analogy to the workflow on *EvoDesigner* since (i) the system is integrated into widely popular software; (ii) it takes advantage of the software's functionalities; (iii) the users must start by inserting content and then the system will suggest styling solutions; (iv) both the user and the system can contribute to the results; (v) the user can improve the final results by editing them. Also, *Evolving Layout* [26] might be a noteworthy work due of its integration in *Adobe inDesign* (same way as *EvoDesginer*). The shortcoming on this system is its limited capabilities, only allowing to interactively evolve the position, scale and rotation of the page elements. As mentioned before, *EvoDesginer* aims not only to evolve designs automatically but also to take advantage of a wide range of software functionalities (as much as possible), so it is on par with human designers.

3 Approach

As an approach for aiding the creation of disruptive GD solutions, we propose the development of an automatic evolutionary system for evolving pages — *EvoDesigner*. The system must assist graphic designers during the experimentation stages of their workflow, so both human and machine can contribute to a given project by editing pages (individuals). To do so, *EvoDesigner* is presented to the user in the form of an extension (plug-in) for a broadly used GD desktop-publishing software — *Adobe inDesign*. The code implementation of the built-in extension has been accomplished using HTML, CSS, *JavaScript* and *ExtendScript* (*JavaScript* for *Adobe*'s software).

An evolutionary engine based on a conventional Genetic Algorithm (GA) with automatic fitness assignment makes the core module of *EvoDesigner*. However, the whole system can be described as a composition of several individual modules: (i) the referred evolutionary engine; (ii) several modules for visually evaluating images and which may or not be picked by the user to assess fitness, such as (among other possibilities) modules for assessing how novel, legible, balanced and how related to a given GD style a page is, or how similar it is to a given image; and (iii) a module for translating keywords (defined by the user) into visual features (e.g. colours, geometric transformations, font weights and others), which must be useful for limiting the search space, leading to results that are more visually related to the concept (keywords) of the respective projects.

So far, developments have been done for implementing the referred evolutionary engine using image-similarity for fitness assignment. Thus, this paper does not include developments on the keywords module nor the novelty, legibility, balance and style evaluation ones. Nevertheless, a full schematic representation of the system is presented in Figure 1. Besides allowing the validation of the developed evolutionary engine, using image-similarity metrics for fitness assignment might also be useful in practical GD tasks, such as finding unexpected layouts that approximate given drafts, e.g. the ones in Figure 5, used as target images in some of the experiments presented later in this paper.

To interact with the system, in the *Adobe inDesign* environment, the user must start by creating a blank document and inserting the intended items (text-boxes, images or shapes) into pages, as usual for starting a project in *inDesign*. Then, a user interface can be used for setting up the system variables. At the current stage, the following are allowed: (i) set of pages to evolve; (ii) population

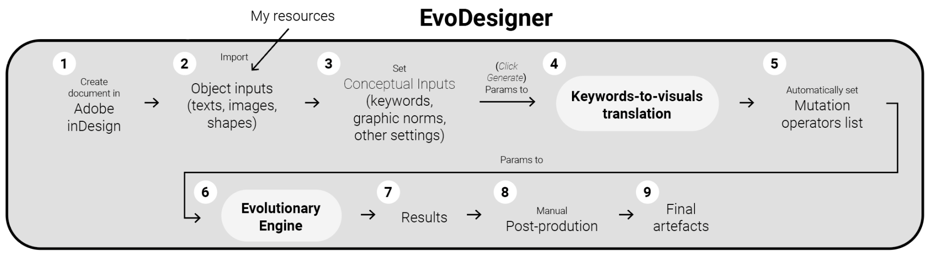

Fig. 1. Schematic representation of *EvoDesigner*. 1) The user must first 1) create a blank document, 2) insert elements into pages, 3) set up desired preferences (e.g. set the pages to evolve and set keywords) and click "Generate" to start; 4) the module *Keywords-to-visuals translation* will try to find properties/tools that match the inserted keywords; 5) Each property/tool will be assigned with a probability to be used by the system to mutate pages (individuals); 6) the evolutionary engine will evolve pages; 7) the resulting pages will be made available as normal and editable *inDesign* pages; lastly, 8) the designer may edit the results and 9) export final artefact. From any stage of the user interaction, the parameters might be changed and the evolution restarted.

size; (iii) number of generations to run; (iv) items that must always be included in any page (e.g. one may define text-boxes to be mandatory and let images be optional). In further developments, other functionalities must be allowed, such as (i) inserting keywords; (ii) defining tools and visual features that the designer wants/desires to be used (e.g. certain colours or typefaces); (iii) defining the hierarchy of the elements (i.e which ones must be emphasised the most); (iv) what fitness modules to use and how important is each of these. After setting up the system preferences, the user must press a "Generate" button for starting the system. Once the evolution terminates, the user can edit the results normally using *inDesign* or evolve again some indented pages, using the same or different parameters.

3.1 Evolutionary Engine

As previously mentioned, before running the system, the user must define which pages (individuals) to evolve, from 1 to any number pages. For example, the user might be working on a document with 10 pages but only wants to consider 3 of them in the evolutionary process. Besides, the desired population size must be set. If the number of selected pages happens to be bigger than the defined population size, the later parameter will be automatically increased to match. Otherwise, if the number of selected pages is smaller than the defined population size, before the evolutionary process starts, the system will automatically create the remaining individuals by crossing over and mutating the selected pages, and it will certify that all the mandatory items are included in every page.

Furthermore, the user might name items, so the system knows which must be treated as equivalent i.e. if several pages have an item named "title" (even if these "title" items have different visual styles among them), the system will assume these are equivalent. However, in the current iteration of the system, this is only useful for mandatory items, as giving the name of a mandatory item to another item will make it mandatory too. Thus, as long as one of these same-name items is on the page, the mandatory items criteria is matched.

After the aforementioned initialisation process, the engine will proceed to the evaluation of the individuals and then check for termination criteria, which might be (i) finding an individual whose fitness equals or exceeds a given satisfactory value (at this stage, not considered), (ii) whether the system had run a given number of generations or (iii) whether the user has ordered the system to stop evolving, by clicking the button "Stop generation".

If no termination criteria was matched, selection will be performed using a tournament method of size 2 and an elite of 1 individual. Lastly, a new population must be created by crossover and mutation, the offspring must be evaluated and the process must repeat.

Representation As briefly suggested already, the phenotype of individuals is the native render of *inDesign* pages themselves, which may contain different types of items, such as text-boxes, shapes or images, which in turn are defined by a number of positioning, geometry and style properties. These properties are stored automatically by *inDesign* in the JSON format. In that sense, in *EvoDesigner*, genotypes consist of JSON objects containing all the properties of the respective pages, as well as the properties of the items contained in them (refer to Figure 2 for a schematic example of the genotype). However, in this first iteration of the system, only the following item-properties were considered: the shape of the surrounding box, size, position, order of the items (z-position), flipping mode, blending mode, opacity, background colour/gradient, background tint, stroke colour/gradient, stroke tint, stroke weight, rotation and shearing angle. Also, for text-boxes, it is available text size, typeface, justification, vertical text alignment, letter spacing and line-height. In further developments, a number of other properties must be available, including page properties such as margins or grid rulers. Furthermore, the properties "name" and "label" are used to keep track of mandatory items, having no visual effect on the phenotypes.

Variation Variation-wise, all the generated individuals go through crossover and then mutation processes. In this first iteration of the system, crossover only shifts whole items and not individual item properties. The crossover process executes as follows: All the items of the first parent (P1) are iterated randomly (are not picked by their order in the page). Each of these items (I1) has a 50% chance to pass directly to the offspring (with the same position, geometry and style). If that is not the case, the system will try to pick, from parent 2 (P2), a random item (I2) that has not been passed to the offspring yet. If no such I2 exists, I1 will be passed anyway. Otherwise, I2 will be passed instead. Thus, offspring can contain a minimum and a maximum number of items respectively equal to the number of items in the smaller and bigger individuals of the initial

```
Page {
    PageItems: [
        Item: {
            Shape_of_surrounding_box: one_of_available_constants,
            size: array_of_numbers,
            position: array_of_numbers,
            z_position: integer,
            flipping_mode: one_of_available_constants,
            blending_mode: one_of_available_constants,
            opacity: number,
            background_colour: colour_or_gradient,
            background_tint: number,
            stroke_colour: colour_or_gradient,
            stroke_tint: number,
            stroke_weight: number,
            rotation: number,
            shearing: number,
            text_size: number,
            typeface: one_of_available_constants,
            justification: one_of_available_constants,
            vertical_alignment: one_of_available_constants,
            letter_spacing: number,
            line_height: number,
            ... (other not-used properties of Item)
        }
        ... (other Items)
    ]
    ... (other properties of Page)
}
```

Fig. 2. Schematic representation of an individual's genotype (this scheme serves only for the sake of the example, so the property names and value-types might not be fully accurate).

population (which, as already referred, might be automatically generated by the system).

Mandatory items can only shift with similar mandatory items. In other words, if I1 is a mandatory item, then I2 must have the same name as I1, or I1 will be always passed. This can be used, for example, so titles (items named "title") can only shift with other titles. Similarly to what happens in the project *Ádea* [29], we refer to such approach as topological crossover, once the shifts happen among similar structural parts. As a further explanation, if an item "title" is mandatory, offspring will always inherit an item "title" either from a parent or the other. A natural analogy might be always inheriting crucial structural parts such as eyes, either from the father or the mother. In the case of our system, this is relevant to guarantee that all the posters include all the mandatory (structural) items, at least. A natural analogy for optional items might be more difficult to pinpoint, but for the sake of the example, one may think of it as inheriting or not a chronic disease.

For each individual, each mutation method might run within a 1% chance, changing one of the position, geometry and style properties referred to in Figure 2. The value assigned to each property is picked randomly. These might be random integers, floats, arrays of numbers or also picked from lists of predefined constants. That is the case of colours, which are picked from a list of colour values. In this iteration of the system, a fixed list of seven colours — black, white, magenta, yellow, red, green and cyan.

Fitness Assignment In this iteration of the system, fitness was assessed through the calculation of an image similarity value between the generated individuals and a given target image. To accomplish that, the individuals are first exported in the PNG format, 72 dpi. The target images share this same format and settings. Then, each individual (PNG) is compared with the given target image through the calculation of the Mean Squared Error (MSE), returning a value m, representative of the difference between the images. Thus, for returning a similarity value (so the bigger the value, the better), the final fitness value equals negative m.

A well-established image similarity metric such as MSE was chosen, first of all, to understand whether or not the developed evolutionary engine is able to evolve correctly. Nevertheless, it also might be useful for practical GD tasks such as the generation of relatively unexpected layouts by approximating the page balance of given images (either sketches or camera-ready images; see Figure 5), using given page-items. As mentioned before in this paper, in further developments other fitness functions must be developed. For instance, from our background in GD, we believe that assessing novelty and balance values might be fundamental for describing disruptive and appealing GD artefacts. However, the latter (as well as MSE) might only be enough for the generation of more artistic artefacts in which legibility might not be important. Thus, for the generation of communication design artefacts (probably, most cases), a legibility value must also be retrieved and considered in fitness assignment. Also, in our perspective, not so crucial yet useful, might be retrieving an additional value for whether an individual belongs to a given GD style (aesthetic movement).

4 Experimental Setup and Results

One of the primary use-cases for *EvoDesigner* might be the generation of posters. Thus, for setting up experiments, the 3 speculative posters of Figure 3 (manually created from blank pages but little stylised) were selected to be evolved. Figure 4 showcases an example of an initial population of 10 individuals generated out of the same 3 selected pages, using crossover and mutation operations.

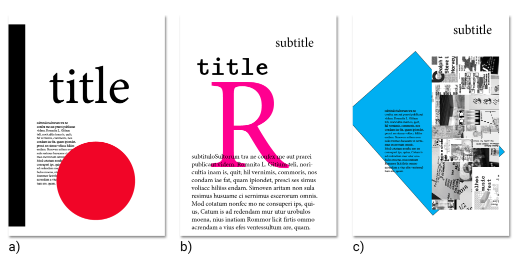

Fig. 3. Pages selected to be evolved (manually created from blank pages; little stylised).

Moreover, the images of Figure 5 were used as targets. Figure 5.b showcases speculative camera-ready posters designed in *inDesign*. These were used at first

Fig. 4. Example of an initial population of 10 individuals, generated out of the 3 selected pages of Figure 3.

instance to assess whether and until what point the system was evolving. Figure 5.a showcases sketches representative of the respective posters of Figure 5.b. These were used as targets for the main experiments, as these might better exemplify an expected target image for whenever using MSE for fitness assignment. For example, a designer might sketch an abstract layout and let the system generate posters that can approximate it, using a given set of page items. Nevertheless, camera-ready posters such as the ones in Figure 5.b might also be useful for different use-cases. For example, if a designer likes a given existent poster, the system might help create new ones with a similar page balance. Even so, these must still be different enough from the target ones once the given page-items must differ, as must do their position, geometry and style.

The remaining parameters were set up as follows: (i) population size: 50 individuals; (ii) tournament size: 2; (iii) elite size: 1; (iv) probability to crossover a page item: 50%; (v) probability for a mutation method to perform: 1%; (vi) mandatory elements: all the text-boxes in the selected pages (the pages of Figure 3); (vi) fitness assignment: MSE; (vii) maximum generations: dependent of the experiment; (viii) termination criteria: achieving the defined number of maximum generations.

In the first experiments, a run with the parameter "maximum generations" set to 1000 was made for assessing whether the fitness values were maximising and what number of generations would be necessary until no major gains were accomplished. To do that, the poster from Figure 5.b.1 was set as a target, once it seemed to be the more easily achievable one, from the presented posters — visually heavier in the top-left and right-bottom corners (black items), and with medium-weight items in the top and bottom areas (red stripes). This run has been manually stopped at the 480th generation, as no major gains were being observed for many generations (see Figure 6.a, which presents the plotted fitness

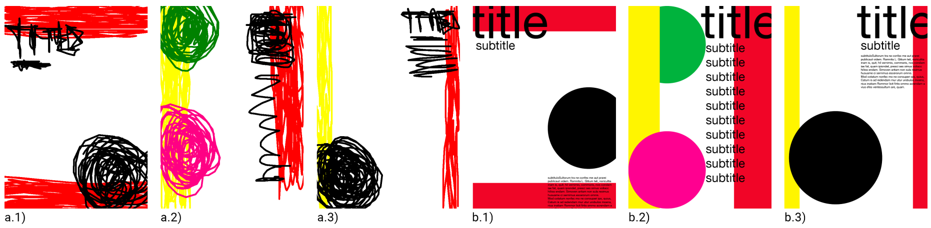

Fig. 5. Possible target images: a) sketched posters; b) camera-ready posters designed in *Adobe inDesign*.

of the best individuals of each generation, along with the average fitness of each population). As a result of this experiment, 100 generations were set for the following runs.

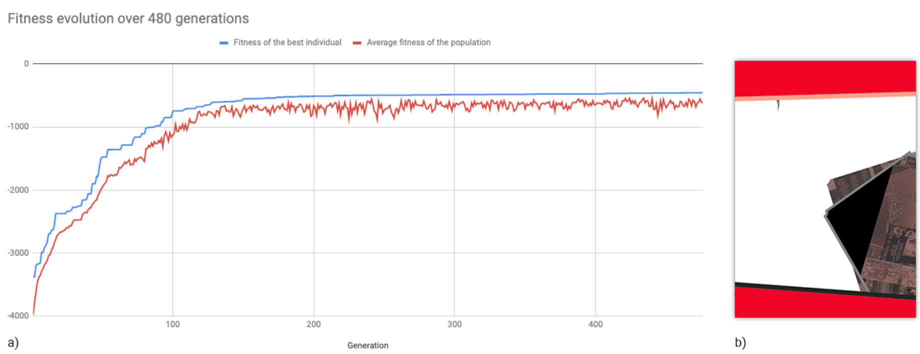

Fig. 6. Results from 480 generations using Figure 5.b.1 as a target; a) the fitness values of the best individuals of each generation and the average fitness for each population; b) the best phenotype form the 480th generation.

The following experiments were performed targeting each of the posters of Figure 5.a and maintaining the parameters mentioned before. For each poster, 4 runs were made. Figure 8 presents the average fitness of the best individuals of each generation, for each target image. Figure 7 showcases some of the resulting images generated by the system, for each of the target posters of Figure 5.a.

The resulting phenotypes suggest that the system has been able to proximate the layout (balance) of the target images, once darker areas in these tend to result in more filled and darker respective areas in the generated posters. Moreover, even the colour pallet tends to be approximated for the respective areas.

As a result, if a user has a preference for the layout of a sketched or existing poster, the presented approach can be considered for creating new posters using some indented page items, by approximating, but not copying, the layouts of the intended images, i.e. getting close but not too close from the the target images.

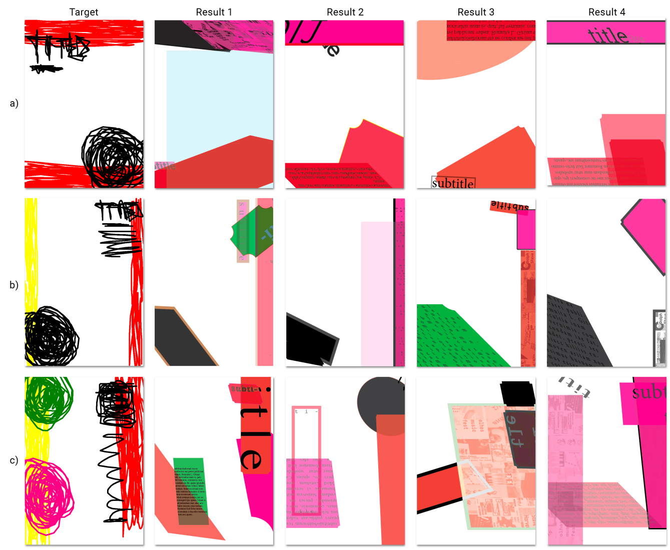

Fig. 7. Best individuals from 4 different runs (100 generations), for 3 different target images: a) Figure 5.a.1; b) Figure 5.a.2; c) Figure 5.a.3

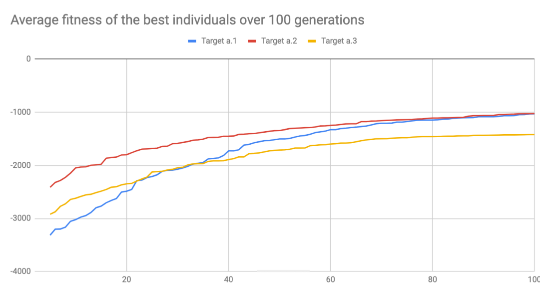

Fig. 8. Average fitness (4 runs) of the best individuals of each generation, for each target image of Figure5.a

Furthermore, as expected, we accomplished to allow the manual edition of the generated results after the termination of the evolutionary process, so human designers and *EvoDesigner* can alternately work in the same *inDesign* project.

An evident and predictable shortcoming on using MSE for fitness assignment is the lack of ability for keeping mandatory items visible (ideally, mandatory items are like so because the designer whats them to be visible to the public, rather than hiding behind other items, being off-page or just too small). In that sense, including legibility assignment in the calculation of fitness could be a possible approach for improving the results regarding this issue. Even so, designers may solve this issue by post-editing the results, so the later can turn into communication artefacts rather than just aesthetic ones.

5 Conclusion

In GD, finding disruptive aesthetics is usually of the utmost importance for getting the attention of the target public. However, often GD artefacts tend to follow trendy solutions/styles, which might result in GD artefacts that might not stand out over competing ones.

EvoDesigner is an automatic system for evolving pages within the *Adobe inDesign* environment. The goal is to assist the creative process of graphic designers by alternately collaborating with these in the edition of pages and page-items, for example, for creating posters or book covers.

This paper has presented the first iteration of *EvoDesigner*, consisting of the implementation of an automatic evolutionary engine based on a conventional GA. So far, experiments have been done for evolving page layouts towards given target images, using the MSE metric for assessing fitness. To do that, the generated pages are exported from *inDesign* in the PNG format and compared to given target images, also in PNG.

In the presented experiments, the used image targets consisted of speculative posters of two kinds: (i) sketched layouts and (ii) camera-ready posters. Sketched targets might be useful, for example, whenever a graphic designer aims to generate artefacts that describe a given page balance and colour pallet. Nevertheless, utilising images of finished GD artefacts might also be useful, for example, for resembling the page balance of the targets without culminating in results that are too similar to the originals. For instance, the generated artefacts might differ in the utilised page items themselves.

The performed experiments suggested the viability of the presented approach in the evolution of GD artefacts that resemble the page balance of the target images, but that still be different enough not to be deemed as the same. In that sense, and besides user testing must be needed to attest the following statement, we believe the presented approach might be worth being included in the GD workflow for assisting the generation of disruptive GD solutions, since the system is able to take given layouts and consider these to dispose and edit page-items in relatively unexpected manners (particularly, in the creation of posters, as it has been tested in this paper).

In future work, several different modules for improving the robustness of the system must be developed, such as: (i) a module for translating keywords into visual properties/tools (e.g. for limiting the search space towards a given creative concept), or (ii) fitness modules that can or not be used for performing novelty, legibility and balance judgements, or assessing how much an image might be in-style with a given GD aesthetic movement. Also, new functionalities must be added, e.g. for positioning items according to page grids, promoting more organised layouts.

References

1. Arteaga, M.: Generative eBook Covers, The New York Public Library (2014), www.nypl.org/blog/2014/09/03/generative-ebook-covers, last accessed 24 Fev2020

2. Birkhoff, G.: Aesthetic Measure. Harvard University Press, Cambridge, MA, USA (1933)
3. Bohnacker, H., Groß, B., Laub, J., Lazzeroni, C.: Generative gestaltung. Verlag Hermann Schmidt **4** (2009)
4. Bychkovsky, V., Paris, S., Chan, E., Durand, F.: Learning photographic global tonal adjustment with a database of input / output image pairs. In: The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition (2011)
5. Campbell, N.D.F., Kautz, J.: Learning a manifold of fonts. ACM Transactions on Graphics **33**(4), 1–11 (jul 2014). https://doi.org/10.1145/2601097.2601212
6. Cleveland, P.: Style based automated graphic layouts. Design Studies **31**(1), 3 – 25 (2010). https://doi.org/10.1016/j.destud.2009.06.003
7. Correia, J., Machado, P., Romero, J., Carballal, A.: Evolving figurative images using expression–based evolutionary art. In: Proceedings of the Fourth International Conference on Computational Creativity. p. 24–31. Sydney, Australia (jun 2013)
8. Correia, J., Vieira, L., Rodriguez-Fernandez, N., Romero, J., Machado, P.: Evolving image enhancement pipelines. In: Romero, J., Martins, T., Rodriguez-Fernandez, N. (eds.) Artificial Intelligence in Music, Sound, Art and Design. p. 82–97. Springer International Publishing, Cham (2021)
9. Cruz, P., Machado, P., Bicker, J.: Data book covers (Jul 2010), https://cdv.dei.uc.pt/data-book-covers/, last accessed 1 Nov 2021
10. Cunha, J., Tiago, M., Bicker, J., Machado, P.: TypeAdviser: a type design aiding-tool. In: Workshop on Computational Creativity, Concept Invention, and General Intelligence (C3GI 2016) (2016)
11. Cunha, J.M., Lourenço, N., Correia, J., Martins, P., Machado, P.: Emojinating: Evolving emoji blends. In: Computational Intelligence in Music, Sound, Art and Design. (EvoMUSART 2019). n/a (2019)
12. Dorris, N., Carnahan, B., Orsini, L., Kuntz, L.A.: Interactive evolutionary design of anthropomorphic symbols. In: Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753). vol. 1, pp. 433–440 Vol.1 (June 2004). https://doi.org/10.1109/CEC.2004.1330889
13. Dozier, G., Carnahan, B., Seals, C., Kuntz, L., Ser-Geon Fu: An interactive distributed evolutionary algorithm (idea) for design. In: 2005 IEEE International Conference on Systems, Man and Cybernetics. vol. 1, pp. 418–422 Vol. 1 (Oct 2005). https://doi.org/10.1109/ICSMC.2005.1571182
14. Duro, L., Machado, P., Rebelo, A.: Graphic narratives (Mar 2013), https://cdv.dei.uc.pt/graphic-narratives/, last accessed 1 Nov 2021
15. Eveillard, L.: Portfolio de louis eveillard — couvertures génératives (2015), www.louiseveillard.com/projets/couvertures-generatives, last accessed 1 Nov 2021
16. Feiner, S.: A grid-based approach to automating display layout. In: Proc. Graphics Interface. vol. 88, pp. 192–197 (1988)
17. Ferreira, D., et al.: Design Editorial Algorítmico. Master's thesis, Universidade de Coimbra (2019)
18. Gambell, T., Hooikaas, A.: Emblemmatic — markmaker (2015), http://emblemmatic.org/markmaker, last accessed 14 Jul 2019
19. Hayashi, H., Abe, K., Uchida, S.: Glyphgan: Style-consistent font generation based on generative adversarial networks (2019)
20. Heath, D., Ventura, D.: Creating images by learning image semantics using vector space models. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
21. den Heijer, E., Eiben, A.E.: Comparing aesthetic measures for evolutionary art. In: Di Chio, C., Brabazon, A., Di Caro, G.A., Ebner, M., Farooq, M., Fink, A.,

Grahl, J., Greenfield, G., Machado, P., O'Neill, M., Tarantino, E., Urquhart, N. (eds.) Applications of Evolutionary Computation. pp. 311–320. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)

22. den Heijer, E., Eiben, A.E.: Evolving art with scalable vector graphics. In: Proceedings of the 13th annual conference on Genetic and evolutionary computation - GECCO '11. p. 427. ACM Press, New York, New York, USA (2011). https://doi.org/10.1145/2001576.2001635
23. Inc., S.: Casa da musica (2007), https://sagmeister.com/work/casa-da-musica, last accessed 1 Nov 2021
24. Jacobs, C., Li, W., Schrier, E., Bargeron, D., Salesin, D.: Adaptive document layout. Communications of the ACM **47**(8), 60–66 (2004)
25. Kitamura, S., Kanoh, H.: Developing support system for making posters with interactive evolutionary computation. In: 2011 Fourth International Symposium on Computational Intelligence and Design. vol. 1, pp. 48–51 (Oct 2011). https://doi.org/10.1109/ISCID.2011.21
26. Klein, D.: Evolving Layout – next generation layout tool, http://www.evolvinglayout.com, last accessed 17 Dec 2018
27. Lewis, M.: Evolutionary visual art and design. In: Romero, J., Machado, P. (eds.) The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music, pp. 3–37. Springer Berlin Heidelberg (2007). https://doi.org/10.1007/978-3-540-72877-1_1
28. Loh, B., White, T.: SpaceSheets: Interactive Latent Space Exploration through a Spreadsheet Interface. In: Workshop on Machine Learning for Creativity and Design. 32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada (2018)
29. Lopes, D., Correia, J., Machado, P.: Adea - evolving glyphs for aiding creativity in typeface design. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion. p. 97–98. GECCO '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3377929.3389964
30. Lopes, R.G., Ha, D., Eck, D., Shlens, J.: A Learned Representation for Scalable Vector Graphics. arXiv preprint arXiv:1904.02632 (2019)
31. Machado, P., Cardoso, A.: All the truth about NEvAr. Applied Intelligence, Special Issue on Creative Systems **16**(2), 101–119 (2002)
32. Machado, P., Cardoso, A.: Computing aesthetics. In: de Oliveira, F.M. (ed.) Advances in Artificial Intelligence. pp. 219–228. Springer Berlin Heidelberg, Berlin, Heidelberg (1998)
33. Martins, T., Correia, J., Costa, E., Machado, P.: Evotype: from shapes to glyphs. In: Proceedings of the Genetic and Evolutionary Computation Conference 2016. pp. 261–268. ACM (2016)
34. Microsoft: Create professional slide layouts with powerpoint designer - office support, https://support.microsoft.com/en-us/office/create-professional-slide-layouts-with-powerpoint-designer-53c77d7b-dc40-45c2-b684-81415eac0617
35. Oeldorf, C., Spanakis, G.: Loganv2: Conditional style-based logo generation with generative adversarial networks. 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA) pp. 462–468 (2019)
36. Oliver, A., Monmarch, N., Venturini, G.: Interactive design of web sites with a genetic algorithm. In: Proceedings IADIS International Conference WWW/Internet. pp. 355–362 (2002)
37. Onduygu, D.C.: Graphagos: evolutionary algorithm as a model for the creative process and as a tool to create graphic design products. Ph.D. thesis, Sabanci University (2010), https://research.sabanciuniv.edu/24145

38. Parente, J., Martins, T., Bicker, J., Bicker, J.: Which type is your type? In: Proceedings of the Eleventh International Conference on Computational Creativity (2020)
39. Pereira, F.A., Martins, T., Rebelo, S., Bicker, J.a.: Generative type design: Creating glyphs from typographical skeletons. In: Proceedings of the 9th International Conference on Digital and Interactive Arts. ARTECH 2019, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3359852.3359866
40. Rebelo, J., Rebelo, S., Rebelo, A.: Experiments in algorithmic design of web pages. In: Kreminski, M., Eisenstadt, V., Pinto, S., Kutz, O. (eds.) Joint Proceedings of the ICCC 2020 Workshops (WS 2020) (2020)
41. Rebelo, S., Martins, P., Bicker, J., Machado, P.: Using computer vision techniques for moving poster design. In: 6.º Conferência Internacional Ergotrip Design (2017)
42. Rebelo, S., Fonseca, C.M.: Experiments in the development of typographical posters. In: 6th Conference on Computation, Communication, Aesthetics and X (2018)
43. Ross, B.J., Ralph, W., Hai Zong: Evolutionary image synthesis using a model of aesthetics. In: 2006 IEEE International Conference on Evolutionary Computation. pp. 1087–1094 (2006)
44. Schmitz, M.: GenoType, interaktivegestaltung.net/genotyp2, last accessed 14 Jul 2019
45. Schmitz, M.: Evolving Logo. In: Bohnacker, H., Gross, B., Laub, J., Lazzeroni, C. (eds.) Generative gestaltung. Verlag Hermann Schmidt, 4 edn. (2009), https://interaktivegestaltung.net/evolving-logo-2/
46. Sorn, D., Rimcharoen, S.: Web page template design using interactive genetic algorithm. In: 2013 International Computer Science and Engineering Conference (ICSEC). pp. 201–206 (Sep 2013). https://doi.org/10.1109/ICSEC.2013.6694779
47. Suveeranont, R., Igarashi, T.: Example-based automatic font generation. In: Taylor, R., Boulanger, P., Kruger, A., Olivier, P. (eds.) Smart Graphics. pp. 127–138. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
48. TheGreenEyl, S.: Mit media lab (2011), www.thegreeneyl.com/mit-media-lab
49. Thoring, K., Muller, R.M.: Understanding the creative mechanisms of design thinking: An evolutionary approach. In: Procedings of the Second Conference on Creativity and Innovation in Design. p. 137–147. DESIRE '11, Association for Computing Machinery, New York, NY, USA (2011). https://doi.org/10.1145/2079216.2079236
50. Toivonen, H., Gross, O.: Data mining and machine learning in computational creativity. Wiley Int. Rev. Data Min. and Knowl. Disc. **5**(6), 265–275 (Nov 2015). https://doi.org/10.1002/widm.1170
51. Unemi, T., Soda, M.: An IEC-based support system for font design. In: SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483). vol. 1, pp. 968–973. IEEE (2003)
52. Yoshida, K., Nakagawa, Y., Køppen, M.: Interactive genetic algorithm for font generation system. In: 2010 World Automation Congress. pp. 1–6. IEEE (2010)
53. Zheng, X., Qiao, X., Cao, Y., Lau, R.W.H.: Content-aware generative modeling of graphic design layouts. ACM Trans. Graph. **38**(4) (Jul 2019). https://doi.org/10.1145/3306346.3322971